Human-Centered Design and User Experience, Vol. 114, 2023, 322–330
https://doi.org/10.54941/ahfe1004250

A Method for Adversarial Example Generation Using Wavelet Transformation

Kanato Takahashi[1], Masaomi Kimura[1], Imam Mukhlash[2], and Mohammad Iqbal[2]

[1]Shibaura Institute of Technology, Koto, Tokyo 135-8548, Japan
[2]Sepuluh Nopember Institute of Technology, Keputih, Sukolilo-Surabaya 60111, Indonesia

ABSTRACT

Deep neural networks (DNN) have improved the accuracy of machine learning tasks, particularly image classification. However, DNNs are vulnerable to adversarial examples, which are small changes made to an image that can cause the DNN model to misclassify the image. This poses a major problem for practical image recognition and has led to research on methods for generating and defending against adversarial examples. The popular approach to attacking DNN models involves adding perturbations to images in the spatial domain, but we propose a new method that focuses on the spatial frequency domain. By adding perturbations to the high-frequency components of images, we generate adversarial examples that appear similar to the original image. This is because the low-frequency component is responsible for the overall color distribution in an image, making any changes more noticeable, while the high-frequency component holds less information and makes changes less apparent. To implement this approach, we apply the discrete wavelet transformation to target images. Our method results in smaller changes to the image, as measured by the peak signal-to-noise ratio (PSNR), and improves the attack accuracy by 9% compared to previous work without using quantization. Our experiments show that our method has a PSNR of about 43, compared to about 32 in previous studies.

Keywords: Adversarial examples, Wavelet transformation, Deep neural networks

INTRODUCTION

Deep neural networks (DNN) have demonstrated exceptional performance in numerous fields, including image generation, search engines, translation, autonomous driving, and face recognition. Image recognition is one of them. Currently, various high-performance models such as ResNet (He, 2016), EfficientNet (Tan, 2019), Vision Transformer (Dosovitskiy, 2020), etc. have appeared. However, the vulnerability of the DNN model has been pointed out in various papers. It was first reported by Szegedy et al. (Szegedy, 2013) that adding a small perturbation to the image causes the DNN model to misclassify the image. Such images are called adversarial examples. The study of adversarial examples involves attack methods to fool the model and defensive methods to prevent the model from being fooled.

Many attack methods perturb images in the spatial domain, but there are also attack methods that focus on the spatial frequency domain (Guo, 2018) (Luo, 2022). Converting the image to the frequency domain reveals locations where pixel values change significantly and locations where they change smoothly. It allows us to separate the elements of the image by frequency band, such as high, medium, and low frequencies, and to separate the processing by the characteristics of each frequency band. This helps to make any perturbations added to the image less noticeable. AdvDrop (Duan, 2021) is another method that focuses on the frequency domain. AdvDrop first applies a discrete cosine transform (DCT) to an image to transform it into the frequency domain. DCT is a method of converting signals on the time axis into signals in the frequency domain, and this technique is also used as an image analysis method. The image is processed by dividing it into blocks and transforming the contents of the blocks into a combination of cosine functions with various frequencies and amplitudes. AdvDrop generates an Adversarial Example by restoring the image through an inverse discrete cosine transform after quantization by DCT. This method is based on JPEG compression (Wallace, 1992). Therefore, this method produces an adversarial example that is closer to the original appearance of the image than existing studies. However, since this method divides the image into 88 pixels to use DCT, there is a problem that the division on the block appears in the generated adversarial example. Additionally, AdvDrop simply reduces image quality through quantization, which ultimately lowers the accuracy of attacks in black-box scenarios. The objective of our research is to overcome the weaknesses of this approach. We propose a new method that uses discrete wavelet transformation (DWT) for frequency analysis. Unlike DCT, DWT transforms a signal into the frequency domain while retaining time information. This means that an image is processed without being separated into blocks during frequency analysis, resulting in adversarial examples without blocky delimiters. DWT uses a function called "mother wavelet" as a basis for frequency analysis, and our method uses Daubechies wavelets, which are well-known mother wavelets. DWT performs finer frequency analysis by using multiple wavelets of different scales. Our method uses level-3 DWT, which enables a more comprehensive frequency analysis of images.

The contributions of this paper are as follows.

- We propose a new adversarial attack method using DWT to generate adversarial examples that do not change much from the original image.
- We show that by implementing our unique method of adding perturbations in the frequency domain, our method has a higher misclassification rate than existing adversary attack methods using frequency analysis techniques.

Related Work

There are various generation methods for adversarial examples (Carlini, 2017) (Dong, 2018) (Goodfellow, 2014) (Kurakin, 2016) (Xiao, 2018). Also, an adversarial example that causes a model to misclassify an image has the property of causing another model to misclassify it as well (Papernot, 2017).

This property is called transferability. By using transferability, the attack can succeed without knowing the internal information of the target model. Such attacks are called black box attacks (Chen, 2017), and various attack techniques are compared in such situations.

While many attack methods perturb images in the spatial domain, some methods perturb images in the spatial frequency domain (Duan, 2021) (Guo, 2018) (Luo, 2022). By processing images in the frequency domain, it is possible to see where in the image the pixel values have changed significantly, making the processing more imperceptible. To the extent that there is a wide variety of attack methods, there is also a wide variety of defense methods (Blau, 2022)(Dhillon, 2018) (Jang, 2019) (Liao, 2018) (Sriramanan, 2020). Therefore, attackers need to consider attack methods, assuming that defensive methods are also applied to the model.

When utilizing DCT, AdvDrop (Duan, 2018) splits the image into blocks and carries out individual quantization for each block. However, due to the independent process of each block, visible lines may appear at the edges where the blocks meet. Since AdvDrop is a method based on JPEG compression, it drops most of the high-frequency component elements. The DWT method enables frequency analysis retaining time information, eliminating the need to divide images into blocks for image processing. Moreover, by applying DWT on various levels, like levels 2 and 3, the image is subdivided into finer frequencies and processed accordingly. This property of DWT is utilized in our method to produce an adversarial example from an original image with minimal modifications.

Proposed Method

Our objective is to propose a method that is superior to AdvDrop in both misclassification rate and image quality. To prevent block delimiters from appearing in the image, our method uses DWT instead of DCT. DWT is an image analysis technique that processes an image without separating it into blocks so that no block separations appear in the image after it is restored. The outline of our method is shown in Figure 1.

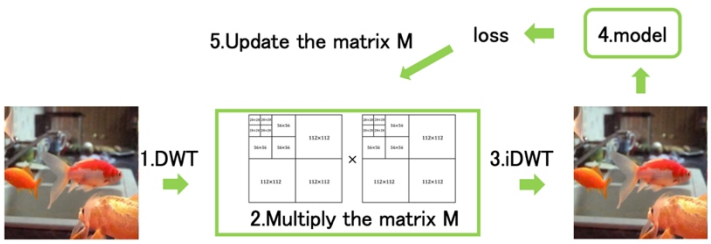

Figure 1: The outline of our method.

Our method consists of the following five steps:

1. Application of DWT to an image
   In AdvDrop, step 1 divides the image into 8x8 blocks and applies DCT to the image to analyze the image frequency.

2. Multiplication of the image by the matrix M
   Our target is square images, whose size is $N \times N$. Using a level-3 DWT on an image of that size would output three $\frac{N}{2} \times \frac{N}{2}$ matrices, three $\frac{N}{4} \times \frac{N}{4}$ matrices, and four $\frac{N}{8} \times \frac{N}{8}$ images. Our method uses a matrix of the same size as the target images, which has the same number of elements as the number of the unified outputs of DWT. These matrices are updated at once by using the gradient of the loss function, as in the following equation.

$$M' = M + \alpha \frac{\partial L}{\partial x}$$

   Where M is the matrix group, L is the loss function, α is the learning rate for updating M, M' is updated M and x is the input image. The matrix group M is initialized to 1 and multiplied by each corresponding element of the image.
3. Application of iDWT to the image
   iDWT is the reverse process of DWT. It restores the frequency-analyzed image to its original form.
4. The input of images to the model
   The restored image is input to the model. If the model misclassifies the image, the attack is successful and the attack ends.
5. Update of M components using the gradient of the loss function.
   Each element of M is updated based on the gradient of the loss function resulting from the input of the image to the model.

M is multiplied by the image transformed into the spatial frequency domain. M is of the same form as the image after DWT is applied, with all elements initially initialized at 1. It is updated by the gradient of the loss function and optimized so that the model misclassifies the image. To minimize changes in the image, M is multiplied only by the high-frequency component. This is because images generally do not retain much information in the high-frequency components and changing the high-frequency components does not change the image significantly. The image looks different when DWT is applied and the pixel values are increased or decreased. If the high-frequency component of the image is increased or decreased, the changes to the original image will be more noticeable when the increase is larger. To avoid significant changes, the M is only updated when its elements decrease. This method effectively suppresses changes in the image.

EXPERIMENT

We conducted two experiments to assess how well our method can generate adversarial examples. Experiment 1 examined how the generated adversarial example differs from the original image. In Experiment 2, we conducted a black-box attack experiment where we input adversarial examples generated by each method into a different model than the one used in Experiment 1. We compared the attack accuracy of our method with that of AdvDrop.

Experiment 1: Generation of Adversarial Examples

AdvDrop has a parameter ϵ to control the magnitude of the perturbation. Since Advdrop used $\epsilon = 100$ in the experiment, the same setting was used in this experiment. In addition, the learning rate α in our method was set to 1.5e-2 in this experiment. α is the updated width of the M. The larger α is, the larger the perturbation applied to the image is likely to be.

In this experiment, we used VGG19, which is pre-trained on ImageNet. We used 2,500 images from ImageNet with a unified image size of 224×224. All 2,500 images were correctly classified by VGG19, which was used in this experiment. The results of this experiment are as follows.

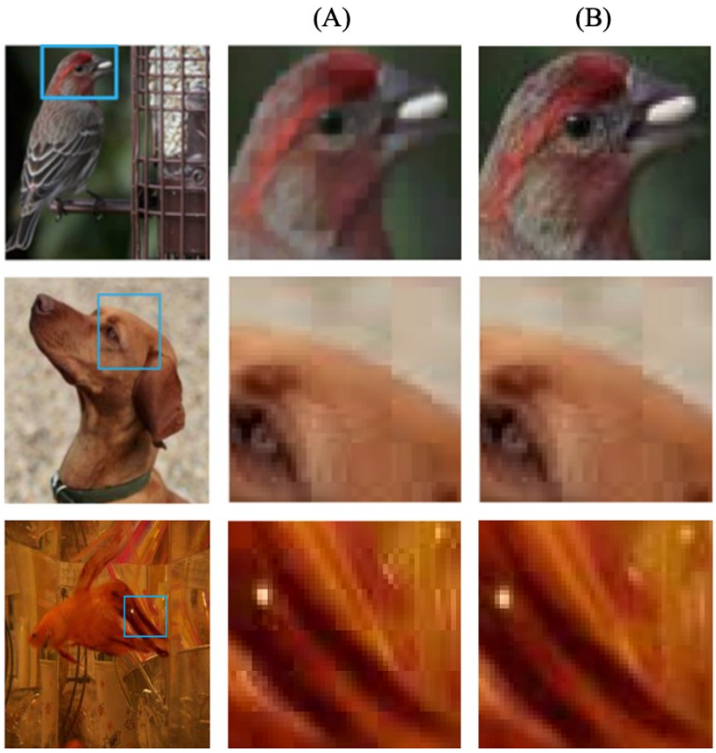

Figure 2: Generation of adversarial examples. The images in (A) are generated by AdvDrop. The images in (B) are generated by our method.

The images in (A) are the adversarial examples generated by AdvDrop and the images in (B) are generated by the proposed method. Comparing them, the images generated by AdvDrop have blocky separations, while the images generated by our method do not have such separations. In Image (A) at the top of Figure 2, we can see that there are unnatural bumps in the outline of the bird image and that many vertical and horizontal lines in the area of the bird are not present in Image (B). AdvDrop uses a DCT that requires image segmentation, while our method uses a DWT that does not require image segmentation. That is why the visual differences shown in Figure 2 occurred.

Experiment 2: Black-Box Attack

In Experiment 2, we checked how well AdvDrop and our method make the model misclassify the images in a Black-Box Attack situation. We used VGG11, which was pre-trained on ImageNet. We set the learning rate of our method to 1.5e-2, 2.0e-2, 2.5e-2, 3.0e-2, and 3.5e-2 with a maximum number of steps of 100. The images used to generate adversarial examples were the same 2,500 images from ImageNet as in Experiment 1. The images used to generate the adversarial examples were the same 2,500 images from ImageNet as in Experiment 1, and we checked the misclassification rate of the model by inputting these images into VGG11 after the generation of the adversarial sample. We used the Peak Signal-to-Noise Ratio (PSNR) to numerically compare the degree of degradation of the generated adversarial example from the original images. The results of Experiment 2 were calculated as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - x_i')^2$$

$$PSNR = 10 \cdot \log_{10}\frac{255^2}{\mathrm{MSE}}$$

MSE is the mean squared error between the original image and the generated adversarial example. The more degraded the images are, the smaller the PSNR value is. PSNR is frequently utilized as an objective evaluation measure because it does not involve intricate computations. The average PSNR values of 2,500 images are as follows.

Table 1. Results of experiment 2.

	Misclassification Rate	Mean of PSNR
AdvDrop	18.68%	32.43
Ours(α:1.5e-2)	21.64%	43.16
Ours(α:2.0e-2)	22.48%	41.25
Ours(α:2.5e-2)	23.88%	39.68
Ours(α:3.0e-2)	26.24%	38.46
Ours(α:3.5e-2)	27.80%	37.30

Our method had the highest misclassification rate with a learning rate of 3.5e-2. The misclassification rate of attacks between AdvDrop and our method differs by up to about 9%. Comparing the average PSNR values of our method and AdvDrop, the proposed method has higher values than AdvDrop at all learning rates. Therefore, the adversarial examples generated by our method are less degraded than those by AdvDrop. Our method shows that the misclassification rate of attacks increases as the learning rate increases, but not the PSNR.

Visualizing *M*

Next, by visualizing the amount of change from the initial value of each element of *M* optimized in Experiment 2 in the same form as Fig. 2, we confirmed whether there is a difference in the perturbation applied depending on the magnitude of the frequency and whether the same pattern of perturbation appears in different images. Figure 3 shows a visualization of the amount of change from the initial value of each element of *M*. We call these matrices perturbation matrices. The figure shows the perturbation matrices for six images generated by the proposed method with a learning rate of 3.5e-2 in Experiment 2.

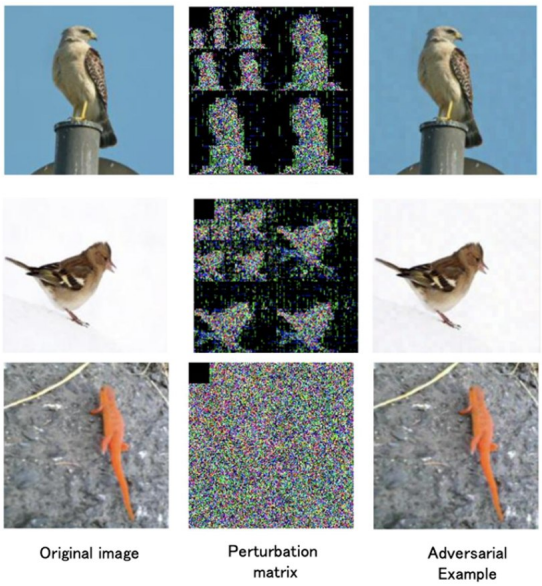

Figure 3: Visualizing *M*.

If the change from the initial value of *M* is 0, that part of the perturbation matrix is black. The displayed images have a size of 224x224, but only the top-left portion of 28x28 is relevant for the low-frequency component. Our method does not affect the low-frequency component, so the perturbation matrix in the upper left part is black. In some cases, perturbations are applied to the entire image, like in the bottom image of Fig. 3, while in others, only the target animal or object is perturbed.

DISCUSSION

Since our method uses DWT, blocky delimiters do not appear in the adversary samples generated by the proposed method, as shown in Fig. 3. The results of Experiment 2 show that the proposed method is more accurate than AdvDrop for Black-Box attacks at all learning rates, and the amount of pixel value change is kept small. Therefore, our method has a higher misclassification rate than AdvDrop, which generates imperceptible adversarial examples.

The results of Experiment 2 show that the misclassification rate increases as the learning rate increases, but the PSNR value decreases. The proposed method can change the pixel values flexibly by multiplying the image by *M*. In addition, by limiting the frequency band of attack to high-frequency components, our method reduced the change in pixel values when perturbing the image. Fig. 3 also shows that our method produces clearer images than AdvDrop.

Figure 3 shows parts of the 2,500 perturbation matrices obtained in Experiment 2. We found that about half of the perturbation matrices have almost no black areas in the high-frequency component region and that perturbations are applied to the entire image as shown in the figure, regardless of the frequency. On the other hand, about half of the perturbation matrices were easily divided into perturbed and unperturbed areas, as shown in some of the perturbation matrices in Figs. 3. The lower the frequency component of such an image, the more perturbations are applied to the entire image. In perturbation matrices where perturbations are visible, perturbations are applied along the shape of the object or animal to be classified. In such cases, it is difficult to extract a pattern of perturbations that is common to many images. However, we found that such an image's background is simple. The DNN model tends to focus on objects correctly in such an image. So, to make the model misclassify the image, it is better to perturb only the part where the object is visible in the image. The relationship between image backgrounds and adversarial examples is an issue that we should investigate in the future.

CONCLUSION

In this paper, we proposed a new method for adversarial example generation using DWT. By multiplying *M* and the image, our method is more flexible in changing pixel values, resulting in an attack with a high misclassification rate. In addition, we could make the perturbations smaller by perturbing only the high-frequency components and by updating each element of cap M only when the value becomes small. However, since the misclassification rate remains in the 20% range, it is necessary to further improve the misclassification rate. Our approach utilizes DWT, with the only perturbation parameter being *M*. This straightforward method involves multiplying *M*. However, since basic attack methods can easily be thwarted by robust models equipped with defensive mechanisms, improving the perturbation addition method and achieving a higher misclassification rate is a future challenge.

REFERENCES

Blau, T., Ganz, R., Kawar, B., Bronstein, A., & Elad, M. (2022). Threat model-agnostic adversarial defense using diffusion models. arXiv preprint arXiv:2207.08089.

Carlini, N., & Wagner, D. (2017, May). Towards evaluating the robustness of neural networks. In *2017 IEEE symposium on security and privacy (sp)* (pp. 39–57). IEEE.

Chen, P. Y., Zhang, H., Sharma, Y., Yi, J., & Hsieh, C. J. (2017, November). Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In Proceedings of the 10th ACM workshop on artificial intelligence and security (pp. 15–26).

Dhillon, G. S., Azizzadenesheli, K., Lipton, Z. C., Bernstein, J., Kossaifi, J., Khanna, A., & Anandkumar, A. (2018). Stochastic activation pruning for robust adversarial defense. arXiv preprint arXiv:1803.01442.

Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., & Li, J. (2018). Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 9185–9193).

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. and Uszkoreit, J., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Duan, R., Chen, Y., Niu, D., Yang, Y., Qin, A. K., & He, Y. (2021). Advdrop: Adversarial attack to dnns by dropping information. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 7506–7515).

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

Guo, C., Frank, J. S., & Weinberger, K. Q. (2018). Low-frequency adversarial perturbation. arXiv preprint arXiv:1809.08758.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770–778).

Jang, Y., Zhao, T., Hong, S., & Lee, H. (2019). Adversarial defense via learning to generate diverse attacks. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 2740–2749).

Kurakin, A., Goodfellow, I., & Bengio, S. (2016). Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236.

Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., & Zhu, J. (2018). Defense against adversarial attacks using high-level representation guided denoiser. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1778–1787).

Luo, C., Lin, Q., Xie, W., Wu, B., Xie, J., & Shen, L. (2022). Frequency-driven imperceptible adversarial attack on semantic similarity. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 15315–15324).

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2017, April). Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia conference on computer and communications security (pp. 506–519).

Sriramanan, G., Addepalli, S., & Baburaj, A. (2020). Guided adversarial attack for evaluating and enhancing adversarial defenses. Advances in Neural Information Processing Systems, 33, 20297–20308.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199.

Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105–6114). PMLR.

Wallace, G. K. (1992). The JPEG still picture compression standard. IEEE transactions on consumer electronics, 38(1), xviii-xxxiv.

Xiao, C., Li, B., Zhu, J. Y., He, W., Liu, M. and Song, D., 2018. Generating adversarial examples with adversarial networks. arXiv preprint arXiv:1801.02610.