Journal of Electronic Research and Application, 2023, Volume 7, Issue 3
http://ojs.bbwpublisher.com/index.php/JERA
ISSN Online: 2208-3510
ISSN Print: 2208-3502

Folding Fan Cropping and Splicing (FFCS) Data Augmentation

Fengyu Yang*

School of Information Science and Technology, Tibet University, Lhasa 850000, China

***Corresponding author:** Fengyu Yang, 1974761685@qq.com

Abstract: Convolutional neural networks (CNNs) are widely used to tackling complex tasks, which are prone to overfitting if the datasets are noisy. Therefore, we propose folding fan cropping and splicing (FFCS) regularization strategy to enhance representation abilities of CNNs. In particular, we propose two different methods considering the effect of different segmentation numbers on classification results. One is the random folding fan method, and the other is the fixed folding fan method. Experimental results showed that FFCS reduced the classification errors both with the value of 0.88% on CIFAR-10 dataset and 1.86% on ImageNet dataset. Moreover, FFCS consistently outperformed Mixup and Random Erasing approaches on classification tasks. Therefore, FFCS effectively prevents overfitting and reduces the impact of background noises on classification tasks.

Keywords: Image processing; Data augmentation; Regularization

Online publication: May 9, 2023

1. Introduction

Convolutional neural networks (CNNs) [1] have enabled breakthroughs in numerous complex learning tasks thanks to their numerous parameters and rich expression ability. However, CNNS are at risk of overfitting because their complexity does not match the number of training sets. Therefore, many regularization methods came into being, including data augmentation [2], L1 regularization [3], L2 regularization [4], Dropout [5], DropConnect [6], and early stopping [7].

Traditional data augmentation methods include self-supervised data augmentation [8,9] and unsupervised data augmentation [10,11]. Self-supervised data augmentation employs preset data transformation rules to augment data on the basis of existing data. Cutout [12] randomly crops a part from the picture. In other words, Cutout randomly selects an area to fill with 0 pixels. Mixup [13] mixes two images proportionally. Hide-and-Seek [14] divides images into S * S grids, and each grid is masked with a probability of 0.5. CutMix [15] randomly selects an area to fill with pixels from other images. The difference between CutMix and Cutout is the pixel value of the filled area. The difference between CutMix and Mixup is the way of mixing. Random Erasing [16] randomly erases an area to fill with random pixels. Random Erasing is between Cutout and CutMix. Random Erasing neither fills 0 pixels nor fills the pixel values of another image. Unlike the previous methods, the GridMask [17] mask area is not random. GridMask determines the mask by setting the side length of each small square and the distance between the two masks. FenceMask [18] is an improvement of GridMask. We proposed a better shape because we believe that using a square mask will have a significant on small objects. AugMix [19] randomly selects several transformations, and then fuses the transformed images together in proportion. The main idea of KeepAugment [20] is

analyzing the least important area for Cutout by obtaining the Saliency map or analyzing the most important area for CutMix.

Compared to self-supervised data augmentation methods, unsupervised data augmentation is more advanced. Cycle Generative Adversarial Network (CycleGAN) [21] uses Generative Adversarial Network (GAN) [22] to generate a mapping relationship between input picture and the enhanced image. Automatic data augmentation works by constructing a search space that contains as many data augmentation methods as possible. When using different networks, the network looks in the search space to find the best algorithm for networks [23-26].

However, these methods do not reduce the impact of background noise on image classification. In this study, we propose a novel method called folding fan cropping and splicing (FFCS). FFCS can effectively prevent overfitting by cutting and splicing of folding fan. Although cutting and stitching is a very simple operation, sometimes the simplest operation can bring us unexpected results. In the case of large background noise, our proposed FFCS can well reduce the noise of background factors. In conclusion, FFCS not only significantly prevents overfitting, but also reduces the impacts of background noises on image classification.

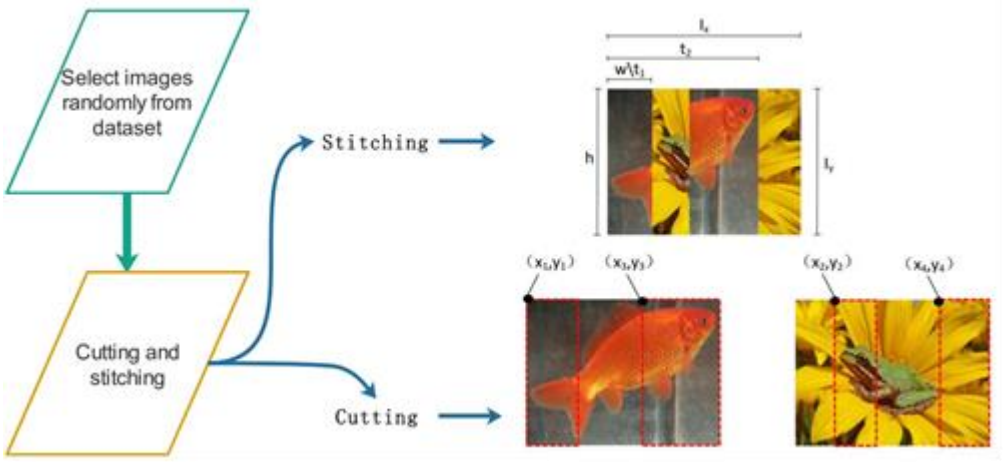

Figure 1. Detailed graphical explanation of R4

2. Our proposed method

In this paper, we propose FFCS for feature fusion extraction tasks with high noises. FFCS is divided into random folding fan method and fixed folding fan method. The random folding fan method means that the size of the outlet is random. The fixed folding fan method means that the size of the outlet is fixed. The number of files is divided into one, two, three, four, five and six. According to the size of the outlet, FFCS was mainly divided into R1–R6, F1–F6. In this section, we take R4 and F4 as examples to explain the operation. We blend the labels of the two pictures in proportion to the area of the picture to achieve the function of label smoothing [27].

2.1. Random folding fan method

R4 included three steps. Firstly, we randomly selected two pictures from the training set. Secondly, we tailored two parts from pictures solely. Thirdly, we joined the cut pieces to a complete picture and input it into the network. R4 means that the size of the outlet is random, and the number of files is 4. In general, the sizes of the cuts were random, and the number of cuts was 4. We randomly selected two pictures from the training ing set and crop two regions from each image. In this way, we cropped out four regions in total. We alternately stitched the four regions together, so that regions from the same image will not be stitched

together. The specific details can be seen in **Figure 1**. To avoid the overlapping of two areas on the same image, we limited the range of parameters settings.

I_x and I_y are breadth and altitude of the input. x_y and y_k are the positions of the upside left horn of the *k*th cropping region. w_k, h_k are the breadth and altitude of the *k*th cut piece. x_k is an integer in the range of 0 to $I_x - w_k + 1$. y_k is set to zero. Limiting the value range of x_k prevents the cropping region from exceeding the range of the original images. Adding one makes x_k take the boundary value $I_x - w_k$. Two variables t_1 and t_2 are involved here. t_1 is an integer within the scope of 0 to $0.5 * I_x$. t_2 is an integer within the scope of $0.5 * I_x$ to I_x. The specific formulas are as follows.

$$t_1 \in \{(0,0.5 * I_x), t_1 \in \mathbb{Z}\} \quad (1)$$
$$t_2 \in \{(0.5 * I_x, I_x), t_2 \in \mathbb{Z}\} \quad (2)$$
$$w_1 = t_1 \quad (3)$$
$$w_2 = 0.5 * I_x - t_1 \quad (4)$$
$$w_3 = t_2 - 0.5 * I_x \quad (5)$$
$$w_4 = I_x - t_2 \quad (6)$$
$$h_1 = h_2 = h_3 = h_4 = I_y \quad (7)$$
$$x_k \in \{(0, I_x - w_k + 1), x_k \in \mathbb{Z}\} \quad (8)$$
$$y_k = 0 \quad (9)$$

The labels of the images were allocated according to the proportion of the cropped area to the previous pictures' area. The widths and heights of the four cropping regions can be obtained from the equation (1) to (9). d_k is the proportion of the *k*th clipping acreage to the total acreage. c_1 is the ratio of the cropped regions on the first image to the total acreage. c_2 is the proportion of the two cropped regions in the second image to the total area. The specific label allocation formulas are as follows:

$$d_k = \frac{w_k * h_k}{I_x * I_y} \quad (10)$$
$$c_1 = d_1 + d_3, c_2 = d_2 + d_4 \quad (11)$$

Table 1. Experimental results on two datasets

Method	CIFAR-10		ImageNet	
	Test error	**Test loss**	**Test error**	**Test loss**
R1/F1	3.63%	0.15	10.52%	0.39
R2	**2.75%**	**0.11**	**8.66%**	**0.33**
F2	4.16%	0.16	10.38%	0.43
R3	3.20%	0.14	9.53%	0.40
F3	4.69%	0.16	14.34%	0.65
R4	4.15%	0.15	10.95%	0.47
F4	4.83%	0.17	14.59%	0.59
R5	3.89%	0.15	15.78%	0.80
F5	4.72%	0.16	14.70%	0.59
R6	3.70%	0.15	11.74%	0.61
F6	5.52%	0.18	18.14%	0.67

2.2. Fixed folding fan method

F4 includes three steps. Firstly, we randomly selected two pictures from the training ing set. Secondly, we tailored two pieces from the picture solely. Thirdly, we joined the cut pieces to a complete picture. F4

means that the size of the outlet is fixed, and the number of files is 4. Generally, the size of the cuts was fixed, and the number of cuts was 4. We randomly selected two pictures from the training set and cropped two pieces from each image. In this way, we cropped out four regions in total. The sizes of the four regions cropped out are the same. We alternately stitched the four regions together, so regions from the same image will not be stitched together. The specific details can be seen in **Figure 2**. We adjusted the parameters settings to avoid the overlapping of two areas on the same image.

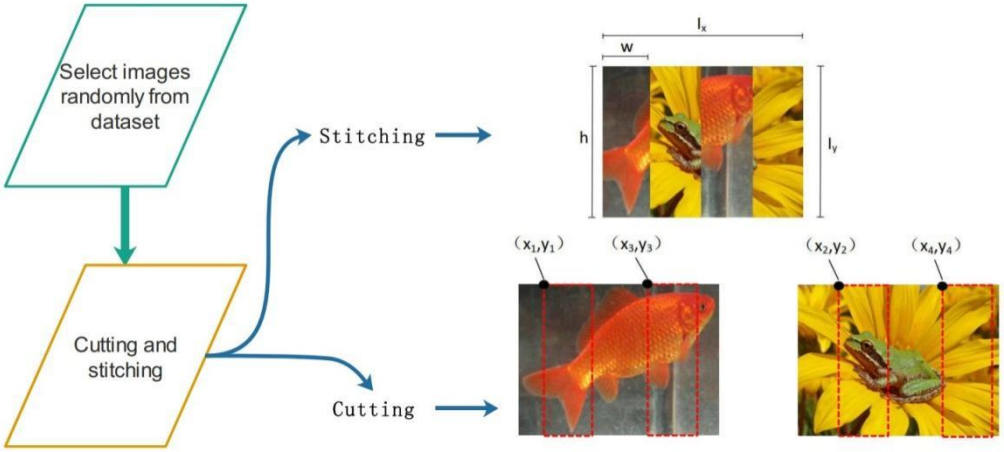

Figure 2. Detailed graphical explanation of F4

I_x and I_y are the widths and heights of the input pictures. x_k, y_k are the positions of the upside left horn of the *k*th cropping region. w_k and h_k are the width and height of the *k*th cropping region. x_k is an integer in the range of 0 to $I_x - w_k + 1$. y_k is set to zero. Limiting the value range of x_k prevents the cropping region from exceeding the range of the original image. Adding one makes x_k take the boundary value $I_x - w_k$. Fixed-size means each of cropping areas has the same width and height. The width of the cropping area is a quarter of the original image. The height of the cropping area is the same as the original image. The specific formulas are as follows:

$$w_1 = w_2 = w_3 = w_4 = 0.25 * I_x \quad (12)$$
$$h_1 = h_2 = h_3 = h_4 = I_y \quad (13)$$
$$x_k \in \{(0, I_x - w_k + 1), x_k \in \mathbb{Z}\} \quad (14)$$
$$y_k = 0 \quad (15)$$

The labels of the images were allocated according to the proportion of the cropping areas to the original image areas. The four cropped regions of F4 had the same area. The label allocation ratio formulas were calculated as follows. Therefore, the labels were allocated according to the ratio of 1:1. d_k is the proportion of the *k*th clipping acreage to the general acreage. c_1 is proportion of the cropped pieces on the first image to the total acreage. c_2 is the proportion of the two cropped regions in the second image to the total acreage. The specific label allocation formulas are as follows.

$$d_k = \frac{w_k * h_k}{I_x * I_y} \quad (16)$$
$$c_1 = d_1 + d_3 = 0.5, c_2 = d_2 + d_4 = 0.5 \quad (17)$$

3. Experimental results

Table 2. Comparison results

Method	CIFAR-10		ImageNet	
	Test error	Test loss	Test error	Test loss
Dropout	3.63%	0.15	10.52%	0.39
Dropout + R2 (Proposed method)	**2.75%**	**0.11**	**8.66%**	**0.33**
Dropout + Random Erasing [16]	3.18%	0.11	11.31%	0.51
Dropout + Mixup [13]	3.02%	0.16	10.07%	0.40

3.1. Experiments on CIFAR-10

We put FFCS into WideResNet 28-10 and assessed it. The dataset had a total of 10 categories. Each pixel contained 3 RGB values ranging from 0 to 255. We conducted experiments on CIFAR-10 to test the effect of FFCS on small datasets with low pixel values.

We used a residual network called WideResNet, also known as [28] WRN, which is wider than ResNet [29]. WideResNet 28-10 means the depth of the network is 28 and the width is 10. We normalized each channel to 0 and standardized the variance. We boasted 4 pixel filling on every way. Besides, we boasted stochastic tailor and stochastic overturn in a longitudinal direction [30-32]. The weight arguments were initialized according to He's article [33]. The weight update algorithm and learning rate were based on RICAP [27].

There are some details about the experimental results that we need to pay attention to. First of all, the backbone of the two datasets was different. More importantly, R1/F1 in **Table 1** is the result of our backbone network output, which means that R1/F1 is the method of original image input. R and F stand for random cropping and fixed cropping. The numbers following the letters represent the number of splices. Based on **Table 1,** there were improvements compared to the original results in R2 and R3 on CIFAR-10, with the best performer being R2. R2 and F2 were dichotomous; however, R2 worked much better than F2. The label allocation ratio of 0.5 is the minimum point. It was F2 when the label allocation ratio was 0.5. The last two points were misclassified due to the background factor. Besides, the classification accuracy of F2 is lower than that of all R2's. We also found that random cropping performed better than fixed cropping under the same conditions. Below is a sample illustration and caption for a multimedia file.

3.2. Experiments on ImageNet

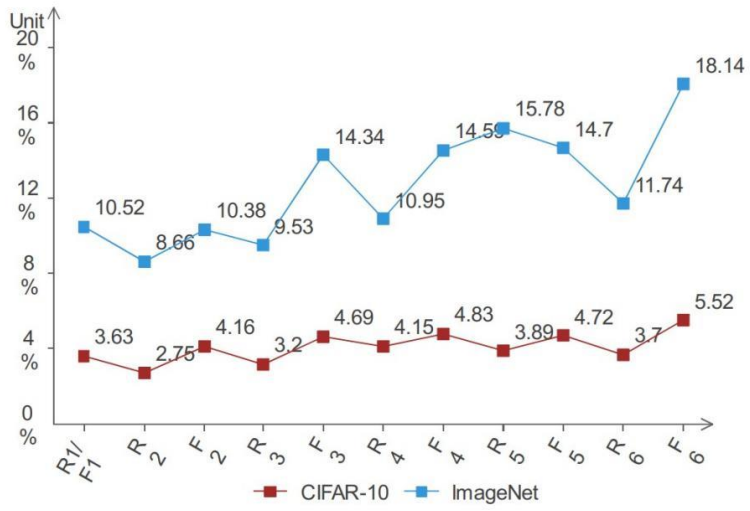

Figure 3. Test error on two datasets

We put FFCS into WideResNet-50-2 and assessed it. The ImageNet is a well-known dataset and it covers most picture categories, which comes up to a thousand categories [34]. In tis study, we did not use all of ImageNet due to the large number of classes in its dataset. Therefore, we randomly selected 40 classes in this dataset as our dataset, which not only reduces the training time, but also improves the transfer learning ability of the networks. The ImageNet contains 1300 images per class. We divided the dataset according to the ratio of 5:1, which meant that each class had 1085 pictures for drilling and 215 images for measurement. In total, the training set contained 43280 pictures and the test set contained 8600 pictures.

WideResNet-50-2 means the depth of the network is 50 and the width is 2. The lot size was set to 64, each channel was set to 0, and the variance was standardized. We also boasted stochastic tailoring, hue dithering, illumination, and stochastic overturn in the longitudinal direction [13,28]. The weight arguments were initialized according to He's article [33]. The weight update algorithm and learning rate were based on RICAP.

3.3. Result analysis

Experiments were conducted on FFCS, which were explained in Section **3.1.** and **3.2.** In this section, we compared FFCS with other methods. In these experiments, R2 was compared with other methods, and R2 performed well on both CIFAR-10 and ImageNet, with an accuracy improvement of 0.88% and 1.86%. It is obvious from **Table 2** that FFCS is performed the best compared to the other two methods. To put it clearly, we also plotted the variation curve of the test error rate for different datasets under different methods in **Figure 3**.

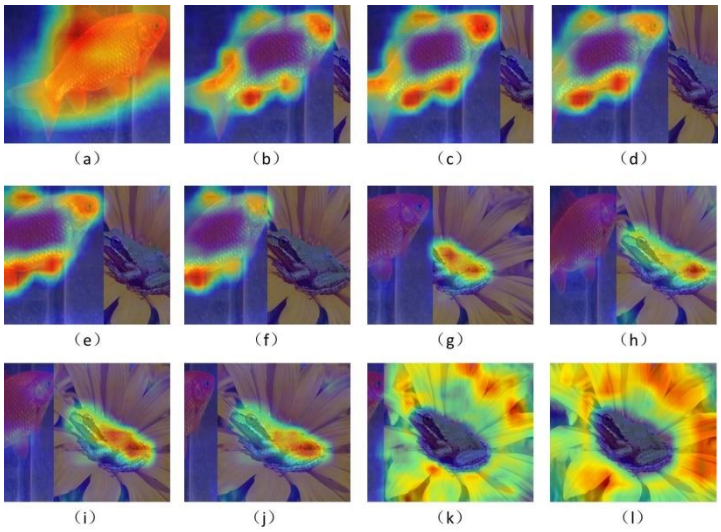

Figure 4. CAM of the images

To evaluate FFCS, we adopted a technology named class activation map (CAM). CAM generates a heatmap of class activations on input images. CAM determines the part of the image that has the greatest impact on the final classification result. The redder the area, the higher the impact on the classification results, the higher the precision of the cropped images. We identified the problem by comparing the differences of CAM. Taking R2 method as example, in **Figure 4**, different stitched images have different heatmaps. **Figure 4** consists of 12 small heatmaps which represents different proportions of the original image to the composite image. The ratio was 0 to 1 in steps of 0.1. When pictures were sent to mesh immediately, their classification results were easily affected by background noises, as shown in **Figure 4(a)**. In the original images, the network focused more on the patterns. The areas emphasized are very different for pictures with different pixel ratios. In conclusion, different splicing methods have a great

influence on the CAM. In conclusion, the FFCS in this paper is flexible and reduces the influence of background noise on the results to a certain extent.

4. Conclusion

In this study, we proposed a regularization strategy called folding fan cropping and splicing (FFCS) for images classification tasks. Our proposed FFCS not only prevents overfitting, but also reduces the influence of background noises on classification tasks. Besides, FFCS can also perform class label smoothing. FFCS has performed well on both CIFAR-10 and ImageNet, with an accuracy improvement of 0.88% and 1.86%. Meanwhile, the experimental results showed that our FFCS is not inferior to other methods. We may focus on more tailoring methods and selection optimization methods in our future works.

Disclosure statement

The author declares no conflict of interest.

References

[1] LeCun Y, Bottou L, Bengio Y, et al., 2021, Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE, 2278–2324.

[2] Steiner A, Kolesnikov A, Zhai X, et al., 2021, How to Train Your ViT? Data, Augmentation, and Regularization in Vision Transformers. Transactions on Machine Learning Research. ArXiv. https://doi.org/10.48550/arXiv.2106.10270

[3] Thanh DNH, Thanh LT, Nguyen NH, et al., 2020, Adaptive Total Variation L1 Regularization for salt and Pepper Image Denoising. Optik, 208: 163677.

[4] Lewkowycz A, Gur-Ari G, 2020, On the Training Dynamics of Deep Networks with L2 Regularization. 34th Conference on Neural Information Processing Systems, 4790–4799.

[5] Morerio P, Cavazza R, Volpi R, et al., 2017, Curriculum Dropout.2017 IEEE International Conference on Computer Vision (ICCV), 3564–3572.

[6] Wan L, Zeiler M, Zhang S, et al., 2013, Regularization of Neural Networks using DropConnect. Proceedings of International Conference on Machine Learning, 1058–1066.

[7] Murugan P, Shanmugasundaram D, 2017, Regularization and Optimization Strategies in Deep Convolutional Neural Network. ArXiv. https://doi.org/10.48550/arXiv.1712.04711

[8] Zhang Z, Li X, Zhang H, et al., 2021, Triplet Deep Subspace Clustering via Self-Supervised Data Augmentation. Proceedings of 2021 IEEE International Conference on Data Mining (ICDM), 946–955.

[9] Sivaraman A, Kim S, Kim M, 2021, Personalized Speech Enhancement Through Self-Supervised Data Augmentation and Purification. Proceedings of INTERSPEECH, 2676–2680.

[10] Bari Saiful M, Mohiuddin T, Joty ST, 2020, UXLA: A Robust Unsupervised Data Augmentation Framework for Zero-Resource Cross-Lingual NLP. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, 1978–1992. https://doi.org/10.18653/v1/2021.acl-long.154

[11] Lowell D, Howard B, Liptopn ZC, et al., 2021, Unsupervised Data Augmentation with Naive Augmentation and without Unlabeled Data. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 4992–5001.

[12] Devries T, Taylor GW, 2017, Improved Regularization of Convolutional Neural Networks with Cutout.

ArXiv. https://doi.org/10.48550/arXiv.1708.04552

[13] Zhang H, Cisse M, Dauphin YN, et al., 2018, mixup: Beyond Empirical Risk Minimization. ArXiv. http:/doi.org/10.48550/arXiv.1710.09412

[14] Singh KK, Yu H, Sarmasi A, 2018, Hide-and-Seek: A Data Augmentation Technique for Weakly-Supervised Localization and Beyond. ArXiv. https://doi.org/10.48550/arXiv.1811.02545

[15] Yun S, Han D, Oh SJ, et al., 2019, CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 6022–6031.

[16] Zhong Z, Zheng L, Kang G, et al., 2017, Random Erasing Data Augmentation. Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20), 13001–13008.

[17] Chen P, Liu S, Zhao H, et al., 2020, GridMask Data Augmentation. ArXiv. https://doi.org/10.48550/arXiv.2001.04086

[18] Li P, Li X, Long X, et al., 2020, FenceMask: A Data Augmentation Approach for Pre-extracted Image Features. ArXiv. https://doi.org/10.48550/arXiv.2006.07877

[19] Hendrycks D, Mu N, Cubuk ED, et al., 2019, AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty. ArXiv. https://doi.org/10.48550/arXiv.1912.02781

[20] Gong C, Wang D, Li M, et al., 2021, KeepAugment: A Simple Information-Preserving Data Augmentation Approach. Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 1055–1064.

[21] Zhu J-Y, Park T, Isola P, et al., 2017, Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), 2242–2251.

[22] Mirza M, Osindero S, 2014, Conditional Generative Adversarial Nets. ArXiv. https://doi.org/10.48550/arXiv.1411.1784

[23] Cubuk ED, Zoph B, Mané D, et al., 2019, AutoAugment: Learning Augmentation Strategies from Data. Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 113–123.

[24] Lim S, Kim I, Kim T, et al., 2019, Fast AutoAugment. 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), 1–11.

[25] Cubuk ED, et al., Randaugment: Practical Automated Data Augmentation with a Reduced Search Space. Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 3008–3017.

[26] Buslaev AV, Parinov A, Khvedchenya E, et al., 2018, Albumentations: Fast and Flexible Image Augmentations. ArXiv. https://doi.org/10.48550/arXiv.1809.06839

[27] Takahashi, Ryo et al. RICAP: Random Image Cropping and Patching Data Augmentation for Deep CNNs. Proceedings of the 10th Asian Conference on Machine Learning, 785–798.

[28] Zagoruyko S, Komodakis N, 2015, Wide Residual Networks. ArXiv. https://doi.org/10.48550/arXiv.1605.07146

[29] He K, Zhang X, Ren S, et al., 2016, Deep Residual Learning for Image Recognition. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778.

[30] Lee C-Y, Xie S, Gallagher P, et al., 2014, Deeply-Supervised Nets. ArXiv. https://doi.org/10.48550/arXiv.1409.5185

[31] Romero A, Ballas N, Kahou SE, et al., 2014, FitNets: Hints for Thin Deep Nets. ArXiv. https://doi.org/10.48550/arXiv.1412.6550

[32] Springenberg JT, Dosovitskiy A, Brox T, et al., 2014, Striving for Simplicity: The All Convolutional Net. ArXiv. https://doi.org/10.48550/arXiv.1412.6806

[33] He K, Zhang X, Ren S, et al., 2015, Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV),1026–1034.

[34] Russakovsky O, Deng J, Su H, et al., 2015, ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, 115: 211–252.