DATA AUGMENTATION USING CORNER CUTMIX AND AN AUXILIARY SELF-SUPERVISED LOSS

Fen Fang[1], *Nhat M.Hoang*[2], *Qianli Xu*[1] *and Joo-Hwee Lim*[1,2]

[1]Institute for Infocomm Research, A*STAR, 138632, Singapore

[2]School of Computer Science and Engineering, NTU, 639798, Singapore

ABSTRACT

Deep convolutional neural networks (CNNs) have achieved remarkable success in computer vision tasks, but their training is susceptible to overfitting when the training sample size is insufficient. In this paper, we introduce Corner CutMix, a novel data augmentation technique for CNN training. During training, Corner CutMix randomly selects a region from one of four corner areas in an image and replaces it with a randomly chosen region from a distractor image. Additionally, we design an auxiliary self-supervised loss function to learn the position of the selected corner region, thereby improving the transferability and generalizability of the learned representation. Corner CutMix is easy to implement, adding little computational overhead, and can be combined with other augmentation methods such as random cropping, color distortion, and flipping. Our extensive classification task experiments in self-supervised learning on public datasets (e.g., CIFAR10, CIFAR100, and STL10) demonstrate the effectiveness of Corner CutMix, which consistently outperforms strong baselines such as CutOut and CutMix.

***Index Terms*—** Data augmentation, region cut and mix, auxiliary loss, self-supervised learning.

1. INTRODUCTION

As widely recognized, a greater number of samples used in training a CNN model can lead to better effectiveness. However, obtaining a large dataset can be prohibitively expensive. Therefore, data augmentation techniques are utilized to enhance the diversity of images in the dataset. Three commonly used augmentation methods are random cropping [1], random color distortion [2] and random flipping [3]. These methods manipulate images in a direct way and easy to implement. These basic augmentation methods suffer from constraints that the training data obeys the distribution close to the testing data distribution. Besides, after some particular augmentation, some regions containing important features of the images will be lost.

This research is supported by the Agency for Science, Technology and Research, under the AME Programmatic Funding Scheme (Project#A18A2b0046)

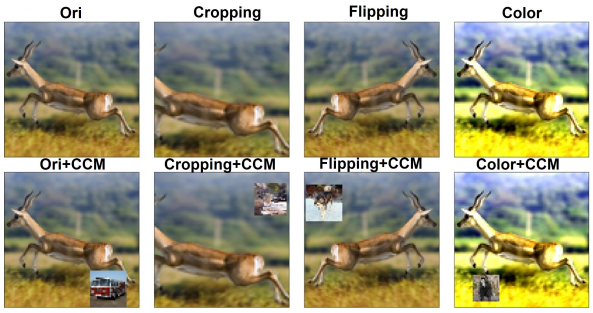

Fig. 1: Visualization examples of Corner CutMix (CCM). The CCM can be directly applied to original images as well as combined with other augmentation techniques including cropping, flipping and color distortion.

To alleviate over-fitting caused by a CNN focusing too much on a small set of activation or small set region on input, effective regularization techniques, Image Mix [4, 5, 6, 7, 8] and random region erasing [9, 10, 11, 12, 13] have been proposed. For the former methods, to improve the robustness and overall performance of CNNs during its training, CutMix [4] implants a portion of an input with a portion of the distractor input. Mixup [5] generates an element-wise convex combination of two inputs using an adaptive mixing policy. Fmix [6] mixes low-frequency images with random binary masks. AugMix [7] utilizes diverse augmentations and a formulation to mixes multiple augmentation operations into three augmentations and the results of the augmentations are mixed in convex combinations. Instead of implementing mix on input image. ManifoldMix [8] manipulates mixing on hidden representations. For the latter methods, CutOut [9] randomly masks out square regions from input ; Hide-and-Seek [10] hides patches of input randomly, then forces the CNNs to seek relevant patches when the most discriminative patch is hidden; Random Erasing [11] randomly selects a rectangle region in an input, and assigns random values to the pixels in the region; GridMask [12] is also region deletion based method, but the deleted regions are neither a continuous region nor randomly selected. Instead, they are a set of squares in spatially uniformly distributed based on density and size. Some advanced augmentations, such as AutoAugment [14] and RandAugment [15], are proposed to automatically search

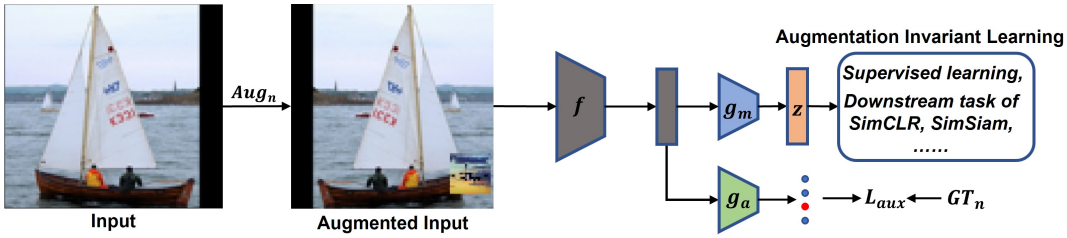

Fig. 2: Illustration of our proposed augmentation method. Given an input image, after applying Corner CutMix augmentation, an auxiliary task that predicts the location of drop out region is learned to enhance the representation learning of the main task. Where f is a feature extractor, g_m and g_a are classifiers for main and auxiliary tasks, L_{aux} is the loss between the predicted location and the automatically generated ground truth location (GT_n), z is the learned representation for main task, which can be transferred to augmentation invariant learning.

augmentation approaches to improve performance.

While the methods mentioned above have shown remarkable improvements in supervised learning, they may not be optimal for self-supervised learning. Using predefined augmentations in self-supervised learning can hurt the performance of learned representations in downstream tasks that rely on the same information related to those augmentations. For instance, using color distortion augmentation may reduce the amount of color information in the learned representations, which can negatively affect their performance in downstream tasks involving color-sensitive objects, such as Food classification [16]. To avoid such information loss and obtain more generalizable representations, previous work [17] has shown that adding an auxiliary self-supervised loss to the augmentation can be useful.

In this study, we introduce a new data augmentation method called Corner CutMix, which replaces a rectangular region from one of the four corners of an input image with a distractor image. Our aim is to improve representation learning in CNNs, particularly in unsupervised/self-supervised training. Fig. 1 shows some examples of the proposed augmentation. The most closely related method to our approach is CutMix [4]. However, Corner CutMix differs in that it only replaces regions within the four corners of the input image. Our method is based on two key ideas. Firstly, CutMix addresses the problem of occlusion, which is a significant factor affecting the generalizability of CNNs [18]. By generating new images with varying degrees of occlusion, a robust classifier should be able to recognize object categories even when some parts of the object are occluded. Secondly, the central portion of an image often contains discriminative features that are critical for classification, particularly for fine-grained classification. Therefore, by replacing off-center regions, we aim to reduce the risk of losing important features.

Meanwhile, we introduce an auxiliary loss on the region location prediction to strengthen the transferability and generalizability of the learned representation. Such a loss is designed to preserve augmentation-aware information in learned representations, which could be helpful for their transferability to downstream tasks. Our method is briefly illustrated in Fig. 2.

2. CORNER CUTMIX AUGMENTATION

2.1. New Training Sample Generation

Let $x \in \mathrm{R}^{\mathrm{WxHxC}}$ and y indicate a training image and its label. The Corner CutMix aims at creating a new training sample (x_{aug}, y_{aug}) by combining an image (x_{ori}, y_{ori}) and distractor one (x_{dis}, y_{dis}). The new training sample (x_{aug}, y_{aug}) can be obtained by

$$\begin{aligned} x_{aug} &= x_{ori} \odot m + x_{dis} \odot (1 - m) \\ y_{aug} &= y_{ori} \end{aligned} \tag{1}$$

where $\odot$ is element-wise multiplication. m is a binary rectangular mask specify where to drop out and fill in content from the distractor image (e.g., '1' or '0' means it will be preserved or dropped out in the new image). Given an image, the drop out region is within one of corner regions (see regions separated by dotted yellow lines in Fig.3), which are defined by four corner points of the image and its center point. The aspect ratio of the drop out region doesn't have to be proportional to the original image, but it is supposed to ranged in commonly used ones (e.g., 1:1, 3:2). The size of the region doesn't have to be fixed whereas it is deliberately set so that it does not extend outside the current corner (quadrant).. From Eq.1, one can note the major difference from CutMix is that, the most of discriminative features are more likely to be preserved in the new image using our method because the drop out region is limited to be within corners of an image. Thus, the label of the new training sample is the same as the one of the original image. This is alleged augmentation-invariant. Given two random augmentations $v_1 = t_{\omega 1}(x)$ and $v_2 = t_{\omega 2}(x)$ (t_w is a composition of different type of augmentations), their representations $f(v_1) \approx f(v_2)$. In CutMix, the new label, y_{aug}, is generated using a combination ratio between the y_{ori} and y_{dis}.

2.2. Auxiliary Self-supervised Loss

In this section, we introduce an auxiliary self-supervised task associated with the proposed augmentation. More specifically, given a augmented sample, we add an auxiliary self-supervised loss to minimize the difference between the predicted and ground truth drop out region locations. As shown

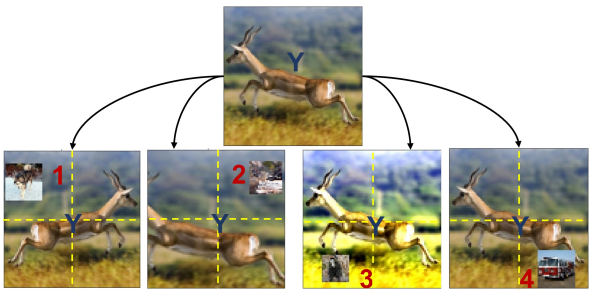

Fig. 3: Examples of four drop out region classes.

in Fig. 3, the regions of an image are divided into four quadrants by yellow dotted grids. Once the drop out region is selected, the ground truth label of region location is automatically generated. Then the auxiliary loss given an augmented sample after applying the proposed augmentation method is defined as

$$L_{aux}(x_{aug};\theta) = -\sum_{i=1}^{n} Y_{rl} log(f(y_{rl})) \quad (2)$$

where $f(x) = \frac{e^x}{\sum_{j=1}^{n} e^{x_j}}$, Y_{rl} is the ground truth drop out location, y_{rl} is the predicted ones by the classifier. θ is the collection of all parameters in the classifier. n is the number of region locations, which is 4 in this work.

Algorithm 1: Corner CutMix Procedure

```
for each iteration do
    Im,label=get_minibatch(training_samples) ;
    Initialize temporary buffer T;
    if mode==training then
        Im_ori,label_ori=shuffle_minibatch(Im,label);
        Augment Im_ori using one ∈(cropping,
          flipping, color distortion);
        Randomly select the quadrant Q_i, i ∈ [1,4] ;
        The drop out region label y_rl=i;
        Randomly select the region R_inpaint in Q_i;
        Randomly select an image Im_1 ∈ training
          samples and Im_1 ≠ Im_ori;
        Take inputs Im and Im_ori to compute the
          augmented image Im_aug using Eq.1;
        Append (Im_aug, label_ori, y_rl) to T;
    end
    for each item in T do
        output = model_forward(Im_aug);
        Compute the auxiliary loss L_aus using Eq.2;
        Compute the self-supervised loss L_sc using
          Eq.3;
        Initialize hyper-parameter λ;
        Compute the total loss L_total using Eq. 4;
    end
end
```

Note that the feature extractors for the main task (representation learning) and the auxiliary task are the same CNNs. The difference is that, in the main task, the feature extractor connects with a projection head. In the auxiliary task, it connects with linear layers. This allows us to incorporate the self-supervised auxiliary loss into the recent state-of-the-art self-supervised learning methods [19, 20, 21]. For instance, the objective of SimCLR [21] computed in Eq.3 with the designed auxiliary loss can be expressed in Eq. 4 to encourages the shared representation $f(x)$ to learn both augmentation-invariant main task and the drop out region location aware auxiliary task.

$$L_{sc}(x_{aug};\theta) = -log\frac{\exp(\text{sim}(z_{aug}, z_{ori})/\tau)}{\sum_{k=1}^{2N} \mathbb{I}_{[k\neq ori]} \exp(\text{sim}(z_{aug}, z_k)/\tau))} \quad (3)$$

$$L_{total}(x_{aug};\theta) = L_{sc}(x_{aug};\theta) + \lambda \cdot L_{aux}(x_{aug};\theta) \quad (4)$$

where $\mathbb{I}$ is an indicator function that is set as '1' if the selected distractor image is not the original image, otherwise '0'. $sim(z_{aug}, z_k)$ is the cosine similarity between representations of the augmented sample and the distractor sample. τ is a temperature parameter set as 0.5 in the paper. $\lambda \in \{0, 1\}$ is a hyper-parameter for balancing losses. N is the number of all augmented samples given an original image.

In this work, the proposed data augmentation method is combined with three commonly used augmentations: Random cropping, random flipping and color distortion. The procedure of implementing Random Corner CutMix and auxiliary loss is shown in Algorithm 1.

3. EXPERIMENTAL SETUP

The Corner CutMix technique is applied for enhancing representation learning in the pre-text task of the self-supervised learning. Its performance is reflected on accuracy of the downstream task, image classification. The evaluation metrics is the top-1 classification accuracy. The datasets used in this work are CIFAR-10 , CIFAR-100 [22], STL-10 [23] and Food-101 [16]. CIFAR-10 contains 10 distinct classes (e.g., cat, dog, car and boat). CIFAR-100 has 100 categories, compared with CIFAR-10, it contains some classes (e.g., maple, oak, palm, pine and willow) visually very similar to each other. Both CIFAR datasts have 60,000 color images of dimension 32×32. Each of them is split into training dataset of 50,000 images and testing dataset of 10,000 images. STL-10 dataset consists of 10 classes (e.g., airplane, bird and horse). It contains a total of 113,000 images with a size of 96×96 pixels. Of all images, only 13,000 images are labeled, in our experiments, the labeled images are split into the training and testing sets with 5,000 and 8,000 images respectively. Food-101 consists of 101 fine-grained food categories, with totally 101,000 images, which are very color-sensitive. For each category, 750 images are for training and 250 are for testing.

To demonstrate the effectiveness of our method, we conducted several experiments. Firstly, we evaluated our method as a standalone augmentation technique on the training set

and compared its performance against other popular augmentation techniques, namely CutOut[9] and CutMix[4]. Next, we evaluated the combined performance of our technique with three basic augmentations, namely Random Cropping (RC), Random Flipping (RF), and Color Distortion (CD), by integrating our method with each of these techniques. To ensure the reliability of our results, we repeated each experiment three times with different random seeds and reported the mean value of these results for each setting.

Two architectures are adopted in our experiments: ResNet-50 [24] (commonly used and robust backbone) and EfficientNet-B2 [25] (achieved both higher accuracy and better efficiency over existing CNNs). The networks are randomly initialized in the pre-text task learning, meaning that they are trained from scratch (e.g., not pretrained on ImageNet.). The learned visual representations from the pre-text are used as weights to initialize the networks in the downstream task. We train the model for 200 epochs with batches of 128 images using LARS optimizer with momentum of 0.9 and weight decay of 1e-6. The learning rate is initially set to 0.1, and is reduced by a factor of 5 after every 20 epochs. All experiments are implemented in Pytorch on a NVIDIA GeForce GTX 2080 GPU machine with 16GB of graphic memory.

4. RESULTS AND DISCUSSIONS

The results of downstream task on different datasets among without augmentation (w/o Aug) and using different augmentations (w/ Cutout, w/CutMix and w/Ours) are shown in Table. 1. One can observe that, by applying our augmentation, the classification accuracy on all datasets with both backbones are improved at least by 3%. This observation verifies that Corner CutMix is useful on strengthening the visual representation learning in CNNs training. In comparison to other augmentation techniques, our method achieves the highest accuracy on all datasets, except for STL-10 with EfficientNet-B2, where our method underperforms CutMix by 0.38%.

Table 2 presents a performance comparison between the application of Corner CutMix in combination with the three basic augmentations, and their use as standalone augmentation techniques. Results indicate that when ResNet-50 is used as the backbone CNN, the integration of Corner CutMix with RC and CD consistently outperforms the use of these techniques individually across all datasets. However, the addition of our augmentation technique to RF yields slightly worse performance than using RF alone. When EfficientNet-B2 is used as the backbone, the performance gains from integrating Corner CutMix with the other augmentation techniques vary across different datasets. On CIFAR-10 and CIFAR-100, combining Corner CutMix with RF and CD leads to higher classification accuracy, while integrating it with RC results in a slight decrease in accuracy. On STL-10, Corner CutMix is better suited for integration with RC and RF, but not with CD. On Food-101, our method yields better performance when combined with all three basic augmentations.

The aforementioned observations demonstrate that Corner CutMix effectively enhances representation learning during CNN training. However, the impact of Corner CutMix on downstream tasks is influenced by several factors, including the specific dataset used, the backbone CNNs employed, and other augmentation techniques implemented. In summary, when employing augmentation techniques such as Corner CutMix, it is important to consider several factors [18]: (1) the notion that more augmentations do not always result in better performance, (2) different combinations of augmentation strategies produce varying effects on CNN training, and (3) different datasets require specific and favorable augmentations to achieve optimal results.

Table 1: Accuracy comparison among using other two similar augmentations and our augmentation on various datasets. Results in gray and white background are using ResNet-50 and EfficientNet-B2 as backbones respectively.

Methods	CIFAR-10	CIFAR-100	STL-10	Food-101
w/o Aug	0.7078	0.4344	0.4826	0.3399
w/ CutOut	0.7710	0.4822	0.5030	0.3593
w/ CutMix	0.7867	0.4890	0.5198	0.3644
w/ Ours	**0.8040**	**0.5028**	**0.5271**	**0.3765**
w/o Aug	0.6339	0.3225	0.3824	0.2226
w/CutOut	0.7033	0.3440	0.3922	0.2540
w/CutMix	0.7246	0.3574	**0.4059**	0.2695
w/ Ours	**0.7335**	**0.3583**	0.4021	**0.2806**

Table 2: Performance comparison between before and after combining our augmentation method to the three basic augmentations on four public datasets.

Augmentation	CIFAR-10	CIFAR-100	STL-10	Food-101
RC	0.8837	0.6708	0.7856	0.5623
RC + Ours	**0.9042**	**0.6790**	**0.7876**	**0.5873**
RF	**0.7442**	**0.4637**	**0.4928**	0.3238
RF + Ours	0.7415	0.4565	0.4908	**0.3253**
CD	0.8297	0.5695	0.6505	0.4574
CD + Ours	**0.8499**	**0.5859**	**0.6617**	**0.4909**
RC	**0.8621**	**0.5902**	0.7238	0.5518
RC + Ours	0.8550	0.5889	**0.7280**	**0.5565**
RF	0.6596	0.3258	0.3959	0.2664
RF+ Ours	**0.7304**	**0.4062**	**0.4340**	**0.3245**
CD	0.8069	0.5208	**0.6210**	0.4479
CD+ Ours	**0.8170**	**0.5239**	0.6045	**0.4759**

5. CONCLUSIONS AND FUTURE WORKS

In this work, we propose a novel and effective augmentation technique, Corner CutMix, to improve the generalization ability of CNNs. Our augmentation is particularly useful in the unsupervised/self-supervised learning paradigm, as it enhances visual representation learning in the pretext task, which in turn leads to improved performance in downstream tasks. Implementation is straightforward: it consists of dropping out a region within one of the four corners of an image and mixing it with a randomly chosen distractor. To further improve the generalizability of representations, we also design an auxiliary task to predict the location of the drop out region. Our experiments have shown the effectiveness of our augmentation algorithm in unsupervised learning and transfer learning on classification task. Currently, the drop out region is forced within in one of four quadrants. In the future, we plan to investigate more fine-grained region selection policies and apply our method to other tasks, such as object detection.

6. REFERENCES

[1] R. Takahashi, T. Matsubara, and K.Uehara, "Data augmentation using random image cropping and patching for deep cnns," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 9, pp. 2917–2931, 2020.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[3] K .Simonyan and A. Zisserman, "Very deep convolutioanl networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[4] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *International Conference on Computer Vision*, 2019, pp. 6023–6032.

[5] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[6] H. Ethan, M. Antonia, P. Matthew, N. Mahesan, P.B. Adam, and H. Jonathon, "Fmix: Enhancing mixed sample data augmentation," in *arXiv:2002.12047*, 2020.

[7] D. Hendrycks, N. Mu, E.D. Cubuk, b. Zoph, J. Gilmer, and B. Lakshminarayanan, "Augmix: A simple data processing method to improve robustness and uncertainty," in *International Conference on Learning Representations*, 2020.

[8] V. Vikas, L. Alex, B. Christopher, N. Amir, M. Ioannis, L.P. David, and B. Yoshua, "Manifold mixup: Better representations by interpolating hidden states," in *International Conference on Learning Representations*, 2019.

[9] D. Terrance and W. T. Graham, "Improved regularization of convolutional neural networks with cutout," in *arXiv:1708.04552*, 2017.

[10] K. K. Singh and Y.J. Lee, "Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization," in *International Conference on Computer Vision*, 2017.

[11] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *AAAI Conference on Artificial Intelligence*, 2020.

[12] P. Chen, S. Liu, H. Zhao, and J. Jia, "Gridmask data augmentation," in *arXiv:2001.04086*, 2020.

[13] P. Li, X. Li, and X. Long, "Fencemask: A data augmentation approach for pre-extracted image features," *ArXiv*, vol. abs/2006.07877, 2020.

[14] E.D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 113–123.

[15] E.D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 702–703.

[16] B. Lukas, G. Matthieu, and V.G.Luc, "Food-101-mining discriminative components with random forests," in *European Conference on Computer Vision*, 2014.

[17] H. Lee, K. Lee, K. Lee, H. Lee, and J. S, "Improving transferability of representations via augmentation-aware self-supervision," in *International Conference on Neural Information Processing Systems*, 2021.

[18] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, "Image data augmentation for deep learning: A survey," in *arXiv:2204.0861*, 2022.

[19] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.

[20] X. Chen and K. He, "Exploring simple siamese representation learning," in *arXiv:2011.10566*, 2020, pp. 9729–9738.

[21] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning*, 2020, pp. 1597–1607.

[22] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[23] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 215–223.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2016.

[25] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, 2019, pp. 6105–6114.