Artificial Intelligence, Social Computing and Wearable Technologies, Vol. 113, 2023, 107–114
https://doi.org/10.54941/ahfe1004184

A Method to Generate Adversarial Examples Based on Color Variety of Adjacent Pixels

Tomoki Kamegawa[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

Perturbing input images to a neural network models can intentionally change the output of the models. Such images are called adversarial examples. Many methods for generating adversarial examples have been studied. However, existing methods can add perturbations to pixels that are large enough to be perceived by the human eyes. In situations where humans see input images to the models, perturbations must be imperceptible. In our previous study, we proposed a method for generating adversarial examples using fixed perturbations. Based on its results, we assume that the perturbation perceivability varies depending on the pixels surrounding the pixel to which the perturbation is added. Because of the amount of perturbation must be adjusted for each pixel. In this study, to add more flexible perturbations to pixels, we use a ratio of a variance of the surrounding pixels to a variance over a large area is used.

Keywords: Adversarial examples, Deep learning, Image recognition, Neural network

INTRODUCTION

Deep neural networks have contributed significantly to the improvement of performance in image and speech recognition. However, deep neural networks can easily be compromised by adding small noise to the input. Adversarial examples for images are generated by perturbing the input image, and make the image classifiers incorrectly predict a label.

Sparse perturbation is one of the ways to add perturbation. It causes misclassification by perturbing some pixels in the image, rather than the whole of the pixels. Sparse perturbation can be found in real world. Raindrops on the surface can fool the image recognition system of an autonomous vehicle (Yang, 2021) (Zhai, 2020). The study of such sparse perturbations can help improve the performance of image classifiers and the robustness of models for noisy images. A well-known sparse perturbation attack is Jacobian-based Saliency Map Attack (JSMA) (Papernot, 2016) (Sethi et al., 2020), which is fast in generating adversarial examples and relatively simple in its algorithm. It is also possible to output targeted labels. However, there is a problem with the way to add perturbation to pixels. The perturbation can be perceived by the human eyes because the large perturbation was added to the pixels.

Some previous methods for generating adversarial examples do not assume that adversarial examples are checked by human eyes and allow many perturbations to be adding to a single pixel. Adversarial examples should not only cause misclassification in the image classifier system but also require less perturbation to avoid human perception of the perturbation, because adversarial example may be checked by human eyes. The large perturbations to the pixels are perceived if the input images are checked by the human eyes, or if the output of the model is judged to be inappropriate and is verified by humans. As a result, the adversarial examples are eliminated from the input, which makes the attack on the image classifier fail.

We propose methods to solve the problems in JSMA. Specifically, it adjusts the amount of perturbation by calculating the variance between the value of the pixel to be perturbed and its surrounding pixels. If a large perturbation is added to the area of an image with a large pixel value variation, the perturbation will be imperceptible. In such case, perceivability does not increase significantly with large perturbation. In contrast, if the large perturbation is added to the area of an image with small pixel value variation, the perturbation will be more perceptible. In such case, the perturbation must be small. In our previous studies, we assumed thresholds to classify perturbations into two classes, large perturbation and small perturbation. If the variance is larger than the threshold, a larger perturbation is added; if the variance is smaller than the threshold, a smaller perturbation is added, which achieved a reduction in the amount of perturbation. However, there are still rooms of improvements of the perturbation to reduce the perceptibility.

In this study, we focus on differences in the perception of perturbations depending on the variance of pixel colors. The amount of perturbation should vary from pixel to pixel, not a fixed amount. Not only the variance of the surrounding pixels but also the variance in a larger area is calculated. Even if the variance with the surrounding pixels is not very large and the variance over a wider area is large, it is not a problem to add larger perturbation. In these situations, fixed perturbation is more likely to be perceived if the variance is extremely small, even for the small perturbation. Moreover, the fixed perturbations, even if it is large, may make efficient misclassification difficult if the variance is extremely large. The smaller variance is, the smaller perturbation should be added; the larger the variance is, the larger perturbation should be added. For more flexible perturbations, we propose to use the ratio of the variance of the surrounding pixels to the variance over a large area.

RELATED WORKS

Szegedy et al. found that perturbations using gradient of prediction errors can cause image classifiers to misclassify (Szegedy et al., 2013). Based on this, methods such as Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014) and Projected Gradient Descent (PGD) (Madry et al., 2017) have been proposed. These methods add a fixed small perturbation to every pixel to increase the prediction error (Moosavi-Dezfooli et al., 2016). This allows the model to output incorrect predictions.

Adversarial examples with sparse perturbations have also been studied (Modas et al., 2019) (Wiyatno and Xu, 2018) (Combey et al., 2020) (Carlini and Wagner, 2017) (Hein and Andriushchenko, 2017). JSMA is a method that uses the gradient of the output. Perturbations are added to the pixels to obtain the desired output. One-Pixel Attack uses a differential evolution algorithm to perturb just one pixel (Su et al., 2019).

There are adversarial examples that are applicable to the real world (Kurakin et al., 2018) (Eykholt et al., 2018). A raindrop can be a perturbation (Yang, 2021) (Zhai, 2020). Adversarial examples for unmanned aerial vehicle (UAV) have been studied (Hickling et al., 2022) (Fu et al., 2022). It is also possible to cause misclassification by using glasses (Sharif et al., 2016) that cause the image recognizer to misrecognize or by attaching patches to the recognition target (Brown et al., 2017).

The authors (Kamegawa and Kimura, 2022) proposed the method to generate adversarial examples with perturbations that are difficult for the human eyes to perceive. It was necessary to reduce the perturbation per pixel. Pixels to be perturbed are selected from the output gradient information, and then fixed perturbations are added to the pixels to generate adversarial examples. Depending on the value of the variance between the perturbed pixel and the surrounding pixels, the amount of perturbation added to the pixels was chosen among two options, ϵ_1 and ϵ_2, where $\epsilon_1 > \epsilon_2$. Compared to JSMA, the previous method achieved reduced perturbation per pixel without changing attack success rates. However, there is a room for improvement in the way to adjust the amount of perturbation.

PROPOSED METHOD

The authors (Kamegawa and Kimura, 2022) proposed method to select perturbed pixels by using Equation (1) and Equation (2).

$$Q_1 = \left\{ x_i \left| \frac{\partial F_s(x)}{\partial x_i} < 0 \; and \frac{\partial F_t(x)}{\partial x_i} > max_{j \neq t} \frac{\partial F_j(x)}{\partial x_i} \right. \right\} \tag{1}$$

$$Q_2 = \left\{ x_i \left| \frac{\partial F_s(x)}{\partial x_i} > 0 \; and \frac{\partial F_t(x)}{\partial x_i} < \min_{j \neq t} \frac{\partial F_j(x)}{\partial x_i} \right. \right\} \tag{2}$$

$F_s(\mathbf{x})$ is the output of the original label and $F_t(\mathbf{x})$ is the output of the target label. i is calculated by Equation (3).

$$i = W \times m + n \, (0 \leq n < W, \; 0 \leq m < H) \tag{3}$$

W is the width of the input image. H is the height of the input image. (n, m) is a coordinate of the pixels with the origin at the upper left pixel of the image.

Since the image classifier takes the label with the largest value of $F(\mathbf{x})$ as the classification label, the output of $F_t(\mathbf{x})$ should be the largest in order to misclassify the image to the target label. In addition, making $F_s(\mathbf{x})$ smaller more effectively causes misclassification because $F_s(\mathbf{x})$ is the largest for the image before the perturbation is added. Q_1 is the selection of pixels to add positive perturbations, and Q_2 is the selection of pixels to add negative perturbations.

The variances of 3 × 3 pixels around pixel x_i obtained by Q_1 and Q_2 and a large area $N \times N$ ($N > 3$)centered on x_i is calculated. Let $s_{i_1}^2$ be the variance obtained from the variance of 3 × 3 pixels around pixel x_i and $s_{i_2}^2$ be the variance obtained from the large area $N \times N$ ($N > 3$)centered on x_i. The perturbation added to pixel x_i is denoted by δ_i. δ_i is calculated by Equation (4):

$$\delta_i = \delta \times \frac{s_{i_1}^2}{s_{i_2}^2} \tag{4}$$

where δ is a hypothetical perturbation that is set a priori. However, if $s_{i_1}^2$ and $s_{i_2}^2$ both become smaller, δ_{i} becomes larger. If $s_{i_1}^2$ and $s_{i_2}^2$ are small, the color variation with respect to the surrounding pixels is small. In this case, a small perturbation must be added.

To solve this problem, a threshold of variance is introduced. For every pixel, $N \times N$ variances are calculated and the median of these is taken as the threshold, T. The variance of all pixels is also calculated as the s_{all}^2. The amount of the perturbation is determined by Equation (5).

$$\delta_{\mathrm{i}} = \begin{cases} \delta \times \frac{s_{i_1}^2}{s_{i_2}^2} \ (s_{i_2}^2 \geq T) \\ \delta \times \frac{s_{i_1}^2}{s_{\mathrm{all}}^2} \ (s_{i_2}^2 < T) \end{cases} \tag{5}$$

The perturbation is added to a pixel according to Equation (6).

$$\widetilde{x}_i = \begin{cases} x_i + \delta_{\mathrm{i}} \ (x_i \in Q_1) \\ x_i - \delta_i \ (x_i \in Q_2) \end{cases} \tag{6}$$

The positive perturbations are added to pixels in Q_1 and the negative perturbations are added to pixels in Q_2.

For Q_1, we perturb are perturbed the pixel with the highest value of $\frac{\partial F_t(\mathbf{x})}{\partial x_i} - \frac{\partial F_s(\mathbf{x})}{\partial x_i}$, and for Q_2, we perturb the pixel with the lowest value of $\frac{\partial F_t(\mathbf{x})}{\partial x_i} - \frac{\partial F_s(\mathbf{x})}{\partial x_i}$. Then Q_1 and Q_2 are calculated again using $\widetilde{\mathbf{x}}$. This is repeated until $F_t(\widetilde{\mathbf{x}})$ takes the greater than any $F_{j \neq t}(\widetilde{\mathbf{x}})$.

The flow of the adversarial example generation is shown in Figure (1).

EXPERIMENTS

Experiments were conducted to measure the performance of the proposed method compared to JSMA and our previous method. To generate perturbations difficult for the human eyes to perceive, the perturbations per pixel must be small. In order to confirm this, we used CIFAR-10 for the image dataset and measured the misclassification success rate, the number of perturbed pixels, the sum of perturbations, and perturbations per pixel. Adversarial examples were generated using 100 images with the label "ship" and the target label "airplane". Pixel values are assumed to be normalized between 0 and 1. The hypothetical perturbation is set to $\delta = 80/255$.

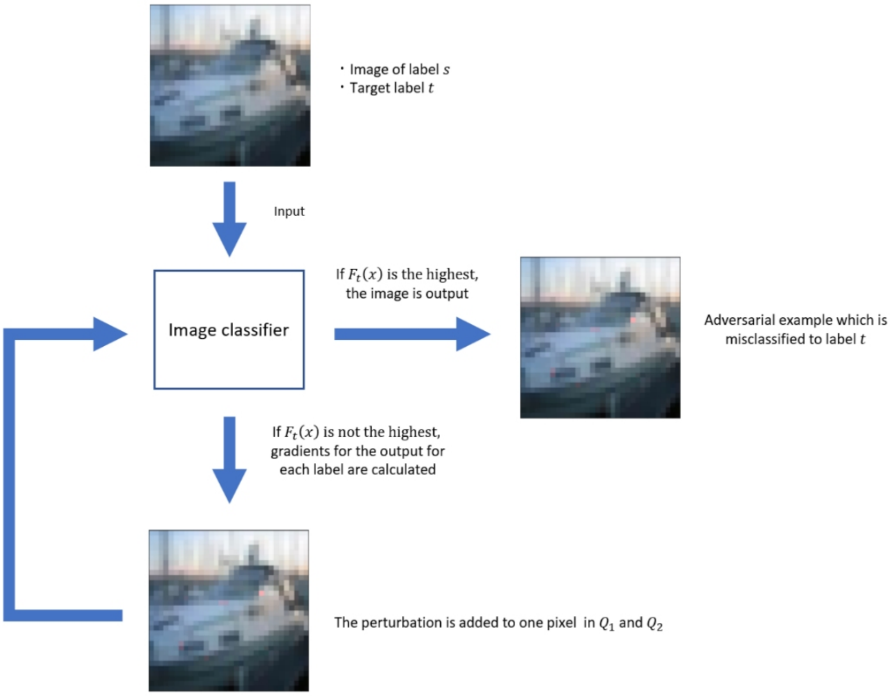

· Image of label s
· Target label t
Input
Image classifier
If $F_t(x)$ is the highest, the image is output
Adversarial example which is misclassified to label t
If $F_t(x)$ is not the highest, gradients for the output for each label are calculated
The perturbation is added to one pixel in Q_1 and Q_2

Figure 1: The flow of the adversarial example generation.

Figure 2 shows adversarial examples for the three methods and Table 1 shows the results of the experiment. "The misclassification success rate" represents the percentage of the generated adversarial samples that were successfully misclassified to the target label, "The number of pixel value changes per image" represents the average of pixels that changed value per successfully misclassified adversarial sample, "The sum of perturbations per image" represents the total number of perturbations per successfully misclassified and "The perturbation per pixel" represents the average of the sum of the perturbations per successfully misclassified adversarial example.

The proposed method had the misclassification success rate comparable to that of JSMA and our previous method. The sum of perturbation per image and the perturbation per pixel were the smallest for the proposed method.

Table 1. Result of comparison experiments.

	Proposed method	Our previous method	JSMA
The misclassification success rate	82%	82%	83%
The number of perturbed pixels	47.05	38.27	32.49
The sum of perturbation per image	4.75	5.27	18.55
The perturbation per pixel	0.13	0.14	0.62

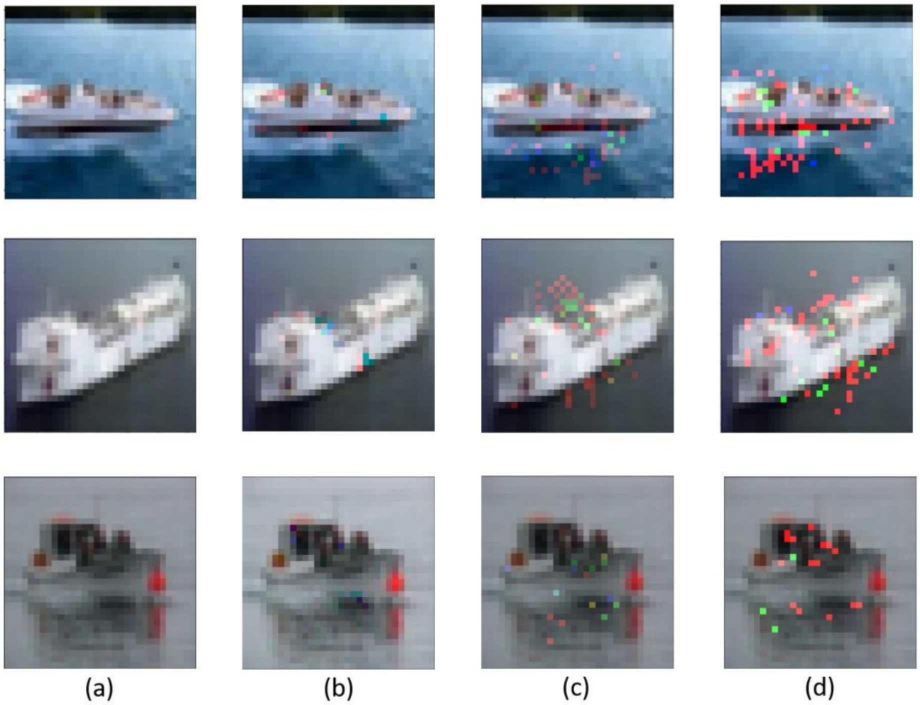
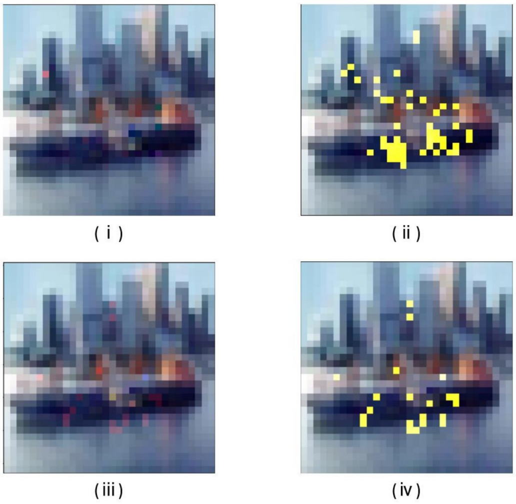


Figure 2: Original image (a) and adversarial examples generated by proposed method (b), our previous method (c) and JSMA (d).

DISCUSSION

The sum of perturbation per image and perturbation per pixel for the proposed method is the lowest among the three methods. It is evident that variable perturbation avoids an excess of perturbation when compared to the other two methods. If the perturbation per pixel is small, then more pixels need to be modified. Table 1 shows that our proposed method has the highest number of perturbed pixels and the smallest perturbation per pixel. This indicates that the alteration to a pixel should be adaptable enough to vary its value depending on the pixel, rather than adding a set value.

Figure 3 shows the adversarial examples and the visualization of their perturbations. The adversarial example with proposed method is Image (i) and the perturbations of it are visualized in image (ii). The adversarial example with our previous method is Image (iii) and the perturbations of it are visualized in Image (iv). We can see perturbations to flat color areas of the image by the proposed method are less perceptible. In both methods, perturbations are added to the black part of the ship's hull. In our previous method, fixed perturbations are added, which makes perturbations in black areas more perceptible. In contrast, the proposed method introduces a small perturbation to the pixels in the black areas where the variance is smaller. This makes perturbations in these areas less perceptible. It shows that adjusting perturbations using the ratio of variances can be effective in reducing the perceivability of perturbations.



(i) (ii) (iii) (iv)

Figure 3: Adversarial examples from the proposed and previous method and visualization of their perturbations.

CONCLUSIONS

In this study, we generated adversarial examples focused on differences in the perception of perturbations depending on the variance of pixel colors. The reduction in the perturbation per pixel by the proposed method made perturbation less perceptible than our previous methods and JSMA.

In future study, it is essential to explore potential enhancements to perturbations that consider the cognitive characteristics of human in relation to color, as well as the required security measures to safeguard against such attacks.

REFERENCES

Brown, T. B., Mane, D., Roy, A., Abadi, M., and Gilmer, J. (2017). Adversarial patch. arXiv preprint arXiv:1712.09665.

Carlini, N., and Wagner, D. (may 2017). Towards evaluating the robustness of neural networks. in 2017 IEEE Symposium on Security and Privacy (SP). Los Alamitos, CA, USA: IEEE Computer Society, pp. 39–57.

Combey, T., Loison, A., Faucher, M., and Hajri, H. (2020). Probabilistic jacobian-based saliency maps attacks. Machine learning and knowledge extraction, 2(4), pp. 558–578.

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C. et al. (2018). Robust physical-world attacks on deep learning visual classification. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Fu, C., Li, S., Yuan, X., Ye, J., Cao, Z., and Ding, F. (2022, May). Ad 2 attack: Adaptive adversarial attack on real-time uav tracking. In 2022 International Conference on Robotics and Automation (ICRA), pp. 5893–5899.

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

Hein, M., and Andriushchenko, M. (2017). Formal guarantees on the robustness of a classifier against adversarial manipulation. Advances in neural information processing systems, 30.

Hickling, T., Aouf, N., and Spencer, P. (2022). Robust adversarial attacks detection based on explainable deep reinforcement learning for uav guidance and planning.arXiv preprint arXiv:2206.02670.

Kamegawa, T., and Kimura, M. (2022, November). A Method for Adversarial Example Generation by Perturbing Selected Pixels. In 2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 1109–1114.

Kurakin, A., Goodfellow, I. J., and Bengio, S. (2018). Adversarial examples in the physical world. In Artificial intelligence safety and security. Chapman and Hall/CRC, pp. 99–112.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks.arXiv preprint arXiv:1706.06083.

Modas, A. Moosavi-Dezfooli, S. and Frossard, P. (jun 2019). Sparsefool: A few pixels make a big difference. in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Los Alamitos, CA, USA: IEEE Computer Society, pp. 9079–9088.

Moosavi-Dezfooli, S. M., Fawzi, A., and Frossard, P. (2016). Deepfool: a simple and accurate method to fool deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2574–2582.

Papernot, N. McDaniel, P and Jha, S. Fredrikson, M. Celik, Z. B. and Swami, A. (2016). The limitations of deep learning in adversarial settings. in 2016 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 372–387.

Sethi, K., Sai Rupesh, E., Kumar, R., Bera, P., and Venu Madhav, Y. (2020). A context-aware robust intrusion detection system: a reinforcement learning-based approach. International Journal of Information Security, 19, pp. 657–678.

Sharif, M., Bhagavatula, S., Bauer, L., and Reiter, M. K. (2016, October). Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In Proceedings of the 2016 acm sigsac conference on computer and communications security, pp. 1528–1540.

Su, J., Vargas, D. V., and Sakurai, K. (2019). One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation, vol. 23, no. 5, pp. 828–841.

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

Wiyatno, R., and Xu, A. (2018). Maximal jacobian-based saliency map attack.arXiv preprint arXiv:1808.07945.

Yang, H.-D. (2021). Restoring raindrops using attentive generative adversarial networks. Applied Sciences, vol. 11, p. 7034.

Zhai, L., Juefei-Xu, F., Guo, Q., Xie, X., Ma, L., Feng, W. et al. (2020). Adversarial rain attack and defensive deraining for DNN perception. arXiv preprint arXiv:2009.09205.