SmoothMix: a Simple Yet Effective Data Augmentation to Train Robust Classifiers

Jin-Ha Lee[1,2], Muhammad Zaigham Zaheer[1,2], Marcella Astrid[1,2], Seung-Ik Lee[1,2]
[1]University of Science and Technology, [2]Electronics and Telecommunications Research Institute,
Daejeon, South Korea
{jhlee, mzz, marcella.astrid}@ust.ac.kr, the_silee@etri.re.kr

Abstract

Data augmentation has been proven effective which, by preventing overfitting, not only enhances the performance of a deep neural network but also leads to a better generalization even with limited dataset. Recently introduced regional dropout based data augmentation strategies remove (or replace) some parts of an input image with a desideratum to make the network focus on less discriminative portions of an image, which results in an improved performance. However, such approaches usually possess 'strong-edge' problem caused by an obvious change in the pixels at the positions where the image is manipulated. It may not only impact on the local convolution operation but can also provide clues for the network to latch on to, which do not align well with the fundamental philosophy of augmentation. In order to minimize such peculiarities, we introduce $Smoothmix$ in which blending of images is done based on soft edges and the training labels are computed accordingly. Extensive analysis performed on CIFAR-10, CIFAR-100 and ImageNet for image classification demonstrates state-of-the-art results. Furthermore, $Smoothmix$ significantly increases the robustness of a network against image corruption which is validated by the experiments carried out on CIFAR-100-C & ImageNet-C corruption datasets.

1. Introduction

Due to the recent advancements in deep convolutional neural networks (DCNNs), significant performance improvements have been observed in various computer vision tasks such as image classification [9, 35, 22, 18], object detection [36, 21], tracking [38, 53], anomaly detection [57, 31, 12], and action recognition [14, 2].

These improvements can be attributed to various network structures [35, 18, 22], sophisticated training algorithms [16, 33, 26] as well as rapid increase in computation power [29]. However, tricky optimization methods, elevated training time consumption and expensive data collection process are still among the major challenges in training deep networks.

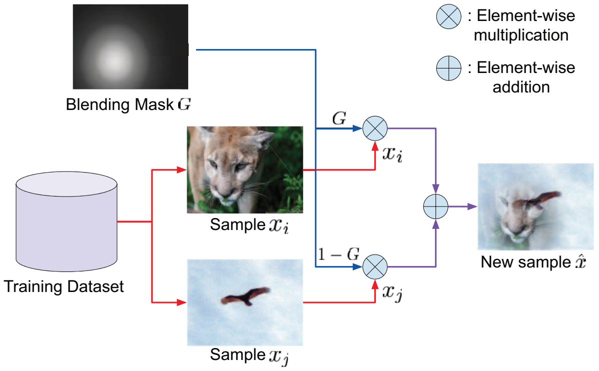

Figure 1. Overview of our proposed$Smoothmix$ framework. Two images (x_i, x_j), sampled from the training dataset, are added after an element-wise multiplication with masks G and $1 - G$ respectively. The resultant image $\hat{x}$ is a mix of the two input images, smoothly merged into each other.

Specifically, given that the DCNNs are capable to learn complex underlying representations of data through vast amounts of parameters and the number of parameters usually increase with the complexity of tasks, it is important to have sufficient amount of data in order to train these deep networks successfully and to achieve good performance. However, obtaining data is not easy as it involves laborious efforts in terms of acquisition and annotation. Therefore, the capabilities of deep networks is often limited by the available amount of training data. This may leads a network towards over-fitting, hence degrading its generalization performance. Thus, to alleviate the over-fitting and memorization problems, various data augmentation strategies [10, 58, 56, 47, 59, 50, 45, 25, 49, 7, 32] have been proposed to increase the total amount of training data by manipulating the existing data in various ways. Data augmentation not only allows more information to be extracted

2160-7516/20/$31.00 ©2020 IEEE
DOI 10.1109/CVPRW50498.2020.00386

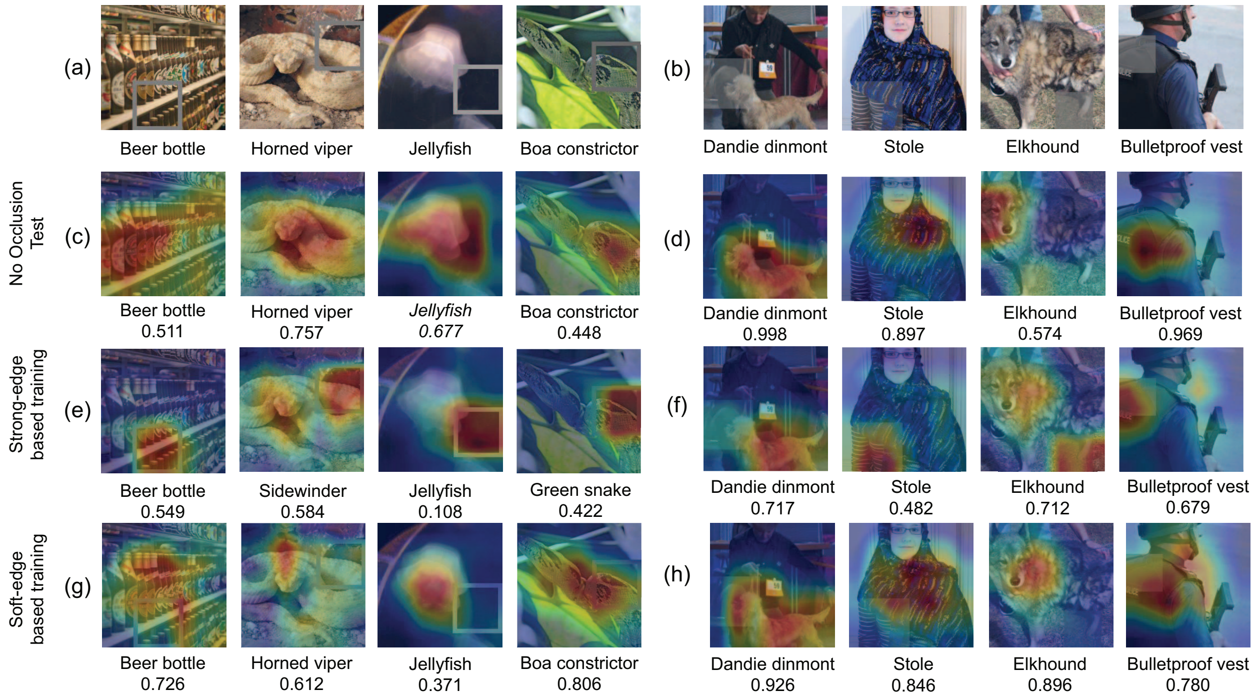

Figure 2. Attention visualization of three different models with Class Activation Map (CAM) using ImageNet test set to examine the impacts of strong edges on training. (a): Test images occluded with square windows. (b): Test images occluded with translucent square boxes. (c) & (d): Results of the vanilla ResNet model tested without any occlusion (clean image). (e) & (f): Results of the model trained using strong-edge based regional dropout method [56]. The CAM results show that the network often latches on to the occlusions due to the presence of strong edges of the occluded areas. (g) & (h): Results of our proposed method (SmoothMix) trained using soft-edge based regional dropout. It can be observed that the model is not effected by the occlusions as much as its other counterpart.

from limited data which usually is not possible otherwise without additional structural changes to the network [44], but it also can be applied together with other existing regularization techniques such as *Dropout* [46], *Batch Norm* [30], *Pretraining* [13], or *Transfer learning* [54] to further enhance the overall effects.

Basic augmentation methods can be as simple as manipulations in a single image space such as flip, rotate, crop, scale, translate, color transform, geometric transform, etc. which have been widely experimented in various existing methods [18, 35, 43]. In addition, regional dropout based methods [10, 11, 15, 6, 46] have also been proposed which prevent the network from focusing only on the characteristic points and encourage it to recognize an object by considering the entire information within an input. Generally, it is done by intentionally dropping random areas of the input [10, 59, 45]. Though regional dropout methods help to improve network generalization, Yun *et al*. [56] pointed out that such deletion of regions may also result in a loss of information, and suggested to cut and paste several portions of different training data in order to prevent this loss as well as to minimize the addition of meaningless pixels such as zero padding [10, 45] or random noise [59].

Overall, these regional dropout methods may lead towards a 'strong-edge' problem due to the drastic change in pixels because of the dropped (or pasted) regions [47, 10, 56, 15, 59, 45]. Consequently, two side-effects can be induced: first, unusual sudden change in pixels may affect local convolution operations. Second, such appearances become a characteristic that a network can latch on to, therefore conflicting with the primary purpose of regional dropout.

In order to investigate the impacts of training with strong-edge based regional dropout, we devised an experiment in which a square box is drawn randomly over test images before inferring a trained model. Figure 2 shows the Class Activation Map (CAM [60]) results as well as the class prediction probability output of three different models including vanilla ResNet [22], a model trained using strong-edge based regional dropout, and a similar but soft-edges based trained model. Two types of tests have been conducted, one occluded with randomly positioned boxes having 50% transparency and no borders while the other occluded with randomly positioned boxes having 100% transparency and thick borders, as shown in Figures 2a & 2b. The purpose of this experiment is to examine whether strong-edge based dropout contributes towards a deteriorated performance of the model. It can be observed that in the case of the model trained with strong-edge based regional dropout (Figures 2e & 2f), the focus of the network is diverted and it can be seen latching on to the occluded region. While the results also show that the class prediction of this strong-edge based regional dropout model is often similar to the other two models, the probability output is not as high as the approach with soft-edge based regional dropout. For example, in the Beer Bottle case, the Resnet model in Figure 2c predicts correct class with a probability of 0.511. Strong-edge based trained model, although improves over the vanilla ResNet model, gets distracted by the box occlusion and does not show significant improvement in the confidence scores. In contrary, the soft-edge based model in Figure 2g, shows a significant improvement in class prediction scores over its counterpart models.

Based on these observations, we propose to minimize the strong-edge problem by introducing smooth change in the boundary between the two contributing images. Therefore, instead of cropping and pasting one portion of an image onto the other, we generate a smoothly transitioning mask and use this to blend two different training images to form an augmented sample. Example shown in Figure 1 demonstrates that our proposed image blending approach does not leave any strong-edge, hence removes the aforementioned vulnerabilities present in the conventional systems. The proposed approach avoids strong-edge and partially preserves pixel information from both images by shifting gradually rather than a sudden change in the values, meanwhile maintaining the regional focusing effect of the dropout based augmentation methods. It demonstrates state-of-the-art results for image classification task by achieving error rates of 2.98% on CIFAR-10, 14.47% on CIFAR-100 and 22.25% on ImageNet Datasets. Moreover, due to the pixel-level manipulation of training images, the method also demonstrates robustness against image corruptions by yielding 2.96% and 1.03% performance gains over its counterpart approaches on ImageNet-C and CIFAR-100-C datasets, respectively.

In summary, the contributions of our paper are as follows:

- It is among the first few to discus the strong-edge problem in regional dropout methods and devise a solution for it.
- The proposed method achieves state-of-the-art performance in image classification task on CIFAR-10 [34], CIFAR-100 [34] and ImageNet [8].
- Extensive analysis on CIFAR-100-C and ImageNet-C [24] corruption datasets demonstrates that our proposed approach also provides significant robustness against image corruption, making it well suited towards real-world applications where noise is susceptible.

2. Related Works

Data augmentation can help in preventing overfitting and improving generalization capability of a network. The widely popular augmentation methods are simple manipulations in image space such as flip, rotate, crop, scale, translate, color transform, geometric transform, etc. which have been utilized by various researchers[18, 35, 43].

Furthermore, several regularization techniques such as *Dropout* [46], *Batch Normalization* [30], *Pretraining* [13], *Transfer learning* [54] are also employed for the same objective of improving generalization. Some researchers also proposed feature dropout as a regularization [55, 28] whereas Ghiasi *et al*. [15] used dropout in convolution filters. Another domain of research is to find optimal combinations of augmentation techniques for a robust training on diverse datasets [7, 19, 25]. Our approach is based on regional dropout in which portions of images are replaced with either noise or another image. However, in this section, we also introduce several other types of data augmentation strategies such as kernel filter, noise injection, and image mixing, adopted in the recent works.

Kernel filter augmentation is to modify a training image by using different kernels to achieve effects such as blurring [17] to reduce noise effect or sharpening [5] to highlight clear edges. He *et al*. [20] used Gabor function as kernel filters with an aim to catch directional representations in images. Kang *et al*. [32] proposed *PatchShuffle* to randomly shuffle the pixels with a kernel window to regularize the network by distorting the original form of an object.

Noise injection is a method of injecting noises into an image with a goal to increase the robustness of a network against occlusions and resolution degradation [42, 4, 39]. Vincent *et al*. [52] proposed to train an auto-encoder with noise injection to learn strong representations that are robust to partial corruption in input. He *et al*. [23] introduced *Parametric Noise* in which a trainable layer is used to inject Gaussian noise into activation.

Image mixing combines several images to create new samples as well as labels which are usually proportional to the ratio between labels of the contributing images. Such mixing can be performed either in image space [58, 50] or feature space [51]. Raphael *et al*. [37] proposed to overlay a Gaussian noise patch on an image to generate local disturbance. However, its original label is preserved as the operation does not involve other images. Summers and Dinneen [47] suggested to adopt several mixup methods and to create region-focused labels.

Regional dropout is commonly performed as image-level

```
Algorithm 1: SmoothMix pseudo code.
Input  : Input images $x_i$, $x_j$, and corresponding
         labels $y_i$, $y_j$
Output: Blended image $\hat{x}$ and label vector $\hat{y}$ of the
         blended image
if $probability > p$ then
    $G, \lambda$ = generate_mask($shape\_properties$)
    $\hat{x}$ = (G$\otimes x_i$) $\oplus$ ((1−G) $\otimes x_j$)
    $\hat{y} = \lambda\, y_i + (1-\lambda)\, y_j$
    return $\hat{x}$, $\hat{y}$
end
```

augmentation which randomly replaces part of an image with zeros or random noise while preserving the original label [10, 59]. Choe *et al*. [6] proposed attention based dropout to improve the localization of a network in which *important* regions of an image are deleted and the network is forced to learn the representations of *less-important* regions. [45] also proposed a grid mask based approach in which portions of images corresponding to several boxes of grids are deleted randomly. The architectures proposed in [56, 49] take advantage of both image mixing and regional dropout methods, by filling the dropped regions with patches from other images to improve the utilization of information as well as the computations.

Overall, our proposed method is different from all kernel filters and noise injection based methods as well as most of the image mixing and regional dropout based methods as our approach takes two different images to create an augmented input. In essence, our method is similar to the works in [37, 56, 47]. However, we attempt to reduce the effects of strong edges which are commonly present in these conventional approaches. We propose to smoothly blend two different images from the training dataset not only to avoid the problems of training based on strong edges but also make the network robust to noise in the test images.

3. Proposed Method

In this section, we present our $Smoothmix$ approach. The overall process is shown in Algorithm 1. At the beginning of each iteration, whether to perform $Smoothmix$ is randomly decided by a probability p. Then, given that the $Smoothmix$ is being applied, a smoothly transitioning mask G is generated based on the shape properties such as width, height, spread, etc of the mask. Two training images x_i and x_j are then element-wise multiplied with G and $1-G$, respectively, and the outputs are element-wise added to create a new training sample $\hat{x}$. The corresponding label vector $\hat{y}$ is then generated using the labels y_i and y_j of the input images x_i and x_j. Details of each step are discussed next:

3.1. Mask Generation

Generation of a mask G is dependent on several factors such as width, height, spread of the smoothing area, etc. Moreover, based on the shape of the mask, various properties may be included or excluded. For example, in our proposed approach, we conduct experiments with two different kinds of masks i.e. circular and square. More details on this are provided in the subsequent portions of this section. Generally, G can be defined using its center point coordinates $[\mu_w, \mu_h]$ and spread σ in the image space. μ_w and μ_h are uniformly sampled from within the range of width W and height H respectively, of the input images. The overall formulation is given as:

$$\mu_w \sim \text{Unif}(0, W) \quad , \quad \mu_h \sim \text{Unif}(0, H). \tag{1}$$

Furthermore, σ defines the spread of a mask which means increasing its value will increase the size of the masked region and widen the smoothing area between x_i and x_j.

Given the *shape_properties* $\in \{\sigma, \mu, W, H\}$, a mask G can then be defined as: $G \in [0, 1]^{W \times H}$. Moreover, as given in Algorithm 1, λ is an accumulated sum of the mask area representing the ratio between the two contributing images. It is a critical variable as it helps in determining the label of the resultant image created after the augmentation. For a mask G, λ can be calculated as:

$$\lambda = \frac{\sum_{i=1}^{W} \sum_{j=1}^{H} G_{ij}}{W \times H}, \tag{2}$$

where G_{ij} denotes pixel values of ith row and jth column in G. λ can also be seen as the average strength of all active pixel values within G.

3.2. Blending Images and Labeling

Given G and λ, an augmented training sample $\hat{x}$ and its label $\hat{y}$ can be generated as shown in Algorithm 1. Given, the dataset D contains images $x \in R^{W \times H \times C}$, where each image x_i is paired with its label y_i as $(x_i, y_i) \sim D$. An augmented image $\hat{x}$ and its label $\hat{y}$ for training can be generated from two arbitrary images x_i, x_j and the corresponding labels y_i and y_j, as:

$$\hat{x} = (G \otimes x_i) \oplus ((1 - G) \otimes x_j), \tag{3}$$

where $\otimes$ and $\oplus$ denote element-wise multiplication and addition, respectively. The resultant image $\hat{x}$ contains portions of the two input images blended smoothly into each other.

Label $\hat{y}$ of the augmented image $\hat{x}$ is generated based on the proportional ratio λ between the labels y_i and y_j of the two contributing images x_i and x_j as:

$$\hat{y} = \lambda y_i + (1 - \lambda) y_j, \tag{4}$$

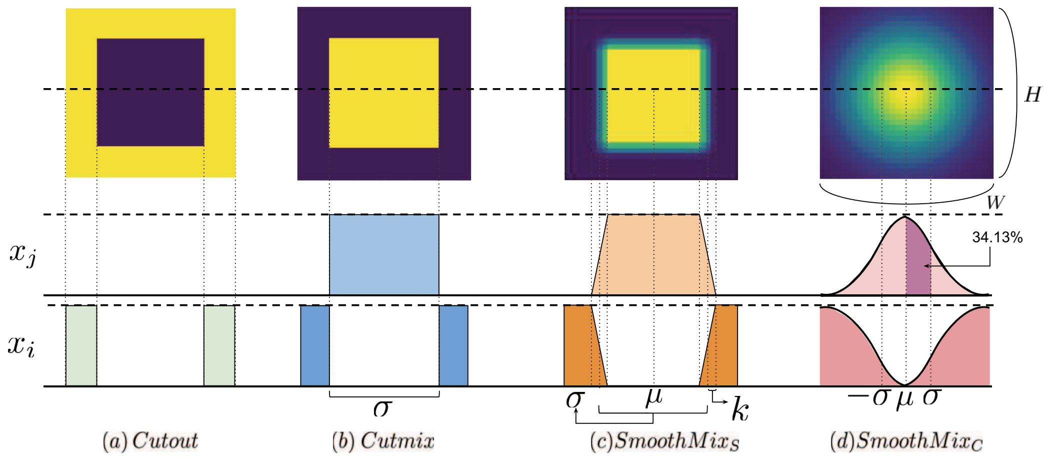

Figure 3. Comparison of the masks used in conventional architectures with the masks used in our proposed approach. x_i and x_j are the two images used to create an augmented image. The graph below each mask shows the ratio between x_i and x_j on a horizontally sliced plane placed in the center of the image. (a) Cutout [58]: Random rectangular region of an input image is filled with zero pixels. (b) Cutmix [56]: Cut and paste a patch of x_j onto x_i. Strong edges appear around the boundaries of these pasted patches. (c) $SmoothMix_S$: Our proposed approach to cut and paste patches. The images are linearly interpolated around the boundaries, hence alleviating the strong-edge problem. (d) $SmoothMix_C$: Our proposed approach to use a Gaussian circle mask to blend x_i and x_j. As a result, the boundary between images become smooth and ambiguous.

3.3. Types of Masks

As previously explained, the strong-edge phenomenon can often impact negatively on the performance of regional dropout based augmentation approaches. Examples in Figure 2 show that the network trained using such fashion may latch on to the boundaries or occluded regions due to the presence of strong edges.

To overcome this phenomenon, we propose to soften the edges which can effectively hide the locations of merger and helps in better generalization of a network. Among the various design choices possible for the shape of the mask in our proposed approach, we systematically selects two different kinds in our experiments.

Square mask: Most of the existing augmentation methods are based on square or rectangular shapes [10, 56, 58, 47, 49, 59, 45, 15]. Therefore, to have a fair comparison, we adopt a similar masking technique as proposed in conventional approaches [56, 47] with the only difference of smoothed borders. Throughout the rest of our paper, this approach will be referred to as $SmoothMix_S$. As shown in Figure 3c, $SmoothMix_S$ contains smooth border values which gradually dissipate in the outward direction. The range of smooth region k is kept proportional to the patch size. From one end of the border to the other, images x_i and x_j are linearly blended, resulting in a smooth transition. One dimensional mask in $SmoothMix_S$ is given as:

$$G_{dim} = \begin{cases} max(0, -\frac{1}{2k}(x-(\mu_{dim}+\frac{\sigma}{2}))+\frac{1}{2}) & \text{, if } x \geq \mu_{dim}+\frac{\sigma}{2}-\frac{\sigma}{k} \\ max(0, \frac{1}{2k}(x-(\mu_{dim}-\frac{\sigma}{2}))+\frac{1}{2}) & \text{, if } x \leq \mu_{dim}-\frac{\sigma}{2}+\frac{\sigma}{k} \\ 1 & \text{, otherwise} \end{cases} \tag{5}$$

where, σ is the span of the mask and $dim \in \{w, h\}$. Finally, the two-dimensional G is created as:

$$G = G_w \otimes G_h \tag{6}$$

This approach preserves the same amount of blended pixel information as compared to the similar conventional methods [10, 56], however due to the blended borders, it reduces the chances of a network to latch on to a property that may help in locating the added regions.

Circle mask: In addition to the $SmoothMix_S$, we also exploit the Gaussian distribution to generate circular masks, which are intuitively more viable in our proposed setting. This configuration will be referred to as $SmoothMix_C$ throughout the rest of our paper. Similar to the $SmoothMix_S$, $SmoothMix_C$ also blends two images with pixel-level ratio estimated using a mask G. However, G in this case takes the shape of a Gaussian distribution, generated by an outer product of two one-dimensional Gaussian distributions as follows:

$$G_w = e^{-\frac{(x-\mu_w)^2}{2\sigma^2}} \quad , \quad G_h = e^{-\frac{(x-\mu_h)^2}{2\sigma^2}} \tag{7}$$

Finally, the two-dimensional G is created using Equation 6. A visualization of this mask is provided in Figure 3d.

4. Experiments

The evaluation results of $Smoothmix$ for image classification on three different image datasets are reported in this section. In addition, robustness tests on two image corruption datasets as well as ablation studies for hyper-parameter selection are also provided.

Model	Top-1 ERR(%)	Top-5 ERR(%)
Baseline(PyramidNet-200) [18]	16.45	3.69
+ Stochdepth [28]	16.73	3.37
+ Cutout [10]	16.53	3.65
+ DropBlock [15]	15.73	3.26
+ Mixup [58]	15.63	3.99
+ Manifold Mixup [51]	16.14	4.07
+ Shakedrop [55]	15.08	**2.72**
+ Cutmix [56]	**14.47**	2.97
+ $SmoothMix_S$	14.74	3.3
+ $SmoothMix_C$	**14.47**	2.99

Table 1. Image classification results on CIFAR-100 dataset. Best and second-best are highlighted as bold and italic respectively.

Evaluation criteria Considering its popularity of usage in many existing works [28, 10, 58, 56, 47, 25, 59], our performance comparison of the image classification results is based on top-1 and top-5 error rates (ERR).

Parameters and Implementation Details. All models are implemented and experimented in Pytorch [40]. Image samples x_i and x_j are arbitrarily selected within a mini-batch, with a random permutation of the index to obtain pairs. In all our experiments related to $SmoothMix_S$, k is set to 0.2 and σ is uniformly sampled between 0 to 1. Whereas for experiments related to $SmoothMix_C$, sigma is obtained randomly in the range $0.25 \sim 0.5$ of the training image size. Until stated otherwise, above mentioned experimental settings have been used. However, in an ablation study presented in the subsequent parts of this section, we also examine the effects of these hyper-parameters.

4.1. Image Classification

4.1.1 Dataset

CIFAR-100 [34] This small-scale but challenging image dataset contains 60,000 images belonging to 100 different classes such as aeroplane, bird, cat, etc. Its train split contains $50,000$ images whereas test split consists of $10,000$ images. Each image is of 32×32 pixels resolution. Our approach is implemented using PyramidNet-200 [18] with the widening factor $\tilde{\alpha} = 240$. Training is conducted for 300 epochs with the mini-batch of size 32. The initial learning rate is set to 2.5×10^{-1} and decayed after every 75 epochs by a factor of 10^{-1}. Cross entropy loss [41] and stochastic gradient decent [3] are adopted for optimization. Probability p to perform $Smoothmix$ is set to 0.5.

CIFAR-10 [34] It is also a small-scale dataset which contains a total of $60,000$ images belonging to 10 classes. All images are of 32×32 pixels resolution. Similar to the CIFAR-100 dataset, the training split in CIFAR-10 also has 50000 images whereas the test split contains 10000 images. Training properties of our models are also kept identical to the experimental setting in CIFAR-100 dataset.

Imagenet [8] It is a large-scale dataset of 1.2 million images belonging to 1,000 different classes. The dataset also contains 50,000 publicly available validation images. To keep the consistency in experimental setup with other related works [15, 18, 48, 27], five standard image augmentation techniques i.e. scaling, cropping, flipping, jittering and lighting are also used. Resnet-50 [22] with $\alpha = 1$ is used for training. Training is carried out for 300 epochs with a mini-batch size of 128. The initial learning rate is set to 10^{-1} and decayed after every 75 epoch by a factor of 10^{-1}. Cross entropy loss [41] and stochastic gradient decent [3] are also employed. The probability p to perform $Smoothmix$ is set to 1.

Model	Top-1 ERR(%)
Baseline(PyramidNet-200 [18])	3.85
+ Cutout [10]	3.1
+ Mixup [58]	3.09
+ Manifold mixup [51]	3.15
+ Cutmix [56]	**2.88**
+ $SmoothMix_C$	2.98

Table 2. Image classification results on CIFAR-10 dataset.

4.1.2 Image classification results

Image classification is among the challenging problems in computer vision in which a network should recognize the objects inside images and predict the classes accurately. The results of the experiments carried out using CIFAR-100, CIFAR-10 [34] and ImageNet [8] datasets are provided with comparisons against other conventional approaches [28, 10, 15, 58, 51, 7, 55, 56, 25, 45].

Results on CIFAR-100. The experiments conducted on CIFAR-100 dataset serves two purposes. In addition to evaluate our method against the conventional architectures, we also compare the performances of our proposed two approaches $SmoothMix_S$ and $SmoothMix_C$. As seen in Table 1, while $SmoothMix_S$ outperforms several strong edge based regional dropout methods [10, 15], it falls slightly behind $SmoothMix_C$. Based on this observation, for the rest of the evaluation in this paper, we only employ $SmoothMix_C$ for standard comparisons. It can also be noticed in Table 1 that $SmoothMix_C$ demonstrates state-of-the-art results by achieving a Top-1 EER of 14.47% and a Top-5 EER of 2.99%.

Results on CIFAR-10. Table 2 provides comparison of our approach with other state-of-the-art methods [10, 58, 51, 56]. $SmoothMix_C$ shows a slight degradation of 0.1% in performance compared to [56] by achieving a Top-1 EER of 2.98%. However, not only it achieves a performance gain of 0.87% over baseline, it also surpasses other related methods trained for the said task.

Results on ImageNet. Table 3 provides the image classification performance results of $SmoothMix_C$ and its comparison against state-of-the-art methodologies [10, 45, 22, 58, 51, 7, 25, 15, 56]. $SmoothMix_C$ yields an absolute

Model	Top-1 ERR(%)	Top-5 ERR(%)
Baseline(Resnet-50) [22]	23.68	7.05
+ Cutout [10]	22.93	6.66
+ Hide-and-Seek [45]	22.8	x
+ StochDepth [28]	22.46	6.27
+ Mixup [58]	22.58	6.4
+ Manifold Mixup [51]	22.5	6.21
+ AutoAugment [7]	22.4	x
+ AugMix [25]	22.47	6.06
+ Drop block [15]	21.87	5.98
+ Cutmix [56]	**21.4**	**5.92**
+ $SmoothMix_C$	22.34	6.37

Table 3. Image classification results on Imagenet dataset.

gain of 1.34% in Top-1 ERR and a gain of 0.68% in Top-5 ERR. Overall, our approach demonstrates comparable results. In addition, we also demonstrate in the following portions of this section that our model shows superior performance in terms of stability and robustness against noise.

4.2. Corrupted Image classification

4.2.1 Datasets

As proposed by Hendrycks *et al*. [24], several corruptions are systematically introduced in the test splits of CIFAR-100 [34] and ImageNet [8] datasets to generate CIFAR-100-C and ImageNet-C. This method is generally employed to measure resilience of a trained model against data shift.

ImageNet-C It consists of 15 different image corruptions sub-categorized into four major types i.e. noise, blur, weather, and digital corruptions. Each of these 15 corruptions are applied at five different levels, showing different intensities of the corruption added to the dataset.

CIFAR-100-C It consists of 19 different corruptions, sub-categorized into the same four major types i.e. noise, blur, weather, and digital corruptions. There is only one level of corruption intensity in this dataset.

In order to evaluate $SmoothMix_C$, we define a baseline architecture which employs strong-edge based regional dropout approach. In essence, the baseline is kept similar to the model introduced in [56] and the performance of $SmoothMix_C$ against this baseline is reported in the following subsections. The reason we select this architecture is because it has shown exceptional performance against several state-of-the-art methods in image classification tasks, as shown in the previous section. For implementation and experimentation, the model trained with ImageNet dataset, provided online [1] by Yun *et al*. [56], is used.

4.2.2 Corrupted image classification results

Results on CIFAR-100-C Results from the experiments on this dataset are reported in Table 4. It can be seen that $SmoothMix_C$ performs robustly against several types of image corruptions. Overall, it yields an average improvement of 1.03% top-1 ERR when compared with the baseline. In total, $SmoothMix_C$ outperforms the baseline in fourteen out of the nineteen total image corruption categories which shows its robustness against noise in the images.

Corruption type		Baseline[56]	$SmoothMix_C$
Noise	Gaussian_noise	92.38	**88.4**
	Shot_noise	85.29	**80.67**
	Impulse_noise	**83.13**	91.94
	Speckle_noise	82.99	**79.49**
Blur	Defocus_blur	29.78	**30.5**
	Glass_blur	**74.47**	79.71
	Motion_blur	36.27	**34.52**
	Zoom_blur	**33.05**	35.07
	Gaussian_blur	39.53	**39.05**
Weather	Brightness	18.89	**18.01**
	Fog	26.19	**21.52**
	Frost	40.73	**35.41**
	Snow	32.06	**30.12**
	Spatter	**23.82**	31.05
Digital	Saturate	26.9	**25.81**
	Pixelate	46.53	**42.86**
	Contrast	33.72	**26.06**
	Elastic_transform	**31.85**	32.37
	Jpeg_compression	54.1	**49.6**
	Average	46.93	**45.90**

Table 4. Image classification results on CIFAR-100-C corruption dataset.

Results on ImageNet-C The detailed comparisons between the results from our proposed $SmoothMix_C$ and the baseline are provided in Table 5. It can be seen that $SmoothMix_C$ shows better performance in all five levels of corruptions added to the test images. Over the baseline, it also yields an average improvement of 2.97% in Top-1 ERR. Overall, our proposed approach shows superior average results in nine out of fifteen image corruption categories, while demonstrating comparable performance in the other six.

The robustness of our proposed approach against noise can be attributed to the smooth edge based training which encourages our network to overlook the deteriorated image quality at the edges of mask and focus on the rest of the image. Therefore, in terms of noisy images, our proposed approach overlook the noise regions and yet classifies the object successfully. It may also be the reason why our model does not perform well in the case of blurred dataset, as shown in Table 4 and 5.

4.2.3 System Stability

In order to evaluate the capability of our system to produce similar results on several training runs, we conducted an experiment on CIFAR-100 dataset. $SmoothMix_C$ and the baseline models are repeatedly trained five times and an av-

ImageNet-C		level 1		level 2		level 3		level 4		level 5		Average.Level	
Corruption Type		Baseline[56]	$SmoothMix_C$	Baseline[56]	$SmoothMix_C$	Baseline[56]	$SmoothMix_C$	Baseline[56]	$SmoothMix_C$	Baseline[56]	$SmoothMix_C$	Baseline[56]	$SmoothMix_C$
Noise	Gaussian_noise	37.94	**35.46**	47.46	**43.56**	62.71	**58.02**	78.49	**74.00**	**99.64**	99.69	65.25	**62.15**
	Shot_noise	40.41	**37.59**	51.95	**48.55**	65.77	**62.72**	82.47	**81.02**	90.23	**90.20**	66.17	**64.02**
	Impulse_noise	52.49	**46.88**	61.39	**54.84**	67.25	**61.21**	80.64	**76.73**	91.08	**90.12**	70.57	**65.96**
	Average.Noise	43.61	**39.98**	53.60	**48.98**	65.24	**60.65**	80.53	**77.25**	93.65	**93.34**	67.33	**64.04**
Blur	Defocus_blur	**40.52**	42.89	**48.29**	49.36	**64.84**	66.17	**76.95**	78.06	**85.08**	86.45	**63.14**	64.59
	Glass_blur	**47.03**	47.75	**62.08**	62.78	**84.83**	85.77	**89.44**	90.31	**92.66**	93.63	**75.21**	76.05
	Motion_blur	**35.00**	35.95	**45.10**	45.95	**62.17**	62.23	78.36	**78.32**	**86.01**	86.10	**61.33**	61.71
	Zoom_blur	**45.82**	45.98	55.68	**54.79**	63.60	**62.33**	69.97	**68.40**	76.03	**74.56**	62.22	**61.21**
	Average.Blur	**42.09**	43.14	**52.79**	53.22	**68.86**	69.13	**78.68**	78.77	**84.95**	85.19	**65.47**	65.89
Weather	Snow	44.88	**41.87**	67.26	**60.02**	64.11	**58.50**	75.27	**70.09**	81.06	**73.95**	66.52	**60.89**
	Frost	39.24	**35.75**	56.06	**47.49**	67.50	**56.95**	69.49	**58.03**	76.27	**64.49**	61.71	**52.54**
	Fog	35.51	**30.87**	40.56	**32.34**	49.14	**35.78**	56.00	**40.36**	73.62	**52.60**	50.97	**38.39**
	Brightness	**26.70**	27.55	**27.85**	28.57	**29.80**	30.42	**33.59**	33.67	39.25	**38.88**	**31.44**	31.82
	Average.Weather	36.58	**34.01**	47.93	**42.11**	52.64	**45.41**	58.59	**50.54**	67.55	**57.48**	52.66	**45.91**
Digital	Contrast	31.44	**30.82**	35.40	**33.39**	44.59	**39.10**	71.94	**57.80**	92.78	**80.86**	55.23	**48.39**
	Elastic_transform	**32.10**	33.12	**52.94**	54.52	47.48	50.59	**61.63**	64.72	**86.15**	88.23	**56.06**	58.24
	Pixelate	**35.06**	35.76	**38.43**	39.32	50.89	**46.56**	70.74	**61.06**	82.61	**71.99**	55.55	**50.94**
	Jpeg_compression	**34.62**	35.98	**38.16**	39.55	**40.70**	42.19	49.60	49.87	62.26	**59.32**	**45.07**	45.38
	Average.Digital	**33.31**	33.92	**41.23**	41.70	45.92	**44.61**	63.48	**58.36**	80.95	**75.10**	52.98	**50.74**
	Average.All	38.90	**37.76**	48.89	**46.50**	58.16	**54.95**	70.32	**66.23**	81.77	**77.78**	59.61	**56.64**

Table 5. Image classification results on ImageNet-C dataset.

Model	Average Top-1 ERR
Baseline[56]	14.95 ± 0.410
$SmoothMix_C$	$\mathbf{14.82 \pm 0.292}$

Table 6. Stability comparison on CIFAR-100 dataset.

Model	Top-1 ERR(%)
Baseline(PyramidNet-200 [18])	16.45
+ Center Gaussian($sigma = 4$)	$15.22 \pm 0.20(Best : 15.05)$
+ Center Gaussian($sigma = 12$)	$15.03 \pm 0.21(Best : 14.79)$
+ $SmoothMix$	$\mathbf{14.81} \pm 0.26(Best : \mathbf{14.47})$

Table 7. Impact of mask location tested on CIFAR-100 dataset.

Model	Top-1 ERR(%)
Baseline(PyramidNet-200 [18])	16.45
+ $SmoothMix(\sigma = 8)$	$14.94 \pm 0.18(Best : 14.67)$
+ $SmoothMix(\sigma = 12)$	$14.88 \pm 0.21(Best : 14.57)$
+ $SmoothMix(\sigma = 16)$	$\mathbf{14.79} \pm 0.23(Best : 14.49)$
+ $SmoothMix(\sigma = [8, 16])$	$14.81 \pm 0.26(Best : \mathbf{14.47})$

Table 8. Impact of mask size tested on CIFAR-100 dataset.

erage of the performance is reported in Table 6. The training setting is kept same as described in Section 4.1.2. It can be seen that our proposed approach shows stable performance by not only demonstrating low standard deviation but also a smaller average Top-1 ERR %.

4.3. Ablation on hyper-parameter selection

In a series of experiments, we explore different possible settings of $Smoothmix$ in terms of μ and σ. The performance of these experiments are compared against the baseline PyramidNet-200 [18] network, as described in Section 4.1.2. The experiments are conducted on CIFAR-100 dataset for image classification. Each experiment is repeatedly performed five times and statistics such as average, standard deviation, and top performance are reported in Tables 7 and 8. Table 7 shows the effects of changing μ values. μ, which is the center of G in our model, is sampled based on a Gaussian distribution of size sigma. It can be seen that with this expansion of sampling Gaussian distribution (increasing sigma as in Table 7), the model shows superior average performance. It depicts the importance of more randomization in the system to select several locations for masking.

Table 8 shows the effects of changing mask size. As explained in Section 3.1, σ defines the spread of G. Using fixed size sigma ranging from 8 to 16 in each experiment shows comparable performances. However, in the case we randomize the sigma value in one experiment, the method shows better performance than the other counterparts having fixed sigma values. This also demonstrates the importance of randomization in augmentation methods such as ours.

5. Conclusion

This paper presents $Smoothmix$, a data augmentation approach that generates mask with softened edges to smoothly blend two images with an aim to avoid 'strong-edge' problem. Alleviating the sudden change in pixel around the image boundary resulted in an overall improved performance. The proposed method demonstrates state-of-the-art results by yielding a top-1 error rate of 14.47% on CIFAR-100, 2.98% on CIFAR-10, and 22.25% on Imagenet dataset. Moreover, our model also depicts stability in training as well as robustness against image corruption by achieving 1.03% and 2.98% improved performance from the baseline on CIFAR-100-C and ImageNet-C corruption datasets, respectively.

6. Acknowledgment

This work was supported by the ICT R&D program of MSIP/IITP. [2017-0-00306, Development of Multimodal Sensor-based Intelligent Systems for Outdoor Surveillance Robots].

References

[1] clovaai/cutmix-pytorch. https://github.com/clovaai/CutMix-PyTorch. Accessed: 31- Mar- 2020. 7

[2] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016. 1

[3] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010. 6

[4] Stefan Braun, Daniel Neil, and Shih-Chii Liu. A curriculum learning method for improved noise robustness in automatic speech recognition. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 548–552. IEEE, 2017. 3

[5] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 3

[6] Junsuk Choe and Hyunjung Shim. Attention-based dropout layer for weakly supervised object localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2219–2228, 2019. 2, 4

[7] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 113–123, 2019. 1, 3, 6, 7

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3, 6, 7

[9] Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg, et al. Imagenet large scale visual recognition challenge. *arXiv preprint arXiv:1409.0575*, 2014. 1

[10] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 1, 2, 4, 5, 6, 7

[11] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Modeling visual context is key to augmenting object detection datasets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 364–380, 2018. 2

[12] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1801–1810, 2019. 1

[13] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660, 2010. 2, 3

[14] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12046–12055, 2019. 1

[15] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. In *Advances in Neural Information Processing Systems*, pages 10727–10737, 2018. 2, 3, 5, 6, 7

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1

[17] Richard A Haddad, Ali N Akansu, et al. A class of fast gaussian binomial filters for speech and image processing. *IEEE Transactions on Signal Processing*, 39(3):723–727, 1991. 3

[18] Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5927–5935, 2017. 1, 2, 3, 6, 8

[19] Ryuichiro Hataya, Jan Zdenek, Kazuki Yoshizoe, and Hideki Nakayama. Faster autoaugment: Learning augmentation strategies using backpropagation. *arXiv preprint arXiv:1911.06987*, 2019. 3

[20] Bo He, Yan Song, Yuemei Zhu, Qixin Sha, Yue Shen, Tianhong Yan, Rui Nian, and Amaury Lendasse. Local receptive fields based extreme learning machine with hybrid filter kernels for image classification. *Multidimensional systems and signal processing*, 30(3):1149–1169, 2019. 3

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 2, 6, 7

[23] Zhezhi He, Adnan Siraj Rakin, and Deliang Fan. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. pages 588–597, 2019. 3

[24] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 3, 7

[25] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 1, 3, 6, 7

[26] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *Neural computation*, 9:1735–80, 12 1997. 1

[27] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 6

[28] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016. 3, 6, 7

[29] Tim Hwang. Computational power and the social impact of artificial intelligence. *Available at SSRN 3147971*, 2018. 1

[30] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Proceedings of The 32nd International Conference on Machine Learning*, 2015. 2, 3

[31] Radu Tudor Ionescu, Fahad Shahbaz Khan, Mariana-Iuliana Georgescu, and Ling Shao. Object-centric auto-encoders and dummy anomalies for abnormal event detection in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7842–7851, 2019. 1

[32] Guoliang Kang, Xuanyi Dong, Liang Zheng, and Yi Yang. Patchshuffle regularization. *arXiv preprint arXiv:1707.07103*, 2017. 1, 3

[33] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014. 1

[34] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 3, 6, 7

[35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1, 2, 3

[36] Yudong Liu, Yongtao Wang, Siwei Wang, TingTing Liang, Qijie Zhao, Zhi Tang, and Haibin Ling. Cbnet: A novel composite backbone network architecture for object detection. *arXiv preprint arXiv:1909.03625*, 2019. 1

[37] Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin D Cubuk. Improving robustness without sacrificing accuracy with patch gaussian augmentation. *arXiv preprint arXiv:1906.02611*, 2019. 3, 4

[38] Matthias Minderer, Chen Sun, Ruben Villegas, Forrester Cole, Kevin P Murphy, and Honglak Lee. Unsupervised learning of object structure and dynamics from videos. In *Advances in Neural Information Processing Systems*, pages 92–102, 2019. 1

[39] Francisco J Moreno-Barea, Fiammetta Strazzera, José M Jerez, Daniel Urda, and Leonardo Franco. Forward noise adjustment scheme for data augmentation. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 728–734. IEEE, 2018. 3

[40] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 6

[41] Reuven Y Rubinstein and Dirk P Kroese. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2013. 6

[42] MJ Sáiz-Abajo, B-H Mevik, VH Segtnan, and T Næs. Ensemble methods and data augmentation by noise addition applied to the analysis of spectroscopic data. *Analytica chimica acta*, 533(2):147–159, 2005. 3

[43] Ikuro Sato, Hiroki Nishimura, and Kensuke Yokoi. Apac: Augmented pattern classification with neural networks. *arXiv preprint arXiv:1505.03229*, 2015. 2, 3

[44] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019. 2

[45] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *2017 IEEE international conference on computer vision (ICCV)*, pages 3544–3553. IEEE, 2017. 1, 2, 4, 5, 6, 7

[46] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 2, 3

[47] Cecilia Summers and Michael J Dinneen. Improved mixed-example data augmentation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1262–1270. IEEE, 2019. 1, 2, 3, 4, 5, 6

[48] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 6

[49] Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. Ricap: Random image cropping and patching data augmentation for deep cnns. In *Asian Conference on Machine Learning*, pages 786–798, 2018. 1, 4, 5

[50] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. Between-class learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5486–5494, 2018. 1, 3

[51] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, Aaron Courville, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. *International Conference on Machine Learning*, pages 6438–6447, 2019. 3, 6, 7

[52] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. 3

[53] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1328–1338, 2019. 1

[54] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):9, 2016. 2, 3

[55] Yoshihiro Yamada, Masakazu Iwamura, Takuya Akiba, and Koichi Kise. Shakedrop regularization for deep residual learning. *arXiv preprint arXiv:1802.02375*, 2018. 3, 6

[56] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6023–6032, 2019. 1, 2, 4, 5, 6, 7, 8

[57] Muhammad Zaigham Zaheer, Jin-ha Lee, Marcella Astrid, and Seung-Ik Lee. Old is gold: Redefining the adversarially learned one-class classifier training paradigm. *arXiv preprint arXiv:2004.07657*, 2020. 1

[58] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. 1, 3, 5, 6, 7

[59] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. 2020. 1, 2, 4, 5, 6

[60] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 2