Journal of information and
communication convergence engineering
http://jicce.org

J. Inf. Commun. Converg. Eng. 16(3): 148-152, Sep. 2018 **Regular paper**

Ensemble of Degraded Artificial Intelligence Modules Against Adversarial Attacks on Neural Networks

Richard Evan Sutanto and Sukho Lee*, ***Member*, *KIICE***

Department of Computer Engineering, Dongseo University, Busan 47011, Korea

Abstract

Adversarial attacks on artificial intelligence (AI) systems use adversarial examples to achieve the attack objective. Adversarial examples consist of slightly changed test data, causing AI systems to make false decisions on these examples. When used as a tool for attacking AI systems, this can lead to disastrous results. In this paper, we propose an ensemble of degraded convolutional neural network (CNN) modules, which is more robust to adversarial attacks than conventional CNNs. Each module is trained on degraded images. During testing, images are degraded using various degradation methods, and a final decision is made utilizing a one-hot encoding vector that is obtained by summing up all the output vectors of the modules. Experimental results show that the proposed ensemble network is more resilient to adversarial attacks than conventional networks, while the accuracies for normal images are similar.

Index Terms: Adversarial attack, Artificial Intelligence, Image classification

I. INTRODUCTION

With the recent advance in deep learning methods, artificial intelligence (AI) systems have become widely used in various fields, such as autonomous driving, banking, and smart homes, to name a few. However, as AI systems are becoming increasingly prevailing, the cost associated with their failure or improper operation is also increasing.

Recently, it has been shown that AI systems can fail to correctly recognize objects in even slightly degraded images [1]. Such images—with the level of degradation that is not captured by the human eye but is sufficient to induce the AI systems' failure—are called adversarial examples. Problems arise when such adversarial examples are intentionally used for misleading AI systems in adversarial attacks.

In a typical adversarial attack, the attacker first generates an adversarial example to the attacked AI system. If the model and its parameters are known to the attacker, it becomes easy to generate adversarial examples. For example, in [1], it was shown that a simple calculation of the gradient of the loss function with respect to the input can be used for generating efficient adversarial examples. Such attacks, which use gradient information about the attacked network, are called gradient-based adversary attacks. In [2], an L_2 attack on the logit value of the neural network was proposed, instead of a direct attack that alters the network's output. This attack is known as one of the strongest adversarial attack methods. In [3], an algorithm for generating adversarial examples was proposed, based on the adversarial saliency map. The study in [4] concluded that adversarial examples may not be critically dangerous for automatic driving applications since adversarial examples are only effective from a specific point of view. However, in [5], it was shown that adversarial examples can indeed pose a threat to automatic driving applications, because adversarial examples stable with respect to affine transformations could be generated.

Received 30 March 2018, **Revised** 19 June 2018, **Accepted** 20 June 2018
***Corresponding Author** Sukho Lee (E-mail: petrasuk@gmail.com, Tel: +82-51-320-1744)
Department of Computer Engineering, Dongseo University, 47, Jurye-ro, Sasang-gu, Busan 47011, Korea.

Open Access **https://doi.org/10.6109/jicce.2018.16.3.148** print ISSN: 2234-8255 online ISSN: 2234-8883

When the parameters of the attacked neural network are unknown, the gradient cannot be directly computed. However, as has been shown in [6], even in such black-box cases, it is possible to generate adversarial examples using differential evolution or model extraction methods. Furthermore, in [6] it was also shown that a single pixel attack is sufficient for deceiving an AI system. The authors in [7] introduced a measure of the robustness of an AI system by suggesting how to measure the minimal length of a vector that causes the attacked AI system to make false decisions. At the same time, the authors in [8] and [9] suggested methods for detecting adversarial attacks before they affect the attacked AI systems.

However, to the best of our knowledge, no AI system has been designed that would be resilient to all possible adversarial attacks, although it is hoped that CapsuleNets [10] can deal with the problem to a certain extent. The inability to come up with an all-encompassing AI system has been attributed to the fact that there exist infinitely many adversarial examples and adversarial example-generating algorithms.

Here, we claim that the vulnerability of AI systems to adversarial examples may stem from the fact that AI systems are sensitive to small details in the test data. Therefore, we propose a degraded AI system, which is trained on degraded training data and works on degraded test data. However, by working with degraded data, the accuracy of the AI system is likely to decrease. Therefore, we use an ensemble of degraded AI modules for which the final decision is made based on the vote across all degraded AI modules.

A similar approach using an ensemble of several modules for defending against adversarial attacks has been proposed in [11]. The main difference between the work in [11] and our work is that in [11] the authors perturbed the data to a more significant extent by adding noise to each module, while we degraded our data by eliminating high-frequency components from our images. We explain the difference between the two models in more detail in Section III.

Experimental results show that the proposed system is more robust to adversarial examples while the accuracy is similar to that of a normal AI system.

II. ADVERSARIAL ATTACKS ON NEURAL NETWORKS

Fig. 1 illustrates the difference between a hacking attack and an adversarial attack. In hacking, an attacker tries to attack an AI system via an abnormal route which the attacker has explored, whereas in an adversarial attack, an adversarial example is shown to the AI system via a normal route, i.e., through the sensors of the AI system.

Fig. 2 shows an adversarial example adopted from [1], where the left image is a normal image while the right image is an adversarial example obtained by adding a small amount of designed noise to the left image. Even though the left and right images look the same to the human eye, the neural network assigns a confidence of 57.7% for the left image to be a panda, while it assigns a confidence of 99.3% for the right image to be a gibbon.

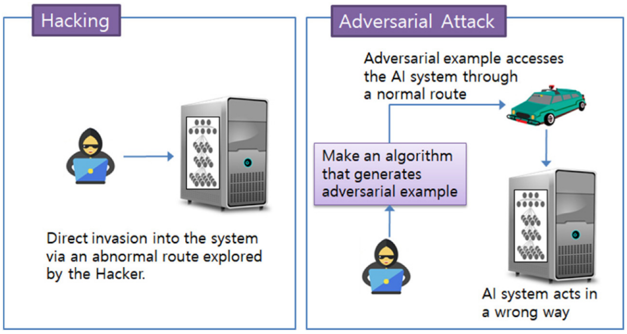
Hacking
Direct invasion into the system via an abnormal route explored by the Hacker.
Adversarial Attack
Adversarial example accesses the AI system through a normal route
Make an algorithm that generates adversarial example
AI system acts in a wrong way

Fig. 1. Comparison between hacking and adversarial attack.

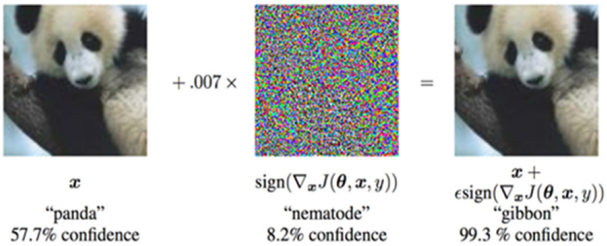
+ .007 ×
=
$\boldsymbol{x}$
"panda"
57.7% confidence
$\mathrm{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} + \epsilon\mathrm{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Fig. 2. Example of an adversarial example. Adapted from Goodfellow *et al.*, Explaining and harnessing adversarial examples, 2014 [1] with permission.

The small amount of noise was not random noise, but was designed carefully to deceive the neural network as desired. For example, the adversarial example in Fig. 2 was generated by just taking the sign of the gradient vector, which provides information about the direction in which the error increases, and multiplying this vector by a small number:

$$\bar{\mathbf{x}} = \mathbf{x} + \varepsilon sign\,(\nabla_{\mathbf{x}} J(\mathbf{x}))\,, \tag{1}$$

where $\bar{\mathbf{x}}$ is the generated adversarial example, $\mathbf{x}$ is the original image, and $\nabla_{\mathbf{x}} J(\mathbf{x})$ denotes the gradient of the cost function with respect to the input image $\mathbf{x}$.

As can be seen from (1), the gradient information is the most important information for generating a successful adversarial example. This is also true for other adversarial example-generating algorithms, although some methods do not use the gradient information, for example the method proposed in [6].

III. PROPOSED ENSEMBLE SYSTEM ROBUST TO ADVERSARIAL ATTACKS

Neural networks may be vulnerable to adversarial examples because these networks are trained to consider even fine details in the presented images. Therefore, to increase the network's resilience to adversarial examples, one defense method amounts to degrading the training set images and train the network on degraded images. However, by doing so, the performance of the network is likely to decrease. Furthermore, if the presented training set images are degraded using only one method, some adversarial example-generating algorithms can still be very effective on this network. Therefore, in this paper we propose an ensemble of convolutional neural network (CNN) models, where each model is trained on degraded images using different methods. We are aware of a similar approach, which uses images with different amounts of noise to reduce the gradient effect in adversarial examples [11]. Adding of extra noise to the network is the same as adding high-frequency components to images. The main idea in [11] is that if the adversarial gradient is hidden in noise (using many high-frequency components), the attacked network will not be able to discriminate the adversarial gradient from pure noise; therefore, the network will be more robust to the adversarial gradient. For comparison, we considered several image degradation transformations, such as several types of resizing and blurring, as well as image de-colorization. This degrading was the same as subtracting high-frequency components from the presented images, which to some extent also subtracts the adversarial gradient.

By doing so, the network was trained to perceive objects by not considering too strongly the image's high-frequency components (as is the case in conventional CNNs).

Each of these degrading transformations made images more resistant to adversarial examples. However, because the accuracy of individual degraded models is relatively low, we make a decision based on adding all of the resulting output vectors of all the models.

Fig. 3 shows the overall diagram of the proposed system. The different CNN models are trained on different degraded images. Then, in test time, a test image is shown to all of the component CNN models, and a decision is made based on the votes of the component CNNs.

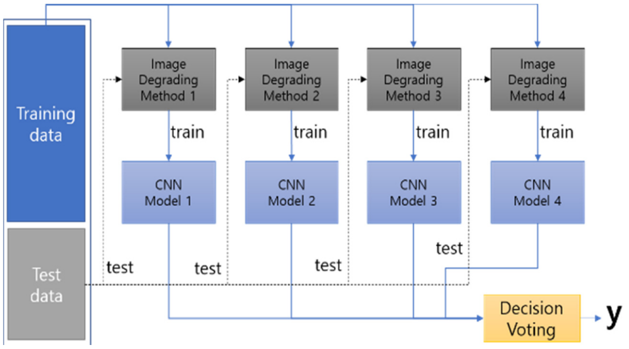

Fig. 3. Schematic of the proposed model.

Here, we mainly use degradation of spatial information, as it has been shown that an ensemble of feature-squeezing networks is not quite efficient [12]. The key components of the model are explained below

A. Degradation of Color Information

For every CNN model, we discard the color information. In other words, we calculate the luminance values for the RGB channels, and then copy the luminance values into the RGB channels. This discards the gradient across the channels. The effect is small, because the spatial information of the gradient is kept intact, but such de-colorization also simplifies the degradation of spatial information.

B. Degradation of Spatial Information

We use two different methods for the degradation of spatial information. In the first approach, the image is filtered using a Gaussian kernel:

$$\hat{\mathbf{x}} = \mathbf{x} * \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\mathbf{x}-\boldsymbol{\mu})^2}{2\sigma^2}}, \qquad (2)$$

where $\hat{\mathbf{x}}$ denotes the convolved image, $\mathbf{x}$ denotes the original image, * denotes the convolution operator, and μ and σ denote the mean and the standard deviation of the training images, respectively.

As is known from the scale-space theory, such filtering using a Gaussian kernel smooths out the fine-scale details in the images, and eliminates adversarial gradients.

The second method amounts to resizing the image several times. We assume that the adversarial gradient vector has mainly high-frequency components, as shown in the shaded part in Fig. 4(a). The down-sampling process widens the frequency spectrum of the input vector and the adversarial gradient vector, so that aliasing occurs in the high-frequency part (Fig. 4(b)). This mixes the frequency components of the adversarial gradient vector, which weakens the adversarial attacking effect. Moreover, when a low-pass filter is applied to the down-sampled image, before up-sampling, the high-frequency components are eliminated as shown in Fig. 4(c), and most of the adversarial gradient vector is eliminated from the input vector. However, as the high-frequency components of the original image are also lost, the recognition accuracy decreases, and should be compensated for by the ensemble decision summing.

C. Decision Summing

After the degraded CNN models output their prediction vectors, we add up the prediction vectors to yield the ensemble

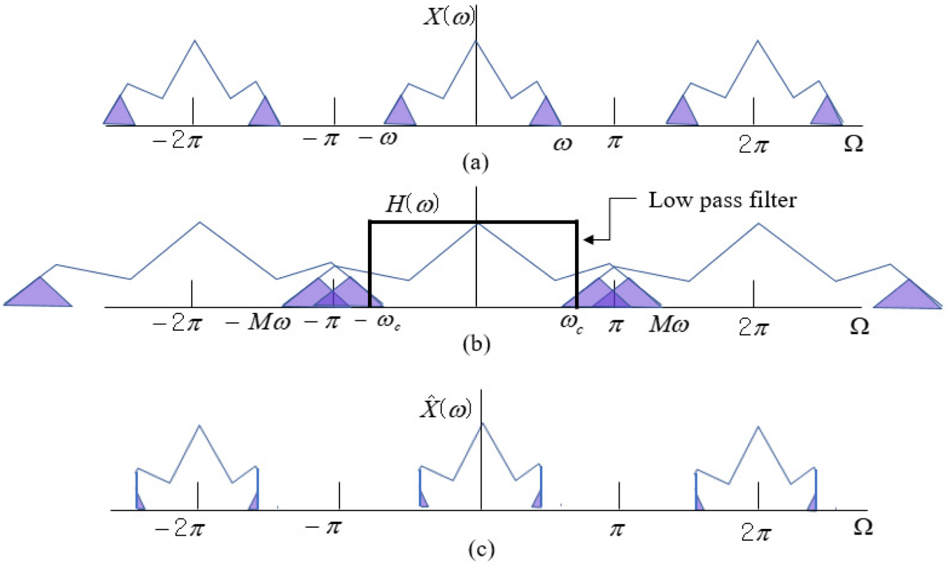

Fig. 4. Analysis of the elimination of the gradient vector in the adversarial example in the Fourier domain. (a) Spectral response to the adversarial example, where the shaded parts show the region in which most of the spectral components of the adversarial gradient reside. (b) Spectral domain after down-sampling of the adversarial example. (c) Spectral domain after low-pass filtering and up-sampling of the adversarial example.

decision of these degraded models. The final ensemble vector is then compared with the true label one-hot encoding vector:

$$compare(truelabel, arg\ \ max(\sum_{k=1}^{N} \mathbf{y}_k)), \tag{3}$$

where $\mathbf{y}_k$ denotes the output vector of the *k*-th degradation model. As will be shown when we describe our experimental results below, the output vectors of the degradation models do not yield higher accuracy on the test data. This is owing to the fact that, although the adversarial gradient information has been eliminated, high-frequency components have been also removed from the signal, which diminishes the network's precision. However, the ensemble vector accumulates the prediction values of all the degraded output vectors, which strengthens the correctly predicted value, while false predictions owing to the adversarial gradient are averaged out, because different models were trained by different methods (and thus give different false predictions). We further demonstrate this point in what follows.

IV. EXPERIMENTAL RESULTS

We used the CIFAR-10 dataset to test the effect of an adversarial attack generated by the fast gradient sign method (FGSM). In the CIFAR-10 dataset, the data are divided into two sets; 50000 data samples are used for training and 10000 data samples are used for testing. The images' size in these experiments was 32 × 32, and the number of classes was 10.

We used four different degraded modules. The first module down-sampled the adversarial images to the size of 8 × 8, and then up-sampled them back to 32 × 32. The second and the third modules resized the adversarial images into sizes of 16 × 8 and 8 × 16, respectively, and then up-sampled them to the normal size. These different resizing transformations distorting the adversarial examples with different geometries; as a result, the low-pass filter eliminated different high-frequency components in each module. The last module modified the adversarial images into grayscale and convolved them with a Gaussian kernel, where we used a kernel with a standard deviation of 0.1. Table 1 shows the accuracy achieved by each module, and also the accuracy achieved without using any defense method.

As can be seen, individual degrading processes do not improve the system's accuracy with respect to adversarial examples, but achieve lower accuracy when no degrading methods are used. We believe that this is true because the degrading process eliminates not only the adversarial gradient but also the important frequency components from the image, which makes the prediction somewhat inaccurate.

Table 1. Experimental result for 4 modules

	Accuracy (%) for adversarial example	
	Epoch = 10	**Epoch = 20**
Without degradation	62.12	63.33
Module 1	55.25	55.58
Module 2	58.40	56.93
Module 3	59.43	57.61
Module 4	55.67	57.43
Ensemble accuracy	67.18	71.02

However, the adversarial gradient is also degraded to some extent. Therefore, if we take the ensemble result for all the output vectors, the effect of the adversarial gradient is reduced while the accuracy increases as the prediction value for the true label increases, as can be seen from Table 1. Compared with the original CNN, the accuracy has increased by 5%-10%.

V. CONCLUSION

In this paper, we proposed an ensemble network that can resist adversarial attacks and can yield high-accuracy performance even in the presence of such attacks. The degradation transformations distorted the adversarial gradient, which in turn eliminated the effect of the adversarial attack, while using the ensemble increased the system's prediction accuracy by accumulating the value associated with the correct prediction. In future work, we will extend the ensemble system by including more diverse degrading modules, and will further test the system on a diverse set of adversarial attacks using adversarial examples generated by different methods.

ACKNOWLEDGMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (No. NRF-2016R1D1A3B03931875) and by the Institute for Information and Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2018-0-00245, Development of prevention technology against AI dysfunction induced by deception attack).

REFERENCES

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014 [Internet], Available: https://arxiv.org/abs/1412.6572.

[2] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proceedings of IEEE Symposium on Security and Privacy*, San Jose, CA, pp. 39-57, 2017. DOI: 10.1109/SP.2017.49.

[3] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proceedings of IEEE European Symposium on Security and Privacy*, Saarbrucken, Germany, pp. 372-387, 2016. DOI: 10.1109/EuroSP.2016.36.

[4] J. Lu, H. Sibai, E. Fabry, and D. Forsyth, "No need to worry about adversarial examples in object detection in autonomous vehicles," 2017 [Internet], Available: https://arxiv.org/abs/1707.03501.

[5] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," 2018 [Internet], Available: https://arxiv.org/abs/1707.07397.

[6] J. Su, D. V. Vargas, and S. Kouichi, "One pixel attack for fooling deep neural networks," 2018 [Internet], Available: https://arxiv.org/abs/1710.08864.

[7] S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: a simple and accurate method to fool deep neural networks," 2016 [Internet], Available: https://arxiv.org/abs/1511.04599.

[8] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner. "Detecting adversarial samples from artifacts," 2017 [Internet], Available: https://arxiv.org/abs/1703.00410.

[9] X. Li and F. Li, "Adversarial examples detection in deep networks with convolutional filter statistics," 2017 [Internet], Available: https://arxiv.org/abs/1612.07767.

[10] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," 2017 [Internet], Available: https://arxiv.org/abs/1710.09829.

[11] T. Strauss, M. Hanselmann, A. Junginger, and H. Ulmer, "Ensemble methods as a defense to adversarial perturbations against deep neural networks," 2018 [Internet], Available: https://arxiv.org/abs/1709.03423.

[12] W. He, J. Wei, X. Chen, N. Carlini, and D. Song, "Adversarial example defenses: ensembles of weak defenses are not strong," 2017 [Internet], Available: https://arxiv.org/abs/1706.04701.

Richard Evan Sutanto
received the M.S. degree in computer engineering from Dongseo University, Busan, Korea, in 2016, where he is now a Ph.D. candidate. His current research interests include image and video processing, human-computer interaction, deep learning, and image representation based on sparse approaches.

Sukho Lee
received the B.S., M.S., and Ph.D. degrees in electronics engineering from Yonsei University, Seoul, Korea, in 1993, 1998, and 2003, respectively. He was a Researcher with the Impedance Imaging Research Center from 2003 to 2006 and was an Assistant Professor with Yonsei University from 2006 to 2008. He has been with the Division of Computer Engineering, Dongseo University, Busan, Korea, since 2008, where he is currently a Professor. His current research interests include deep learning, image and video filtering based on PDEs, and computer vision.