Random Erasing Data Augmentation

Zhun Zhong,[1,2] **Liang Zheng,**[3] **Guoliang Kang,**[4] **Shaozi Li,**[1*] **Yi Yang**[2]
[1]Department of Artificial Intelligence, Xiamen University
[2]ReLER, University of Technology Sydney
[3]Research School of Computer Science, Australian National University
[4]School of Computer Science, Carnegie Mellon University

Abstract

In this paper, we introduce Random Erasing, a new data augmentation method for training the convolutional neural network (CNN). In training, Random Erasing randomly selects a rectangle region in an image and erases its pixels with random values. In this process, training images with various levels of occlusion are generated, which reduces the risk of over-fitting and makes the model robust to occlusion. Random Erasing is parameter learning free, easy to implement, and can be integrated with most of the CNN-based recognition models. Albeit simple, Random Erasing is complementary to commonly used data augmentation techniques such as random cropping and flipping, and yields consistent improvement over strong baselines in image classification, object detection and person re-identification. Code is available at: https://github.com/zhunzhong07/Random-Erasing.

1 Introduction

The ability to generalize is a research focus for the convolutional neural network (CNN). When a model is excessively complex, such as having too many parameters compared to the number of training samples, over-fitting might happen and weaken its generalization ability. A learned model may describe random error or noise instead of the underlying data distribution (Zhang et al. 2017). In bad cases, the CNN model may exhibit good performance on the training data, but fail drastically when predicting new data. To improve the *generalization ability* of CNNs, many data augmentation and regularization approaches have been proposed, such as random cropping (Krizhevsky, Sutskever, and Hinton 2012), flipping (Simonyan and Zisserman 2015), dropout (Srivastava et al. 2014), and batch normalization (Ioffe and Szegedy 2015).

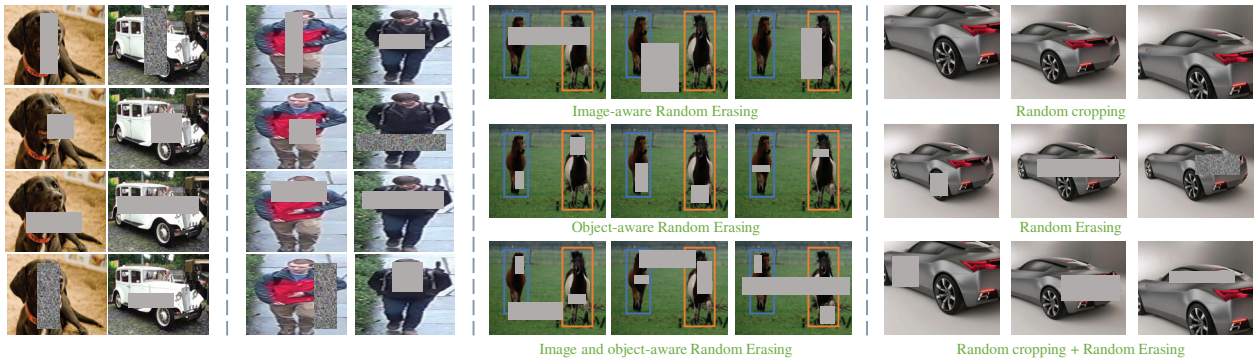
Occlusion is a critical influencing factor on the *generalization ability* of CNNs. It is desirable that invariance to various levels of occlusion is achieved. When some parts of an object are occluded, a strong classification model should be able to recognize its category from the overall object structure. However, the collected training samples usually exhibit limited variance in occlusion. In an extreme case when all the training objects are clearly visible, *i.e.,* no occlusion happens, the learned CNN will probably work well on testing images without occlusion, but, due to the limited generalization ability of the CNN model, may fail to recognize objects which are partially occluded. While we can manually add occluded natural images to the training set, it is costly and the levels of occlusion might be limited.

To address the occlusion problem and improve the generalization ability of CNNs, this paper introduces a new data augmentation approach, Random Erasing. It can be easily implemented in most existing CNN models. In the training phase, an image within a mini-batch randomly undergoes either of the two operations: 1) kept unchanged; 2) we randomly choose a rectangle region of an arbitrary size, and assign the pixels within the selected region with random values (or the ImageNet (Deng et al. 2009) mean pixel value). During Operation 2), an image is partially occluded in a random position with a random-sized mask. In this manner, augmented images with various occlusion levels can be generated. Examples of Random Erasing are shown in Fig. 1.

Two commonly used data augmentation approaches, *i.e.,* random flipping and random cropping, also work on the image level and are closely related to Random Erasing. Both techniques have demonstrated the ability to improve the image recognition accuracy. In comparison with Random Erasing, random flipping does not incur information loss during augmentation. Different from random cropping, in Random Erasing, 1) only part of the object is occluded and the overall object structure is preserved, 2) pixels of the erased region are re-assigned with random values, which can be viewed as adding block noise to the image.

Working primarily on the fully connected (FC) layer, Dropout (Srivastava et al. 2014) is also related to our method. It prevents over-fitting by discarding (both hidden and visible) units of the CNN with a probability p. Random Erasing is somewhat similar to performing Dropout on the image level. The difference is that in Random Erasing, 1) we operate on a continuous rectangular region, 2) no pixels (units) are discarded, and 3) we focus on making the model more robust to noise and occlusion. The recent A-Fast-RCNN (Wang, Shrivastava, and Gupta 2017) pro-

*Corresponding author (szlig@xmu.edu.cn).



(a) Image classification (b) Person re-ID (c) Object detection (d) Different augmentation methods

Figure 1: Examples of Random Erasing in image classification (a), person re-identification (re-ID) (b), object detection (c) and comparing with different augmentation methods (d). In CNN training, we randomly choose a rectangle region in the image and erase its pixels with random values or the ImageNet mean pixel value. Images with various levels of occlusion are thus generated.

poses an occlusion invariant object detector by training an adversarial network that generates examples with occlusion. Comparison with A-Fast-RCNN, Random Erasing does not require any parameter learning, can be easily applied to other CNN-based recognition tasks and still yields competitive accuracy with A-Fast-RCNN in object detection.

To summarize, Random Erasing has the following advantages:

- A lightweight method that does not require any extra parameter learning or memory consumption. It can be integrated with various CNN models without changing the learning strategy.
- A complementary method to existing data augmentation and regularization approaches. When combined, Random Erasing further improves the recognition performance.
- Consistently improving the performance of recent state-of-the-art deep models on image classification, object detection, and person re-identification.
- Improving the robustness of CNNs to partially occluded samples. When we randomly adding occlusion to the CIFAR-10 testing dataset, Random Erasing significantly outperforms the baseline model.

2 Related Work

Overfitting is a long-standing problem for the convolutional neural network (CNN). In general, methods of reducing the risk of overfitting can be divided into two categories: regularization and data augmentation.

Regularization. Regularization is a key component in preventing over-fitting in the training of CNN models. Various regularization methods have been proposed (Krizhevsky, Sutskever, and Hinton 2012; Wan et al. 2013; Ba and Frey 2013; Zeiler and Fergus 2013; Xie et al. 2016; Kang et al. 2017). Dropout (Krizhevsky, Sutskever, and Hinton 2012) randomly discards (setting to zero) the output of each hidden neuron with a probability during the training and only considers the contribution of the remaining weights in forward pass and back-propagation. Latter, Wan *et al.* (Wan et al. 2013) propose a generalization of dropout approach, DropConect, which instead randomly selects weights to zero during training. In addition, Adaptive dropout (Ba and Frey 2013) is proposed where the dropout probability for each hidden neuron is estimated through a binary belief network. Stochastic Pooling (Zeiler and Fergus 2013) randomly selects activation from a multinomial distribution during training, which is parameter free and can be applied with other regularization techniques. Recently, a regularization method named "DisturbLabel" (Xie et al. 2016) is introduced by adding noise at the loss layer. DisturbLabel randomly changes the labels of small part of samples to incorrect values during each training iteration. PatchShuffle (Kang et al. 2017) randomly shuffles the pixels within each local patch while maintaining nearly the same global structures with the original ones, it yields rich local variations for training of CNN.

Data augmentation. Data augmentation is an explicit form of regularization that is also widely used in the training of deep CNN (Krizhevsky, Sutskever, and Hinton 2012; Simonyan and Zisserman 2015; He et al. 2016a). It aims at artificially enlarging the training dataset from existing data using various translations, such as, translation, rotation, flipping, cropping, adding noises, *etc*. The two most popular and effective data augmentation methods in training of deep CNN are random flipping and random cropping. Random flipping randomly flips the input image horizontally, while random cropping extracts random sub-patch from the input image. As an analogous choice, Random Erasing may discard some parts of the object. For random cropping, it may crop off the corners of the object, while Random Erasing may occlude some parts of the object. Random Erasing maintains the global structure of object. Moreover, it can be viewed as adding noise to the image. The combination of random cropping and Random Erasing can produce more various training data.

Blocking-based approach. Our work is also closely related to blocking-based approaches (Murdock et al. 2016; Fong and Vedaldi 2017; Wei et al. 2017; Wang, Shrivas-

tava, and Gupta 2017; Kumar Singh and Jae Lee 2017). Murdock *et al.* (Murdock et al. 2016) propose "Blockout", a method for simultaneous regularization and model selection via masked weight matrices on CNN layers. The hyper-parameters need to learn during training. In contrast, our approach is applied on image-level and does not need any extra parameter learning. Fong and Vedaldi (Fong and Vedaldi 2017) blur an image to suppress the SoftMax probability of the target class to learn saliency region. In contrast, our approach does not rely on any supervision information. In (Wei et al. 2017), by erasing the most discriminative region, a sequence of models is trained iteratively for weakly supervised semantic segmentation. In comparison, our approach only needs to train a single model once. Recently, Wang *et al.* (Wang, Shrivastava, and Gupta 2017) learn an adversary with Fast-RCNN (Girshick 2015) detection to create hard examples on the fly by blocking some feature maps spatially. Instead of generating occlusion examples in feature space, Random Erasing generates images from the original images with very little computation which is in effect and does not require any extra parameters learning. Singh and Lee (Kumar Singh and Jae Lee 2017) randomly hide patches (with black pixels) in a training image to force the network to seek discriminative parts as many as possible for object localization. Instead, our approach randomly selects a rectangle region in an image and erases its pixels with random values. Comparison with the above mentioned methods, our approach aims to reduce the risk of over-fitting, which is model-agnostic, does not require extra parameter learning and can be easily applied to various vision tasks. Our method and Cutout (DeVries and Taylor 2017) are contemporary works. Different from Cutout, we evaluate our method on more vision tasks. We also investigate the impact of different erasing values and different aspect ratios of the erased region.

3 Datasets

For **image classification**, we evaluate on four image classification datasets, including two well-known datasets, CIFAR-10 and CIFAR-100 (Krizhevsky and Hinton 2009), a new dataset Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017), and a large-scale dataset ImageNet2012 (Deng et al. 2009). **CIFAR-10** and **CIFAR-100** contain 50,000 training and 10,000 testing 32×32 color images drawn from 10 and 100 classes, respectively. **Fashion-MNIST** consists of 60,000 training and 10,000 testing 28x28 gray-scale images. Each image is associated with a label from 10 classes. **ImageNet2012** consists of 1,000 classes, including 1.28 million training images and 50k validation images. For CIFAR-10, CIFAR-100 and Fashion-MNIST, we evaluate top-1 error rates in the format "mean ± std" based on 5 runs. For ImageNet2012, we evaluate the top-1 and top-5 error rates on the validation set.

For **object detection**, we use the **PASCAL VOC 2007** (Everingham et al. 2010) dataset which contains 9,963 images of 24,640 annotated objects in training/validation and testing sets. We use the "trainval" set for training and "test" set for testing. We evaluate the performance using mean average precision (mAP).

Algorithm 1: Random Erasing Procedure

Input : Input image I; Image size W and H; Area of image S; Erasing probability p; Erasing area ratio range s_l and s_h; Erasing aspect ratio range r_1 and r_2.
Output: Erased image I^*.
Initialization: $p_1 \leftarrow$ Rand (0, 1).

```
if p_1 ≥ p then
    I* ← I;
    return I*.
else
    while True do
        S_e ← Rand (s_l, s_h)×S;
        r_e ← Rand (r_1, r_2);
        H_e ← sqrt(S_e × r_e),  W_e ← sqrt(S_e / r_e);
        x_e ← Rand (0, W),  y_e ← Rand (0, H);
        if x_e + W_e ≤ W and y_e + H_e ≤ H then
            I_e ← (x_e, y_e, x_e + W_e, y_e + H_e);
            I(I_e) ← Rand (0, 255);
            I* ← I;
            return I*.
        end
    end
end
```

For **person re-identification (re-ID)**, the **Market-1501** dataset (Zheng et al. 2015) contains 12,936 images with 751 identities for training, 19,732 images with 750 identities and 3,368 query images for testing. **DukeMTMC-reID** (Zheng, Zheng, and Yang 2017; Ristani et al. 2016) includes 16,522 training images of 702 identities, 2,228 query images of the other 702 identities and 17,661 gallery images. For **CUHK03** (Li et al. 2014), we use the **new training/testing protocol** proposed in (Zhong et al. 2017). There are 767 identities in the training set and 700 identities in the testing set. We conduct experiment on both "detected" and "labeled" sets. Rank-1 accuracy and mean average precision (mAP) are evaluated on these three datasets.

4 Our Approach

This section presents the Random Erasing data augmentation method for training the convolutional neural network (CNN). We first describe the detailed procedure of Random Erasing. Next, the implementation of Random Erasing in different tasks is introduced. Finally, we analyze the differences between Random Erasing and random cropping.

4.1 Random Erasing

In training, Random Erasing is conducted with a certain probability. For an image I in a mini-batch, the probability of it undergoing Random Erasing is p, and the probability of it being kept unchanged is $1 - p$. In this process, training images with various levels of occlusion are generated.

Random Erasing randomly selects a rectangle region I_e in an image, and erases its pixels with random values. Assume

that the size of the training image is $W \times H$. The area of the image is $S = W \times H$. We randomly initialize the area of erasing rectangle region to S_e, where $\frac{S_e}{S}$ is in range specified by minimum s_l and maximum s_h. The aspect ratio of erasing rectangle region is randomly initialized between r_1 and r_2, we set it to r_e. The size of I_e is $H_e = \sqrt{S_e \times r_e}$ and $W_e = \sqrt{\frac{S_e}{r_e}}$. Then, we randomly initialize a point $\mathcal{P} = (x_e, y_e)$ in I. If $x_e + W_e \leq W$ and $y_e + H_e \leq H$, we set the region, $I_e = (x_e, y_e, x_e + W_e, y_e + H_e)$, as the selected rectangle region. Otherwise repeat the above process until an appropriate I_e is selected. With the selected erasing region I_e, each pixel in I_e is assigned to a random value in [0, 255], respectively. The procedure of selecting the rectangle area and erasing this area is shown in Alg. 1.

4.2 Random Erasing for Image Classification and Person Re-identification

In image classification, an image is classified according to its visual content. In general, training data does not provide the location of the object, so we could not know where the object is. In this case, we perform Random Erasing on the whole image according to Alg. 1.

Recently, the person re-ID model is usually trained in a classification network for embedding learning (Zheng, Yang, and Hauptmann 2016). In this task, since pedestrians are confined with detected bounding boxes, persons are roughly in the same position and take up the most area of the image. In this scenario, we adopt the same strategy as image classification, as in practice, the pedestrian can be occluded in any position. We randomly select rectangle regions on the whole pedestrian image and erase it. Examples of Random Erasing for image classification and person re-ID are shown in Fig. 1(a, b).

4.3 Random Erasing for Object Detection

Object detection aims at detecting instances of semantic objects of a certain class in images. Since the location of each object in the training image is known, we implement Random Erasing with three schemes: 1) Image-aware Random Erasing (**IRE**): selecting erasing region on the whole image, the same as image classification and person re-identification; 2) Object-aware Random Erasing (**ORE**): selecting erasing regions in the bounding box of each object. In the latter, if there are multiple objects in the image, Random Erasing is applied on each object separately. 3) Image and object-aware Random Erasing (**I+ORE**): selecting erasing regions in both the whole image and each object bounding box. Examples of Random Erasing for object detection with the three schemes are shown in Fig. 1(c).

4.4 Comparison with Random Cropping

Random cropping is an effective data augmentation approach, it reduces the contribution of the background in the CNN decision, and can base learning models on the presence of parts of the object instead of focusing on the whole object. In comparison to random cropping, Random Erasing retains the overall structure of the object, only occluding some parts of object. In addition, the pixels of erased region are re-assigned with random values, which can be viewed as adding noise to the image. When jointly employing random cropping and Random Erasing during training, more various images can be generated for data augmentation. In our experiment (Section 5.2), we show that these two methods are complementary to each other for improving the discriminative ability of CNN. The examples of Random Erasing, random cropping, and the combination of them are shown in Fig. 1(d).

5 Image Classification

5.1 Experiment Settings

In all of our experiment, we compare the CNN models trained with or without Random Erasing. For the same deep architecture, all the models are trained from the same weight initialization. Note that some popular regularization techniques (*e.g.*, weight decay, batch normalization and dropout) and various data augmentations (*e.g.*, flipping, padding and cropping) are employed. The compared CNN architectures are summarized as below.

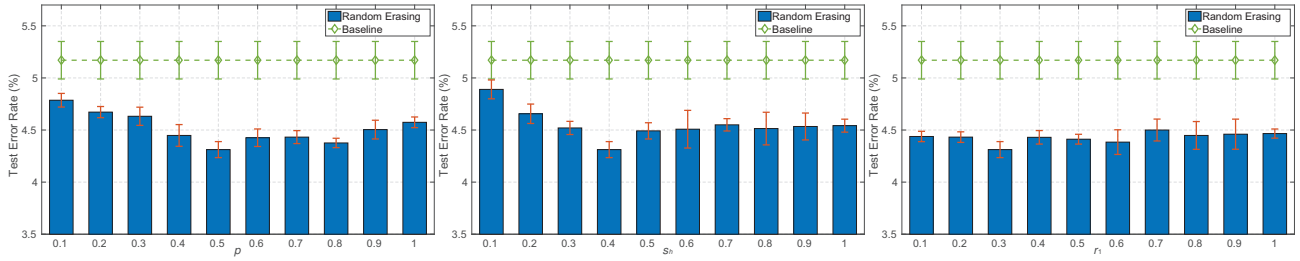
Architectures and Settings. Four architectures are adopted on CIFAR-10, CIFAR-100 and Fashion-MNIST: ResNet (He et al. 2016a), pre-activation ResNet (He et al. 2016b), ResNeXt (Xie et al. 2017), and Wide Residual Networks (Zagoruyko and Komodakis 2016). We use the 20, 32, 44, 56, 110-layer network for ResNet. The 18-layer network is also adopted for pre-activation ResNet. We use ResNeXt-29-8×64 and WRN-28-10 in the same way as (Xie et al. 2017) and (Zagoruyko and Komodakis 2016), respectively. The training procedure follows (He et al. 2016a). Specially, the learning rate starts from 0.1 and is divided by 10 after the 150th and 225th epoch. We stop training by the 300th epoch. If not specified, all models are trained with data augmentation: randomly performs horizontal flipping, and takes a random cropping with 32×32 for CIFAR-10 and CIFAR-100 (28×28 for Fashion-MNIST) from images padded by 4 pixels on each side. For Imagenet-2012 (Deng et al. 2009), we follow the training strategy of ResNet and conduct experiment on ResNet-34, ResNet-50 and ResNet-101. Random cropping, random flipping and label smoothing regularization (Szegedy et al. 2016) are used during model training.

5.2 Classification Evaluation

Classification accuracy on different datasets. We first evaluate Random Erasing on medium-scale datasets. The results on CIFAR-10 ,CIFAR-100 and Fashion-MNIST with different architectures are shown in Table 1. We set $p = 0.5$, $s_l = 0.02$, $s_h = 0.4$, and $r_1 = \frac{1}{r_2} = 0.3$. Results indicate that models trained with Random Erasing have significant improvement, demonstrating that our method is applicable to various CNN architectures. For CIFAR-10, our method improves the accuracy by 0.49% using ResNet-110. In particular, our approach obtains 3.08% error rate using WRN-28-10, which improves the accuracy by 0.72% and achieves new state of the art. For CIFAR-100, our method obtains 17.73% error rate which gains 0.76% than the WRN-28-10 baseline. Our method also works well for gray-scale images: Random erasing improves WRN-28-10 from 4.01% to

Table 1: Test errors (%) with different architectures on CIFAR-10, CIFAR-100 and Fashion-MNIST. **RE:** Random Erasing.

Model	CIFAR-10		CIFAR-100		Fashion-MNIST	
	Baseline	RE	Baseline	RE	Baseline	RE
ResNet-20	7.21 ± 0.17	6.73 ± 0.09	30.84 ± 0.19	29.97 ± 0.11	4.39 ± 0.08	4.02 ± 0.07
ResNet-32	6.41 ± 0.06	5.66 ± 0.10	28.50 ± 0.37	27.18 ± 0.32	4.16 ± 0.13	3.80 ± 0.05
ResNet-44	5.53 ± 0.08	5.13 ± 0.09	25.27 ± 0.21	24.29 ± 0.16	4.41 ± 0.09	4.01 ± 0.14
ResNet-56	5.31 ± 0.07	4.89 ± 0.07	24.82 ± 0.27	23.69 ± 0.33	4.39 ± 0.10	4.13 ± 0.42
ResNet-110	5.10 ± 0.07	4.61 ± 0.06	23.73 ± 0.37	22.10 ± 0.41	4.40 ± 0.10	4.01 ± 0.13
ResNet-18-PreAct	5.17 ± 0.18	4.31 ± 0.07	24.50 ± 0.29	24.03 ± 0.19	4.31 ± 0.06	3.90 ± 0.06
WRN-28-10	3.80 ± 0.07	**3.08 ± 0.05**	**18.49 ± 0.11**	**17.73 ± 0.15**	**4.01 ± 0.10**	**3.65 ± 0.03**
ResNeXt-8-64	**3.54 ± 0.04**	3.24 ± 0.03	19.27 ± 0.30	18.84 ± 0.18	4.02 ± 0.05	3.79 ± 0.06



(a) probability p (b) area ratio s_h (c) aspect ratio r_1

Figure 2: Test errors (%) under different hyper-parameters on CIFAR-10 with using ResNet18 (pre-act).

Table 2: Test errors (%) on ImageNet-2012 validation set.

Model	Baseline		Random Erasing	
	Top-1	Top-5	Top-1	Top-5
ResNet-34	25.22	8.01	**24.89**	**7.71**
ResNet-50	23.39	6.89	**22.75**	**6.69**
ResNet-101	20.98	5.73	**20.43**	**5.30**

3.65% in top-1 error on Fashion-MNIST.

We then evaluate our approach on large-scale dataset. Results on ImageNet-2012 with different architectures are reported in Table 2. Our method consistently improves the results on all three architectures, demonstrating the effectiveness of our method on large-scale dataset.

The impact of hyper-parameters. When implementing Random Erasing on CNN training, we have three hyper-parameters to evaluate, *i.e.*, the erasing probability p, the area ratio range of erasing region s_l and s_h, and the aspect ratio range of erasing region r_1 and r_2. To demonstrate the impact of these hyper-parameters on the model performance, we conduct experiment on CIFAR-10 based on ResNet18 (pre-act) under varying hyper-parameter settings. To simplify experiment, we fix s_l to 0.02, $r_1 = \frac{1}{r_2}$ and evaluate p, s_h, and r_1. We set $p = 0.5$, $s_h = 0.4$ and $r_1 = 0.3$ as the base setting. When evaluating one of the parameters, we fixed the other two parameters. Results are shown in Fig. 2.

Notably, Random Erasing consistently outperforms the

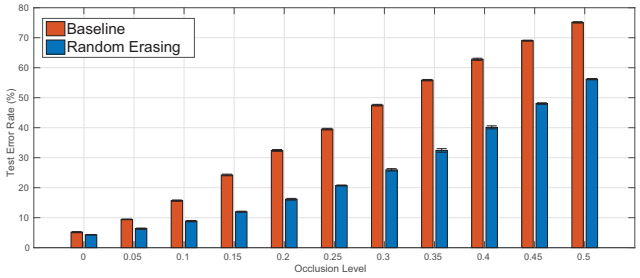


Figure 3: Test errors (%) under different levels of occlusion on CIFAR-10 based on ResNet18 (pre-act). The model trained with Random Erasing is more robust to occlusion.

ResNet18 (pre-act) baseline under all parameter settings. For example, when $p \in [0.2, 0.8]$ and $s_h \in [0.2, 0.8]$, the average classification error rate is 4.48%, outperforming the baseline method (5.17%) by a large margin. Random Erasing is also robust to the aspect ratios of the erasing region. Specifically, our best result (when $r_1 = 0.3$, error rate = 4.31%) reduces the classification error rate by 0.86% compared with the baseline. In the following experiment for image classification, we set $p = 0.5$, $s_l = 0.02$, $s_h = 0.4$, and $r_1 = \frac{1}{r_2} = 0.3$, if not specified.

Four types of random values for erasing. We evaluate Random Erasing when pixels in the selected region are erased in four ways: 1) each pixel is assigned with a random

Table 3: Test errors (%) on CIFAR-10 based on ResNet18 (pre-act) with four types of erasing value. **Baseline:** Baseline model, **RE-R:** Random Erasing model with random value, **RE-M:** Random Erasing model with mean value of ImageNet 2012, **RE-0:** Random Erasing model with 0, **RE-255:** Random Erasing model with 255.

Erasing Value	Baseline	RE-R	RE-M	RE-0	RE-255
Test Errors(%)	5.17 ± 0.18	$\mathbf{4.31 \pm 0.07}$	4.35 ± 0.12	4.62 ± 0.09	4.85 ± 0.13

Table 4: Comparing Random Erasing with dropout and random noise on CIFAR-10 with using ResNet18 (pre-act).

Method	Test error (%)	Method	Test error (%)
Baseline	5.17 ± 0.18	Baseline	5.17 ± 0.18
Ours	$\mathbf{4.31 \pm 0.07}$	Ours	$\mathbf{4.31 \pm 0.07}$
Dropout	Test error (%)	Noise	Test error (%)
$\lambda_1 = 0.001$	5.37 ± 0.12	$\lambda_2 = 0.01$	5.38 ± 0.07
$\lambda_1 = 0.005$	5.48 ± 0.15	$\lambda_2 = 0.05$	5.79 ± 0.14
$\lambda_1 = 0.01$	5.89 ± 0.14	$\lambda_2 = 0.1$	6.13 ± 0.12
$\lambda_1 = 0.05$	6.23 ± 0.11	$\lambda_2 = 0.2$	6.25 ± 0.09
$\lambda_1 = 0.1$	6.38 ± 0.18	$\lambda_2 = 0.4$	6.52 ± 0.12

Table 5: Test errors (%) with different data augmentation methods on CIFAR-10 based on ResNet18 (pre-act). **RF**: Random flipping, **RC**: Random cropping, **RE**: Random Erasing.

Method	RF	RC	RE	Test errors (%)
Baseline				11.31 ± 0.18
	✓			8.30 ± 0.17
		✓		6.33 ± 0.15
			✓	10.13 ± 0.14
	✓	✓		5.17 ± 0.18
	✓		✓	7.19 ± 0.10
		✓	✓	5.21 ± 0.14
	✓	✓	✓	$\mathbf{4.31 \pm 0.07}$

value ranging in [0, 255], denoted as RE-R; 2) all pixels are assign with the mean ImageNet pixel value *i.e.,* [125, 122, 114], denoted as RE-M; 3) all pixels are assigned with 0, denoted as RE-0; 4) all pixels are assigned with 255, denoted as RE-255. Table 3 presents the result with different erasing values on CIFAR10 using ResNet18 (pre-act). We observe that, 1) all erasing schemes outperform the baseline, 2) RE-R achieves approximately equal performance to RE-M, and 3) both RE-R and RE-M are superior to RE-0 and RE-255. If not specified, we use RE-R in the following experiment.

Comparison with Dropout and random noise. We compare Random Erasing with two variant methods applied on image layer. 1) Dropout: we apply dropout on image layer with probability λ_1. 2) Random noise: we add different levels of noise on the input image by changing the pixel to a random value in [0, 255] with probability λ_2. The probability of whether an image undergoes dropout or random noise is set to 0.5 as Random Erasing. Results are presented in Table 4. It is clear that applying dropout or adding random noise at the image layer fails to improve the accuracy. As the probability λ_1 and λ_2 increase, performance drops quickly. When $\lambda_2 = 0.4$, the number of noise pixels for random noise is approximately equal to the number of erasing pixels for Random Erasing, the error rate of random noise increases from 5.17% to 6.52%, while Random Erasing reduces the error rate to 4.31%.

Comparing with data augmentation methods. We compare our method with random flipping and random cropping in Table 5. When applied alone, random cropping (6.33%) outperforms the other two methods. Importantly, **Random Erasing and the two competing techniques are complementary**. Particularly, combining these three methods achieves 4.31% error rate, a 7% improvement over the baseline without any augmentation.

Robustness to occlusion. Last, we show the robustness of Random Erasing against occlusion. In this experiment, we add different levels of occlusion to the CIFAR-10 dataset in testing. We randomly select a region of area and fill it with random values. The aspect ratio of the region is randomly chosen from the range of [0.3, 3.33]. Results as shown in Fig. 3. Obviously, the baseline performance drops quickly when increasing the occlusion level l. In comparison, the performance of the model training with Random Erasing decreases slowly. Our approach achieves 56.36% error rate when the occluded area is half of the image ($l = 0.5$), while the baseline rapidly drops to 75.04%. It demonstrates that Random Erasing improves the robustness of CNNs against occlusion.

6 Object Detection

6.1 Experiment Settings

Experiment is conducted based on the Fast-RCNN (Girshick 2015) detector. The model is initialized by the ImageNet classification models, and then fine-tuned on the object detection data. We experiment with VGG16 (Simonyan and Zisserman 2015) architecture. We follow A-Fast-RCNN (Wang, Shrivastava, and Gupta 2017) for training. We apply SGD for 80K to train all models. The training rate starts with 0.001 and decreases to 0.0001 after 60K iterations. With this training procedure, the baseline mAP is slightly better than the report mAP in (Girshick 2015). We use the selective search proposals during training. For Random Erasing, we set $p = 0.5$, $s_l = 0.02$, $s_h = 0.2$, and $r_1 = \frac{1}{r_2} = 0.3$.

6.2 Detection Evaluation

We report results with using IRE, ORE and I+ORE during training Fast-RCNN in Table 6. The detector is trained with VOC07 trainval and the union of VOC07 and VOC12 trainval. When training with VOC07 trainval, the baseline is 69.1% mAP. The detector learned with IRE scheme achieves

Table 6: **VOC 2007 test** detection average precision (%). ⋆ refers to training schedule in (Wang, Shrivastava, and Gupta 2017).

Method	train set	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv
FRCN	07	66.9	74.5	78.3	69.2	53.2	36.6	77.3	78.2	82.0	40.7	72.7	67.9	79.6	79.2	73.0	69.0	30.1	65.4	70.2	75.8	65.8
FRCN⋆	07	69.1	75.4	80.8	67.3	59.9	37.6	81.9	80.0	84.5	50.0	77.1	68.2	81.0	82.5	74.3	69.9	28.4	71.1	70.2	75.8	66.6
A-Fast-RCNN	07	71.0	74.4	81.3	67.6	57.0	46.6	81.0	79.3	86.0	52.9	75.9	73.7	82.6	83.2	77.7	72.7	37.4	66.3	71.2	78.2	74.3
Ours (IRE)	07	70.5	75.9	78.9	69.0	57.7	46.4	81.7	79.5	82.9	49.3	76.9	67.9	81.5	83.3	76.7	73.2	40.7	72.8	66.9	75.4	74.2
Ours (ORE)	07	71.0	75.1	79.8	69.7	60.8	46.0	80.4	79.0	83.8	51.6	76.2	67.8	81.2	83.7	76.8	73.8	43.1	70.8	67.4	78.3	75.6
Ours (I+ORE)	07	**71.5**	76.1	81.6	69.5	60.1	45.6	82.2	79.2	84.5	52.5	78.7	71.6	80.4	83.3	76.7	73.9	39.4	68.9	69.8	79.2	77.4
FRCN	07+12	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4
FRCN⋆	07+12	74.8	78.5	81.0	74.7	67.9	53.4	85.6	84.4	86.2	57.4	80.1	72.2	85.2	84.2	77.6	76.1	45.3	75.7	72.3	81.8	77.3
Ours (IRE)	07+12	75.6	79.0	84.1	76.3	66.9	52.7	84.5	84.4	88.7	58.0	82.9	71.1	84.8	84.4	78.6	76.7	45.5	77.1	76.3	82.5	76.8
Ours (ORE)	07+12	75.8	79.4	81.6	75.6	66.5	52.7	85.5	84.7	88.3	58.7	82.9	72.8	85.0	84.3	79.3	76.3	46.3	76.3	74.9	86.0	78.2
Ours (I+ORE)	07+12	**76.2**	79.6	82.5	75.7	70.5	55.1	85.2	84.4	88.4	58.6	82.6	73.9	84.2	84.7	78.8	76.3	46.7	77.9	75.9	83.3	79.3

Table 7: Person re-identification performance with Random Erasing (RE) on Market-1501, DukeMTMC-reID, and CUHK03 based on different models. We evaluate CUHK03 under the new evaluation protocol in (Zhong et al. 2017).

Method	Model	RE	Market		Duke		CUHK03 (L)		CUHK03 (D)	
			Rank-1	mAP	Rank-1	mAP	Rank-1	mAP	Rank-1	mAP
IDE	ResNet-18	No	79.87	57.37	67.73	46.87	28.36	25.65	26.86	25.04
		Yes	82.36	62.06	70.60	51.41	36.07	32.58	34.21	31.20
	ResNet-34	No	82.93	62.34	71.63	49.71	31.57	28.66	30.14	27.55
		Yes	84.80	65.68	73.56	54.46	40.29	35.50	36.36	33.46
	ResNet-50	No	83.14	63.56	71.99	51.29	30.29	27.37	28.36	26.74
		Yes	85.24	68.28	74.24	56.17	41.46	36.77	38.50	34.75
TriNet	ResNet-18	No	77.32	58.43	67.50	46.27	43.00	39.16	40.50	37.36
		Yes	79.84	61.68	71.81	51.84	48.29	43.80	46.57	43.20
	ResNet-34	No	80.73	62.65	72.04	51.56	46.00	43.79	45.07	42.58
		Yes	83.11	65.98	72.89	55.38	53.07	48.80	53.21	48.03
	ResNet-50	No	82.60	65.79	72.44	53.50	49.86	46.74	50.50	46.47
		Yes	83.94	68.67	72.98	56.60	**58.14**	**53.83**	**55.50**	**50.74**
SVDNet	ResNet-50	No	84.41	65.60	76.82	57.70	42.21	38.73	41.85	38.24
		Yes	**87.08**	**71.31**	**79.31**	**62.44**	49.43	45.07	48.71	43.50

an improvement to 70.5% mAP and the ORE scheme obtains 71.0% mAP. The ORE performs slightly better than IRE. When implementing Random Erasing on overall image and objects, the detector training with I+ORE obtains further improved in performance with 71.5% mAP. Our approach (I+ORE) outperforms A-Fast-RCNN (Wang, Shrivastava, and Gupta 2017) by 0.5% in mAP. Moreover, our method does not require any parameter learning and is easy to implement. When using the enlarged 07+12 training set, the baseline is 74.8% which is much better than only using 07 training set. The IRE and ORE schemes give similar results, in which the mAP of IRE is improved by 0.8% and ORE is improved by 1.0%. When applying I+ORE during training, the mAP of Fast-RCNN increases to 76.2%, surpassing the baseline by 1.4%.

7 Person Re-identification

7.1 Experiment Settings

Three baselines are used in person re-ID, *i.e.*, the ID-discriminative Embedding (IDE) (Zheng, Yang, and Hauptmann 2016), TriNet (Hermans, Beyer, and Leibe 2017), and SVDNet (Sun et al. 2017). IDE and SVDNet are trained with the Softmax loss, while TriNet is trained with the triplet loss. The input images are resized to 256×128. We use the ResNet-18, ResNet-34, and ResNet-50 architectures for IDE and TriNet, and ResNet-50 for SVDNet. We fine-tune them on the model pre-trained on ImageNet (Deng et al. 2009). We also perform random cropping and random horizontal flipping during training. For Random Erasing, we set $p = 0.5$, $s_l = 0.02$, $s_h = 0.2$, and $r_1 = \frac{1}{r_2} = 0.3$.

7.2 Person Re-identification Performance

Random Erasing improves different baseline models. As shown in Table 7, when implementing Random Erasing in these baseline models, Random Erasing consistently improves the rank-1 accuracy and mAP. Specifically, for Market-1501, Random Erasing improves the rank-1 by 3.10% and 2.67% for IDE and SVDNet with using ResNet-50. For DukeMTMC-reID, Random Erasing increases the rank-1 accuracy from 71.99% to 74.24% for IDE (ResNet-50) and from 76.82% to 79.31% for SVDNet (ResNet-50). For CUHK03, TriNet gains 8.28% and 5.0% in rank-1 accuracy when applying Random Erasing on the labeled and detected settings with ResNet-50, respectively. We note that, due to lack of adequate training data, over-fitting tend to oc-

cur on CUHK03. For example, a deeper architecture, such as ResNet-50, achieves lower performance than ResNet-34 when using the IDE mode on the detected subset. However, with our method, IDE (ResNet-50) outperforms IDE (ResNet-34). This indicates that our method can reduce the risk of over-fitting and improves the re-ID performance.

8 Conclusion

In this paper, we propose a new data augmentation approach named "Random Erasing" for training the convolutional neural network (CNN). It is easy to implemented: Random Erasing randomly occludes an arbitrary region of the input image during each training iteration. Experiment conducted on CIFAR10, CIFAR100, Fashion-MNIST and ImageNet with various architectures validate the effectiveness of our method. Moreover, we obtain reasonable improvement on object detection and person re-identification, demonstrating that our method has good performance on various recognition tasks. In the future work, we will apply our approach to other CNN recognition tasks, such as, image retrieval, face recognition and fine-grained classification.

Acknowledgment

We wish to thank the anonymous reviewers for their helpful comments. We also would like to thank Ross Wightman for reproducing our method on ImageNet and providing important experimental results for this paper. Zhun Zhong thanks Wenjing Li for encouragement. This work is supported by the National Nature Science Foundation of China (No. 61876159, 61806172, 61572409, U1705286 & 61571188), the National Key Research and Development Program of China (No. 2018YFC0831402).

References

Ba, J., and Frey, B. 2013. Adaptive dropout for training deep neural networks. In *NIPS*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

DeVries, T., and Taylor, G. W. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *IJCV*.

Fong, R., and Vedaldi, A. 2017. Interpretable explanations of black boxes by meaningful perturbation. In *ICCV*.

Girshick, R. 2015. Fast r-cnn. In *ICCV*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016a. Deep residual learning for image recognition. In *CVPR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016b. Identity mappings in deep residual networks. In *ECCV*. Springer.

Hermans, A.; Beyer, L.; and Leibe, B. 2017. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*.

Kang, G.; Dong, X.; Zheng, L.; and Yang, Y. 2017. Patchshuffle regularization. *arXiv preprint arXiv:1707.07103*.

Krizhevsky, A., and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

Kumar Singh, K., and Jae Lee, Y. 2017. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*.

Li, W.; Zhao, R.; Xiao, T.; and Wang, X. 2014. Deepreid: Deep filter pairing neural network for person re-identification. In *CVPR*.

Murdock, C.; Li, Z.; Zhou, H.; and Duerig, T. 2016. Blockout: Dynamic model selection for hierarchical deep networks. In *CVPR*.

Ristani, E.; Solera, F.; Zou, R.; Cucchiara, R.; and Tomasi, C. 2016. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCVW*.

Simonyan, K., and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*.

Sun, Y.; Zheng, L.; Deng, W.; and Wang, S. 2017. SVDNet for pedestrian retrieval. In *ICCV*.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *CVPR*.

Wan, L.; Zeiler, M.; Zhang, S.; Cun, Y. L.; and Fergus, R. 2013. Regularization of neural networks using dropconnect. In *ICML*.

Wang, X.; Shrivastava, A.; and Gupta, A. 2017. A-fast-rcnn: Hard positive generation via adversary for object detection. In *CVPR*.

Wei, Y.; Feng, J.; Liang, X.; Cheng, M.-M.; Zhao, Y.; and Yan, S. 2017. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *CVPR*.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Xie, L.; Wang, J.; Wei, Z.; Wang, M.; and Tian, Q. 2016. Disturblabel: Regularizing cnn on the loss layer. In *CVPR*.

Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated residual transformations for deep neural networks. In *CVPR*.

Zagoruyko, S., and Komodakis, N. 2016. Wide residual networks. In *BMVC*.

Zeiler, M. D., and Fergus, R. 2013. Stochastic pooling for regularization of deep convolutional neural networks. In *ICLR*.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2017. Understanding deep learning requires rethinking generalization. In *ICLR*.

Zheng, L.; Shen, L.; Tian, L.; Wang, S.; Wang, J.; and Tian, Q. 2015. Scalable person re-identification: A benchmark. In *ICCV*.

Zheng, L.; Yang, Y.; and Hauptmann, A. G. 2016. Person re-identification: Past, present and future. *arXiv preprint arXiv:1610.02984*.

Zheng, Z.; Zheng, L.; and Yang, Y. 2017. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *ICCV*.

Zhong, Z.; Zheng, L.; Cao, D.; and Li, S. 2017. Re-ranking person re-identification with k-reciprocal encoding. In *CVPR*.