Defense Against Adversarial Examples
Using Quality Recovery for Image Classification

Motohiro TAKAGI* · Masafumi HAGIWARA*

Adversarial examples can be used to exploit vulnerabilities in neural networks and threaten their sensitive applications. Adversarial attacks are evolving daily, and are rapidly rendering defense methods that assume specific attacks obsolete. This paper proposes a new defense method that does not assume a specific adversarial attack, and shows that it can be used efficiently to protect a network from a variety of adversarial attacks. Adversarial perturbations are small values; consequently, an image quality recovery method is considered to be an effective way to remove adversarial perturbations because such a method often includes a smoothing effect. The proposed method, called the denoising-based perturbation removal network (DPRNet), aims to eliminate perturbations generated by an adversarial attack for image classification tasks. DPRNet is an encoder–decoder network that excludes adversarial images during training and can reconstruct a correct image from an adversarial image. To optimize DPRNet’s parameters for eliminating adversarial perturbations, we also propose a new perturbation removal loss (PRloss) metric, which consists of a reconstructed loss and a Kullback–Leibler divergence loss that expresses the class probability distribution difference between an original image and a reconstructed image. To remove adversarial perturbation, the proposed network is trained using various types of distorted images considering the proposed PRloss metric. Thus, DPRNet eliminates image perturbations, allowing the images to be classified easily. We evaluate the proposed method using the MNIST, CIFAR-10, SVHN, and Caltech 101 datasets and show that the proposed defense method invalidates 99.8%, 95.1%, 98.7%, and 96.0% of the adversarial images that are generated by several adversarial attacks in the MNIST, CIFAR-10, SVHN, and Caltech 101 datasets, respectively.

Keywords: deep neural network, adversarial example, image recognition, image quality recovery

1. Introduction

Machine learning applications that use deep neural networks (DNNs) are known to be vulnerable to adversarial examples [1, 2]. Therefore, addressing adversarial examples is crucial for applications that must ensure security. For example, adversarial attacks are extremely dangerous in the field of autonomous driving, potentially causing networks to misrecognize stop signs [2] or remove pedestrians from recognition targets [3]. In voice recognition, voice agents can be fooled by adversarial examples of voice commands [4, 5]. In addition, adversarial examples can fool multiple models [1, 6], indicating that it is possible to attack a target model without considering its structure and weights. This feature increases the security risk involved in real-world applications, highlighting the urgency of developing defenses against adversarial attacks.

Many methods have been proposed for addressing security risks brought about by adversarial examples. In image recognition, adversarial images are constructed by adding small values to original images. The small values generated by adversarial attacks, called perturbations, can be removed using a smoothing filter such as a median filter because perturbations are small and often scattered [7]. However, smoothing filters cause blurring that significantly reduces classification accuracy when convolutional neural networks (CNNs) are used [8]. In addition, adversarial attacks are constantly evolving, and the defense methods that assume specific attacks quickly become obsolete. Therefore, it is important that a defense method should not have to assume a specific type of adversarial attack.

In this paper, we propose a defense network for image classification tasks, namely the denoising-based perturbation removal network (DPRNet). DPRNet is an encoder–decoder neural network that can eliminate perturbations without assuming a specific type of adversarial attack.

Most quality recovery methods deal with a single distortion or are not optimized for an image classifier [9, 10]. In contrast, our proposed DPRNet reconstructs images that are optimized for the classifier. Moreover, the images are used as input to the classifier. To optimize the parameters of DPRNet for eliminating adversarial perturbations, we propose a new loss metric called the perturbation removal loss (PRloss) metric, which consists of a reconstructed loss and a loss expressing the class probability distribution difference between an original image and a reconstructed image. Using the PRloss metric, the input image is reconstructed to suppress degradation of classification accuracy. DPRNet can generate reconstructed images from which perturbations are removed by using distorted images for training. When training, DPRNet tries to recover original images from distorted images. By training to recover image quality, DPRNet can eliminate the perturbations generated by adversarial attacks.

The remainder of this paper is organized as follows: Section 2 discusses related work. Section 3 presents the proposed DPRNet, which is based on an encoder–decoder network that can eliminate
perturbations generated by several adversarial attacks. Section 4 describes evaluations of the proposed method conducted on four public datasets: MNIST [11], CIFAR-10 [12], SVHN [13], and Caltech 101 [14]. Finally, Section 5 summarizes the paper and discusses the potential for the future development of DPRNet.

2. Related Work

In this section, we first give an overview of typical adversarial attacks and then describe the means used to defend against them.

2.1 Adversarial Attacks

CNNs are robust to random noise; however, they are vulnerable when faced with adversarial perturbation. Researchers have recently proposed adversarial attacks on CNNs. Szegedy et al. devised an adversarial attack that generates adversarial examples with the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm [1], which is the first adversarial attack described against recent CNNs. The L-BFGS attack causes a classifier to misclassify an image by creating an input image that has a perturbation belonging to a specific incorrect target label; however, the L-BFGS attack requires an iteration in which to minimize a perturbation. Goodfellow et al. proposed the fast gradient sign method (FGSM) [15], which can find a perturbation by calculating the model gradients once. However, it is known that a one-step attack can be easily protected against; hence, Kurakin et al. introduced a stronger attack method, the basic iterative method (BIM) [16], which repeatedly calculates the model gradients.

More sophisticated adversarial attacks have been proposed recently. Deepfool [17] generates an adversarial image by considering the discrimination boundary nearest the original image. Deepfool performs attacks more successfully than FGSM on several datasets. The C&W attack (hereinafter C&W) [18] was devised by Carlini et al. C&W can attack a distillation network proposed as a defense model. It is one of the most powerful attack methods reported in recent years. Zeroth-order optimization (ZOO) [19] generates an adversarial image using the estimated gradient. Although the estimation cost is high, ZOO outperforms C&W.

Additional recent attack methods include the universal adversarial perturbation technique, which finds common perturbations that can imitate multiple classes [20], and one-pixel attack [21], which generates an adversarial image by changing only one pixel.

2.2 Defense Methods

When adversarial attacks are proposed, corresponding defense methods are often derived soon after. Several defense methods work by training a model with adversarial images [15, 22, 23]. For example, BANG training [23] increases the gradients of the correct samples in a minibatch when updating the weights. Papernot et al. proposed a defense method to improve generalization performance and prevent adversarial images by distilling the network [24].

Defense methods that prevent attacks by detecting adversarial images in advance have also been proposed [25–28]. To extract features for detecting adversarial images, SafetyNet [25] uses the outputs of rectified linear units (ReLUs) [29] in VGG19 [30] or ResNet [31]. PixelDefend [27] focuses on finding differences in distribution between adversarial and unperturbed images. To detect adversarial images, PixelDefend calculates a \( p \) value for target images and discriminates whether their distribution is similar to that of training images. PixelCNN [32] is used as the generation model for calculating a \( p \) value. Although the cost of calculating \( p \) is high, it has a high detection rate for several adversarial attacks.

Defense methods that filter or reconstruct adversarial images have also been proposed. Xu et al. proposed a feature squeezing method that detects adversarial images by smoothing them and reducing their bit depth [33]. MagNet [34] has been proposed as a method that combines the detection and reconstruction of adversarial images. In its detection stage, MagNet enlists an auto-encoder (AE) trained with the mean square error (MSE) between the original and reconstructed images. If an image reconstructed by the AE has a high Jensen–Shannon divergence, the detector considers the image to be adversarial. In the reconstruction stage, an image that the detector considers to be correct is reconstructed by an AE that is trained to generate a more general image. The reconstructed image is then used as the input to a classifier. However, because MagNet needs the two networks for detecting and removing adversarial perturbations, training costs are high. In addition, reconstructing images using an AE before they are input to a classifier involves smoothing the images, thus risking degradation of classification accuracy. As another reconstruction-based method, Liao et al. proposed a high-level representation-guided denoiser [35] using a reconstruction network similar to U-Net [36]. Their method uses high-level classifier features as a loss to train the reconstruction network. This technique exhibits high defense accuracy but must be retrained for each new adversarial attack and requires adversarial images for training. PuVAE [37], which takes a variational auto-encoder (VAE)-based approach, has also been proposed. However, the VAE-based approach often causes blur and decreased accuracy.

3. Denoising-Based Perturbation Removal Network

In this paper, we show that DPRNet can eliminate perturbations generated by several adversarial attacks by training from distorted images to which noise, such as Gaussian noise and Gaussian blurring, has been added. Compared to most of the conventional methods, DPRNet does not need adversarial examples for training. In addition, training with the PRloss metric, which considers differences in class probability distribution, allows reconstructed images generated by DPRNet to be easily classified.

Figure 1 shows the framework for training DPRNet. DPRNet has an encoder–decoder-type architecture and is trained with the proposed combined loss metric. L1 distance expresses a distance between the original image and the reconstructed image. Reconstructed images are recovered from the original or distorted image. Kullback–Leibler (KL) divergence expresses the difference in the softmax layer outputs of a classifier. Here, we present the details of the PRloss metric and the network structure of DPRNet. Then,
where $x$ less blurring. The reconstruction loss is expressed as follows: the input images are either distorted or original images. The L1 distance is used as the reconstruction loss because it encourages them to the classifier. To reconstruct images from distorted images, we use the L1 distance as the reconstruction loss between the original and reconstructed images and the Kullback–Leibler (KL) divergence loss, which expresses the class probability distribution difference between the original and reconstructed images. The classifier parameters are fixed during training.

### 3.1 Perturbation Removal Loss

A smoothing filter such as a median filter is effective for eliminating adversarial perturbations that are tiny and cannot be recognized by human vision [33]. Hence, it is expected that a reconstruction network trained to recover an image quality from distorted images is also effective for adversarial perturbations. In addition, Song et al. [27] showed that reconstructed images generated by PixelCNN [32] from adversarial images have a different probability compared to non-adversarial images. Adversarial perturbations are designed to cause a specific classifier to have a different class probability distribution compared to that of the original images. Therefore, the difference in the class probability distribution between the original and adversarial images is the key for disabling adversarial images. Based on the above point of interest, we train DPRNet with the combination of a reconstruction loss and a difference of class probability distribution loss. The reconstruction loss is used to eliminate perturbations and the difference of class probability distribution loss seeks to ensure that the class probability distribution of reconstructed images is similar to that of original images. DPRNet can thus reconstruct input images without detecting adversarial images before inputting them to the classifier.

To reconstruct images from distorted images, we use the L1 distance as the reconstruction loss between the original images and the reconstructed images generated from the input images. The input images are either distorted or original images. The L1 distance is used as the reconstruction loss because it encourages less blurring. The reconstruction loss is expressed as follows:

$$L_{rec} = E_{x,x'}[||x - D(x')||_1]$$

$$= E_{x,x'}[||x - \tilde{x}||_1].$$

where $x$ is the original image, $D$ expresses DPRNet, $x'$ is the input image for reconstruction, and $\tilde{x}$ is the reconstruction image generated by DPRNet. The input image for reconstruction corresponds to the original image or to the distorted image.

To incorporate the class probability distribution difference between the original image and the reconstructed image, we consider the softmax layer output of the classifier as the posterior probability distribution. We use the softmax layer output of the classifier from the original image and from the reconstructed image for calculating KL divergence. Using this divergence, we expect the reconstructed image to have the correct class probability distribution, i.e., like that of the original image.

$$L_{KL} = E_{x,\tilde{x}}[D_{KL}(S(x)||S(\tilde{x}))].$$

Finally, we use the combination of the reconstruction loss and the class probability distribution difference loss as the PRloss metric to optimize the parameters of DPRNet:

$$Loss = L_{rec} + \alpha L_{KL},$$

where $\alpha$ is the hyperparameter that adjusts the balance between the reconstruction loss and the class probability distribution difference loss.

### 3.2 Network Structure

The structure of DPRNet is based on an encoder–decoder architecture, similar to U-Net [36], which has a skip connection structure for preserving image information. The skip connection structure reduces blurring in the reconstructed images. Table 1 shows the structure of DPRNet. The network encoder has a repeated structure with two $3 \times 3$ convolutions, batch normalization [38], and a $2 \times 2$ maxpooling operation. After convolution, ReLUs [29] are used for activation. The convolutional filters are initialized using He’s initializer [39]. The network decoder has a repeated structure with a two-dimensional (2D) transposed convolution with stride 2, a concatenation, two $3 \times 3$ convolutions, and batch normalization. After convolution, we also use a ReLU as the activation function. The sigmoid function is used to activate the output of the network. The concatenation concatenates the outputs of the 2D transposed convolution and the encoder convolution, which have the same dimensions. Dropout [40] is used to

---

**Fig. 1** DPRNet’s training framework. DPRNet is optimized with a new PRloss metric consisting of the L1 distance between the original and reconstructed images and the Kullback–Leibler (KL) divergence loss, which expresses the class probability distribution difference between the original and reconstructed images. The classifier parameters are fixed during training.
improve the generalization.

3.3 Training Procedure
3.3.1 Quality Recovery from a Distorted Image

To recover image quality in distorted images, we adopt the training procedure proposed for a training quality recovery network [41]. We now briefly describe this training procedure. To train DPRNet to eliminate noise, noise is added to the original images. When training, we use a batch that contains an equal number of original and distorted images in the form of Gaussian noise and blur. When training, we train DPRNet with the PRloss metric by using the images for pretraining. Second, we train DPRNet with the PRloss metric by using the images for training.

Table 1 Structure of DPRNet

| Layer         | Channels | Filter Size | Stride |
|---------------|----------|-------------|--------|
| Conv.         | 16       | 3 × 3       | -      |
| Conv.         | 16       | 3 × 3       | -      |
| BatchNorm.    | -        | -           | -      |
| Maxpool.      | -        | 2 × 2       | -      |
| Conv.         | 32       | 3 × 3       | -      |
| Conv.         | 32       | 3 × 3       | -      |
| BatchNorm.    | -        | -           | -      |
| Maxpool.      | -        | 2 × 2       | -      |
| Conv.         | 64       | 3 × 3       | -      |
| Conv.         | 64       | 3 × 3       | -      |
| BatchNorm.    | -        | -           | -      |
| Transposed Conv. | 32   | 2 × 2       | 2      |
| Concat.       |          |             |        |
| Conv.         | 32       | 3 × 3       | -      |
| Conv.         | 32       | 3 × 3       | -      |
| BatchNorm.    | -        | -           | -      |
| Transposed Conv. | 16   | 2 × 2       | 2      |
| Concat.       |          |             |        |
| Conv.         | 16       | 3 × 3       | -      |
| Dropout       | -        | -           | -      |
| Conv.         | 16       | 3 × 3       | -      |
| BatchNorm.    | -        | -           | -      |
| Conv.         | 3        | 1 × 1       | -      |
| Sigmoid.      | -        | -           | -      |

is trained robustly for various distortions and is expected to eliminate adversarial perturbations.

3.3.2 Two-Stage Training that Considers Class Probability Distribution Differences

We adopt KL divergence as a loss metric to express the difference in class probability distribution that the original and reconstructed images can have. The class probability distribution of each desirably expresses the distribution when testing. In other words, the original image and the reconstructed image that are used to calculate the KL loss may not be sufficient to express the distribution. If the original and reconstructed images are in the training data, the class probability distribution will differ from that obtained during testing. Hence, we propose a two-stage training procedure to generate a reconstructed image that can have the appropriate class probability distribution during testing. First, we train a classifier by using the images for pretraining. Second, we train DPRNet with the PRloss metric by using the images for training.

The training procedure is as follows:
1. Train the classifier by using the images for pretraining.
2. Train DPRNet with the reconstruction loss (Equation (2)) by using the images for pretraining.
3. Train DPRNet with the PRloss metric (Equation (4)) by using the images for training.

When using the PRloss metric, the classifier parameters are fixed. To adopt the two-stage training procedure, we expect to obtain class probability distributions from the reconstructed images that are similar to those of the original images when testing. If the amount of data is small, the images for pretraining and training may be used together when training DPRNet with the PRloss metric. Moreover, we do not use any labels in this procedure to train DPRNet; we only use the labels to train the classifier. DPRNet therefore has a reasonable training cost.

4. Results

We evaluated the proposed method on the MNIST [11], CIFAR-10 [12], SVHN [13], and Caltech 101 [14] datasets. We made adversarial examples using several adversarial attacks and applied DPRNet to these images to generate reconstructed images. In this section, we describe experimental settings and evaluate the classification of the reconstructed images. Moreover, we analyze the characteristics of the reconstructed images generated by DPRNet and show differences in class probability distributions compared to adversarial images.

4.1 Experimental Setting

We used the MNIST, CIFAR-10, SVHN, and Caltech 101 datasets as evaluation data. The MNIST and CIFAR-10 datasets each have 60,000 training images and 10,000 test images. In these two datasets, we split the training images into three parts: 30,000 for pretraining, 20,000 for training, and 10,000 for validation. In the SVHN dataset, we also split the training images into three
Table 2. CNN structure for MNIST classification

| Layer   | Channels | Filter Size | Stride |
|---------|----------|-------------|--------|
| Conv.   | 32       | 3 x 3       | -      |
| Conv.   | 32       | 3 x 3       | -      |
| Maxpool. | -       | 2 x 2       | -      |
| Conv.   | 64       | 3 x 3       | -      |
| Dense   | -        | -           | -      |
| Dense   | -        | -           | -      |
| Softmax | -        | -           | -      |

Parts: 30,000 for pretraining, 33,257 for training, and 10,000 for validation. The classifier is trained with the pretraining images. The parameters of the classifier are fixed during the training of the DPRNet parameters. DPRNet is first trained with the pretraining images with data augmentation (rotation and flip) and then with the training images with the fixed classifier parameters. We used the Caltech 101 dataset to evaluate multi-class classification with more classification classes compared to the MNIST, CIFAR-10, and SVHN datasets. The details are presented in Section 4.3. The images were normalized to the range [0, 1] for training and testing.

In this study, we used two distortion types, Gaussian noise and Gaussian blur. We randomly added noise to the original images during training. The range of standard deviation for the Gaussian noise was (0, 100) and that for the Gaussian blur was (0, 10). The kernel size of the Gaussian blur is defined as follows:

\[ K = 4\sigma_{\text{blur}} - 1. \]  

(5)

\( K \) denotes the kernel sizes of the Gaussian blur. \( \sigma_{\text{blur}} \) is the standard deviation of the Gaussian blur. We set five noise levels: for the Gaussian noise, we used \{20, 40, 60, 80, 100\} as the standard deviations and for the Gaussian blur, we used \{2, 4, 6, 8, 10\} as the standard deviations. Therefore, the kernel sizes of the Gaussian blur \( K \) were \{7, 15, 23, 31, 39\}.

In this study, we used Adam [42] to optimize the classifier and DPRNet. During pretraining, we set the learning rate as 0.001, the dropout rate as 0.2, the batch size as 32, and the maximum number of epochs as 200. In each epoch, we used early stopping when the validation loss was not updated after five epochs. During training, we set the learning rate as 0.0001 and the other parameters kept the same values as during pretraining. In Equation (4), we evaluated different \( \alpha \) values \{0.5, 0.05, 0.005\} on the validation data, ultimately using \( \alpha = 0.05 \) in Equation (4).

To make adversarial images, we used three adversarial attacks: BIM, C&W, and Deepfool. For BIM, we set \( \epsilon = 0.3 \) and we endeavored to minimize the L2 distance for C&W and Deepfool. The structure of the CNN used for classifying MNIST data is shown in Table 2. This CNN uses ReLU as an activation function after convolution. The dimension of outputs in the fully connected layers before the softmax layer were 500 and 10. We used ResNet32 to classify the CIFAR-10 and SVHN datasets.

4.2 Classification of Adversarial Images

Table 3 shows the classification results of reconstructed images generated by DPRNet from adversarial images in the MNIST, CIFAR-10, and SVHN datasets. The “w/o Reconstruction” column shows the result of classifying the adversarial images using the classifiers directly, without any reconstruction. The “GF,” “MF,” and “BM3D” columns list the classification results of denoised images to which a Gaussian filter, a median filter with a kernel size of 3 x 3, and a BM3D filter [43] with a standard deviation of seven, respectively, were applied to eliminate adversarial perturbations. The “AE” column lists the result of an AE with the same structure as that of DPRNet. In this experiment, the AE was trained with Equation (2) only, not with Equation (3). The “DPRNet” column is the classification result of reconstructed images generated by DPRNet from the adversarial images.

The classification accuracy of images denoised using a Gaussian filter is degraded considerably. For the CIFAR-10 dataset, the result of applying the Gaussian filter to the no attack (original) image is 0.2119, and the result of the image perturbed by C&W is also 0.2119. This result implies that the adversarial perturbation itself has been eliminated by the Gaussian filter; however, the accuracy degradation due to blurring is large.

The classification accuracy of applying the median filter produces better results compared to those of the Gaussian filter. However, the classification accuracy is degraded compared to the results of the no attack (original) images, especially in the case of CIFAR-10 data. The accuracies of the no attack (original) images using the median filter are 0.9858 and 0.9482 on the MNIST and SVHN datasets, respectively. Classification accuracy is less degraded when the median filter is applied because the images in MNIST and SVHN have simple textures compared to those in the CIFAR-10 dataset.

The BM3D denoising filter yields better accuracy than do the smoothing filters. In the no attacked images of the MNIST dataset, the accuracy of the BM3D filter is 0.9919, which is higher than the accuracy of DPRNet. However, in the attacked images of all datasets, DPRNet obtains higher accuracy than the BM3D filter for several attacks. In particular, on the CIFAR-10 dataset, DPRNet achieves better performance than the other denoising methods. This implies that DPRNet can eliminate the adversarial perturbations efficiently and generate images that retain the signals important for classification.

In MNIST, the classification accuracy of BIM images is degraded compared to those of C&W and Deepfool, suggesting that the adversarial perturbations of BIM are larger than those of the other attacks. The larger perturbations are difficult for the Gaussian and median filters to eliminate. In fact, the MSEs of the BIM, C&W, and Deepfool images are 0.0028, 0.0013, and 0.0020, respectively. MSEs were calculated between the original images and the adversarial images.

In SVHN, the classification accuracy of DPRNet is also better than the classification accuracy of AE. Moreover, the improvement is larger than that obtained for the MNIST dataset. Because SVHN includes more complicated textures than can be found in the MNIST dataset, we consider this improvement to be substantial. However, the BM3D filter has a similar performance to that of DPRNet, and the accuracy of BM3D is better than that of...
DPRNet in BIM. Although the MNIST and SVHN datasets have a simple texture compared to the CIFAR-10 dataset, the BM3D filter of the two datasets results in better performance than the performance of the CIFAR-10 dataset. The BM3D filter has good performance. However, this filter is computationally expensive.

Compared to the Gaussian and median filters, the AE and DPRNet are more accurate. The AE and DPRNet can eliminate adversarial perturbation without degrading the classification accuracy for each attack. DPRNet exhibits higher classification accuracy than AE for both datasets. Compared to the AE, which is trained only with L1 distance as the loss, DPRNet can reconstruct an image that is more suitable for classification. In addition to the effect of removing adversarial perturbations by training with L1 distance, the adversarial image is reconstructed with the correct class probability distribution by the DPRNet trained with KL divergence. Although the difference between the class probability distributions is expressed by KL divergence, DPRNet can reconstruct the image with the correct class probability distribution. The classification result of the no attack (original) images using the AE is 0.8751 and that of DPRNet is 0.8872. The performance degrades that occurs when the original images are transformed with the BM3D filter. The BM3D filter has good performance. However, this filter is computationally expensive.

The classification result of the no attack (original) images using the DPRNet trained with KL divergence. Although the difference between the class probability distributions is expressed by KL divergence, DPRNet can reconstruct the image with the correct class probability distribution. The classification result of the no attack (original) images using the AE is 0.8751 and that of DPRNet is 0.8872. The performance degrades that occurs when the original images are transformed with the BM3D filter. The BM3D filter has good performance. However, this filter is computationally expensive.

Table 3 Adversarial examples classification results

| Datasets | Attack Type | w/o Reconstruction | GF | MF | BM3D | AE | DPRNet |
|----------|-------------|---------------------|----|----|------|----|---------|
| MNIST    | No attack   | 0.9919              | 0.6901 | 0.9858 | 0.9919 | 0.9914 | 0.9915 |
|          | BIM         | 0.0000              | 0.5824 | 0.9524 | 0.9847 | 0.9888 | 0.9890 |
|          | C&W         | 0.0000              | 0.6269 | 0.9679 | 0.9909 | 0.9908 | 0.9912 |
|          | Deepfool    | 0.3002              | 0.5849 | 0.9514 | 0.9003 | 0.9900 | 0.9906 |
| CIFAR-10 | No attack   | 0.9044              | 0.2119 | 0.7412 | 0.8726 | 0.8751 | 0.8872 |
|          | BIM         | 0.0005              | 0.2114 | 0.7119 | 0.8461 | 0.8573 | 0.8664 |
|          | C&W         | 0.0000              | 0.2119 | 0.7228 | 0.8481 | 0.8582 | 0.8658 |
|          | Deepfool    | 0.0129              | 0.2106 | 0.7062 | 0.8325 | 0.8411 | 0.8503 |
| SVHN     | No attack   | 0.9537              | 0.8999 | 0.9482 | 0.9505 | 0.9496 | 0.9520 |
|          | BIM         | 0.0002              | 0.8898 | 0.9336 | 0.9436 | 0.9342 | 0.9416 |
|          | C&W         | 0.0000              | 0.8895 | 0.9327 | 0.9426 | 0.9406 | 0.9429 |
|          | Deepfool    | 0.0068              | 0.8856 | 0.9208 | 0.9333 | 0.9328 | 0.9335 |

Table 4 Adversarial examples classification results on the Caltech 101 dataset

| Attack Type | w/o Reconstruction | AE | DPRNet |
|-------------|--------------------|----|--------|
| No attack   | 0.8807             | 0.8802 | 0.8840 |
| C&W         | 0.0000             | 0.8386 | 0.8457 |

4.3 Evaluation on Caltech 101 Dataset

To evaluate DPRNet for a multi-class dataset that has more classes compared to MNIST, CIFAR-10, and SVHN, we used the Caltech 101 dataset as evaluation data. We randomly split the data as 80% for training and 20% for testing and used 20% of the training data for validation. We used VGG16 to classify the Caltech 101 dataset. When training and testing, we resized the images of the dataset to 224 × 224 for training. We set the parameters of DPRNet to the same value as the parameters of the other datasets except for α in Equation (4). We used α = 0.5 in the Caltech 101 dataset.

Table 4 shows the classification results of reconstructed images generated by DPRNet from adversarial images. Adversarial images were generated by the C&W attack. Compared to the accuracy of the AE, the accuracy of DPRNet is higher; specifically, 0.8457. DPRNet invalidates 96.0% of the adversarial images. Hence, our proposed DPRNet is effective on the multi-class dataset that has more classes than the other datasets. More interestingly, the accuracy of DPRNet in no attack is 0.8840, which is higher than the accuracy of the classifier without reconstruction. In this dataset, DPRNet with PRloss is able to generate signals that enable more accurate recognition compared to the classifier that was trained and tested using the original images.

4.4 Comparison with Knowledge Distillation

Our proposed DPRNet does not need class labels and adversarial examples during training. Knowledge distillation (KD) [24] is also a defense method that does not need class labels and adversarial examples. Hence, KD is suitable for comparison with DPRNet.

Table 5 shows the classification results of KD and DPRNet on...
the CIFAR-10 dataset. The results of DPRNet are the same as in Table 3. ResNet32 was used as the classifier. In the experiment, we used the ResNet32 model as the teacher of KD and trained the same model from scratch as the student of KD. We used the loss metric of KD, which is the same as the proposed loss metric in [24]. As the accuracy of the original classifier in the CIFAR-10 is 0.9919 and the average accuracy of KD in each attack is 0.8141, KD invalidates 90.0% of the adversarial images. The invalidation rate of DPRNet is 95.1%. Therefore, DPRNet outperforms KD in view of the invalidation rate.

Furthermore, we investigated the effect of the proposed training procedure on KD. When training the classifier, Gaussian noise and blurred images are used as training data in the proposed training procedure. The half images in minibatch are the clean images, and the other images are the distorted images. Because distorted images are included in minibatch, it is more difficult for the classifier to learn network parameters. Hence, the accuracy of the classifier that is trained with the distorted images as a teacher model is degraded. In our experiment, the accuracy of the ResNet32 trained with our proposed training procedure from scratch was 0.7509. The classifier trained with the proposed training procedure had degraded accuracy compared to the classifier trained with the clean images only. In addition, we trained the distillation network from the ResNet32 trained with the procedure to defend against adversarial images. The accuracy of the distillation network was 0.7496 for the adversarial images generated by the C&W attack. The accuracy was slightly degraded compared to the accuracy of the ResNet32 trained with the procedure. Knowledge distillation is very effective for adversarial images; however, the accuracy of the teacher classifier is low because of the distorted images. To obtain high accuracy on the teacher model, we therefore consider that a suitable pretrained model and a different training procedure are necessary.

### 4.5 Quality of the Reconstructed Images

Figure 2 shows the adversarial images attacked by BIM and the same images reconstructed by DPRNet. Both the adversarial and reconstructed images are virtually indistinguishable from the original images by human vision. Although DPRNet incorporates KL divergence into the loss metric, its effect is not visible in the reconstructed images.

Figure 3 shows difference images between the original images and the adversarial images attacked by C&W, and the reconstructed images generated by the AE and DPRNet. Each image is shown in grayscale and is enhanced to highlight the differences. Note that the differences in the DPRNet images are stronger in the recognition target and its surroundings than the differences in the AE images. Because the edge of the recognition object is important for the classifier, DPRNet appears to have tried to recover the edges of the recognition object and its surroundings. In addition, because the textures around the recognition object are complex, the reconstructed images generated by DPRNet are virtually indistinguishable from the original images. The reconstructed images generated by DPRNet have the effect of adding
image signals that are necessary for classification; however, the signals cannot be seen by human vision.

Our objective is to defend against adversarial images in image classification tasks by using an image quality recovery method. In general, improving image quality has a good effect on image classification accuracy. However, classification accuracy and image quality are not always correlated. Hence, we not only need an image quality metric but also a recognition accuracy metric combined in a loss metric. The PRloss satisfies this by combining the MSE and the KL divergence. In this context, conventional image quality metrics are less meaningful because image quality and classification accuracy are not necessarily correlated. However, when there is no noise, image quality metrics are important to achieve high accuracy. The effect of image quality on accuracy has not been studied extensively, and it is not well known whether the accuracy of classification increases with better performance of image quality recovery methods. Therefore, we evaluated the image quality of adversarial and reconstructed images to investigate the relationship between image quality and classification accuracy.

Table 6 shows the image quality of the adversarial, smoothed, and reconstructed images. We used the peak signal-to-noise-ratio (PSNR) and structural similarity index measure (SSIM) [44] as image quality metrics. The PSNR of the adversarial images attacked by BIM is 45.52. Because the adversarial perturbation is small, the image quality is high. Compared with the adversarial images, the PSNR of the images after filtering by the Gaussian and median filters is degraded. Moreover, the PSNR of the reconstructed images generated by DPRNet is also degraded. However, although applying the Gaussian and median filters substantially reduces classification accuracy, applying DPRNet has less effect on it. In other words, despite the low PSNR, the classification accuracy of the reconstructed images generated by DPRNet is high. In addition, although the images filtered by the BM3D filter have a higher PSNR compared to DPRNet, the classification accuracy of the reconstructed images generated by DPRNet is also higher than that of the images filtered by the BM3D filter. This means that image quality and classification accuracy are not correlated, and by incorporating KL divergence into the loss, DPRNet generates the signals that are necessary for classification. Further, the SSIM of the reconstructed images generated by DPRNet is higher than that of the adversarial images attacked by BIM. This implies that the structure of an image is important for image classification.

### 4.6 Class Probability Distributions of the Reconstructed Images

Figure 4 shows the class probability distributions, which are the outputs of the softmax layer of the classifier, of the images shown in Table 6.
in Fig. 2. Figs. 4(a)–(d) correspond to the images in rows 1–4 in Fig. 2. The adversarial images using the C&W attack had a higher class probability in a wrong class than in the target class. After applying DPRNet to the adversarial images, the reconstructed images have the highest class probability in the correct class because the adversarial perturbations can be removed properly. Moreover, using the original images, which are not included among the classifier training images, we can obtain a similar class probability distribution during testing when training DPRNet, with its KL divergence loss.

5. Conclusion

We proposed DPRNet, an encoder–decoder network that removes perturbations from adversarial images by training with a metric called PRloss, which consists of a reconstruction loss and a KL divergence loss that expresses the class probability difference between an original image and a reconstructed image. Using the PRloss metric, adversarial images are reconstructed to suppress degradation of classification accuracy compared to the AE that uses the reconstruction loss only. A classifier can classify the reconstructed images accurately without retraining. Moreover, by training with distorted images, the proposed method does not need adversarial images for training compared to some defense methods for adversarial images. Experimental results show that training with the distorted images to which Gaussian noise and blur have been added is effective in removing perturbations generated from several adversarial attacks. In addition, the classification accuracy of reconstructed images is improved by considering the class probability difference between an original image and a reconstructed image.

In the future, we will investigate the effect of an attacker who knows which defense method is being used and will construct a classifier that is difficult to attack even with tailored adversarial examples.

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus: “Intriguing properties of neural networks,” Proc. of Int. Conf. on Learning Representations (ICLR), Apr. 14-16, Banff, 2014.

[2] A. Kurakin, I. J. Goodfellow, and S. Bengio: “Adversarial examples in the physical world,” arXiv preprint, arXiv:1607.02533, 2016.

[3] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille: “Adversarial examples for semantic segmentation and object detection,” Proc. of IEEE Int. Conf. on Computer Vision (ICCV), Oct. 22-29, Venice, pp. 1378-1387, 2017.

[4] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou: “Hidden voice commands,” Proc. of the 25th USENIX Conf. on Security Symp., Aug. 10-12, Austin, pp. 513-530, 2016.

[5] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu: “DolphinAttack: Inaudible voice commands,” Proc. of ACM SIGSAC Conf. on Computer and Communications Security, Oct. 30-Nov. 3, Dallas, pp. 103-117, 2017.

[6] N. Papernot, P. D. McDaniel, and I. J. Goodfellow: “Transferability in machine learning: from phenomena to black-box attacks using adversarial samples,” arXiv preprint, arXiv:1605.07277, 2016.

[7] W. He, J. Wei, X. Chen, N. Carlini, and D. Song: “Adversarial example defenses: Ensembles of weak defenses are not strong,” Proc. of USENIX Conf. on Offensive Technologies, Aug., Berkeley, 2017.

[8] S. Dodge and L. Karam: “Understanding how image quality affects deep neural networks,” Proc. of the 5th Int. Conf. on Quality of Multimedia Experience (QoMEX), Jun. 6-8, Lisbon, pp. 1-6, 2016.

[9] X.-J. Mao, C. Shen, and Y.-B. Yang: “Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections,” Proc. of Advances in Neural Information Processing Systems (NIPS 2016), Dec. 5-10, Barcelona, pp. 2810-2818, 2016.

[10] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, “Noise2Noise: Learning image restoration without clean data,” arXiv preprint, arXiv:1803.04189, 2018.

[11] The MNIST database of handwritten digits: http://yann.lecun.com/exdb/mnist [accessed Jan. 22, 2020]

[12] A. Krizhevsky: “Learning multiple layers of features from tiny images,” Tech Report, pp. 1-58, 2009.

[13] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng: “Reading digits in natural images with unsupervised feature learning,” NIPS 2011 Workshop on Deep Learning and Unsupervised Feature Learning, Dec., Granada, 2011.

[14] L. Fei-Fei, R. Fergus, and P. Perona: “Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories,” 2004 Conf. on Computer Visionand Pattern Recognition Workshop, Jun. 27-Jul. 2, Washington, pp. 178-178, 2004.

[15] I. J. Goodfellow, J. Shlens, and C. Szegedy: “Explaining and harnessing adversarial examples,” Proc. of the 3rd Int. Conf. on Learning Representations (ICLR), May 7-9, San Diego, 2015.

[16] A. Kurakin, I. J. Goodfellow, and S. Bengio: “Adversarial machine learning at scale,” Proc. of the 5th Int. Conf. on Learning Representations (ICLR 2017), Apr. 24-26, Toulon, 2017.

[17] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard: “Deepfool: A simple and accurate method to fool deep neural networks,” Proc. of the 2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Jun. 26-Jul. 1, Las Vegas, pp. 2574-2582, 2016.

[18] N. Carlini and D. Wagner: “Towards evaluating the robustness of neural networks,” Proc. of the 35th IEEE Symp. on Security and Privacy (SP), May 22-24, San Jose, pp. 39-57, 2017.

[19] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh: “ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models,” Proc. of the 10th ACM Workshop on Artificial Intelligence and Security (AISec), Nov. 3, Dallas, pp. 15-26, 2017.

[20] S. M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard: “Universal adversarial perturbations,” Proc. of 2017 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Jul. 21-26, Honolulu, pp. 86-94, 2017.

[21] J. Su, D. V. Vargas, and K. Sakurai: “One pixel attack for fooling deep neural networks,” IEEE Trans. on Evolutionary Computation, Vol.23, No.5, pp. 828-841, 2019.

[22] R. Huang, B. Xu, D. Schuurmans, and C. Szepesvári: “Learning with a strong adversary,” arXiv preprint, arXiv:1511.03034, 2015.

[23] A. Rozsa, M. Günther, and T. E. Boult: “Towards robust deep neural networks with BANG,” Proc. of IEEE Winter Conf. on Applications of Computer Vision (WACV), Mar. 12-15, Lake Tahoe, pp. 803-811, 2018.

[24] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami: “Distillation as a defense to adversarial perturbations against deep neural networks,” Proc. of IEEE Symp. on Security and Privacy (SP), May, San Jose, pp. 582-597.

[25] J. Lu, T. Issaranon, and D. A. Forsyth: “Safetynet: Detecting and
rejecting adversarial examples robustly,” *Proc. of IEEE Int. Conf. on Computer Vision (ICCV)*, Oct. 22-29, Venice, pp. 446-454, 2017.

[26] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff: “On detecting adversarial perturbations,” *Proc. of Int. Conf. on Learning Representations (ICLR)*, Apr. 24-26, Toulon, 2017.

[27] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman: “PixelDefend: Leveraging generative models to understand and defend against adversarial examples,” *Proc. of Int. Conf. on Learning Representations (ICLR)*, Apr. 30-May 3, Vancouver, 2018.

[28] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. McDaniel: “On the (statistical) detection of adversarial examples,” *arXiv preprint, arXiv:1702.06280*, 2017.

[29] V. Nair and G. E. Hinton: “Rectified linear units improve representations (ICLR),” Apr. 24-26, Toulon, 2017.

[30] K. Simonyan and A. Zisserman: “Very deep convolutional networks for large-scale image recognition,” *Proc. of Int. Conf. on Learning Representations (ICLR)*, May 7-9, Sandiego, 2015.

[31] K. He, X. Zhang, S. Ren, and J. Sun: “Deep residual learning for image recognition,” *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 27-30, Las Vegas, pp. 770-778, 2016.

[32] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu: “Pixel recurrent neural networks,” *Proc. of the 33rd Int. Conf. on Machine Learning (ICML)*, Jun. 19-24, pp. 1747-1756, 2016.

[33] W. Xu, D. Evans, and Y. Qi: “Feature squeezing: Detecting adversarial examples in deep neural networks,” *Proc. of Network and Distributed Systems Security Symp. (NDSS)*, Feb. 18-21, San Diego, 2018.

[34] D. Meng and H. Chen: “Magnet: A two-pronged defense against adversarial examples,” *Proc. of the 2017 ACM SIGSAC Conf. on Computer and Communications Security*, Oct., Dallas, pp. 135-147, 2017.

[35] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu: “Defense against adversarial attacks using high-level representation guided denoiser,” *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 18-23, Salt Lake City, pp. 1778-1787, 2018.

[36] O. Ronneberger, P. Fischer, and T. Brox: “U-net: Convolutional networks for biomedical image segmentation,” *Lecture Notes in Computer Science: The Int. Conf. on Medical Image Computing and Computer-Assisted Intervention Proc. (MICCAI)*, Vol.9351, Oct. 5-9, Munich, pp. 234-241, 2015.

[37] U. Hwang, J. Park, H. Jang, S. Yoon, and N. I. Cho: “PaVAE: A variational autoencoder to purify adversarial examples,” *IEEE Access*, Vol.7, pp. 126582-126593, 2019.

[38] S. Ioffe and C. Szegedy: “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *Proc. of the 32nd Int. Conf. on Machine Learning (ICML)*, Jul. 6-11, Lille, Vol.37, pp. 448-456, 2015.

[39] K. He, X. Zhang, S. Ren, and J. Sun: “Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification,” *Proc. of IEEE Int. Conf. on Computer Vision (ICCV)*, Dec. 7-13, Santiago, pp. 1026-1034, 2015.

[40] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov: “Dropout: A simple way to prevent neural networks from overfitting,” *J. of Machine Learning Research*, Vol.15, No.56, pp. 1929-1958, 2014.

[41] M. Takagi, A. Sakurai, and M. Hagiwara: “Quality recovery for image recognition,” *IEEE Access*, Vol.7, pp. 105851-105862, 2019.

[42] D. Kingma and J. Ba: “Adam: A method for stochastic optimization,” *Proc. of Int. Conf. on Learning Representations (ICLR)*, May 7-9, San Diego, 2015.

[43] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian: “Image denoising by sparse 3-D transform-domain collaborative filtering,” *IEEE Trans. on Image Processing*, Vol.16, No.8, pp. 2080-2095, 2007.

[44] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli: “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. on Image Processing*, Vol.13, No.4, pp. 600-612, 2004.