Journal of Autonomous Intelligence (2024) Volume 7 Issue 5
doi: 10.32629/jai.v7i5.1518

ORIGINAL RESEARCH ARTICLE

HardMix: Considering Difficult Examples in Mixed Sample Data Augmentation

A. F. M. Shahab Uddin[1], Md Delowar Hosen[1], Md. Nasim Adnan[1], Syed Md Galib[1], Md. Alam Hossain[1], Sung-Ho Bae[2,*]

[1] *Department of Computer Science and Engineering, Jashore University of Science and Technology, Jashore 7408, Bangladesh*

[2] *School of Computing, Kyung Hee University, Seoul 17104, Republic of Korea*

*** Corresponding author:** Sung-Ho Bae, shbae@khu.ac.kr

ABSTRACT

Mixed sample data augmentation (MSDA) techniques enhance the generalization ability of deep learning models where the training samples and their labels are mixed to generate new samples. Those mixed (augmented) samples increase data diversity and combined with mixed labels, offer better localization and generalization ability of the model. The performance of MSDA highly depends on the selection of source patch to be mixed. Consequently, several methods, from random to careful selection of source patch using prior knowledge have been studied, to propose better augmentation strategy. We argue that besides the careful selection of the source patch, selecting the source sample from where the source patch will be cut, also plays an important role. Based on that, we propose HardMix that selects the source patch from hard samples (which are frequently being miss-classified by a model) to let the model better learn the feature of hard samples. We conduct comprehensive experiments on image classification task on several benchmark datasets using various state-of-the-art architectures to verify the effectiveness of the proposed method. HardMix achieves the best known top-1 error of 3.62%, and 3.54% for ResNet-18 and ResNet-50 architectures on CIFAR-10 classification dataset, respectively. Also, it achieves the best known top-1 error of 19.33%, 18.31%, and 16.21% for ResNet-18, ResNet-50, and WideResNet architectures on CIFAR-100 classification dataset, respectively. Moreover, the proposed HardMix data augmentation strategy outperforms state-of-the-art methods with a best known top-1 error of 21.20% and 20.01% on ImageNet validation dataset when applied using ResNet-50 and ResNet-101 architectures, respectively.

Keywords: HardMix; data augmentation; hard sample based data augmentation; generalization; mixed sample data augmentation; MSDA

ARTICLE INFO

Received: 2 January 2024
Accepted: 18 February 2024
Available online: 23 July 2024

1. Introduction

Deep learning models have shown outstanding performance in numerous fields, especially in vision based tasks such as image classification[1–3], object detection[4,5], and semantic segmentation[6–8], etc. With the advent of modern devices for parallel computing and improved training algorithms, it has become possible to increase models' depth significantly. As a result, today's Convolution Neural Networks (CNNs) typically have 10 to 100 million of learnable parameters. Such a huge number of parameters enable the deep CNNs to solve complex problems. However, besides the powerful representation ability, a huge number of parameters increase the probability of overfitting when the number of training examples is insufficient, which results in poor generalization of the model. Moreover, collecting and labeling data is an expensive and tedious job. As an alternative, advanced data augmentation strategies have widely been studied.

Random feature removal is one of the widely used methods which directs the CNNs to avoid focusing on certain tiny areas of the input images or on particular portions of internal activations, thereby improves the robustness of the model. Dropout[9,10] is a well-known feature-level training strategy which randomly disables some internal activations of a network to avoid overfitting. To enjoy similar benefits, regional dropout, an image-level data augmentation method known as Cutout[11], has been proposed that introduces the dropout effect by eliminating or modifying random regions of the input image. Cutout[11] guides a model to learn the full object region rather than focusing on a small part or activation and increase the model's generalization. However, the feature removal is undesirable since it eliminates informative pixels of the training images.

After regional dropout, several MSDA strategies have been proposed e.g., Mixup[12], CutMix[13], SaliencyMix[14], PuzzleMix[15], etc. MSDA techniques have shown impressive performance in enhancing models' generalization ability by generating more diverse samples. This technique mixes two or more training samples in order to increase data diversity. Besides that, since the augmented samples result in the occlusion of various portions of the mixed objects, it guides the model to learn the less discriminative part of an object instead of always learning the most important features such as the head of a dog, the face of a human, etc. As a result, it enhances the generalization and localization ability. Also, it improves model robustness to adversarial attacks[16].

Mixup[12] is a data augmentation technique that involves creating a new sample by linearly interpolating a pair of training samples to enhance data diversity. Although this approach enhances the generalization ability of a model, the augmented image seems locally ambiguous and unnatural[13]. However, it introduces a new set of data augmentation techniques and numerous methods have been proposed since then.

Yun et.al. proposed a successful MSDA technique called CutMix[13] that cuts a patch of a training sample (known as target image) in a random fashion and then instead of keeping it blank, replaces that region with a patch (known as source patch) from another training sample (known as source image). In addition, they suggest to mix the labels of the target and source images based on the ratio of the mixed patches. However, the random selection may select uninformative image regions e.g., a background patch and when mixing the image labels, it introduces label error since the background patch does not represent the corresponding object[14]. In order to solve that problem researchers have suggested to use some prior information when selecting the source patch and also several studies have been conducted to find the optimal mixing strategy[14,15].

SaliencyMix[14] proposed to select the source patch based on the most salient part of an image and then mix it to the target image. It prevents the method from selecting any uninformative patch and helps to solve the label error problem. PuzzleMix[15] proposed a very similar approach but solved a dual optimization problem to maximize the saliency information in the augmented image. Although this strategy also performs well, it increases the computational complexity due to the dual optimization problem. ResizeMix[17] proposed not to cut any patch from the source image but rather to down-sample the source image and mix it to the target image and also mix their labels in order to avoid label error. All of the abovementioned works focused on better selection of the source patch and their mixing location.

Following the success of MSDA, some of the recent works applied MSDA in semi-supervised learning, especially, in medical image domain[18,19]. Wang et al.[18] proposed a semi supervised learning framework for 3D medical image detection where they proposed to mix a labeled image with a pseudo labeled image segmentation for a better training. Since, the actual labels for detection tasks in medical imaging are bounding boxes, it is not possible to get something as meaningful as soft classes in classification by taking the linear interpolation of two sets of boxes. As a result, they used image level and object level mixing strategy with focal loss for training the model. Similarly, Qiao et al.[19] also used MSDA for the semi-supervised Computed Tomography (CT) lesion segmentation in medical image domain. Similar to Wang et al.[18], the authors proposed to make a pair of labeled and pseudo labeled images based on an uncertainty score predicted by Monte Carlo method and then used an existing MSDA method called SwapMix. Both of the above-mentioned works utilized MSDA for semi-supervised medical image domain to increase the labeled segmentation map information in the training data with a predicted pseudo label.

However, in this study, we find that besides selecting an optimal source patch it is also important to carefully select the source images from where the patch will be cut. It is well known that some samples within a dataset are challenging for machine learning models to predict or classify accurately, which are called hard samples[20–25]. We propose considering those hard samples of a mini-batch as the source images in an MSDA technique and guiding a model to learn a better feature representation of those samples and enhance the model's performance. This more effective data augmentation strategy is what we call "HardMix".

To evaluate the effectiveness of the proposed HardMix data augmentation, we perform extensive experiments on image classification task on various benchmark datasets, using state-of-the-art CNN architectures. In summary, the proposed method has obtained the top-1 error of 3.62%, and 3.54% for ResNet-18 and ResNet-50 architectures on CIFAR-10 classification dataset, respectively. Also, it achieves the top-1 error of 19.33%, 18.31%, and 16.21% for ResNet-18, ResNet-50, and WideResNet architectures on CIFAR-100 classification dataset, respectively. All of these results clearly indicate the effectiveness of the proposed HardMix data augmentation strategy. The contributions of this paper can be summarized as follows:

- To the best of our knowledge, we are the first to investigate the effect of source image selection in Mixed Sample-based Data Augmentation (MSDA).
- We propose HardMix, which suggests using only hard samples (that are difficult for a model to predict accurately) as source images to cut patches in MSDA. This technique helps to increase the appearance frequency of hard samples and enhances the feature representation learning of a model.
- The proposed method outperforms state-of-the-art data augmentation methods.

The rest of this paper is organized as follows. Section 2 presents related works including traditional data augmentations and MSDAs. Section 3 explains the proposed method. In section 4, we verify the proposed data augmentation method by performing experiments on image classification task. Section 4.7 discusses the strengths and weaknesses of the proposed method. Finally, section 5 summarizes the importance of proposed method and future opportunities.

2. Related works

2.1. Data augmentation

Data augmentation has become one of the most useful training strategies to enhance generalization ability of deep learning methods[11–15,17]. Traditional data augmentation techniques include horizontal and vertical flipping, rotation, random cropping and resizing, contrast changing, translation, random noise addition, etc. Those traditional data augmentation techniques have widely been used in training deep neural networks to enhance their generalization ability. Considering their effectiveness, several advanced data augmentation techniques have been studied since then[11–15,17].

2.2. Regional dropout

Dropout[9,10] is another effective approach to prevent overfitting problem. This technique randomly erases some neural connections of a network in order to restrain the model from memorising the training data distribution. In order to enjoy similar benefit as dropout, regional dropout methods have been proposed which offer similar mechanism of dropping internal connections but at the input space. Specifically, regional dropout randomly makes some portion of the input training samples blank to restrict information flow from that region. Cutout[11] is a regional dropout technique which randomly removes a region from the input image which makes the model not to focus on a small part of the internal activation, i.e., instead of focusing only on the important parts of an image, it lets the model to learn other less import features. As a result, it prevents overfitting, enhances the generalization ability and localization ability of the model.

2.3. MSDA

MSDA methods[12–15,17,26–29] have drawn much attention due to their promising augmentation performance. We divide them into two groups: (i) basic MSDA[12–14,17] and (ii) dual optimization based MSDA[15,26–29].

2.3.1. Basic MSDA

Mixup[12] is one such methods that suggests to mix two training samples instead of removing some regions, so that there is no blank pixels in the augmented image like as in regional dropout methods. It also proposed to mix the labels of the two samples which have been mixed up. The mixing ratio is controlled by a hyper-parameter λ.

However, the augmented image produced by Mixup[12] looks locally ambiguous. To solve the uninformative pixel problem caused by Cutout[11] and local ambiguity problem caused by Mixup[12], CutMix[13] proposed to cut a source patch and then mix it to the target image. The patch size is controlled by a mixing ratio λ. Then their labels are also mixed based on λ. This approach shown very promising performance in various tasks including image classification, object detection, and adversarial robustness. The source patch has been selected in a random fashion. However, SaliencyMix[14] has shown that the random selection of the source patch may introduce a severe problem of selecting a background region as a source patch and mixing the labels based on that patch introduces label error. Since the background patch may not represent the source object, it may mislead the classifier.

As a remedy, SaliencyMix[14] suggests selecting the source patch based on the most salient region of the source image instead of random selection. Specifically, a saliency map is extracted from the source image using an existing saliency detection algorithm and then the location of the most salient part of that map is determined to crop the source patch. It guarantees that source patch represents the object. This strategy enhances the generalization ability of a model.

However, we argue that besides the sophisticated selection of the source patch, it is also very crucial to select the appropriate source images from where the patches would be cut. As a result, we propose to select the hard samples from a training mini batch and use them as source images. Our method shows promising performance in various tasks.

2.3.2. Dual optimization based MSDA

This kind of augmentation methods[15,28,29] aim to minimize two loss functions, one is to maximize the saliency region in an image and another is to perform the actual task. PuzzleMix[15] proposed to maximize the saliency information in the augmented image and in order to do that, it solved a dual optimization problem. AutoMix[28] focused to build a bridge between the mixup generation and classification task with a unified optimization framework to improve the mixup training efficiency. AutoMix reformulates the mixup classification into two sub-tasks i.e., (i) mixed sample generation and (ii) mixup classification, with corresponding sub-networks and solves them in a bi-level optimization framework. For the generation, a learnable lightweight mixup generator is designed to generate mixed samples by modeling patch-wise relationships under the direct supervision of the corresponding mixed labels. SuperMix[29] proposed a supervised mixing augmentation method which exploits the salient regions within input images to construct mixed training samples. This work also applied a dual optimization framework and a knowledge distillation approach to maximize the saliency information in the augmented sample to obtain rich visual features. Although this strategy performs well, it increases the computational complexity due to the dual optimization problem.

2.4. Hard samples

Several works have been proposed to generate hard samples or investigate the effect of hard samples in various tasks[30–33]. Hu et al.[30] found out that the number of negative samples is larger than that of positive ones in a limited training set sample in Voice Spoofing Detection, which can bias the loss function. As a result, they proposed to exclude those hard samples (non-informative) from the training losses to alleviate the imbalance between simple and hard samples. Feng et al.[31] investigates different data augmentation techniques that can be used to generate sufficient data for training CNN-based facial landmark localization systems. First, they prepared the augmented samples by applying two types of augmentation as (i) textural augmentation: Gaussian blur, noise, jitter, and occlusion; (ii) geometric transformation: Flip, and bounding box perturbation. Those samples are generated by applying random textural and geometric variations to the original labeled training images. Some augmented samples may be harder and more effective for the training of a deep neural network and some may be less effective. To select the most effective augmented training samples, they proposed to select the hard samples which largely contribute to the loss.

In self-supervised image anomaly detection, augmenting samples is a promising approach. Usually, geometric transformation and adding noise are the commonly used methods to generate images with anomalies. Following a contrastive learning framework, Wang et al.[33] proposed to destroy the global semantic information of the original image for this task but keep the local semantic information intact, so that the model can learn the local underlying feature of those images. They denote those non-semantic information as hard negative samples. All of the above-mentioned works used the concept of hard samples. Among them, Hu and Zhou[30], Feng et al.[31], Wang et al.[32] proposed to filter out hard samples in some scale to prevent data imbalance problems. On the other hand, Wang et al.[33] proposed another augmentation process that destroys the global semantic information and produces non-semantic hard samples.

We focus on preparing a better source batch to cut source patches for Mixed Sample based Data Augmentation (MSDA). Recent studies suggest that the mixing strategy plays an important role in MSDA methods i.e., how to cut a patch from a source sample and how to mix it into a target image to make the data augmentation more effective. However, we argue that, besides the mixing strategy, preparing a better source sample pool is also important to make an MSDA method more effective. To do that, we proposed to use the hard samples to prepare the source batch and then apply the existing state-of-the-art (SOTA) mixing strategy. Instead of filtering out the hard samples to prevent data imbalance problem as in the abovementioned works, our method utilizes the hard samples to cut patches and then mix them to the original images to prepare more effective samples for image classification tasks.

2.5. Tricks for training deep networks

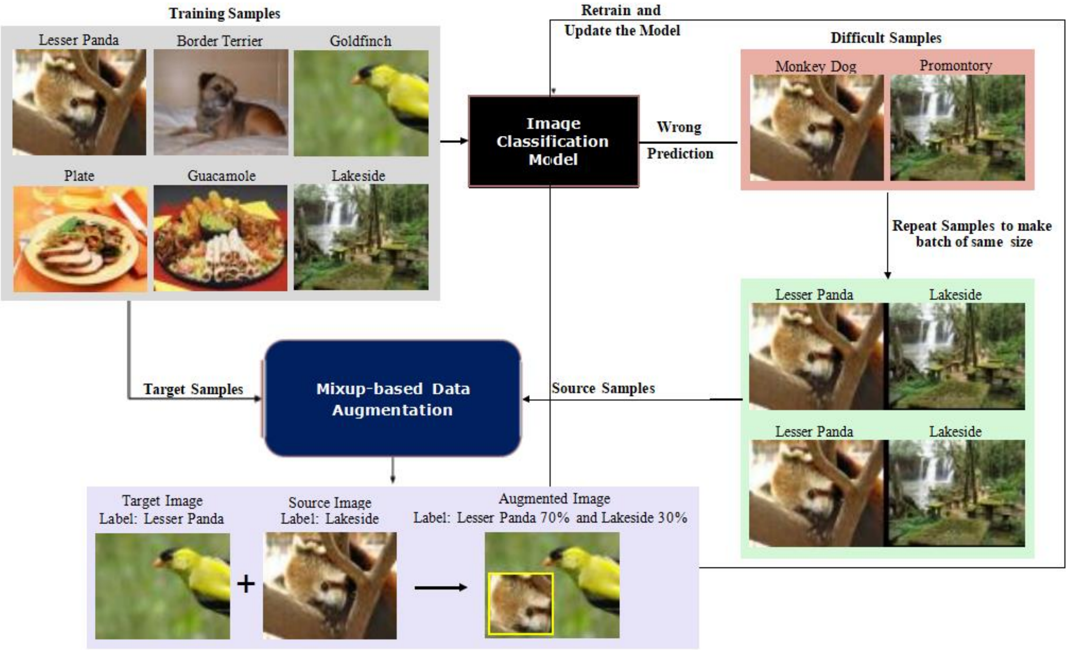
Since deep networks need a lot of computation power and data, efficient network training is one of the most crucial challenges facing the computer vision community. To effectively train deep networks, techniques like weight decay[34], dropout[9,10], and batch normalization[35] are frequently employed. Recent techniques to improve models' performance include adding noise to CNNs' internal features[35–37] or adding new paths to the architecture[38,39]. On the other hand, MSDA techniques, including the proposed HardMix, operate at the data level without modifying internal representations or architecture while significantly enhancing the models' performance and generalization ability.

2.6. Label smoothing

The class labels in object classification are often represented by a one-hot code, meaning that the true labels are anticipated to have a probability of precisely 1, while the others are anticipated to have a probability of exactly 0. In other words, it implies that the model is too sure of itself, which leads to overfitting to the training set. As a result, the models perform poorly on an unknown test dataset as a result. Label smoothing offers a solution to this issue by allowing the model confidence in the correct label to be relaxed by lowering the class probability to a little lower value, such as less than 1. It thus directs the model to be more adaptable rather than overconfident and eventually enhances the model's resilience and performance[40]. The proposed data augmentation method and other MSDA techniques offer label smoothing since they blend the class labels of the mixed samples.

3. Proposed method

The proposed data augmentation strategy follows the principle of MSDA techniques. In addition, we suggest to find to out the hard samples from a training mini-batch and use them as source images so that the underlying model can better learn the features of those difficult samples. **Figure 1** graphically presents the proposed method. We explain the proposed data augmentation strategy in this section.



Training Samples
Lesser Panda
Border Terrier
Goldfinch
Plate
Guacamole
Lakeside
Retrain and Update the Model
Image Classification Model
Wrong Prediction
Difficult Samples
Monkey Dog
Promontory
Repeat Samples to make batch of same size
Lesser Panda
Lakeside
Lesser Panda
Lakeside
Target Samples
Mixup-based Data Augmentation
Source Samples
Target Image Label: Lesser Panda
Source Image Label: Lakeside
Augmented Image Label: Lesser Panda 70% and Lakeside 30%
+

Figure 1. The proposed HardMix data augmentation. We first extract the hard samples of a minibatch to create a source batch. We call the original input batch "target batch" and the hard samples based batch as "source batch". Then cut a patch from the source image and mix it to the target image to generate an augmented image. We also mix their labels according to the ratio of the mixed patches.

3.1. Selecting the hard samples

Nowadays, deep learning models are highly capable of learning high dimensional complex data distribution, which is a key to their success. It becomes possible due to the sophisticated training process and the available training data. However, it is well known that not all the training samples are equally understandable by the models[20–25,41,42]. The samples which are difficult for the models to learn are known as "hard samples". Several approaches have been introduced to solve this problem. In this study, we reconsider this problem from the perspective of data augmentation technique. Our hypothesis is that if the hard samples are presented to the model at a higher frequency than easy samples, the model can gain more attention to those hard samples. As a result, this selective attention may enhance the model's performance. So, we propose to cut the source patches from hard samples of a mini-batch and mix them to the target images in order to increase the frequency of those hard samples to the model.

Suppose, we have an image classification model that needs to be trained on a dataset D. Our goal is to prepare a source mini-batch $\mathrm{I}_{\mathrm{S}}^{\mathrm{h}}$ containing only hard samples from where we shall cut the patches. In order to do that, we pass a mini-batch I to model M and find out the wrongly classified images denoted by I^{h}. Then, we apply refinement to correct the labels. We call it hard-sample mini-batch and denote it by $\mathrm{I}_{\mathrm{s}}^{\mathrm{h}}$. This process can bedefined as follows:

$$\hat{y}=\mathrm{M}(\mathrm{I}),\ I^h = I_{\hat{y}=y} \tag{1}$$

It is worth noting that the number of instances in I and $\mathrm{I}_{\mathrm{s}}^{\mathrm{h}}$ should be the same to apply augmentation. However, the number of samples in $\mathrm{I}_{\mathrm{s}}^{\mathrm{h}}$ may be smaller than the number of samples in I. So, we randomly duplicate the hard samples in the mini-batch $\mathrm{I}_{\mathrm{s}}^{\mathrm{h}}$ and shuffle them.

Algorithm 1 Pseudo code of HardMix

```
Data: $Dataset\ \mathcal{D}$
Output: $Augmented\ minibatch\ \tilde{I}$
Initialization: $Model\ \mathcal{M}$
for $each\ iteration$ do
    if $model == training$ then
        input I, target y = get_minibatch($\mathcal{D}$);
        model_prediction $\hat{y}$ = model_forward $\mathcal{M}(I)$;
        hard_samples $I^h$ = miss_classified_samples $I_{\hat{y}} \neq y$;
        if $length\ of\ I^h < I$ then
            length_diff $d$ = length($I$) − length($I^h$);
            $I_s^h$ = randomly repeat $d$ samples in $I^h$;
        end
        $I_s^h, y_s^h$ = shuffle_minibatch($I_s^h, y_s^h$);
        $\lambda \sim U(0,1)$;
        $P_x \sim U(0, W)$;
        $P_y \sim U(0, H)$;
        $P_w \sim \sqrt{(1-\lambda}$;
        $P_h \sim \sqrt{(1-\lambda}$;
        $x_1$ = round(clip(($P_x - P_w$)/2), min = 0));
        $x_2$ = round(clip(($P_x + P_w$)/2), max = $W$));
        $y_1$ = round(clip(($P_y - P_h$)/2), min = 0));
        $y_2$ = round(clip(($P_y + P_h$)/2), max = 0));

        /* Create the augmented samples I_a by mixing source and target samples */
        generate $I_a$ as follows:
        $I[:,:,x_1:x_2,y_1:y_2] = I_s^h[:,:,x_1:x_2,y_1:y_2]$;

        /* Adjust lambda to the exact ratio of the mixed areas. */
        $\lambda = 1 - (x_2 - x_1) * (y_2 - y_1)/(W * H)$;
        $y_a = \lambda * y + (1 - \lambda) * y_s^h$;
    end
    output = model_forward($input$);
    loss = compute_loss($output, target$);
    model_update();
end
```

3.2. Applying augmentation

Let $I_s \in \mathbb{R}^{W\times H\times C}$ denote a randomly selected training (source) image taken from the mini-batch $\mathrm{I}_{\mathrm{s}}^{\mathrm{h}}$ from where a source patch will be cut and y_s denote its label. Also, let $I_t \in \mathbb{R}^{W\times H\times C}$be another randomly selected training (target) image with label y_t, to where the source patch will be mixed. The goal is to partially mix I_t and I_s to produce a new training sample I_a, the augmented image, with label y_a. The mixing of two images can be defined as follows:

$$I_a = M \odot I_s + M' \odot I_t \tag{2}$$

where I_a denotes the augmented image, $M \in \{0,1\}^{W\times H}$ represents a binary mask, M' is the complement of M and $\odot$ represents element-wise multiplication. In doing so, first, a source patch location is randomly selected based on the size of the patch defined by a mixing ratio λ, where λ is sampled from the beta distribution

$Beta(\alpha, \alpha)$. Following CutMix[13] and SaliencyMix[14], α is set to 1 for all experiments. Then the corresponding location of the mask M is set to 1 and others to 0. The element-wise multiplication of M with the source image removes everything except the selected region. In contrast, M' does the opposite of M, i.e., the element-wise multiplication of M' with the target image keeps all the regions except the selected patch. Finally, the addition of those two creates a new training sample that contains the target image with the selected source patch in it (see **Figure 1**). Besides mixing the images we also mix their labels based on the size of the mixed patches as follows.

$$y_a = \lambda y_t + (1 - \lambda) y_s \tag{3}$$

where y_a denotes the label for the augmented sample and λ is the combination ratio. Following CutMix[13], we sample the binary mask M by randomly selecting a source patch region P defined by the bounding box coordinates $P = (P_x, P_y, P_w, P_h)$. Here, P_x and P_y denote the starting point of the bounding box on the x-axis and y-axis, respectively and P_w and P_h denote the width and height of the bounding box, respectively. The box coordinates are sampled from a uniform distribution. A source patch cropped from the source image Is based on P and mixed with the target image I_t, on the same location as specified by P. An algorithm of the proposed data augmentation strategy is presented in Algorithm 1.

4. Experiments and analysis

The effectiveness of the proposed data augmentation strategy has been evaluated for image classification using popular state-of-the-art (SOTA) architectures on benchmark datasets. Two NVIDIA GeForce RTX 2080 Ti GPUs have been used to perform all the experiments using PyTorch deep learning framework.

4.1. Datasets and models

We conduct comprehensive experiments using a variety of well-known models and image classification datasets to demonstrate the efficacy of our data augmentation technique. We choose ResNet[43] architectures including ResNet-18, ResNet-50, ResNet-101, and WideResNet[44]. Considering that they offer a wide range of architectural concepts, we choose these specific architectural designs. We evaluate the approach across various depths/sizes of each architecture with the help of the unique variants in each architecture. We pick CIFAR-10, CIFAR-100[45] and ImageNet[46] as the image classification datasets since they are well-known benchmarks that can be used to measure the effectiveness of different techniques.

4.2. Experimental setup

To avoid having any pretraining bias effect on the outcomes of our evaluation, we train each model from scratch. We train the baselines using the hyperparameter setups following their original publications. In addition, the proposed method is trained following the same training setup as the baselines. Given that our primary goal is to compare our data augmentation technique to others rather than to achieve state-of-the-art results, all data augmentation techniques for a specific architecture and dataset are run for a fixed number of epochs, which is sufficient for the models to converge. Specifically, we train the models using Stochastic Gradient Descent (SGD) with weight decay of 5×10^4, and Nesterov momentum of 0.9, for 200 epochs. The learning rate was primarily set to 0.1 and after 60, 120, and 160 epochs, the learning rate was decreased by a factor of 0.2 from its initial value of 0.1. Using the per-channel mean and standard deviation, the images are normalized. We conduct tests both with and without the use of a conventional data augmentation technique, which includes zero-padding, random cropping, and horizontal flipping. The Pytorch[47] framework is used to implement all the models, and to evaluate all data augmentation techniques. Top-1 and top-5 errors are used as performance metrics for comparison where the lower value represents better performance.

4.3. Results on CIFAR-10

CIFAR-10 dataset consists of 6000 images per class in 10 different classes. Those 60,000 color images are of size 32 × 32. The 10 classes include airplane, ships, trucks, frogs, horses, deer, cats, birds, cats, and cars. The dataset contains 10,000 test images and 50,000 training images. 1000 randomly chosen images from each class make up the test set.

Experimental results on CIFAR-10[45] dataset is presented in **Table 1** where the top-1 error is reported for all the methods in comparison. "CIFAR-10" and "CIFAR-10+" columns in the table represents the top-1 error of the corresponding methods with and without traditional data augmentation, respectively. For a better comparison, the results are reported on five-runs average. "CIFAR-10" in **Table 1** represents the results that have been reported when the methods were trained without applying any traditional data augmentation technique. On the other hand, "CIFAR-10+" in **Table 1** represents the results that have been reported when the methods were trained with traditional data augmentation techniques such as rotation, flipping, etc. It can be seen that for each of the architectures, the proposed HardMix data augmentation strategy outperforms all other methods in comparison. For ResNet-18 architecture, the proposed HardMix achieves the best known top-1 error of 8.51% and 3.62% when trained with and without traditional data augmentation, respectively. Similarly, for ResNet-50 architecture which is deeper than ResNet-18 architecture, the proposed method achieves a top-1 error of 8.78% and 3.54%, respectively. Finally, for a wider network WideResNet, our method achieves a top-1 error of 5.21% and 2.72% when trained with and without traditional data augmentation, respectively.

Table 1. Performance comparison of the SOTA data augmentation methods for image classification task on CIFAR-10 and CIFAR-100 datasets. The results are reported on five runs average. The dataset name followed by an extra "+" sign denotes that standard data augmentation methods were also applied during training.

Method	**Top 1 Error (%)**			
	CIFAR-10	**CIFAR-10+**	**CIFAR-100**	**CIFAR-100+**
ResNet-18 (Baseline)	10.63 ± 0.26	4.72 ± 0.21	36.68 ± 0.57	22.46 ± 0.31
ResNet-18 + Cutout	9.31 ± 0.18	3.99 ± 0.13	34.98 ± 0.29	21.96 ± 0.24
ResNet-18 + CutMix	9.44 ± 0.34	3.78 ± 0.12	34.42 ± 0.27	19.42 ± 0.23
ResNet-18 + SaliencyMix	8.63 ± 0.17	3.77 ± 0.08	33.89 ± 0.23	19.47 ± 0.21
ResNet-18 + HardMix	**8.51 ± 0.17**	**3.62 ± 0.08**	**33.74 ± 0.23**	**19.33 ± 0.21**
ResNet-50 (Baseline)	12.14 ± 0.95	4.98 ± 0.14	36.48 ± 0.50	21.58 ± 0.43
ResNet-50 + Cutout	8.84 ± 0.77	3.86 ± 0.25	32.97 ± 0.74	21.38 ± 0.69
ResNet-50 + CutMix	9.16 ± 0.38	3.61 ± 0.13	31.65 ± 0.61	18.72 ± 0.23
ResNet-50 + SaliencyMix	8.90 ± 0.35	4.01 ± 0.15	30.33 ± 0.43	18.42 ± 0.22
ResNet-50 + HardMix	**8.78 ± 0.35**	**3.54 ± 0.15**	**30.16 ± 0.43**	**18.31 ± 0.22**
WideResNet-28-10 (Baseline)	6.97 ± 0.22	3.87 ± 0.08	26.06 ± 0.22	18.80 ± 0.08
WideResNet-28-10 + Cutout	5.54 ± 0.08	3.08 ± 0.16	23.94 ± 0.15	18.41 ± 0.27
WideResNet-28-10 + AutoAugment	-	**2.60 ± 0.10**	-	17.10 ± 0.30
WideResNet-28-10 + PuzzleMix (200 epochs)	-	-	-	16.23
WideResNet-28-10 + CutMix	5.18 ± 0.20	2.87 ± 0.16	23.21 ± 0.20	16.66 ± 0.20
WideResNet-28-10 + SaliencyMix	5.35 ± 0.09	2.82 ± 0.09	22.43 ± 0.13	16.34 ± 0.14
WideResNet-28-10 + HardMix	**5.21 ± 0.09**	2.72 ± 0.09	**22.26 ± 0.13**	**16.21 ± 0.14**

4.4. Results on CIFAR-100

The CIFAR-100 dataset consists of 60,000 32 × 32 color images divided into 100 classes with 600 images each. Per class, there are 500 training images and 100 test images. There are 50,000 training images and 10,000 test images.

Experimental results on CIFAR-100[45] dataset is presented in **Table 1** where the top-1 error is reported for all the methods in comparison. For a better comparison, the results are reported on five-runs average. “CIFAR-100” in **Table 1** represents the results that have been reported when the methods were trained without applying any traditional data augmentation technique. On the other hand, “CIFAR-10+” in **Table 1** represents the results that have been reported when the methods were trained with traditional data augmentation techniques such as rotation, flipping, etc. It can be seen that for each of the architectures, the proposed HardMix data augmentation strategy outperforms all other methods in comparison. For ResNet-18 architecture, the proposed HardMix achieves the best known top-1 error of 33.74% and 19.33% when trained with and without traditional data augmentation, respectively. Similarly, for ResNet-50 architecture which is deeper than ResNet-18 architecture, the proposed method achieves a top-1 error of 30.16% and 18.31%, respectively. Finally, for a wider network WideResNet, our method achieves a top-1 error of 22.26% and 16.21% when trained with and without traditional data augmentation, respectively.

4.5. Results on ImageNet

ImageNet[46] contains 1.2 million training images and 50,000 validation images of 1000 classes. To perform experiments on ImageNet dataset, we apply the same settings as used in the study of Yun et al.[13], for a fair comparison. We have trained our HardMix for 300 epochs with an initial learning rate of 0.1 and decayed by a factor of 0.1 at epochs 75, 150, and 225, with a batch size of 256. Also, the traditional data augmentation techniques such as resizing, cropping, flipping, and jitters have been applied during the training process. **Table 2** presents the ImageNet experimental results where the best performance of each method is reported. HardMix outperforms all other methods in comparison. It drops top-1 error for ResNet-50 by 1.76%, 1.38%, 0.20%, 0.06% and 0.04% over Cutout[11], Mixup[12], CutMix[13], SaliencyMix[14] and PuzzleMix[15] data augmentation, respectively. For ResNet-101 architecture, the proposed HardMix achieves the new best result of 20.01% top-1 error and 5.09% top-5 error and outperforms all state-of-the-art data augmentation methods.

Table 2. Performance comparison (the best performance) of SOTA data augmentation strategies on ImageNet classification with standard model architectures.

Method	Top-1 Error (%)	Top-5 Error (%)
ResNet-50 (Baseline)	23.68	7.05
ResNet-50 + Cutout	22.93	6.66
ResNet-50 + StochasticDepth	22.46	6.27
ResNet-50 + Mixup	22.58	6.40
ResNet-50 + Manifold Mixup	22.50	6.21
ResNet-50 + AutoAugment	22.40	6.20
ResNet-50 + DropBlock	21.87	5.98
ResNet-50 + CutMix	21.40	5.92
ResNet-50 + PuzzleMix	21.24	5.71
ResNet-50 + SaliencyMix	21.26	5.76
ResNet-50 + HardMix	**21.20**	**5.69**
ResNet-101 (Baseline)	21.87	6.29
ResNet-101 + Cutout	20.72	5.51
ResNet-101 + Mixup	20.52	5.28
ResNet-101 + Cutmix	20.17	5.24
ResNet-101 + SaliencyMix	20.09	5.15
ResNet-101 + HardMix	**20.01**	**5.09**

4.6. Comparison with Dual Optimization based MSDA:

We also compare the image classification performance of the proposed HardMix data augmentation with dual optimization based MSDA methods. **Table 3** presents the performance comparison with dual optimization based data augmentation methods for image classification on benchmark datasets using standard architectures. It can be seen that the proposed method shows competitive performance with dual optimization based data augmentation methods which are computationally expensive. While the proposed HardMix achieves SOTA performance with a lower computational complexity as discussed in section 4.8.

Table 3. Performance comparison of the proposed data augmentation method with dual optimization based MSDA for image classification task on CIFAR-10, CIFAR-100, and ImageNet datasets.

Dataset	Baseline model	PuzzleMix[15]	AutoMix[28]	SuperMix[29]	HardMix
CIFAR-10	ResNet-18	2.9	2.66	-	3.62
	ResNet-50	2.73	2.35	-	3.54
CIFAR-100	ResNet-18	18.87	17.96	-	19.33
	ResNet-50	17.15	16.14	-	18.31
ImageNet	ResNet-50	21.14	20.75	22.4	21.20
	ResNet-101	19.33	19.02	-	20.01

4.7. Discussion

The experimental results show that the proposed method outperforms existing MSDA methods in terms of classification accuracy on every dataset. This performance enhancement can be attributed to the fact that the proposed HardMix guides a model to improve the implicit representation for the samples that were previously difficult to predict by the model. Specifically, when applying the augmentation, source patches are sampled from the hard samples for mixing into the target images. As a result, HardMix increases the appearance frequency of the hard samples during the training process and thereby helps the model to better learn the representation of hard samples. This phenomenon enhances the overall performance of the model.

4.8. Computational complexity

We inspect the computational complexity of all data augmentation methods in comparison, in terms of training time. The experiments are performed on CIFAR-10 dataset, where all the models were trained over 200 epochs using the ResNet-18 architecture. The time complexity comparison is shown in **Table 4**. The results suggest that the training time of the proposed HardMix is lower than SaliencyMix[14] and slightly higher than CutMix[13] due to finding out the difficult samples. However, it could be negligible considering the performance improvement compared to CutMix[13].

Table 4. On the CIFAR-10 dataset, a comparison of the training times for several data augmentation strategies utilizing the ResNet-18 architecture.

Time complexity						
Method	**ResNet-18 (Baseline)[43]**	**Mixup[12]**	**CutOut[11]**	**CutMix[13]**	**SaliencyMix[14]**	**Proposed method**
Time (hour)	0.83	0.87	0.84	0.89	0.90	0.91

5. Conclusion

MSDA has shown promising performance in enhancing generalization and localization ability of deep learning models. Following the success of MSDA techniques, we have proposed HardMix to cut patches only from hard samples in the augmentation process. Specifically, we consider the fact that some samples are challenging for machine learning models to predict or classify accurately, including data points that are outliers, have low-quality data, or have ambiguous labels. Considering this fact, HardMix finds the hard samples in a mini-batch based on a model's prediction, cuts patches from those hard samples, and then mixes them with the target samples to create augmented samples. Incorporating hard samples into the augmentation strategy can help the model to learn a better representation of those samples and improves overall performance. Extensive experiments on several tasks using a various state-of-the-art architectures, verifies the effectiveness of the proposed method. HardMix outperforms other SOTA methods in terms of top-1-error on image classification task using various standard architectures on several benchmark datasets.

Author contributions

Conceptualization, AFMSU, MNA and SHB; methodology, AFMSU and MDH; software, AFMSU and MDH; validation, MNA, SMG, MAH and SHB; formal analysis, AFMSU and SHB; investigation, MDH and MAH; resources, SMG and SHB; data curation, MDH; writing—original draft preparation, AFMSU and MDH; writing—review and editing, MNA, MAH and SMG; visualization, MDH; supervision, SHB, MNA and MAH; project administration, AFMSU; funding acquisition, SHB. All authors have read and agreed to the published version of the manuscript.

Funding

This work was supported in part by the IITP grant funded by the Korea Government (MSIT) (Artificial Intelligence Convergence Innovation Human Resources Development, Kyung Hee University) under Grant RS-2022-00155911; in part by the IITP grant funded by the Korea Government (MSIT) (Artificial Intelligence Innovation Hub) under Grant 2021-0-02068; in part by the Korea Research Institute for Defense Technology (KRIT) grant funded by the Korea Government [Defense Acquisition Program Administration (DAPA)] under Grant V220023; and in part by the Research Cell, Jashore University of Science and Technology, Jashore-7408, under Grant no. [22 FoET 02/02].

Conflict of interest

The authors declare no conflict of interest.

References

1. Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998; 86(11), 2278–2324. doi: 10.1109/5. 726791
2. Krizhevsky A, Sutskever I, Hinton G. Imagenet classification with deep convolutional neural networks. In: Neural Information Processing Systems (NeurIPS). 2012, pp. 1097–1105. doi: 10.1145/ 3065386
3. Lu D, Weng Q. A survey of image classification methods and techniques for improving classification performance. International Journal of Remote Sensing. 2007; 28(5): 823-870. doi: 10.1080/01431160600746456
4. Shaoqing R, Kaiming H, Ross G, Jian S. Faster r-CNN: Towards real-time object detection with region proposal networks. In: Neural Information Processing Systems (NeurIPS). 2015.
5. Zou Z, Chen K, Shi Z, et al. Object Detection in 20 Years: A Survey. Proceedings of the IEEE. 2023; 111(3): 257-276. doi: 10.1109/jproc.2023.3238524
6. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015. pp. 3431–3440.
7. Yu C, Wang J, Peng C, et al. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV). 2018. pp. 325–341.
8. Guo Y, Liu Y, Georgiou T, et al. A review of semantic segmentation using deep neural networks. International Journal of Multimedia Information Retrieval. 2018; 7(2): 87-93. doi: 10.1007/s13735-017-0141-z
9. Nitish S, Geoffrey H, Alex K, et al. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research. 2014; 15:1929–1958
10. Tompson J, Goroshin R, Jain A, et al. Efficient object localization using convolutional networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015; pp. 648–656.
11. Devries T, Taylor GW. Improved regularization of convolutional neural networks with cutout. ArXiv abs/1708.04552. 2047.
12. Zhang H, Ciss´e M, Dauphin YN, LopezPaz D. Mixup: Beyond empirical risk minimization. arXiv preprint. 2017.
13. Yun S, Han D, Chun S, et al. Cutmix: Regularization strategy to train strong classifiers with localizable features.

In: International Conference on Computer Vision (ICCV). 2019.

14. Shahab AFM, Mst Sirazam U, Wheemyung M, et al. Saliencymix: A saliency guided data augmentation strategy for better regularization. arXiv preprint arXiv:2006.01791. 2020.
15. Kim JH, Choo W, Song HO. Puzzle Mix: Exploiting Saliency and Local Statistics for Optimal Mixup. 2020.
16. Muhammad A, Zhou F, Xie C, et al. Mixacm: Mixup-based robustness transfer via distillation of activated channel maps. Advances in Neural Information Processing Systems. 2021; 34: 4555–4569.
17. Qin J, Fang J, Zhang Q, et al. Resizemix: Mixing data with preserved object information and true labels. arXiv preprint arXiv:2012.11101. 2020.
18. Wang D, Zhang Y, Zhang K, et al. Focalmix: Semi-supervised learning for 3d medical image detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020. pp. 3951–3960.
19. Qiao P, Li H, Song G, et al. Semi-Supervised CT Lesion Segmentation Using Uncertainty-Based Data Pairing and SwapMix. IEEE Transactions on Medical Imaging. 2023; 42(5): 1546-1562. doi: 10.1109/tmi.2022.3232572
20. Shrivastava A, Gupta A, Girshick R. Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016. pp. 761–769.
21. Tang W, Huang S, Zhang X, et al. Multiple instance learning framework with masked hard instance mining for whole slide image classification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023. pp. 4078–4087.
22. Wang Y, Peng T, Duan J, et al. Pathological Image Classification Based on Hard Example Guided CNN. IEEE Access. 2020; 8: 114249-114258. doi: 10.1109/access.2020.3003070
23. Wu T, Ding X, Zhang H, Gao J, et al. Discrimloss: a universal loss for hard samples and incorrect samples discrimination. IEEE Transactions on Multimedia. 2023.
24. Yang C, Hou B, Chanussot J, et al. N-Cluster Loss and Hard Sample Generative Deep Metric Learning for PolSAR Image Classification. IEEE Transactions on Geoscience and Remote Sensing. 2022; 60: 1-16. doi: 10.1109/tgrs.2021.3099840
25. Zhu C, Chen W, Peng T, et al. Hard Sample Aware Noise Robust Learning for Histopathology Image Classification. IEEE Transactions on Medical Imaging. 2022; 41(4): 881-894. doi: 10.1109/tmi.2021.3125459
26. Cubuk ED, Zoph B, Man´e D, et al. Autoaugment: Learning augmentation strategies from data. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019; pp. 113– 123. doi: 10.1109/CVPR. 2019.00020
27. Lim S, Kim I, Kim T, et al. Fast autoaugment. Advances in Neural Information Processing Systems. 2019; 32.
28. Liu Z, Li S, Wu D, et al. Automix: Unveiling the power of mixup for stronger classifiers. In: European Conference on Computer Vision. 2022. pp. 441–458.
29. Dabouei A, Soleymani S, Taherkhani F, et al. Supermix: Supervising the mixing data augmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021. pp. 13794–13803.
30. Hu C, Zhou R. Synthetic voice spoofing detection based on online hard example mining. arXiv preprint arXiv:2209.11585. 2022.
31. Feng ZH, Kittler J, Wu XJ. Mining Hard Augmented Samples for Robust Facial Landmark Localization with CNNs. IEEE Signal Processing Letters. 2019; 26(3): 450-454. doi: 10.1109/lsp.2019.2895291
32. Wang Y, Lu H, Qin X, et al. Residual Gabor convolutional network and FV-Mix exponential level data augmentation strategy for finger vein recognition. Expert Systems with Applications. 2023; 223: 119874. doi: 10.1016/j.eswa.2023.119874
33. Wang M, Zhu Y, Li G, et al. Image anomaly detection with semanticenhanced augmentation and distributional kernel. In: 2022 IEEE 24th Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys). 2022. pp. 163–170.
34. Krogh A. Hertz J. A simple weight decay can improve generalization. Advances in Neural Information Processing Systems. 1991; 4.
35. Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. International Conference on Machine Learning. 2015. pp. 448–456.
36. Huang G, Sun Y, Liu Z, et al. Deep networks with stochastic depth. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam; October 11–14, 2016. The Netherlands. pp. 646–661.
37. Yamada Y, Iwamura M, Akiba T, et al. Shakedrop Regularization for Deep Residual Learning. IEEE Access. 2019; 7: 186126-186136. doi: 10.1109/access.2019.2960566
38. Hu J, Shen L, Sun G. Squeeze-andexcitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018. pp. 7132–7141.
39. Hu J, Shen L, Albanie S, et al. Gather-excite: Exploiting feature context in convolutional neural networks. Advances in Neural Information Processing Systems. 2018. 31.
40. Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016. pp. 2818–2826.
41. Katharopoulos A, Fleuret F. Not all samples are created equal: Deep learning with importance sampling. In: International Conference on Machine Learning. 2018. pp. 2525–2534.
42. Chang HS, Learned-Miller E, McCallum A. Active bias: Training more accurate neural networks by emphasizing

high variance samples. Advances in Neural Information Processing Systems. 2017. 30.

43. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. pp. 770–778. doi: 10.1109/CVPR. 2016.90
44. Zagoruyko S, Komodakis N. Wide residual networks. Procedings of the British Machine Vision Conference. 2016.
45. Krizhevsky A. Learning multiple layers of features from tiny images. University of Toronto. 2012.
46. Olga R, Jia D, Hao S, et al. Imagenet large scale visual recognition challenge. International Journal of Computer Vision. 2015; 115(3): 211–252.
47. Imambi S, Prakash KB, Kanagachidambaresan G. Pytorch. Programming with TensorFlow: Solution for Edge Computing Applications. 2021; 87–104.