Proceedings of Machine Learning Research 95:786-798, 2018 ACML 2018

RICAP: Random Image Cropping and Patching Data Augmentation for Deep CNNs

Ryo Takahashi TAKAHASHI@AI.CS.KOBE-U.AC.JP
Takashi Matsubara MATSUBARA@PHOENIX.KOBE-U.AC.JP
Kuniaki Uehara UEHARA@KOBE-U.AC.JP
Graduate School of System Informatics, Kobe University, 1-1 Rokko-dai, Nada, Kobe, Hyogo 657-8501, Japan

Editors: Jun Zhu and Ichiro Takeuchi

Abstract

Deep convolutional neural networks (CNNs) have demonstrated remarkable results in image recognition owing to their rich expression ability and numerous parameters. However, an excessive expression ability compared to the variety of training images often has a risk of overfitting. Data augmentation techniques have been proposed to address this problem as they enrich datasets by flipping, cropping, resizing, and color-translating images. They enable deep CNNs to achieve an impressive performance. In this study, we propose a new data augmentation technique called *random image cropping and patching* (*RICAP*), which randomly crops four images and patches them to construct a new training image. Hence, RICAP randomly picks up subsets of original features among the four images and discard others, enriching the variety of training images. Also, RICAP mixes the class labels of the four images and enjoys a benefit similar to label smoothing. We evaluated RICAP with current state-of-the-art CNNs (e.g., shake-shake regularization model) and achieved a new state-of-the-art test error of 2.23% on CIFAR-10 among competitive data augmentation techniques such as cutout and mixup. We also confirmed that deep CNNs with RICAP achieved better results on CIFAR-100 and ImageNet than those results obtained by other techniques.

Keywords: Data Augmentation, Image Classification, Convolutional Neural Network

1. Introduction

Deep convolutional neural networks (CNNs) (LeCun et al., 1989) have yielded significant achievements in image classification and image processing tasks thanks to their numerous parameters and rich expression ability (Zeiler and Fergus, 2014; Sermanet et al., 2014). However, CNNs with numerous parameters have a risk of overfitting because they learn detailed features of training images that do not generalize to others (Zeiler and Fergus, 2014; Zintgraf et al., 2017). Data augmentation has been used to address this problem (Krizhevsky et al., 2012; He et al., 2016a; DeVries and Taylor, 2017). Data augmentation increases the variety of images by manipulating them in several ways; flipping, resizing, random-cropping, and color-translating (He et al., 2016a). Dropout, proposed by Hinton et al. (2012), is another common data augmentation technique that injects noise into an image by dropping pixels. Differently from conventional data augmentation techniques, dropout could disturb and mask out the features in original images. Many recent

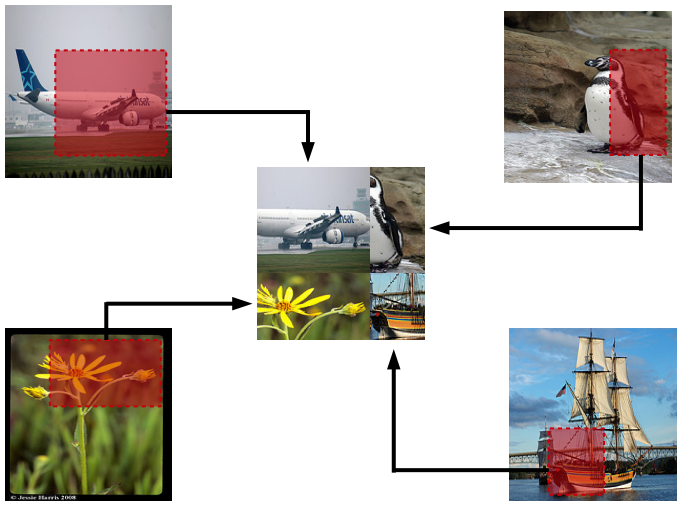

Figure 1: A conceptual explanation of the proposed *random image cropping and patching* (*RICAP*) data augmentation. We randomly crop four training images as denoted by the red shaded areas, and patch them to construct a new training image (at the center of this figure). The size of the final image is identical to that of the original one (e.g., 32×32 for the CIFAR dataset (Krizhevsky, 2009)). We collected the images in this figure from the training set of ImageNet dataset (Russakovsky et al., 2014).

studies have proposed new network architectures of CNN that have much more parameters (Zagoruyko and Komodakis, 2016; Huang et al., 2017; Han et al., 2017), and these traditional data augmentation techniques have become insufficient for such deeper CNNs.

Therefore, these days, data augmentation techniques attract rising attention (DeVries and Taylor, 2017; Zhong et al., 2017; Zhang et al., 2018). DeVries and Taylor (2017) proposed cutout, which randomly masks out a square region in an image at every training step and thus changes the apparent features. Cutout is an extension of dropout that can achieve better performance than it. Random erasing, proposed by Zhong et al. (2017), also masks out a subregion in an image like cutout. However, it randomly determines whether or not to mask out as well as the size and aspect ratio of the masked region. Mixup, proposed by Zhang et al. (2018), α-blends two images to form a new image, regularizing the CNN to favor a simple linear behavior in-between training images. Not limited to an increase in the variety of images, mixup also behaves like class label smoothing as it mixes the class labels of two images with the ratio $\alpha : 1 - \alpha$ (Szegedy et al., 2016). These new data augmentation techniques have been applied to modern deep CNNs and have broken the state-of-the-art records, demonstrating the importance of data augmentation.

In this study, as a further advance in data augmentation, we propose a novel method called *random image cropping and patching* (*RICAP*). RICAP crops four training images and patches them to construct a new training image; it selects the images and determines the cropping sizes randomly, where the size of the final image is identical to that of the original.

A conceptual explanation is shown in Fig. 1. RICAP also mixes class labels with ratios proportional to the areas of the four images like the label smoothing in mixup. Compared to mixup, RICAP has three clear distinctions; it mixes images spatially, it uses images partially by cropping, and it does not create features that absent from the original dataset except for patching boundary. We apply RICAP to existing deep CNNs and evaluate them on the CIFAR-10, CIFAR-100 (Krizhevsky, 2009), and ImageNet (Russakovsky et al., 2014) datasets. The experimental results demonstrate that RICAP outperforms the existing data augmentation techniques and achieves new state-of-the-art performances.

2. Related Works

RICAP is a novel data augmentation technique and can be applied to deep CNNs in the same manner as conventional techniques. In addition, RICAP is related to the class label smoothing technique. In this section, we explain about data augmentation and label smoothing as related works.

2.1. Data Augmentation

Data augmentation increases the variety of training samples and prevents overfitting. Krizhevsky et al. (2012) used random-cropping and horizontal-flipping for a deep CNN evaluated on the CIFAR dataset (Krizhevsky, 2009). Random-cropping prevents the overfitting to specific features by changing remarking points in an image. Horizontal-flipping doubles the variation in image with specific orientations, such as an airplane taken from one side. Krizhevsky et al. (2012) performed principal components analysis (PCA) on a set of RGB values to alter the intensities of the RGB channels for the evaluation on the ImageNet dataset (Russakovsky et al., 2014). This kind of color translation is useful for colorful objects, such as flowers. Facebook AI Research employed another method of color translation called color jitter for the reimplementaion of ResNet (He et al., 2016a) that is available at `https://github.com/facebook/fb.resnet.torch`. Color jitter manipulates the brightness, contrast, and saturation of an image instead of the RGB channels. These traditional data augmentation techniques play an important role in training deep CNNs. However, after He et al. (2016a) proposed ResNet, many studies proposed new network architectures (Zagoruyko and Komodakis, 2016; Huang et al., 2017; Han et al., 2017). The number of parameters is ever-growing, and the risk of overfitting is also ever-increasing. With this background, further data augmentation techniques attract much attention.

Dropout, proposed by Hinton et al. (2012), is a data augmentation that disturbs and masks out the original information of the given data by dropping pixels. Pixel-dropping can be considered an injection of noise into the image (Sietsma and Dow, 1991). It makes the CNN robust to noisy images and contributes to the generalization rather than enriching dataset.

Cutout randomly masks out a square region in an image at every training step (DeVries and Taylor, 2017). It is an extension of dropout, where the masking out of regions behaves as injected noise and makes the CNNs robust to noisy images. In addition to this, cutout can mask out the whole main part of an object in an image, such as the face of a cat. In this case, the CNNs need to learn other parts that are usually ignored, such as the tail of the cat in this case. This prevents deep CNNs from overfitting to features of the

main part of the object. In other words, cutout increases the variety of features by changing the apparent features at every training step. A similar method, random erasing, has been proposed by Zhong et al. (2017). It also masks out a certain area of an image, but it has clear differences; it randomly determines whether or not to mask out as well as the size and aspect ratio of the masked region.

Mixup α-blends two images to construct a new training image (Zhang et al., 2018). α-blending not only increases the variety of training images but also works like adversarial perturbation (Goodfellow et al., 2015). Mixup can train deep CNNs on convex combinations of pairs of training samples and their labels, and it enables deep CNNs to favor a simple liner behavior in-between training samples. This behavior makes the prediction confidence transit linearly from a class to another class, thus providing smoother estimation and margin maximization. Thereby, mixup makes deep CNNs robust to adversarial examples and stabilizes the training of generative adversarial networks. In addition, it behaves like class label smoothing by mixing of class labels with the ratio $\alpha : 1 - \alpha$ (Szegedy et al., 2016). We explain the label smoothing in detail below.

AutoAugment Cubuk et al. (2018) is a framework that exploring the best hyperparameters of existing data augmentations using reinforcement learning Zoph and Le (2017). It achieved the significant results on the CIFAR-10 classification and proved the importance of data augmentation for learning of deep CNN.

2.2. Label Smoothing

In classification tasks, class labels are often expressed as 0 and 1 probabilities. Deep CNNs commonly employ the softmax function, which never predicts an exact probability of 0 or 1. Thus, deep CNNs continue to learn ever-larger weight parameters and make an unjustly high confidence. Label smoothing sets the class probabilities to intermediate values, such as 0.9 and 0.8. It prevents the endless pursuit of the hard 0 and 1 probabilities for the estimated classes and enables the weight parameters to converge to certain values without discouraging correct classification (Szegedy et al., 2016). Mixup also mixes class labels of the α-blended images in the ratio $\alpha : 1 - \alpha$ and has a similar contribution to label smoothing (Zhang et al., 2018).

3. Methods

3.1. Random Image Cropping and Patching

In this paper, we propose a novel data augmentation technique called *random image cropping and patching* (*RICAP*) for deep convolutional neural networks (CNNs). The conceptual explanation of RICAP is shown in Fig. 1. It consists of three steps of data manipulation. First, we randomly select four images from the training set. Second, we crop the images separately. Third, we patch the cropped images to construct a new image and feed it to the CNNs. Despite this simple procedure, RICAP increases the variety of images drastically and prevents overfitting of deep CNNs having numerous parameters. We mix the class labels of the four images with the ratios proportional to the image areas. This label mixing works as label smoothing and prevents the endless pursuit of the hard 0 and 1 probabilities in deep CNNs using the softmax function.

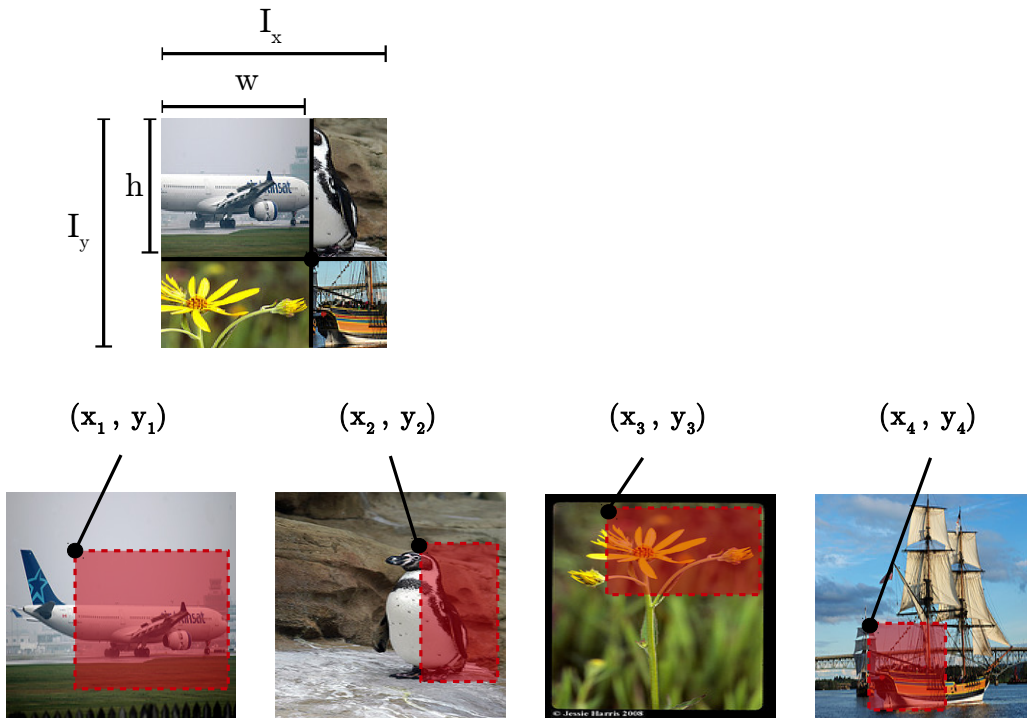

Figure 2: Detailed explanation of RICAP. I_x and I_y are the width and height of the image. We randomly crop four images, shown by the area shaded in red, and patch them based on the boundary position (w, h). We select (w, h) based on methods explained in detail in subsection 3.2. Based on the value of (w, h), we select (x_i, y_i) such that it does not increase image size.

A specific explanation of the implementation is shown in Fig. 2. We randomly select four images $k \in \{1, 2, 3, 4\}$ from the training set and patch them on the upper left, upper right, lower left, and lower right sides. I_x and I_y denote the width and height of the training image, respectively. We draw the boundary position (w, h) of the four images k from an uniform distribution; we explain the optimization of the distributions in the next subsection. We then automatically obtain the cropping sizes (w_k, h_k) of the images k, i.e., $w_1 = w_3 = w$, $w_2 = w_4 = I_x - w$, $h_1 = h_2 = h$, and $h_3 = h_4 = I_y - h$. For cropping the four images k following the sizes (w_k, h_k), we randomly determine the coordinates (x_k, y_k) of the upper left corners of the cropped areas as $x_k \sim \mathcal{U}(0, I_x - w_k)$ and $y_k \sim \mathcal{U}(0, I_y - h_k)$. Finally, we define the target label c by mixing one-hot coded class labels c_k of the four patched images with ratios W_i proportional to their areas in the new constructed image;

$$c = \sum_{k \in \{1,2,3,4\}} W_k c_k \quad \text{for} \quad W_k = \frac{w_k h_k}{I_x I_y}, \tag{1}$$

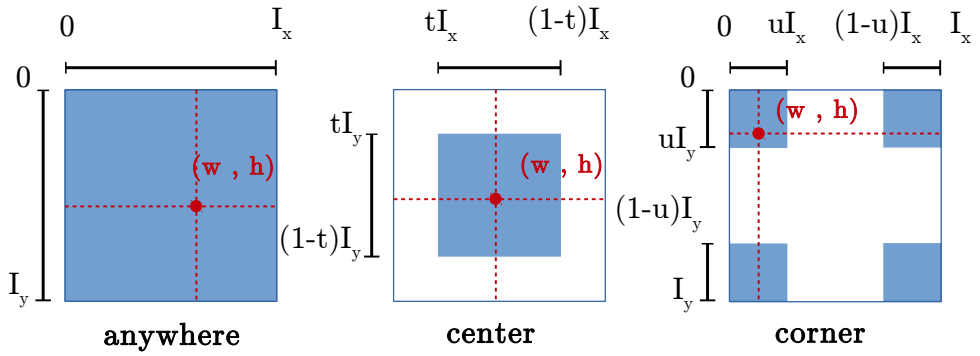

Figure 3: Three variants of the distribution of the boundary position (w, h). (left panel) The most trivial case called *anywhere-RICAP*. The boundary position (w, h) is selected from all possible values. (middle panel) *center-RICAP* narrows the range of boundary position (w, h) using the parameter t. (right panel) *corner-RICAP* restricts the boundary position (w, h) within ranges close to the four corners.

where $w_k h_k$ is the area of the cropped image k and $I_x I_y$ is the area of the original image.

3.2. Optimization of RICAP

In this section, we describe the optimization of the distribution of the boundary position (w, h). We first introduce the simplest method, in which we select the boundary position (w, h) from the range of all possible values $[0, I_x]$ and $[0, I_y]$, as shown in the left panel of Fig. 3. The distribution of the boundary position (w, h) is

$$w \sim \mathcal{U}(0, I_x),$$
$$h \sim \mathcal{U}(0, I_y).$$

We call this variant *anywhere-RICAP*, hereafter. Moreover, we introduce two other variants because we can move the range of boundary position. First, we restrict the boundary position (w, h) to the center of the patched image as shown in the middle panel of Fig. 3. The distribution of the boundary position (w, u) is

$$w \sim \mathcal{U}(tI_x, (1-t)I_x),$$
$$h \sim \mathcal{U}(tI_y, (1-t)I_y),$$
$$t \in [0, 0.5],$$

where the parameter value $t = 0.0$ denotes the same range as that of the anywhere-RICAP and a larger parameter value t restricts the boundary position (w, h) within a narrower range. We call this variant *center-RICAP*, hereafter. However, using *center-RICAP* with $t > 0.0$, the target class probabilities c never get the value of 1.0 but often have values close to 0.25. This has a risk of excessive label smoothing and discourages correct classification. We introduce a variant with the opposite tendency. We restrict the boundary position (w, h)

within ranges close to the four corners as shown in the right panel of Fig. 3. Specifically, the distribution of the boundary position (w, u) is

$$
\begin{aligned}
w &\sim \frac{1}{2}\left(\mathcal{U}(0, uI_x) + \mathcal{U}((1-u)I_x, I_x)\right), \\
h &\sim \frac{1}{2}\left(\mathcal{U}(0, uI_y) + \mathcal{U}((1-u)I_y, I_y)\right), \\
u &\in [0, 0.5],
\end{aligned}
$$

where the parameter value $u = 0.5$ denotes the same range as the anywhere-RICAP, a smaller parameter value u restricts the boundary position (w, h) within ranges close to the four corners, and the parameter value $u = 0.0$ indicates the case without RICAP. We call this variant *corner-RICAP*, hereafter.

4. Experiments and Results

To evaluate the performance of RICAP, we apply it to deep CNNs and evaluate them on the CIFAR-10, CIFAR-100, and ImageNet datasets.

4.1. Classification of CIFAR-10 and CIFAR-100

In this section, we apply RICAP to an existing deep CNN and evaluate it on the CIFAR-10 and CIFAR-100 datasets (Krizhevsky, 2009). CIFAR-10 and CIFAR-100 consist of 32×32 RGB images of objects in natural scenes. 50,000 images are used for training and 10,000 for evaluation. Each image is manually given one of 10 class labels in CIFAR-10 and one of 100 in CIFAR-100. The number of images per class is thus reduced in CIFAR-100. According to previous studies (Lee et al., 2015; Romero et al., 2015; Springenberg et al., 2015), we normalized each channel of all images to zero mean and unit variance as preprocessing. We also employed 4-pixel padding on each side, 32×32 random cropping, and random-flipping in the horizontal direction as conventional data augmentation techniques.

We used a residual network called WideResNet proposed by Zagoruyko and Komodakis (2016). We used architecture called *WideResNet 28-10*, which consists of 28 convolution layers with a widen factor of 10 and employs dropout of a drop probability of $p = 0.3$ in the intermediate layers. This architecture achieved the highest accuracy on the CIFAR datasets in Zagoruyko and Komodakis (2016). The hyperparameters were set to the same as those used in the original study. Batch normalization (Ioffe and Szegedy, 2015) and the ReLU activation function (Nair and Hinton, 2010) were used. The weight parameters were initialized following the algorithm proposed by He et al. (2016b). The weight parameters were updated using the momentum SGD algorithm with a momentum parameter of 0.9 and weight decay of 10^{-4} over 200 epochs with batches of 128 images. The learning rate was initialized to 0.1, and then, it was reduced to 0.02, 0.004 and 0.0008 at the 60th, 120th and 160th epochs, respectively.

We evaluated RICAP with the WideResNet 28-10 to explore the best variant among *anywhere-RICAP*, *center-RICAP* and *corner-RICAP*, and the best hyperparameter t and u. I_x and I_y were 32 for the CIFAR datasets. Fig. 4 shows the results on CIFAR-10 and CIFAR-100. The baselines denote the WideResNet 28-10 without RICAP. Anywhere-RICAP, which does not have a hyperparameter, obtained better test error rates than the baselines on both

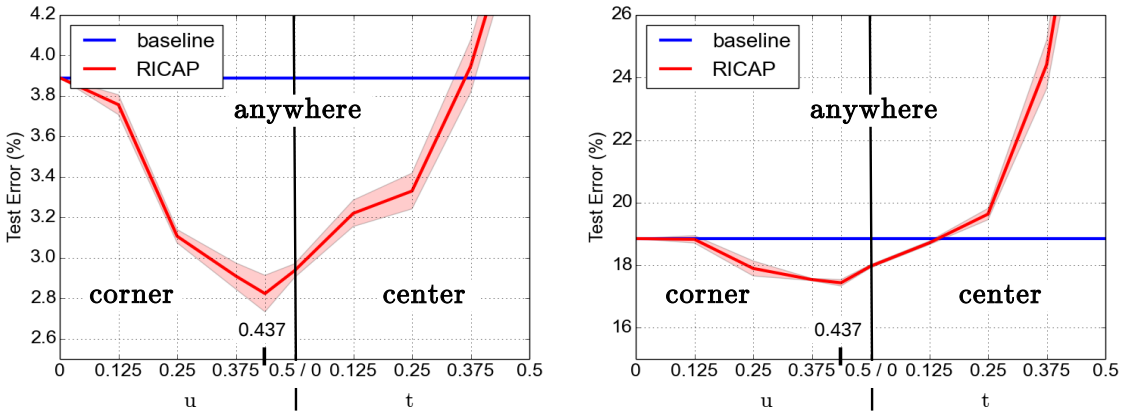

Figure 4: The optimization of RICAP using the WideResNet 28-10. We arranged *corner-RICAP* using the hyperparameter u, *anywhere-RICAP*, and *center-RICAP* with hyperparameter t along one axis. (left panel) The test error rate on CIFAR-10 and (right panel) that of CIFAR-100. We performed three runs and depicted the means and standard deviations using the red lines and shaded areas, respectively. Baseline indicates the results of the WideResNet 28-10 without RICAP.

datasets. Corner-RICAP obtained better test error rates than the baselines over the entire range of the value of hyperparameter u and the best test error rates with $u = 0.437$ on both datasets. Center-RICAP obtained better test error rates than the baselines only with small values of u. Results of large values of u demonstrated a negative influence of excessive label smoothing. We also summarized the results of RICAP in Table 1 as well as the results of competitive methods; dropout (Hinton et al., 2012), cutout (DeVries and Taylor, 2017), random erasing (Zhong et al., 2017), and mixup (Zhang et al., 2018). Competitive results denoted by the † symbols were obtained from our experiments and the other results were cited from the original studies. In our experiments, each value following the ± symbol is the standard deviation over three runs. Recall that WideResNet 28-10 usually employs dropout in intermediate layers. For dropout, we added dropout to the input layer as data augmentation for comparison. Drop probability was set to $p = 0.2$ according to Hinton et al. (2012). For other competitive methods, we set the hyperparameters to the values with witch the CNNs achieved the best result in each study; cutout size 16×16 (CIFAR-10) and 8×8 (CIFAR-100) for cutout, and $\alpha = 1.0$ for mixup. Anywhere-RICAP achieved test error rates superior to or comparable to the competitive methods, and Corner-RICAP clearly outperformed them. Corner-RICAP is expected to be the best variant because it can acquire the benefit of cropping and patching of RICAP without harms by the extreme label smoothing.

Table 1: Test Error Rates using WideResNet.

Method	CIFAR-10	CIFAR-100
Baseline	3.89	18.85
+ dropout (input)	4.69†	21.54†
+ cutout	3.08	18.41
+ random erasing	3.08 ±0.05	17.73 ±0.15
+ mixup	3.02 ±0.04†	17.62 ±0.25†
+ anywhere-RICAP	**2.94** ±0.03	17.97 ±0.05
+ corner-RICAP ($u = 0.437$)	**2.82** ±0.09	**17.44** ±0.10

† indicates the results of our experiments.

Table 2: Test Error Rates on CIFAR-10.

Method	Pyramidal ResNet 272-200	ShakeShake 26 2x96d
Baseline	3.31 ±0.08	2.86
+ dropout (input)	4.06†	3.79†
+ cutout	2.82†	2.56
+ mixup	2.61†	2.48†
+ anywhere-RICAP	**2.61** ±0.06	**2.42** ±0.13
+ corner-RICAP ($u = 0.437$)	**2.59** ±0.05	**2.23** ±0.07

† indicates the results of our experiments.

4.2. Classification by Other Architectures

We also evaluated RICAP with the pyramidal ResNet, proposed by Han et al. (2017), and the shake-shake regularization model, proposed by Gastaldi (2017). For the pyramidal ResNet, we used the architecture called *Pyramidal ResNet 272-200*; as the name implies, it consists of 272 convolution layers using bottleneck residual blocks with a widening factor of $\alpha = 200$. For the shake-shake regularization model, we used the architecture called *ShakeShake 26 2×96d*; that is a ResNet of 26 convolution layers and 2×96d channels with Shake-Shake-Image regularization. Each architecture had achieved the highest test accuracies in the corresponding paper. Data normalization and data augmentation were performed in the same way as for the Sec. 4.1. The hyperparameters were the same as in the original studies by Han et al. (2017) and Gastaldi (2017).

We also summarized the results in Table 2. Anywhere-RICAP outperformed the competitive methods and Corner-RICAP achieved even better results. In particular, the shake-shake regularization model with *corner-RICAP* ($u = 0.437$) achieved a test error rate of 2.23%. This is a new state-of-the-art result on the CIFAR-10. These results also indicate that RICAP is applicable to various CNN architectures and the choice of appropriate hyperparameter depends on the dataset but not on the CNN architectures.

4.3. Classification of ImageNet

In this section, we evaluate RICAP on the ImageNet dataset (Russakovsky et al., 2014). ImageNet consists of 1.28 million training images and 50,000 validation images. Each image is given one of 1,000 class labels. We normalized each channel of all images to the zero mean

Table 3: Single Crop Test Error Rates on ImageNet using WideResNet-50-2-bottleneck.

Network	Epochs	top-1 Error(%)	top-5 Error(%)
Baseline	100	21.90	6.03
+ cutout	100	22.45†	6.22†
+ mixup	100	21.83†	**5.81**†
+ anywhere-RICAP	100	**21.70**	5.83
Baseline	200	22.88†	6.61†
+ anywhere-RICAP	200	**21.38**	**5.89**

† indicates the results of our experiments.

and unit variance as preprocessing. We also employed random-resizing, random 224×224 cropping, color jitter, lighting, and random-flipping in the horizontal direction following previous studies (Zagoruyko and Komodakis, 2016; Zhang et al., 2018).

To evaluate RICAP, we apply it to the architecture called *WideResNet 50-2-bottleneck*, consisting of 50 convolution layers using bottleneck residual blocks with a widen factor of 2 and dropout with a drop probability of $p = 0.3$ (Zagoruyko and Komodakis, 2016). This architecture had achieved the highest accuracy on ImageNet in Zagoruyko and Komodakis (2016). The hyperparameters and other conditions were the same as those used in the baseline study. WideResNet 50-2-bottleneck was trained using the momentum SGD algorithm with a momentum parameter of 0.9 and weight decay of 10^{-4} over 100 or 200 epochs with batches of 256 images. The learning rate was initialized to 0.1, and then, it was reduced to 0.01, 0.001 and 0.0001 at the 30th, 60th, and 90th epochs in the case of 100 epoch training. The learning rate was reduced at the 65th, 130th, and 190th epochs in the case of 200 epoch training.

Table 3 summarizes the results of RICAP with WideResNet 50-2-bottleneck as well as the results of competitive methods: cutout DeVries and Taylor (2017) and mixup Zhang et al. (2018). Competitive results denoted by † symbols are obtained from our experiments and the other results are cited from the original studies. For the competitive methods, we set the hyperparameters to specific values according to each study: cutout size 56×56 for cutout, and $\alpha = 0.2$ for mixup. We did not optimize the hyperparameter of RICAP but used *anywhere-RICAP* because of limited computational resources. Even without hyperparameter tuning, *anywhere-RICAP* reduced the test error rates. About the competitive methods, only mixup can reduce the test error rates and outperformed the RICAP only the case of top-5 error. In the case of 200 epochs training without *anywhere-RICAP*, WideResNet achieved worse results than in the case of 100 epochs training. Deep CNNs including WideResNet sometimes obtained worse results through a longer training, as reported by Zagoruyko and Komodakis (2016). In contrast, WideResNet with *anywhere-RICAP* improved the top-1 test error rate by 200 epochs training. This result indicates that RICAP prevents deep CNNs from overfitting or other harmful effects in a longer training.

Cutout did not attempt to classify ImageNet in DeVries and Taylor (2017) and requires further hyperparameter adjustment. Mixup classified ImageNet in Zhang et al. (2018) using other CNNs but the appropriate hyperparameter was considerably different from those for the CIFAR datasets. On the other hand, *anywhere-RICAP* worked well for all CIFAR datasets and ImageNet even though *corner-RICAP* could achieve better results.

5. Conclusion

In this study, we proposed a novel data augmentation method called *random image cropping and patching* (*RICAP*) to improve the accuracy of the classification of images. RICAP selects four training images randomly, crops them randomly, and patches them to construct a new training image. Experimental results demonstrated that RICAP improves classification accuracy of various datasets by increasing the variety of training images and preventing overfitting. Future works include a more detailed evaluation by applying it to other tasks, such as image-caption retrieval.

Acknowledgments

This study was partially supported by the MIC/SCOPE #172107101.

References

Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. AutoAugment: Learning Augmentation Policies from Data. *arXiv*, pages 1–14, 2018.

Terrance DeVries and Graham W. Taylor. Improved Regularization of Convolutional Neural Networks with Cutout. *arXiv*, pages 1–8, 2017.

Xavier Gastaldi. Shake-Shake regularization. In *ICLR Workshop*, pages 1–10, 2017.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In *Proc. of International Conference on Learning Representations (ICLR2015)*, pages 1–11, 2015.

Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep Pyramidal Residual Networks. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2017)*, pages 6307–6315, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2016)*, pages 770–778, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. of the IEEE International Conference on Computer Vision (ICCV2016)*, volume 11-18-Dece, pages 1026–1034, 2016b.

G E Hinton, N Srivastava, A Krizhevsky, I Sutskever, and R R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv*, pages 1–18, 2012.

Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely Connected Convolutional Networks. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2017)*, pages 2261–2269, 2017.

Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proc. of the 32th International Conference on Machine Learning (ICML2015)*, pages 448–456, 2015.

Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. *Technical report, University of Toronto*, pages 1–60, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems (NIPS2012)*, pages 1097–1105, 2012.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551, 1989.

C Y Lee, S Xie, P Gallagher, Z Zhang, and Z Tu. Deeply-supervised nets. In *Proc. of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS2015)*, volume 2, pages 562–570, 2015.

Vinod Nair and Geoffrey E Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proc. of the 27th International Conference on Machine Learning (ICML2010)*, number 3, pages 807–814, 2010.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for Thin Deep Nets. In *Proc. of International Conference on Learning Representations (ICLR2015)*, pages 1–13, 2015.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, and C V Jan. ImageNet Large Scale Visual Recognition Challenge. *arXiv*, pages 1–43, 2014.

Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. In *Proc. of International Conference on Learning Representations (ICLR2014)*, pages 1–16, 2014.

Jocelyn Sietsma and Robert J.F. Dow. Creating artificial neural networks that generalize. *Neural Networks*, 4(1):67–79, 1991.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for Simplicity: The All Convolutional Net. In *Proc. of International Conference on Learning Representations (ICLR2015)*, pages 1–14, 2015.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Z B Wojna. Rethinking the Inception Architecture for Computer Vision. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2016)*, pages 2818–2826, 2016.

Sergey Zagoruyko and Nikos Komodakis. Wide Residual Networks. *Proc. of the British Machine Vision Conference (BMVC2016)*, pages 87.1–87.12, 2016.

Matthew D. Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In *Proc. of European Conference on Computer Vision (ECCV2014)*, pages 818–833, 2014.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond Empirical Risk Minimization. In *Proc. of International Conference on Learning Representations (ICLR2018)*, pages 1–13, 2018.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random Erasing Data Augmentation. *arXiv*, pages 1–10, 2017.

Luisa M Zintgraf, Taco S Cohen, Tameem Adel, and Max Welling. Visualizing Deep Neural Network Decisions: Prediction Difference Analysis. *Proc. of International Conference on Learning Representations (ICLR2017)*, pages 1–12, 2017.

Barret Zoph and Quoc V. Le. Neural Architecture Search with Reinforcement Learning. In *Proc. of International Conference on Learning Representations (ICLR2017)*, pages 1–16, 2017.