GradAug: A New Regularization Method for Deep Neural Networks

Taojiannan Yang, Sijie Zhu, Chen Chen
University of North Carolina at Charlotte
{tyang30,szhu3,chen.chen}@uncc.edu

Abstract

We propose a new regularization method to alleviate over-fitting in deep neural networks. The key idea is utilizing randomly transformed training samples to regularize a set of sub-networks, which are originated by sampling the width of the original network, in the training process. As such, the proposed method introduces self-guided disturbances to the raw gradients of the network and therefore is termed as Gradient Augmentation (GradAug). We demonstrate that GradAug can help the network learn well-generalized and more diverse representations. Moreover, it is easy to implement and can be applied to various structures and applications. GradAug improves ResNet-50 to 78.79% on ImageNet classification, which is a new state-of-the-art accuracy. By combining with CutMix, it further boosts the performance to 79.58%, which outperforms an ensemble of advanced training tricks. The generalization ability is evaluated on COCO object detection and instance segmentation where GradAug significantly surpasses other state-of-the-art methods. GradAug is also robust to image distortions and adversarial attacks and is highly effective in the low data regimes.

1 Introduction

Deep neural networks have achieved great success in computer vision tasks such as image classification [1, 2], object detection [3, 4] and semantic segmentation [5, 6]. But deep neural networks are often over-parameterized and easily suffering from over-fitting. Regularization [7, 8] and data augmentation [1, 9] are widely used techniques to alleviate the over-fitting problem. Many data-level regularization methods [8, 10, 11] have achieved promising performance in image classification. These methods are similar to data augmentation where they put constraints on the input images. Although effective in image classification, these methods are hard to apply to downstream tasks such as object detection and segmentation due to their special operations. For example, the state-of-the-art CutMix [11] can not be directly applied to object detection because first, mixing samples will destroy the semantics in images; second, it is hard to interpolate the labels in these tasks. Another category of regularization methods impose constraints on the network structures. [12] proposes that adding noises to the network gradients can improve generalization. Other methods [7, 13, 14] randomly drop some connections in the network, which implicitly introduce random noises in the training process. These methods are usually more generic but not as effective as data-level regularization.

In this paper, we introduce Gradient Augmentation (GradAug), which generates meaningful disturbances to the gradients by the network itself rather than just adding random noises. The idea is that when a random transformation (e.g., random rotation, random scale, random crop, etc.) is applied to an image, a well-generalized network should still recognize the transformed image as the same object. Different from the regular data augmentation technique which only regularizes the full-network, we regularize the representations learned by a set of sub-networks, which are randomly sampled from the full network in terms of the network width (number of channels in each layer). Since the representation of the full network is composed of sub-networks’ representations due to weights

Preprint. Under review.
sharing, we expect the sub-networks to learn different representations from different transformations, which will lead to a well-generalized and diversified full network representation.

We conduct a comprehensive set of experiments to evaluate the proposed regularization method. Using a simple random scale transformation, GradAug can improve the ImageNet Top-1 accuracy of ResNet-50 from 76.32% to 78.79%, which is a new state-of-the-art accuracy. By leveraging a more powerful data augmentation technique – CutMix [11], we can further push the accuracy to 79.58%. The representation’s generalization ability is evaluated on COCO object detection and instance segmentation (Section 4.4). Our ImageNet pretrained model alone can improve the baseline MaskRCNN-R50 by +1.2 box AP and +1.2 mask AP. When applying GradAug to the detection framework, it can outperform the baseline by +1.7 box AP and +2.1 mask AP. Moreover, we demonstrate that GradAug is robust to image corruptions (e.g., noise, blur, etc.) and adversarial attacks (Section 4.5) and is also highly effective in the low data settings (Section 4.6).

2 Related Work

Data augmentation. Data augmentation [1, 9, 15] increases the amount and diversity of training data by linear or non-linear transformations over the original data. In computer vision, it usually includes rotation, flipping, etc. Recently, a series of regularization methods use specially-designed operations on the input images to alleviate over-fitting in deep neural networks. These methods are similar to data augmentation. Cutout [8] randomly masks out a squared region on the image to force the network to look at other image context. Dropblock [16] shares a similar idea with Cutout but it drops a region in the feature maps. Although they have achieved improvements over the regular data augmentation, such region dropout operations may lose information about the original images. Mixup [10] mixes two samples by linearly interpolating both the images and labels. CutMix [11] combines Cutout and Mixup to replace a squared region with a patch from another image. Other mixed sample variants [17, 18] all share similar ideas. While effective in image classification, the mixed sample augmentation is not natural to be applied to tasks such as detection and segmentation due to semantic and label ambiguities. In contrast, the proposed GradAug is a task-agnostic approach which leverages the most common image transformations to regularize sub-networks. This allows the method to be directly applied to different vision tasks and easily amenable for other applications.

Structure regularization. Another category of regularization methods impose constraints on the network weights and structure to reduce over-fitting. [12] points out that adding random noises to the gradients during training can help the network generalize better. Dropout [7] randomly drops some connections during training to prevent units from co-adapting. The random dropping operation also implicitly introduces random noises into the training process. Many following works share the idea of Dropout by randomly dropping network layers or branches. Shake-Shake [19] assigns random weights to residual branches to disturb the forward and backward passes. But it is limited to three-branch architectures. ShakeDrop [20] extends Shake-Shake to two-branch architectures (e.g., ResNet [2] and PyramidNet [21]). However, its application is still limited. [13] randomly drops a subset of layers during training. The final network can be viewed as an ensemble of many shallow networks. Although these methods have shown improvements on image classification, they are usually not as effective as data-level regularization strategies. Moreover, their generalization and effectiveness are not validated on other tasks.

GradAug leverages the advantages of both categories of methods. We use different augmentations to regularize different sub-networks. This introduces self-guided disturbances to the gradients rather than adding random noises. Therefore, GradAug is more effective and generic than previous techniques.

3 GradAug

3.1 Algorithm

When applying some random transformations to an image, human can still recognize it as the same object. We expect deep neural networks to have the same generalization ability. GradAug aims to regularize sub-networks with differently transformed training samples. There are various methods to generate sub-networks during training. Previous works [7, 13, 14] usually stochastically drop some neurons, layers or paths, thus the generated sub-networks do not have a clear correlation with each
We provide an in-depth analysis of GradAug from the perspective of gradient flow. For simplicity, we also leverage two training tricks in [24]. First, we use the output of the full-network as soft labels. This way, a larger sub-network always shares the representations of a smaller sub-network in a weights-sharing training fashion, so it can leverage the representations learned in smaller sub-networks. Iteratively, sub-networks can construct a full-network with diversified representations. Figure 1 shows the class activation maps (CAM) [22] of the sub-network and full-network. The full-network pays attention to several regions of the object because it can leverage the representation of the sub-network. For example, when the sub-network (\(w = 0.9\)) focuses on one dog in the image, the full-network shares this attention and uses the other network part to capture the information of another dog. Therefore, the full-network learns richer semantic information in the image, while the baseline model only models a single region and does not fully comprehend the salient information of the image. To make the method simple and generic, we choose among the most commonly used transformations such as random scales, random rotations, random crops, etc. In the experiments, we show that a simple random scale transformation can already achieve state-of-the-art performance on image classification, and it can be directly applied to other applications. Moreover, we can use more powerful augmentations such as CutMix for further enhanced performance.

**Training procedure.** The training procedure of GradAug is very similar to the regular network training. In each training iteration, we first train the full-network with the original images, which is the same as the regular training process. Then we additionally sample \(n\) sub-networks and train them with randomly transformed images. Finally, we accumulate the losses of full-network and sub-networks to update the model weights. This naive training approach achieves good training accuracy but the testing accuracy is very low. This is caused by the batch normalization (BN) [23] layers. The BN layer will collect a moving average of training batches’ means and variances during training. The collected mean and variance will be used during inference. However, the batch mean and variance in the sub-networks can be very different from those in the full-network because the training samples are randomly transformed. This will cause the final BN mean and variance to be inappropriate for the full-network during inference. But in the training phase, BN uses the mean and variance of the current batch, so the training behaves normally. To obtain the correct BN statistics for the full-network, we do not update BN mean and variance when training the sub-networks. Only the full-network is allowed to collect these statistics. However, the weights in BN layer are still updated by sub-networks because they can be shared with full-network. To further improve the performance, we also leverage two training tricks in [24]. First, we use the output of the full-network as soft labels to train the sub-networks. Second, we always sample the smallest sub-network (i.e., \(w = \alpha\)) during training if \(n > 1\). The Pytorch-style pseudo-code of GradAug is presented in Algorithm 1.

### 3.2 Analysis of gradient property

We provide an in-depth analysis of GradAug from the perspective of gradient flow. For simplicity, we consider a fully connected network with 1-D training samples. We define the network as \(N\). The parameter of one layer in the full-network is \(\theta \in \mathbb{R}^{c_1 \times c_2}\). The parameter of sub-networks is \(\theta_w\) as explained in Section 3.1. \(x \in \mathbb{R}^d\) is the training sample and \(y\) is its label. The output of the network is denoted as \(N(\theta, x)\), and the training loss is \(l(N(\theta, x), y)\) where \(l\) is the loss function, which is often the cross entropy in image classification. The loss and gradients in a standard training process are computed as

\[
L_{std} = l(N(\theta, x), y), \quad g_{std} = \frac{\partial L_{std}}{\partial \theta},
\]

![Figure 1: Class activation maps (CAM) of the network trained by GradAug and the baseline. The full-network shares the attention of the sub-network and focuses on multiple semantic regions.](image.png)
We first evaluate the effectiveness of GradAug on image classification. Next, we show the generalization ability of GradAug on object detection and instance segmentation. Finally, we demonstrate that GradAug can improve the model’s robustness to image distortions and adversarial attacks. We also show GradAug is effective in low data settings and can be extended to semi-supervised learning.

\section{Experiments}

We first evaluate the effectiveness of GradAug on image classification. Next, we show the generalization ability of GradAug on object detection and instance segmentation. Finally, we demonstrate that GradAug can improve the model’s robustness to image distortions and adversarial attacks. We also show GradAug is effective in low data settings and can be extended to semi-supervised learning.
Table 1: ImageNet classification accuracy of different techniques on ResNet-50 backbone.

| Model                      | FLOPs | Top-1 (%) | Top-5 (%) |
|----------------------------|-------|-----------|-----------|
| ResNet-50 [2]              | 4.1G  | 76.32     | 92.95     |
| ResNet-50 + Cutout [8]     | 4.1G  | 77.07     | 93.34     |
| ResNet-50 + Dropblock [16] | 4.1G  | 78.13     | 94.02     |
| ResNet-50 + Mixup [10]     | 4.1G  | 77.9      | 93.9      |
| ResNet-50 + CutMix [11]    | 4.1G  | 78.60     | 94.08     |
| ResNet-50 + StochDepth [13]| 4.1G  | 77.53     | 93.73     |
| ResNet-50 + Droppath [14]  | 4.1G  | 77.10     | 93.50     |
| ResNet-50 + ShakeDrop [20] | 4.1G  | 77.5      | -         |
| ResNet-50 + GradAug (Ours) | 4.1G  | 78.79     | 94.38     |
| ResNet-50 + bag of tricks [26] | 4.3G | 79.29 | 94.63 |
| ResNet-50 + GradAug+ (Ours) | 4.1G  | 79.58     | 94.85     |

4.1 ImageNet classification

Implementation details. ImageNet [25] dataset contains 1.2 million training images and 50,000 validation images in 1000 categories. We follow the same data augmentations in [11] to have a fair comparison. On ResNet-50, we train the model for 200 epochs with a batch size of 512. The initial learning rate is 0.2 with cosine decay schedule. We sample \( n = 3 \) sub-networks in each training iteration and the width lower bound is \( \alpha = 0.9 \). For simplicity, we only use random scale transformation for sub-networks. That is the input images are randomly resized to one of \{224 \times 224, 192 \times 192, 160 \times 160, 128 \times 128\}. Note that we report the final-epoch accuracy rather than the highest accuracy in the whole training process as is reported in CutMix [11].

We evaluate GradAug and several popular regularization methods on the widely used ResNet-50 [2]. The results are shown in Table 1. GradAug achieves a new state-of-the-art performance of 78.79\% based on ResNet-50. Specifically, GradAug significantly outperforms the structure regularization methods by more than 1 point. As illustrated in Eq. 3 and Eq. 4, GradAug has a similar form with structure regularization. The difference is that GradAug introduces self-guided disturbances to augment the raw gradients. The large improvement over the structure regularization methods clearly validates the effectiveness of our proposed method.

As shown in Eq. 6, GradAug can be seamlessly combined with data augmentation. We combine GradAug with CutMix (\( p=0.5 \)) and denote this method as GradAug+. We compare GradAug+ with bag of tricks [26] at the bottom of Table 1. We can see that GradAug+ outperforms bag of tricks both in model complexity and accuracy. Note that bag of tricks includes a host of advanced techniques such as model tweaks, training refinements, label smoothing, knowledge distillation, Mixup augmentation, etc., while GradAug is as easy as regular model training.

4.2 Cifar classification

Implementation details. We also evaluate GradAug on Cifar-100 dataset [27]. The dataset has 50,000 images for training and 10,000 images for testing in 100 categories. We choose WideResNet [28] and PyramidNet [21] structures as they achieve state-of-the-art performance on Cifar dataset. We follow the training setting in [21, 28] in our experiments. For WideResNet, we train the model for 200 epochs with a batch size of 128. The initial learning rate is 0.1 with cosine decay schedule. Weight decay is 0.0005. PyramidNet is trained for 300 epochs with a batch size of 64. The initial learning rate is 0.25 and decays by a factor of 0.1 at 150 and 225 epochs. Weight decay is 0.0001. We use random scale transformation where input images are resized to one of \{32 \times 32, 28 \times 28, 24 \times 24\}. The number of sub-networks is \( n = 3 \) and the width lower bound is \( \alpha = 0.8 \).

The results are compared in Table 2. GradAug is comparable with the state-of-the-art CutMix, and it clearly outperforms the best structure regularization method ShakeDrop, which validate the effectiveness of the self-guided augmentation to the raw gradients. We further illustrate this by comparing GradAug+ with CutMix + ShakeDrop. On WideResNet, ShakeDrop severely degrades...
Table 2: Cifar-100 classification accuracy of different techniques on WideResNet and PyramidNet.

| Model | WideResNet-28-10 | PyramidNet-200 ($\tilde{\alpha} = 240$) |
|-------|------------------|---------------------------------------|
|       | Top-1 Acc (%)    | Top-5 Acc (%)                         | Top-1 Acc (%) | Top-5 Acc (%) |
| Baseline | 81.53            | 95.59                                 | 83.49         | 94.31         |
| + Mixup [10] | 82.5             | -                                     | 84.37         | 96.01         |
| + CutMix [11] | **84.08**        | **96.28**                             | 84.83         | 97.08         |
| + ShakeDrop [20] | 81.65            | 96.19                                 | 84.57         | 97.08         |
| + GradAug (Ours) | 83.98            | 96.17                                 | **84.98**     | **97.08**     |
| + CutMix + ShakeDrop | 81.64            | 96.46                                 | 85.93         | 97.63         |
| + GradAug+ (Ours) | **85.25**        | **96.85**                             | **86.24**     | **97.33**     |

The Top-1 accuracy of CutMix by 2.44%, while GradAug consistently improves CutMix by more than 1 point. The reason is that ShakeDrop introduces random noises to the training process, which is unstable and ineffective in some cases. However, GradAug is a self-guided augmentation to the gradients, which makes it compatible with various structures and data augmentation methods.

4.3 Ablation study

We first study the contribution of random width sampling and random scale transformation to the performance, respectively. Next, we show the impact of the number of sub-networks $n$ and the width lower bound $\alpha$ based on WideResNet-28-10 on Cifar-100. Finally, we explore GradAug variants by investigating different image transformations, e.g., random rotation transformation and the combination of random rotation and scale transformations.

Random width sampling and random transformation. We study the effect of one component by abandoning the other one. First, we do not randomly sample sub-networks. Then GradAug becomes multi-scale training in our experiments. In each iteration, we feed different scaled images to the network. Second, we do not conduct random scale transformation. In each iteration, we sample 3 sub-networks and feed them with the original images. The results are shown in Table 3. Random scale and random width sampling only achieve marginal improvements over the baseline, but GradAug remarkably enhances the baseline (+2.43%). This reaffirms the effectiveness of our method, which unifies data augmentation and structure regularization in the same framework for better performance.

Table 3: Contribution of random width sampling and random scale on Cifar-100.

|       | WideResNet-28-10 | Top-1 Acc (%) | Top-5 Acc (%) |
|-------|------------------|--------------|--------------|
| Baseline | 81.53            | 95.59        |              |
| RandScale | 82.27            | 96.16        |              |
| RandWidth | 81.74            | 95.56        |              |
| GradAug | 83.98            | 96.17        |              |

Figure 2: Effect of number of sub-networks and width lower bound.

Number of sub-networks and width lower bound. There are two hyperparameters in GradAug, the number of sub-networks $n$ and sub-network width lower bound $\alpha$. We first explore the effect of $n$. 
Table 4: Effect of different transformations in GradAug on Cifar-100 dataset.

| Model                | Top-1 Acc (%) | Top-5 Acc (%) |
|----------------------|---------------|---------------|
| WideResNet-28-10     | 81.53         | 95.59         |
| + GradAug-RandScale  | 83.98         | 96.17         |
| + GradAug-RandRot    | 83.36         | 96.11         |
| + GradAug-RandRot&Scale | 84.21     | 96.49         |

Other settings are the same as Section 4.2. The results are shown in Figure 2. A larger $n$ tends to achieve higher performance since it involves more self-guided gradient augmentations. The accuracy plateaus when $n \geq 3$. Note that even one sub-network can significantly improve the baseline. Then we investigate the impact of width lower bound $\alpha$ by fixing other settings. As shown in Figure 2, $\alpha = 0.8$ achieves the best accuracy, but all the values clearly outperform the baseline. GradAug is not sensitive to these hyperparameters. Empirically, we can set $n \geq 3$ and $\alpha \in [0.7, 0.9]$.

**Different transformations in GradAug.** In previous experiments, we use random scale transformation to regularize sub-networks because it is simple and effective. In this experiment, we explore other transformations, including random rotation transformation and the combination of scale and rotation transformations, and analyze the effect on the performance. The training setting of random rotation transformation is the same as random scale transformation. The only difference is that input images are randomly rotated by a degree from $\{0\degree, 90\degree, 180\degree, 270\degree\}$. The combination of scale and rotation transformations follows the same settings. The input images are first randomly rotated and then randomly resized. As shown in Table 4, both random scale and random rotation transformations achieve significant improvements over the baseline. And the combination of both can further enhance the performance. This again validates the effectiveness of our proposed self-guided augmentation to the raw gradients.

### 4.4 Object detection and instance segmentation

To evaluate the generalization ability of the learned representations by GradAug, we finetune its ImageNet pretrained model for COCO [29] object detection and instance segmentation. The experiments are based on Mask-RCNN-FPN [4, 30] framework and MMDetection toolbox [31] on ResNet-50 backbone. Mixup and CutMix, two most effective methods in image classification, are employed for comparison. As explained in Section 2, Mixup and CutMix are mixed sample data augmentation methods, which cannot be applied to object detection and segmentation. Therefore, we compare these methods by directly finetuning their ImageNet pretrained models on COCO dataset. All models are trained with $1 \times$ schedule on COCO dataset. The image resolution is $1000 \times 600$. The mean Average Precision (AP at IoU=$0.50\sim 0.95$) is reported in Table 5. We can see that although Mixup and CutMix achieve large improvements on ImageNet classification, the learned representations can barely benefit object detection and segmentation. In contrast, GradAug-pretrained model remarkably improves the performance of Mask-RCNN. This validates that GradAug enables the model to learn well-generalized representations which transfer well to other tasks.

Moreover, the training procedure of GradAug can be directly applied to the detection framework. Following the setting in ImageNet classification, the number of sub-networks $n$ is 4 and the width lower bound $\alpha$ is 0.9. We also use the simple random scale transformation. The shorter edge of the input image is randomly resized to one of $\{600, 500, 400, 300\}$ while keeping the aspect ratio. We only do sub-network sampling on the network backbone. The FPN neck and detection head are shared among different sub-networks. For simplicity, sub-networks are trained with the ground truth rather than soft labels. Other settings are the same as the default settings of $1 \times$ training schedule in MMDetection. The result (last line of Table 5) shows that GradAug further boosts the performance as compared with GradAug-pretrained and significantly improves the baseline by $+1.7$ det mAP and $+2.1$ seg mAP. Some visual examples are shown in Figure 3. GradAug-trained model can effectively detect small scale and large scale objects and is robust to occlusions.
Table 5: COCO object detection and instance segmentation based on Mask-RCNN-FPN.

| Model               | ImageNet Cls Acc (%) | Det mAP | Seg mAP |
|---------------------|-----------------------|---------|---------|
| ResNet-50 (Baseline)| 76.3 (+0.0)           | 36.5 (+0.0) | 33.3 (+0.0) |
| Mixup-pretrained    | 77.9 (+1.6)           | 35.9 (-0.6) | 32.7 (-0.6) |
| CutMix-pretrained   | 78.6 (+2.3)           | 36.7 (+0.2) | 33.4 (+0.1) |
| GradAug-pretrained  | 78.8 (+2.5)           | 37.7 (+1.2) | 34.5 (+1.2) |
| GradAug             | 79.6 (+3.3)           | 38.2 (+1.7) | 35.4 (+2.1) |

Table 6: Corruption error of ResNet-50 trained by different methods. The lower the better.

| Model      | Clean Err | Noise | Blur | Weather | Digital | mCE   |
|------------|-----------|-------|------|---------|---------|-------|
|            | Gauss     | Shot  | Impulse | Defocus | Glass | Motion | Zoom | Snow | Frost | Fog | Bright | Contrast | Elastic | Pixel | JPEG | mCE |
| ResNet-50  | 23.7      | 72    | 75    | 76      | 77     | 91     | 82   | 81   | 78    | 65   | 59     | 65      | 89     | 72    | 75   | 73.5|
| + Cutout   | 22.9      | 72    | 74    | 77      | 77     | 91     | 80   | 80   | 77    | 65   | 58     | 64      | 89     | 75    | 76   | 73.5|
| + Mixup    | 22.1      | 68    | 72    | 72      | 75     | 88     | 75   | 74   | 70    | 70   | 55     | 55      | 61     | 85    | 65   | 72   | 70.5|
| + CutMix   | 21.4      | 72    | 74    | 76      | 77     | 91     | 78   | 78   | 77    | 75   | 62     | 56      | 65     | 87    | 77   | 74   | 74.6|
| + GradAug  | 21.2      | 72    | 72    | 79      | 78     | 90     | 80   | 80   | 73    | 72   | 61     | 55      | 64     | 87    | 64   | 71   | 73.2|
| + GradAug+ | 20.4      | 71    | 73    | 78      | 76     | 91     | 78   | 77   | 72    | 71   | 61     | 53      | 63     | 86    | 76   | 69   | 73.0|

Figure 3: Object detection and instance segmentation visual examples.

4.5 Model robustness

Deep neural networks are easily fooled by unrecognizable changes on input images. Developing robust machine learning models is pivotal for safety-critical applications. In this section, we evaluate the model robustness to two kinds of permutations, image corruptions and adversarial attacks.

Image corruption. ImageNet-C dataset [32] is created by introducing a set of 75 common visual corruptions to ImageNet classification. ImageNet-C has 15 types of corruptions drawn from four categories (noise, blur, weather and digital). Each type of corruption has 5 levels of severity. Corruptions are applied to validation set only. Models trained on clean ImageNet should be tested on the corrupted validation set without retraining. We follow the evaluation metrics in [32] to test ResNet-50 trained by different regularization methods. The mean corruption error (mCE) is reported in Table 6. Mixup has lower mCE than other methods. We conjecture the reason is that Mixup proportionally combines two samples, which is in a similar manner to the generation of corrupted images. GradAug outperforms the second best competing method CutMix by +1.4%. Note that GradAug can also be combined with Mixup to make the model more robust to image corruptions.

Adversarial attack. We also evaluate model robustness to adversarial samples. Different from image corruption, adversarial attack uses a small distortion which is carefully crafted to confuse a classifier. We use Fast Gradient Sign Method (FGSM) [33] to generate adversarial distortions and conduct white-box attack to ResNet-50 trained by different methods. The classification accuracy on adversarially attacked ImageNet validation set is reported in Table 7. Note that here Mixup is not as robust as to image corruptions, which validates our aforementioned conjecture in the image corruption experiment. GradAug and CutMix are comparable and both significantly outperform other methods. GradAug+ remarkably improves over GradAug and CutMix, manifesting superiority of our self-guided gradient augmentation.

Table 6: Corruption error of ResNet-50 trained by different methods. The lower the better.
Table 7: ImageNet Top-1 accuracy after FGSM attack. $\epsilon$ is the attack severity.

| Model             | $\epsilon = 0.05$ | $\epsilon = 0.10$ | $\epsilon = 0.15$ | $\epsilon = 0.20$ | $\epsilon = 0.25$ |
|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| ResNet-50         | 27.90             | 22.65             | 19.50             | 17.04             | 15.09             |
| + Cutout          | 27.22             | 21.55             | 17.51             | 14.68             | 12.37             |
| + Mixup           | 30.76             | 25.59             | 21.63             | 18.44             | 16.19             |
| + CutMix          | 37.73             | 33.42             | 29.69             | 26.29             | 23.26             |
| + GradAug         | 36.51             | 31.44             | 27.70             | 24.93             | 22.33             |
| + GradAug+        | 40.26             | 35.18             | 31.36             | 28.04             | 25.12             |

Table 8: Top-1 accuracy on Cifar-10 and STL-10 with limited labels.

| Model                      | Cifar-10         | STL-10          |
|----------------------------|------------------|-----------------|
|                            | 250              | 1000            | 4000            | 1000             |
| WideResNet-28-2            | 45.23±1.01       | 64.72±1.18      | 80.17±0.68      | 67.62±1.06       |
| + CutMix (p=0.5)           | 43.45±1.98       | 63.21±0.73      | 80.28±0.26      | 67.91±1.15       |
| + CutMix (p=0.1)           | 43.98±1.15       | 64.60±0.86      | 82.14±0.65      | 69.34±0.70       |
| + ShakeDrop                | 42.01±1.94       | 63.11±1.22      | 79.62±0.77      | 66.51±0.99       |
| + GradAug                  | 50.11±1.21       | 70.39±0.82      | 83.69±0.51      | 70.42±0.81       |
| + GradAug-semi             | 52.95±2.15       | 71.74±0.77      | 84.11±0.25      | 70.86±0.71       |
| Mean Teacher [34]          | 48.41±1.01       | 65.57±0.83      | 84.13±0.28      | -                |

4.6 Low data setting

Deep neural network models suffer from more severe over-fitting when there is only limited amount of training data. Thus we expect regularization methods to show its superiority in low data setting. However, we find that state-of-the-art methods are not as effective as supposed. For a fair comparison, we follow the same hyperparameter settings in [35]. The backbone network is WideResNet-28-2. We first evaluate different methods on Cifar-10 with 250, 1000, 4000 labels. Training images are sampled uniformly from 10 categories. We run each model on 5 random data splits and report the mean and standard deviation in Table 8. We observe that CutMix (p=0.5) and ShakeDrop even degrade the baseline model performance, especially when labels are very limited. CutMix mixes images and their labels, which introduces strong noises to the data and ground truth labels. This is effective when there is enough clean labels to learn a good baseline. But when the baseline is weak, this disturbance is too severe. We reduce the impact of CutMix by setting p=0.1, where CutMix is barely used during training. CutMix still harms the baseline when there are only 250 labels, but it becomes beneficial when there are 4000 labels. ShakeDrop has a similar trend with CutMix since it introduces noises to the structure. In contrast, GradAug significantly and consistently enhances the baseline in all cases because it generates self-guided augmentations to the baseline rather than noises. Moreover, GradAug can be easily extended to semi-supervised learning (SSL). For each unlabeled image, we first feed it to the full-network and use the output of the full-network as its pseudo-label. Then we can leverage unlabeled images and their pseudo-labels to train sub-networks as stated in GradAug. Our GradAug-semi can further improve the performance over GradAug. It even achieves comparable performance with Mean Teacher [34], which is a popular SSL algorithm. We also evaluate the methods on STL-10 dataset [36]. The dataset is designed to test SSL algorithms, where the unlabeled data are sampled from a different distribution than labeled data. Similarly, CutMix and ShakeDrop are not effective while GradAug and GradAug-semi achieve clear improvements.

5 Conclusion

In this paper, we propose GradAug which introduces self-guided augmentations to the network gradients during training. The method is easy to implement while being effective. It achieves a new state-of-the-art accuracy on ImageNet classification. The generalization ability is verified on COCO object detection and instance segmentation. GradAug is also robust to image corruption and adversarial attack. We further reveal that current state-of-the-art methods do not perform well in low data settings, while GradAug consistently enhances the baseline in all cases.
References

[1] 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. 1, 2

[2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778. 1, 2, 5

[3] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in Advances in neural information processing systems, 2015, pp. 91–99. 1

[4] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969. 1, 7

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” IEEE transactions on pattern analysis and machine intelligence, vol. 40, no. 4, pp. 834–848, 2017. 1

[6] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440. 1

[7] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting.” Journal of Machine Learning Research, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html 1, 2, 4

[8] T. DeVries and G. W. Taylor, “Improved regularization of convolutional neural networks with cutout,” arXiv preprint arXiv:1708.04552, 2017. 1, 2, 5

[9] P. Y. Simard, D. Steinkraus, J. C. Platt et al., “Best practices for convolutional neural networks applied to visual document analysis.” in Icedar, vol. 3, no. 2003, 2003. 1, 2

[10] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” in International Conference on Learning Representations, 2018. [Online]. Available: https://openreview.net/forum?id=r1Ddp1-Rb 1, 2, 5

[11] 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 Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 6023–6032. 1, 2, 5, 6

[12] A. Neelakantan, L. Vilnis, Q. V. Le, I. Sutskever, L. Kaiser, K. Kurach, and J. Martens, “Adding gradient noise improves learning for very deep networks,” arXiv preprint arXiv:1511.06807, 2015. 1, 2

[13] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, “Deep networks with stochastic depth,” in European conference on computer vision. Springer, 2016, pp. 646–661. 1, 2, 4, 5

[14] G. Larsson, M. Maire, and G. Shakhnarovich, “Fractalnet: Ultra-deep neural networks without residuals,” arXiv preprint arXiv:1605.07648, 2016. 1, 2, 4, 5

[15] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, “Autoaugment: Learning augmentation strategies from data,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2019, pp. 113–123. 2

[16] G. Ghiasi, T.-Y. Lin, and Q. V. Le, “Dropblock: A regularization method for convolutional networks,” in Advances in Neural Information Processing Systems, 2018, pp. 10 727–10 737. 2, 5

[17] C. Summers and M. J. Dinneen, “Improved mixed-example data augmentation,” in 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2019, pp. 1262–1270. 2

[18] R. Takahashi, T. Matsubara, and K. Uehara, “Ricap: Random image cropping and patching data augmentation for deep cnns,” in Asian Conference on Machine Learning, 2018, pp. 786–798. 2

[19] X. Gastaldi, “Shake-shake regularization,” arXiv preprint arXiv:1705.07485, 2017. 2
[20] Y. Yamada, M. Iwamura, T. Akiba, and K. Kise, “Shakedrop regularization for deep residual
learning,” IEEE Access, vol. 7, pp. 186 126–186 136, 2019. 2, 5, 6
[21] D. Han, J. Kim, and J. Kim, “Deep pyramidal residual networks,” in Proceedings of the IEEE
conference on computer vision and pattern recognition, 2017, pp. 5927–5935. 2, 5
[22] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for
discriminative localization,” in Proceedings of the IEEE conference on computer vision and
pattern recognition, 2016, pp. 2921–2929. 3
[23] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing
internal covariate shift,” arXiv preprint arXiv:1502.03167, 2015. 3
[24] J. Yu and T. S. Huang, “Universally slimmable networks and improved training techniques,” in
Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 1803–1811. 3
[25] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for
discriminative localization,” in Proceedings of the IEEE conference on computer vision and
pattern recognition, 2016, pp. 2921–2929. 3
[26] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing
internal covariate shift,” arXiv preprint arXiv:1502.03167, 2015. 3
[27] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, “Imagenet: A large-scale hierarchical
image database,” in 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009,
pp. 248–255. 5
[28] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick,
“Microsoft coco: Common objects in context,” in European conference on computer vision.
Springer, 2014, pp. 740–755. 7
[29] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid
networks for object detection,” in Proceedings of the IEEE conference on computer vision and
pattern recognition, 2017, pp. 2117–2125. 7
[30] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu et al., “Mmdetection: Open mmlab detection toolbox and benchmark,” arXiv preprint arXiv:1906.07153,
2019. 7
[31] D. Hendrycks and T. Dietterich, “Benchmarking neural network robustness to common corrup-
tions and perturbations,” arXiv preprint arXiv:1903.12261, 2019. 8
[32] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,”
arXiv preprint arXiv:1412.6572, 2014. 8
[33] A. Tarvainen and H. Valpola, “Mean teachers are better role models: Weight-averaged consist-
tency targets improve semi-supervised deep learning results,” in Advances in neural information
processing systems, 2017, pp. 1195–1204. 9
[34] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow, “Realistic evaluation of deep
semi-supervised learning algorithms,” in Advances in Neural Information Processing Systems,
2018, pp. 3235–3246. 9
[35] A. Coates, A. Ng, and H. Lee, “An analysis of single-layer networks in unsupervised feature
learning,” in Proceedings of the Fourteenth International Conference on Artificial Intelli-
cence and Statistics, ser. Proceedings of Machine Learning Research, G. Gordon, D. Dunson, and
M. Dudík, Eds., vol. 15. Fort Lauderdale, FL, USA: PMLR, 11–13 Apr 2011, pp. 215–223.
[Online]. Available: http://proceedings.mlr.press/v15/coates11a.html 9