IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 2, 2020, accepted February 2, 2020, date of publication February 17, 2020, date of current version February 26, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2974328

Regional Patch-Based Feature Interpolation Method for Effective Regularization

SOOJIN JANG[1], (Student Member, IEEE), KYOHOON JIN[1], (Student Member, IEEE), JUNHYEOK AN[1], (Student Member, IEEE), AND YOUNGBIN KIM[1], (Member, IEEE)
Department of Image Science and Arts, Chung-Ang University, Seoul 06974, South Korea

Corresponding author: Youngbin Kim (ybkim85@cau.ac.kr)

This work was supported in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) funded by the Korean Government through the Ministry of Science and ICT (MSIT) under Grant IITP-2018-0-00599, in part by the National Research Foundation of Korea (NRF) funded by the Korea Government (MSIT) under Grant NRF-2018R1C1B5046461, and in part by the Seoul R&BD Program under Grant CY190039.

ABSTRACT Deep Convolutional Neural Networks (CNNs) can be overly dependent on training data, causing a generalization problem in which trained models may not predict real-world datasets. To address this problem, various regularization methods such as image manipulation and feature map regularization have been proposed for their strong generalization ability. In this paper, we propose a regularization method that applies both image manipulation and feature map regularization based on patches. The method proposed in this paper has a regularization effect in two stages, which makes it possible to better generalize the model. Consequently, it improves the performance of the model. Moreover, our method adds features extracted from other images in the hidden state stage, which not only makes the model robust to noise but also captures the distribution of each label. Through experiments, we show that our method performs competently on models that generate a large number of parameter and multiple feature maps for the CIFAR and Tiny-ImageNet datasets.

INDEX TERMS Convolutional neural network, manifold, regularization, computer vision.

I. INTRODUCTION

With the application of deep convolutional neural networks (CNNs) in diverse computer vision tasks (e.g., image captioning [1], [2], object recognition [3], [4], and semantic segmentation [5]), a range of models have been explored, including multi-path networks [6], deep networks [7], and networks utilizing the attention mechanism [8]. These models have the necessary tendency to learn more parameters to increase their representational power. Moreover, as deep CNNs learn sparse representations, their decision boundaries are less clear than those of conventional statistical methods [9]. Consequently, excessive dependence on training data causes a generalization failure [10]. Models that fail to generalize may cause a decline in the performance of the test data. To address these challenges, diverse regularization methods have been proposed.

Various regularization methods have been proposed to deal with CNN's excessive dependence on training data. The most widely utilized regularization method involves the direct augmentation of images. Some methods simply crop, rotate, and flip the images [11], whereas others eliminate or combine image information to generate new images. The research into the technique that eliminates partial information of the image considers the entire object area as well as the discriminatory part of the object in training. Thereby, it can improve generalization and localization, but some beneficial information may be removed from the image. Furthermore, the method of combining images has the limitation that the images are visually unnatural. Several methods have been explored to overcome this problem and one of them is CutMix [12]. CutMix preserves the image information by cropping the image in patches and adding them to a new image.

Furthermore, regularization methods for manipulating feature maps in a hidden state have been extensively researched. Regularization methods that eliminate some value from feature maps [13], [14], add noise to feature maps [15]–[17] or combine feature maps in hidden states have been proposed. However, the methods involving the elimination of values and addition of extra noise methods slow the convergence speed because they directly affect gradient. In addition, the methods of combining feature maps via additional networks and obtaining regularization effects may prompt additional computational costs depending on how the feature maps were combined [9], [18].

The associate editor coordinating the review of this manuscript and approving it for publication was Haruna Chiroma.

A method of patch-based regularization that applies both image augmentation and feature map regularization is proposed. The proposed method creates a new image by mixing input images on a patch basis, whereas the label of the new image, the label of the patch, and the label of the existing input image are mixed in proportion to the patch size. Then, a CNN is applied to the generated new image to generate a feature map. Next, the generated feature map undergoes linear interpolation with a feature map of the patch in proportion to the patch size.

The proposed method can be applied effectively in image manipulation and feature map regularization to obtain regularization effects. In the image manipulation stage, two images can be combined, based on the patch to generate a new image without information loss, as in CutMix [12]. Additionally, we added patch-based other image features through interpolation between feature maps of new images in the feature map regularization stage. This allows the model to simultaneously learn the image distribution of other labels and to generate a model robust to noise.

We used the CIFAR [19] and Tiny ImageNet [20] datasets to test the proposed method in ResNet [21] and WideResNet-28 [22] models. The top-1 accuracy performance of CIFAR-10 improved by 8.03% to 15.07% over the baseline and that of CIFAR-100 improved by 18.60% to 29.28% over the baseline. Tiny ImageNet displayed an improved top-1 accuracy performance between 2.54% and 5.28% over the baseline. Particularly, models that involved massive parameters and generated multiple feature maps performed better.

II. RELATED WORKS

Image manipulation is the most widely used regularization method, and it is largely divided into the following two types: Image Manipulation and Regularization in feature maps.

A. IMAGE MANIPULATION

Image manipulation refers to transforming an image to create a new image and involves diverse techniques such as image flipping, cropping, and rotating, as well as color space transformation and random erasing [11], [23]. Cutout [24] is a method that randomly drops a square region from the input image. These techniques are easily applicable but may cause information loss; the noise may also become detrimental to the performance in the case of images biased towards specific textures or shapes or in case of geometrically biased images.

Recently, various methods have been researched to not only convert a single image but also to mix two or more images. Among them, the Mixup [25] method mixes two images on a pixel basis for augmentation. Pairing sample [26] is also a method of mixing two images by average RGB pixels. In addition to mixing pixels, there has been a study to connect images to each region [27]. Despite its effectiveness in regularization, Mixup and Pairing sample have the limitation of an unnatural look of the images. To overcome this problem, devoted the study of mixing images using deep learning model to give intuitiveness of images [28]. However, in this case, it resulted in a large computational cost due to the necessity of learning the models for mixing separately. The CutMix [12] method rectifies the problem of a resultant shortfall by not just eliminating the pixels in the patches, as in Cutout, but also by filling them with pixels from other images. This patch-based image mixing achieves not just augmentation but also localization effects.

B. REGULARIZATION IN FEATURE MAPS

In addition to image augmentation, regularization in feature maps is another method universally used in many models. This method draws on the feature map obtained from the hidden state of a model, not the input image.

Dropout [13] is the most widely used and robust technique for dropping features from the hidden states for achieving regularization. Another popular technique, Dropblocks [14] does not simply drop features into random, but instead drops features into the hidden state via localization. Besides, for some tasks such as object detection, a dropout with various techniques including attention mechanism have been examined [29], [30]. Batch normalization [31] can prevent the decline in performance by solving the gradient vanishing problem in the nonlinearity function with an internal covariate shift, which directly influences the gradient and thereby renders a model robust against noises. Yet, the speed of convergence slows down in the batch normalization compared with other methods.

Furthermore, techniques for mixing noises with feature maps are also being continuously explored. In actual tests, however, unsound data may become involved. Therefore, learning by intentionally adding noises enables the models to focus on the essentials of the tasks rather than texture biases [32], [33]. This method does not simply use random noise values, but also adds probability distribution values based on statistical grounds into noise [16], [17]. Among various noise with different distributions, the Gaussian distribution is most widely used. Still, depending on the modes of application, Gaussian noise has been proven to cause substantial confusion for these models [11]. In addition, an ongoing study is delving into different methods, other than the Gaussian distribution, for adding noises to cause confusion to the distribution of the datasets and enable the models to perform tasks more robustly [15].

C. MIXING IN FEATURE MAP

Applying manipulation techniques to feature maps is also of interest to researchers. Mixing in feature maps is a widely used technique for regularization in hidden states, where the feature maps of the extracted characteristics of the images are mixed through different operations and affect the gradients of the models.

Manifold Mixup [9] addresses challenges such the sharp decision boundaries and the short distance to data by mixing in hidden states. Moreover, a method using several networks for Mixup in hidden states has been suggested [18]. In the

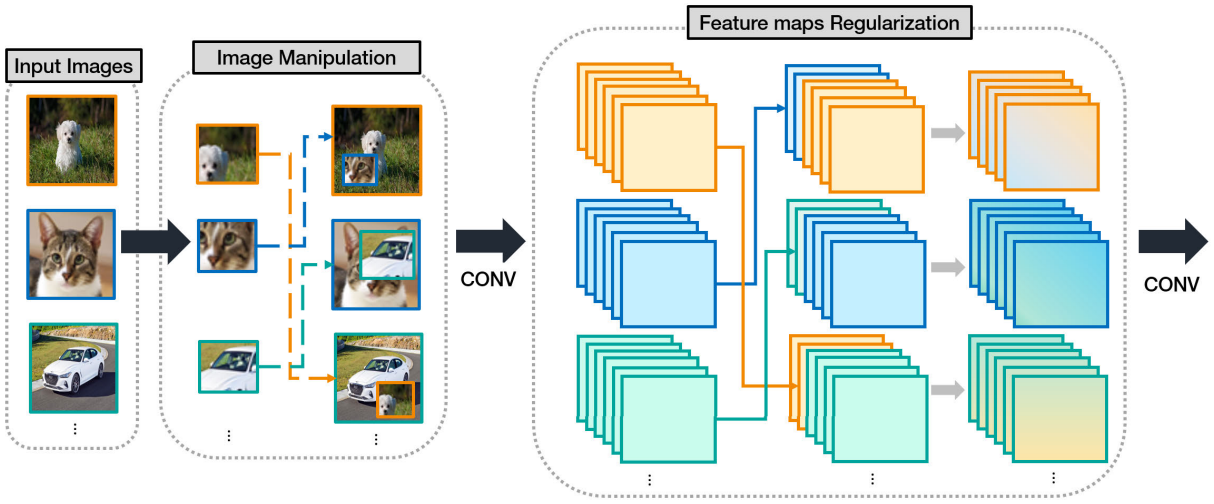


FIGURE 1. Illustration of the regularization method on three images when images are given in batches. The figure shows the process of image manipulation and feature map regularization. In feature map regularization, the feature maps are interpolated in the ratio of λ. Here, Beta(α, α).

method, a triple network structure is used to extract the features of two images with two shallow networks. Then, a new network is used to mix up the two feature maps.

In this paper, we propose a method for efficiently achieving both image manipulation and feature map regularization effects. The proposed method transforms the input images with image manipulation and performs linear interpolation on the feature maps for Mixup. The method applies the regularization to the model in two steps, enabling robust feature representation against noises and better generalization of the models.

III. METHOD

The proposed method consists of two steps. First, the image manipulation step combines the images based on patches to generate new images. Second, the feature map regularization step uses the generated images to perform a convolution network operation. Then, linear interpolation is performed on the feature maps generated in this process. The proposed method is outlined in Fig. 1.

The algorithm relevant to the proposed method is discussed in the following sections.

A. IMAGE MANIPULATION

The image manipulation step uses a ratio λ for mixing the two training samples (as in Cutmix). A training image $x \in \mathbb{R}^{W\times H\times C}$ is combined with the patch $P_x \in \mathbb{R}^{W_{1-\lambda}\times H_{1-\lambda}\times C}$ extracted in the ratio 1-λ from another training image with a different label to generate a new image, $\hat{x} \in \mathbb{R}^{W\times H\times C}$. The new images label $\hat{y}$ is generated by combining y_a with the patch's label P_y. The patch and the new training sample $(\hat{x}, \hat{y})$ are generated as follows:

$$\begin{aligned} P_x, P_y &= x_b(1-\lambda), \quad y_b(1-\lambda) \\ \hat{x} &= x_a \odot B + x_b \odot (1-B) \\ \hat{y} &= y_a\lambda + y_b(1-\lambda) \end{aligned} \tag{1}$$

Here, $(\hat{x}, \hat{y})$ is the image and label from which the patch is extracted. (x_a, y_a) is the training sample extracted from the original mini-batch index. (x_b, y_b) is the training sample extracted following a shuffle in the index within the mini-batch. As in the Mixup method, the combination ratio λ is sampled from the beta distribution Beta(α, α). For (x_b, y_b), a patch is generated in the ratio 1-λ. The patch coordinate is extracted using uniform distribution, whereas the patch size is determined in proportion to λ of the image size. $B \in \{0, 1\}^{W\times H}$ is a binary mask where the patch and position size are filled with 1 and the remaining with 0. The pixel at position P_x in x_a is removed by the element-wise multiplication of B and x_a.

The element-wise multiplication is performed on $1-B$ and x_b to extract the pixels at position P_x from x_b. Thereafter, we create a new image $\hat{x}$ by adding x_a from which the pixel at the patch is removed and x_b containing only the pixel from that patch. $\hat{y}$ is created by combining y_a with P_y.

For the training dataset $D = \{(x_1, y_1), \ldots, (x_k, y_k)\}$, we use (1) to generate a new training dataset $\hat{D} = \{(\hat{x}_1, \hat{y}_1), \ldots, (\hat{x}_k, \hat{y}_k)\}$.

In the training step, we proceed with the test by setting α to 1, that is, by sampling λ in the uniform distribution.

B. FEATURE MAP REGULARIZATION

The images generated in the image manipulation step are used as inputs for the convolutional layer to generate feature maps, which are in turn linearly interpolated to create new feature maps. By inputting the generated images into a convolutional model, we calculate the hidden state vector f derived from the convolutional layer l.

$$f^l_{\hat{x}} = C^l_{(\hat{x})} \tag{2}$$

Here, C is a convolutional model that has l convolutional layers. $f^l_{\hat{x}}$ is a feature map that is calculated by l's convolutional layers, by inputting image $\hat{x}$.

We perform the regularization through linear interpolation as given in (3), on the feature map generated in the convolution layer.

$$f^l_{\hat{x}_{k_a},\hat{x}_{k_b}} = (1-\lambda) f^l_{\hat{x}_{k_a}} + \lambda f^l_{\hat{x}_{k_b}} \tag{3}$$

$\hat{x}_{k_a}$ is the new image generated with images x_a and x_b combined. $\hat{x}_{k_b}$ is the image where x_b is combined with another training image. $f^l_{\hat{x}_{k_a},\hat{x}_{k_b}}$ is the feature map generated for the lth layer and the $\hat{x}_{k_b}$ image. The newly generated image's feature map and the feature map of the patch image used to generate the image are linearly interpolated in the mixing ratio λ to generate a new feature map.

Equation (3) is applied to a new training dataset $\hat{D}$. Instead of simply going through the convolution operation on the existing feature map, the proposed method can cause further confusion to models by combining the feature map with the distribution of another label within the data, which has been experimentally proven to implement a more robust feature representation. In our experiment, l was used with uniform distribution for sampling. The experiments section describes layer l, which is more efficient for the regularization.

IV. EXPERIEMENTS

A. DATASETS

For the experiment, CIFAR-10, CIFAR-100 [19], and Tiny ImageNet [20] data were used. To compare the performance with previous works, CIFAR datasets, which are the most widely used benchmark datasets, were used. We used Tiny ImageNet data with more images and labels than the CIFAR datasets. CIFAR-10 and CIFAR-100 data are intended for image classification and consist of 60,000 color images (50,000 training images and 10,000 test images), where each image size is 32×32. Additionally, the number of data labels in each dataset is 10 and 100, respectively. Tiny ImageNet data are also intended for image classification and consist of 100,000 images, each of which measures 64×64 in size, and has 200 labels.

B. IMPLEMENTATION DETAILS

We used a GTX-1080ti GPU for training the models, with the widely used ResNet [21] as the model. A WideResNet [22] variant of ResNet was used. Specifically, 18-, 34-, and 50-layer ResNet and 28-layer WideResNet were used. The batch size for the CIFAR data was set to 64, and that for Tiny ImageNet data was set to 128. The training epochs of each model were set to 250 and 300 for the CIFAR and Tiny ImageNet data, respectively. We used the Stochastic Gradient Descent (SGD) [34] for optimization. For the CIFAR data, the learning rate was initially set to 0.25 and decayed by a factor of 0.2 at the 60th, 120th, 160th, and 200th epochs, respectively. For the Tiny ImageNet, the learning rate was initially set to 0.1 and decayed by a factor of 0.1 at the 75th, 150th, 225th, and 300th epochs, respectively. Moreover, given that overfitting easily occurs in the baseline in comparison with other models, we used early stoppage as necessary.

TABLE 1. Summary of test top-1 accuracy rates(%) in comparison with CIFAR-10 dataset. For cutmix and manifold mixup, we trained the ResNet models with different hyperparameters and reported the best results.

Method	ResNet-18	ResNet-34	ResNet-50
Baseline	82.67	81.72	84.40
+ Augmentation	91.98	91.00	92.16
+ Cutout	92.87	96.45	94.40
+ Dropblock	94.82	94.89	94.88
+ Mixup	95.92	96.34	96.52
+ CutMix	**96.47**	96.55	**96.62**
+ Manifold Mixup	95.90	93.86	93.88
+ Our method	96.37	**96.79**	96.19

TABLE 2. Summary of the test top-1 accuracy rates(%) in comparison with the CIFAR-100 dataset. For cutmix and manifold mixup, the hyperparameter settings are the same as described in Table 1.

Method	ResNet-18	ResNet-34	ResNet-50
Baseline	55.79	52.56	57.60
+ Augmentation	67.48	77.22	69.97
+ Cutout	71.91	78.70	78.78
+ Dropblock	73.49	76.00	78.59
+ Mixup	78.22	78.16	81.07
+ CutMix	80.13	80.38	80.74
+ Manifold Mixup	78.65	72.54	75.17
+ Our method	**80.34**	**81.84**	**81.69**

We describe the best performances of our method and other methods during training. We used accuracy and Error metrics to evaluate the classification task. These metrics indicate the accuracy and error rate of the predicted values of the trained model concerning the ground-truth values.

C. EXPERIMENT WITH CIFAR DATASET

Each dataset was used for an experiment in the baseline model and other different models. We compared our methods with the baseline, augmentation, and other regularization methods. The augmentation settings were random cropping and random flipping. Other regularization methods used for comparison were Cutout [24], DropBlock [14], Mixup [25], CutMix [12], and Manifold Mixup [9].

Each method was used in the experiment based on the optimal hyper parameter values mentioned in each article. For Cutout, the learning rate was set to 0.1, the number of holes to 1, and the hole length to 16. For Dropblock, the keep-prob was set to 0.9 and the block size to 4. In Mixup, the learning rate was set to 0.1, α to 1.0, and decay to 1e-4. In CutMix, the learning rate was set to 0.25 and α to 1.0. The results are summarized in Table 1, Table 2, and Table 3. The results show the top-1 accuracy achieved by testing each method in the ResNet-18, ResNet-34, ResNet-50, WideResNet-28 models.

Table1 and table2 show the results of the CIFAR-10 and CIFAR-100 datasets in the ResNet-18, ResNet-34, and ResNet-50 models. Our method achieved a 96.79% top-1 accuracy in ResNet-34 on the CIFAR-10 dataset, which is 15.07 higher than the baseline performance of 81.72%. On the CIFAR-100 dataset, our method achieved 80.34%, 81.84%, 81.69% top-1 accuracy in the ResNet-18, ResNet-34, and ResNet-50 models, respectively, which is

TABLE 3. Test top-1 accuracy rates(%) on Test dataset in WideResNet-28 model. C10 means CIFAR10 dataset, and C100 means CIFAR-100 dataset.

Method	# Paramas	C10	C100
WideResNet-28	36.4M	89.25	65.61
+ Augmentation	36.4M	92.48	74.78
+ Cutout	36.4M	95.20	75.40
+ Dropblock	36.4M	94.78	78.77
+ Mixup	36.4M	97.04	82.17
+ CutMix	36.4M	96.91	82.62
+ Manifold Mixup	36.4M	97.24	83.97
+ Our method	36.4M	**97.28**	**84.21**

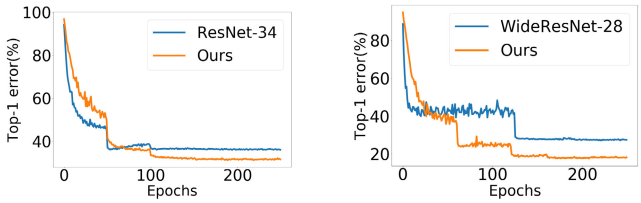


FIGURE 2. Top-1 test error(%) plot on CIFAR-100 classification. On the left (a) is the result of the ResNet-34 model, and on the right (b) is the result of WideResNet-28.

24.55%, 29.28% and 24.09% higher than the baseline performance.

For more details, we visualized the error values of each epoch of ResNet-34 and WideResNet-28 for the CIFAR-100 datasets. Fig. 2(a) shows that our model converges slightly slower than the baseline in the early stage, but becomes more stable after the first learning rate scheduling. Compared with the baseline, the error value increases in some sections. In our model, however, the error values gradually decrease throughout the training. A similar trend can be found in Fig. 2(b). It shows that the convergence is slightly slower in the early stages, as before, but it can be confirmed that it converges faster after the first learning rate scheduling. In particular, the baseline has not shown much difference since the first learning rate scheduling in comparison with our method.

To test the performance of our proposed method, we compared it with state-of-the-art augmentation methods. For the CIFAR-10 data, our method outperformed other existing methods in ResNet-34. In contrast, CutMix and Mixup outperformed our proposed method, respectively, in the ResNet-18 and ResNet-50 models, although the difference was marginal (approximately 0.2), which indicates that the proposed method is sufficiently effective.

Moreover, in CIFAR-100, regardless of the models, the proposed method achieved the highest performance and demonstrated approx. 2% performance improvement, exerting a substantial effect on the model generalization.

Table 3 shows the performance comparison against other state-of-the-art data augmentation and regularization methods of the CIFAR dataset in the WideResNet-28 model. Our method achieves a 97.28% top-1 accuracy on CIFAR-10 and

TABLE 4. Summary of test accuracy with the Tiny ImageNet dataset. For Cutmix and Manifold mixup, we trained the ResNet models with different hyperparameters and reported the best results.

Model	# Params	Top-1 Acc(%)	Top-5 Acc(%)
ResNet18	11.2M	63.22	84.16
+ Cutmix	11.2M	**66.07**	86.11
+ Manifold Mixup	11.2M	65.13	85.26
+ Our method	11.2M	65.76	**86.31**
ResNet34	21.3M	64.02	84.89
+ Cutmix	21.3M	68.47	87.25
+ Manifold Mixup	21.3M	67.41	86.18
+ Our method	21.3M	**68.77**	**87.90**
ResNet50	23.9M	63.93	84.57
+ Cutmix	23.9M	67.38	85.33
+ Manifold Mixup	23.9M	**69.75**	87.50
+ Our method	23.9M	69.21	**87.86**

an 84.21% top-1 accuracy on CIFAR-100. Our method outperforms CutMix and Manifold Mixup, by 0.37% and 0.04%, respectively on CIFAR-10. On CIFAR-100, it surpasses CutMix and Manifold Mixup, by 1.59% and 0.24% respectively.

The experimental results indicate a performance improvement in the deep ResNet or WideResNet models in generating massive feature maps in comparison with the shallow ResNet model. Indeed, with the CIFAR-100 data, ResNet-18 showed an approximately 0.2% performance improvement compared with other methods, whereas the WideResNet-28 achieved an approximately 1% performance improvement.

According to the experimental results, we achieved improved performance for those models where our methods produced multiple feature maps. Moreover, the results confirm that performing regularization in both the input and hidden state phases has a robust regularization effect on the model.

D. EXPERIMENT WITH TINY IMAGENET DATASET

To explore if our method works well with data that are larger than CIFAR and have diverse labels, we used the Tiny ImageNet data set. Each dataset was used in the experiments in the baseline model and other models. As in the aforementioned experiments, each method was applied with the optimal hyperparameters mentioned in each article. Each method was tested in ResNet-18, ResNet-34, and ResNet-50 models. The results are summarized in Table 4.

The experimental results from the Tiny ImageNet data were comparable to those from the earlier experiment. First, compared with the baseline, a performance improvement between 2.54% and 5.28% was achieved in terms of the top-1 accuracy. Apart from the performance improvement, the trend thereof is also comparable to earlier experimental results. For the top-1 accuracy in ResNet-18, our method outperforms other methods, especially in the deeper models.

Particularly, in ResNet-50, our method shows more than 2% performance improvement compared to Cutmix. Hence, in models that are deeper and have more parameters to learn, our method is more effective for generalization.

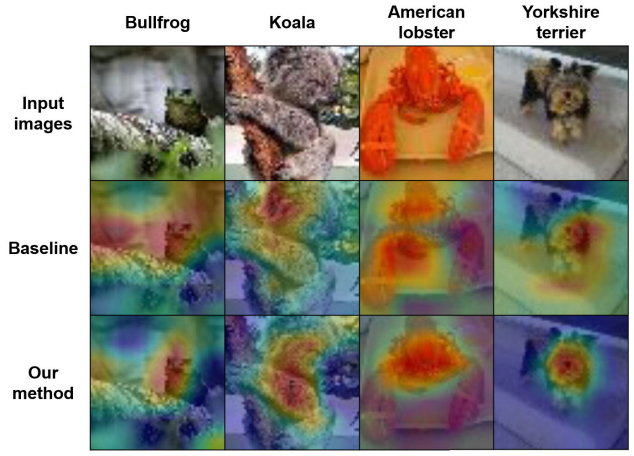


FIGURE 3. The visualization results of class activation mapping applying baseline and our method in ResNet-34 to Tiny ImageNet data.

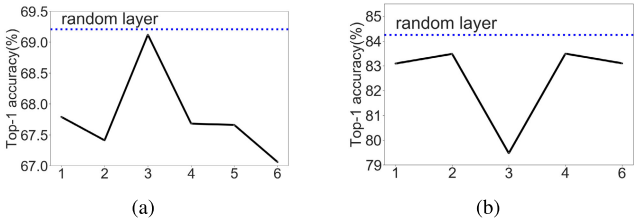


FIGURE 4. Results of experiments with the proposed method under fixed layer depth. Tiny ImageNet data in ResNet-50(a); CIFAR-100 data in WideResNet-28(b).

E. CLASS ACTIVATION MAPPING VISUALIZATION

The proposed method is not only an image manipulation but also a linear interpolation of two feature maps. Because a powerful regularization scheme overlaps, the model may not focus on the main information in the image. Therefore, we plotted a class activation map (CAM) [35] to visually check if the model properly captures the main information of the image.

The result of the CAM according to each label of input images is given in Fig. 3. As seen in Fig. 3, the CAM result of the baseline finds no important features and is widely activated on the entire images. In the case of our method, it can easily be seen that the important features for image classification are activated. The reason for this result is that our method effectively utilizes image manipulation and feature map regularization to learn important features in an image.

F. ABLATION STUDY

1) LAYER IN OUR-METHOD

The proposed method involves linear interpolation on feature maps within hidden states. Here, the layer where the method was applied was randomly selected. Fig. 4 shows the result of fixing the layer onto which our method was applied.

The experimental details are same as those mentioned earlier (Section IV-B). In the experiment, ResNet-50 (Fig. 4 (a)) and WideResNet-28(Fig. 4 (b)) were used for the Tiny ImageNet and CIFAR-100 data, respectively. We denoted 1 in the index for linear interpolation after the first convolution operation, batch normalization and activation function. Values 2 to 5 in the index indicated that linear interpolation was performed after each stage and 6 denoted after average pooling. The results show that random selection of layers led to a better performance than a planned designation of layers.

TABLE 5. Test accuracy of our method for different hyper-parameter α on Tiny ImageNet.

α	Top-1 Acc(%)	Top-5 Acc(%)
0.1	68.90	87.35
0.25	68.78	87.41
0.5	69.55	88.13
1	68.77	87.9
2	70.24	**88.53**
4	**70.87**	88.51

TABLE 6. Performance of different interpolation methods on CIFAR-10.

ResNet-18	Top-1 Acc(%)
linear	90.41%
concat linear	84.62%
nonlinear (tanh)	90.06%
linear interpolation (Our method)	**96.36%**

2) IMPACT OF HYPER-PARAMETER α

Table 5 shows the impact of the hyperparameter in extracting the mixing ratio from the hidden states and the input step in our method. We experimentally used Tiny ImageNet in the ResNet-34 model as described earlier (Section IV-B). As the results indicate, when the size of a patch and the original image were similar in the mixing step, the performance was better and improved with a range of choices.

3) COMPARISONS OF DIFFERENT INTERPOLATION METHOD

In our method, a preset was multiplied in the feature map regularization step and a new feature map was generated with linear interpolation. When generating a new feature map, we performed the experiments with different methods. The results are shown in Table 6.

The linear layer involves interpolation on two different feature maps after going through the linear layer. The concat linear layer involves the concatenation of two different feature maps before going through the linear layer. The nonlinear layer involves using a nonlinear function. We used the hyperbolic tangent function. The experimental results show that when a new feature map is generated, using linear interpolation leads to better results than as compared with adding a linear layer and nonlinearity.

The results of this experiment show that mixing two feature maps is effective, because it has better performance than the basic baseline regardless of the method of mixing the two feature maps. Moreover, we can confirm that mixing in our way is the best performance. In the case of mixing the same way as the concatenation of two feature maps and reduction in dimension by the linear layer, the baseline and performance

changes are insignificant, while our method shows about 6% improvement in performance compared with other mixing methods as well as with the linear method and nonlinear method.

V. CONCLUSION & FUTURE WORK

This paper proposes a method for regularizing both input images and the feature maps thereof. The proposed method, unlike existing ones, mixes a different image distribution, not random noises, with the feature map on a patch basis in the feature map regularization step.

As a result, the proposed method enabled the models to learn the distribution of images with different labels as well as to eliminate noises, and it ultimately outperformed other methods. When applied to WideResNet-28 for the CIFAR data, the top-1 accuracy was 97.28% for CIFAR-10 and 84.21% for CIFAR-100; the improved performance for CIFAR-10 was between 0.04% and 8.03% and that for CIFAR-100 was between 0.24% and 18.6% over other methods. For the Tiny ImageNet dataset, ResNet-34 and ResNet-50 achieved a top-1 accuracy of 68.77% and 69.21%, respectively, and ResNet-34 showed an improved performance between 0.3% and 4.75% compared with other methods.

When using the convolutional network in different practical applications, instead of using a single regularization technique, several techniques are used in combination (e.g., flip + crop, dropout + batch-normalization). Hence, instead of using our proposed method alone, using it in combination with other regularization methods will add to the effects of robustness in regularization. In the future, we plan to expand the method for applying noise to the model and study adversarial attacks.

REFERENCES

[1] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[2] M. Cornia, L. Baraldi, and R. Cucchiara, "Show, control and tell: A framework for generating controllable and grounded captions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8307–8316.

[3] Y. Li, Y. Chen, N. Wang, and Z. Zhang, "Scale-aware trident networks for object detection," 2019, *arXiv:1901.01892*. [Online]. Available: http://arxiv.org/abs/1901.01892

[4] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[5] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," 2019, *arXiv:1909.11065*. [Online]. Available: http://arxiv.org/abs/1909.11065

[6] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1–7.

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[8] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3156–3164.

[9] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, A. Courville, D. Lopez-Paz, and Y. Bengio, "Manifold mixup: Better representations by interpolating hidden states," 2018, *arXiv:1806.05236*. [Online]. Available: http://arxiv.org/abs/1806.05236

[10] R. Caruana, S. Lawrence, and C. L. Giles, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 402–408.

[11] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, p. 60, 2019.

[12] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "CutMix: Regularization strategy to train strong classifiers with localizable features," 2019, *arXiv:1905.04899*. [Online]. Available: http://arxiv.org/abs/1905.04899

[13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[14] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "Dropblock: A regularization method for convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 10727–10737.

[15] Z. You, J. Ye, K. Li, Z. Xu, and P. Wang, "Adversarial noise layer: Regularize neural network by adding noise," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 909–913.

[16] M. C. Ilter, H. U. Sokun, H. Yanikomeroglu, R. Wichman, and J. Hamalainen, "The joint impact of fading severity, irregular constellation, and non-Gaussian noise on signal space diversity-based relaying networks," *IEEE Access*, vol. 7, pp. 116162–116171, 2019.

[17] F. Horger, T. Würfl, V. Christlein, and A. Maier, "Towards arbitrary noise augmentation—Deep learning for sampling from arbitrary probability distributions," in *Proc. Int. Workshop Mach. Learn. Med. Image Reconstruction*. Cham, Switzerland: Springer, 2018, pp. 129–137.

[18] H. Oki and T. Kurita, "Mixup of feature maps in a hidden layer for training of convolutional neural network," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2018, pp. 635–644.

[19] A. Krizhevsky, "Learning multiple layers of features from tiny images," M.S. thesis, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2009.

[20] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[22] S. Zagoruyko and N. Komodakis, "Wide residual networks," 2016, *arXiv:1605.07146*. [Online]. Available: http://arxiv.org/abs/1605.07146

[23] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," 2017, *arXiv:1708.04896*. [Online]. Available: http://arxiv.org/abs/1708.04896

[24] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," 2017, *arXiv:1708.04552*. [Online]. Available: http://arxiv.org/abs/1708.04552

[25] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," 2017, *arXiv:1710.09412*. [Online]. Available: http://arxiv.org/abs/1710.09412

[26] H. Inoue, "Data augmentation by pairing samples for images classification," 2018, *arXiv:1801.02929*. [Online]. Available: http://arxiv.org/abs/1801.02929

[27] R. Takahashi, T. Matsubara, and K. Uehara, "Data augmentation using random image cropping and patching for deep CNNs," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.

[28] D. Liang, F. Yang, T. Zhang, and P. Yang, "Understanding mixup training methods," *IEEE Access*, vol. 6, pp. 58774–58783, 2018.

[29] J. Choe and H. Shim, "Attention-based dropout layer for weakly supervised object localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2219–2228.

[30] K. K. Singh and Y. J. Lee, "Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3544–3553.

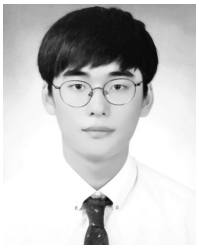
[31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167

[32] C. Gulcehre, M. Moczulski, M. Denil, and Y. Bengio, "Noisy activation functions," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 3059–3068.

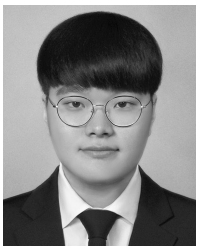
[33] Y. Tang and C. Eliasmith, "Deep networks for robust visual recognition," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 1055–1062.

[34] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," in *Proc. 21st Int. Conf. Mach. Learn. (ICML)*, 2004, p. 116.
[35] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.

SOOJIN JANG (Student Member, IEEE) received the B.S. degree in information and communication engineering from Sun Moon University, Asan, South Korea, in 2018. She is currently pursuing the M.S. degree with the Imaging Engineering Department, Graduate School of Advanced Imaging Science, Multimedia & Film, Chung-Ang University. Her research interests include deep learning and computer vision.



KYOHOON JIN (Student Member, IEEE) received the B.S. degree in applied statistics from Gachon University, Seongnam, South Korea, in 2019. He is currently pursuing the M.S. degree with the Imaging Engineering Department, Graduate School of Advanced Imaging Science, Multimedia & Film, Chung-Ang University. His research interests include deep learning and natural language processing.



JUNHYEOK AN (Student Member, IEEE) received the B.S. degree in computer media information engineering from Kangnam University, Yongin, South Korea, in 2019. He is currently pursuing the M.S. with the Imaging Engineering Department, Graduate School of Advanced Imaging Science, Multimedia & Film, Chung-Ang University. His research interests include deep learning and computer vision.

YOUNGBIN KIM (Member, IEEE) received the B.S. and M.S. degrees in computer science and the Ph.D. degree in visual information processing from Korea University, in 2010, 2012, and 2017, respectively. From August 2017 to February 2018, he has been a Principal Research Engineer at Linewalks. He is currently an Assistant Professor with the Graduate School of Advanced Imaging Science, Multimedia & Film, Chung-Ang University. His current research interests include data mining and deep learning.