Hindawi
Security and Communication Networks
Volume 2022, Article ID 3440123, 11 pages
https://doi.org/10.1155/2022/3440123

Research Article

AEGuard: Image Feature-Based Independent Adversarial Example Detection Model

Mihui Kim and Junhyeok Yun

School of Computer Engineering & Applied Mathematics, Computer System Institute, Hankyong National University, Jungang-ro, Anseong-si, Gyeonggi-do 17579, Republic of Korea

Correspondence should be addressed to Mihui Kim; mhkim@hknu.ac.kr

Received 30 July 2022; Revised 29 October 2022; Accepted 14 November 2022; Published 17 December 2022

Academic Editor: Shah Nazir

With the rapid development of image processing technology, image recognition systems based on massive image data are being developed and deployed. The wrong decision regarding an image recognition system for security-sensitive systems can cause serious problems such as personal accidents and property damage. Furthermore, adversarial attacks, which are security attacks that cause malfunctions in image recognition systems by inserting adversarial noise, have emerged and evolved. Several studies have been conducted to prevent adversarial attacks. However, existing mechanisms have low classification accuracy and low detection accuracy for adversarial examples with small adversarial noise. This paper proposes an adversarial example detection mechanism based on image feature extraction and a deep neural network (DNN) model. The proposed system achieves versatility and independence by detecting adversarial examples based on image features, such as edge noise and discrete cosine transform (DCT) bias, which adversarial examples have in common. The proposed system shows relatively higher detection accuracy than existing mechanisms for various types and amounts of adversarial noise and different sharpness of adversarial examples because the proposed system detects them depending on the characteristics of each type of adversarial example.

1. Introduction

With the rapid development of image processing technology and algorithms, many image recognition systems based on massive image data are being developed and deployed [1]. Security-sensitive systems, such as autonomous vehicles [2] and unattended surveillance [3], have also adopted image recognition systems. These systems require high reliability because the wrong decision in an image recognition system can cause serious problems, such as personal accidents and property damage. Thus, a proper method to avoid a security attack that forces the system to make an incorrect decision is necessary [4]. Image recognition systems can be categorized into classification, object detection, and segmentation models. The classification model classifies the objects that appear in the image. The object detection model detects one or more objects from an image. The segmentation model detects one or more objects and extracts the contours of those objects. An adversarial attack is a security attack that compels the classification model to make wrong decisions by inserting adversarial noise that destroys the original object's characteristics. Recent adversarial attack methods have been significantly improved to perform attacks with only a tiny, invisible level of adversarial noise [5]. Therefore, the system administrator cannot capture adversarial examples by monitoring the input data of the system. The lack of adequate defending mechanisms against adversarial attacks can cause significant problems, such accidents because of image sensor-based autonomous driving [6], and property damage because of the disabled intrusion detection systems [7]. Thus, a proper automated countermeasure to block adversarial attacks is necessary.

Due to the characteristics of current adversarial attacking methods that are hard to recognize with human eyes, research has been conducted to automate the detection of adversarial examples or make the classification model

robust to adversarial examples [8]. Madry et al. [9] proposed an adversarial noise reduction mechanism that neutralizes adversarial noise by adopting image filters, such as Gaussian filters. However, adopting image filters to neutralize adversarial noise can also destroy the characteristics of the original object. Thus, the classification accuracy decreases. Shaham et al. [10] proposed a robust model mechanism that makes the classification model resistant to adversarial examples by adding adversarial examples to the training dataset of the classification model. The robust model mechanism has a higher adversarial example detection accuracy and a smaller decrease in classification accuracy than the adversarial noise reduction mechanism. However, the robust model mechanism requires the generation of many adversarial example data sets for each object class that the model can classify. Thus, the robust model mechanism requires more processing overhead during the model training stage. Grosse et al. [11] proposed a statistics-based adversarial example detection mechanism to avoid a decrease in classification accuracy, a common problem with other mechanisms. The statistics-based adversarial example detection mechanism detects adversarial examples by analyzing the difference in the distribution between normal images and adversarial examples. However, the statistics-based adversarial example detection model has a lower detection accuracy than the other models. Thus, avoiding adversarial attacks using only a statistics-based adversarial example mechanism is not appropriate. In summary, a countermeasure for current adversarial attacks should achieve high detection accuracy and minimum classification accuracy loss.

We propose a detection system that can detect adversarial examples based on the image features that adversarial examples have in common, such as edge noise and discrete cosine transform (DCT) bias, regardless of which object class the classification model can classify. The adversarial example includes adversarial noise, especially around edges, to destroy the edge characteristics of the original objects. Thus, the edge components extracted from adversarial examples include noise around the contours. The degree of edge noise can vary depending on other image features, such as image entropy, color composition, and many other features. Therefore, we propose a detection system that includes an image feature quantification module and a detection model trained using the quantified image features. The proposed system is versatile because it does not require any modification of the classification model, even if the object classes that the classification model can classify are changed. Furthermore, the proposed system can solve the problem of decrease in classification accuracy because the classification models before and after applying the system are precisely the same.

The remainder of this paper is organized as follows: Section 2 introduces related works, such as those related to the image classification model, adversarial attacks, and avoidance of adversarial attacks. Section 3 explains the theoretical background, training data, and training method of the proposed system. Section 4 shows the feasibility of the proposed system by evaluating its performance and comparing it with other mechanisms, and Section 5 presents the conclusions.

2. Related Work

2.1. Image Classification Model. The convolutional neural network (CNN) model comprises one input layer and an output layer, one or more convolutional layers, max-pooling layers, and fully connected layers (see Figure 1(a)). The input layer imports the image and conducts preprocessing, such as image resizing and color channel conversion. The convolution layer extracts the edge features of the object using a convolution filter (see Figure 1(b)) and saves weight for each convolution filter. The fully connected layer flattens the convolutional and max-pooling layers' output values and uses them as input values. The fully connected layer trains the feature characteristics of each object using input values. The output layer makes the final decision of the input image that includes the object. In general, the softmax function is used as the activation function of the output layer [12].

The adversarial noise reduction mechanism requires the modification of the input layer. The classification model can neutralize the adversarial noise of the input image by adding an image filter to the input layer. However, the modification of the input layer can destroy the original object's characteristics and cause a decrease in the classification accuracy. The robust model mechanism requires modifications throughout the classification model. Adding adversarial examples to the training dataset changes the weight of the layers. Thus, the robust model mechanism requires retraining the model when the object class that the classification model can classify has changed. The proposed system detects adversarial examples using the image features of the input image. Thus, the proposed system is versatile and does not require any modification of the classification model. This characteristic means that the proposed system does not modify any of the layers constituting the image classification system. Thus, the proposed system can avoid a decrease in classification accuracy and does not need to retrain the classification model under any circumstances.

2.2. Adversarial Attacks. An adversarial attack is a security attack that causes the classification model to make an incorrect decision by inserting adversarial noise that destroys the original object's characteristics [9]. Adversarial attacks can be classified into white-box and black-box attacks. A white-box attack [13] is an adversarial attack in which an attacker attacks the classification model using the training environment information, such as the training dataset and hyper-parameters. Using a white-box attack, attackers can attack the model with minimal modification of the parts that significantly influence the decision of the classification model. Thus, adversarial examples generated for white-box attacks are difficult to capture by monitoring the input images. However, in the real world, white-box attacks do not frequently occur because the training environment information is generally not disclosed. A black-box attack [14] is

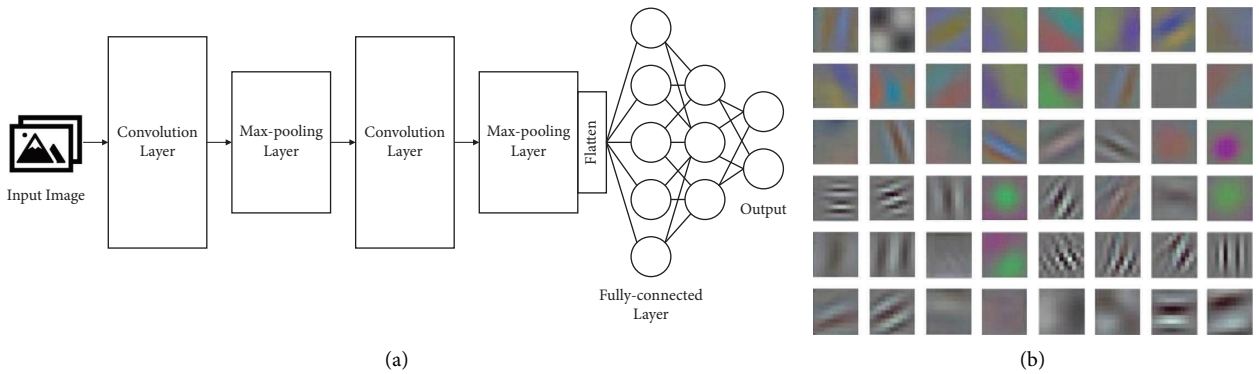


Figure 1: Convolutional neural network (CNN) model structure. (a) CNN model layer composition. (b) Convolution filter visualization.

an adversarial attack that attacks the model using input data and the classification model's decision without training the environment information. The Jacobian saliency map attack (JSMA) [15] and fast gradient sign attack (FGSM) [16] are black-box adversarial attacks. The JSMA attack inserts adversarial noise gradually and determines the point at which the classification model's decision changes. In this way, the attacker can perform an attack with minimal adversarial noise using a black-box attack.

Because of the characteristics of current black-box attack methods, the system administrator cannot capture the adversarial example by monitoring the input images. Figure 2(a) shows an image classified as "fox" based on the classification model. The difference between Figures 2(a) and 2(b) is challenging to see. However, the classification model classifies Figure 2(b) as "cow," which is entirely different from "fox." Figure 2(c) shows the difference between the original image (Figure 2(a)) and the adversarial example (Figure 2(b)). Figure 2(b) was generated by adding adversarial noise (Figure 2(c)) multiplied by 0.01.

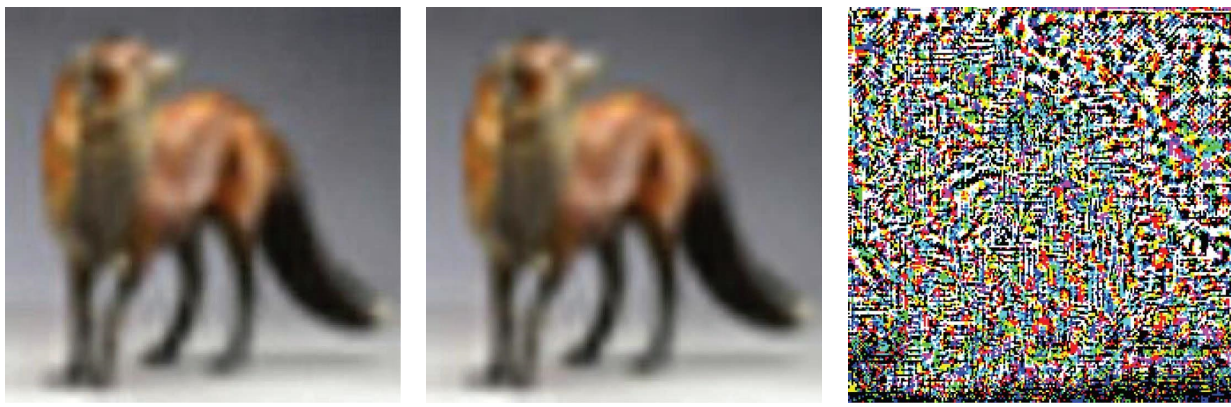
2.3. Existing Works. Adversarial attack block mechanisms can be classified into dependent mechanisms affected by the classification model and independent mechanisms not affected by the classification model (see Table 1). The robust model and the statistics-based detection model are dependent mechanisms. The adversarial noise reduction and the proposed system are independent mechanisms.

Madry et al. [9] proposed an adversarial noise reduction mechanism that neutralizes adversarial noise to compel a classification model to make the correct decision for adversarial examples. The adversarial noise reduction mechanism is versatile because the image filter can be used for all input images and classification models. However, the adversarial noise reduction mechanism has lower detection accuracy than others and has a problem of decrease in classification accuracy because the image filter can destroy the original object's characteristics. Zheng et al. [18] also proposed an adversarial attack block mechanism that neutralizes adversarial noise with input image modification. Yuan et al. used a randomized transform function for adversarial noise neutralization.

Shaham et al. [10] proposed a robust model mechanism that makes the classification model resistant to adversarial examples. The robust model mechanism has a higher detection accuracy and a smaller decrease in classification accuracy than the adversarial noise reduction mechanism because the robust model mechanism neutralizes only the characteristics of trained adversarial examples. The robust model mechanism trains the characteristics of adversarial examples by adding adversarial examples to the training dataset. Thus, the robust model mechanism requires modification throughout the classification model, and the system administrator should obtain sufficient adversarial examples for each object class that the classification model can classify. In addition, the problem of decrease in classification accuracy remains because of the classification model modification. Zhou et al. [19] also proposed an adversarial attack block mechanism that reinforces the image classification model using adversarial examples. Zhou et al. proposed an image classification model training method that can rapidly train a robust model with a small image dataset.

Grosse et al. [11] proposed a statistics-based detection model that detects whether the input image is adversarial based on statistical characteristics. Statistics-based detection can solve the classification accuracy decrease problem because it does not require any modification of the classification model. However, it is difficult to say that the statistics-based detection model is entirely independent of the classification model because the detection is conducted by comparing the statistical characteristics of the original images and the input image. Furthermore, the detection accuracy is lower than that of the robust model mechanism, especially for adversarial examples with small adversarial noise. Thus, it is not appropriate to block adversarial attacks using only a statistics-based detection mechanism.

The proposed system detects adversarial examples based on image features such as edge noise and DCT bias. The proposed system does not have the problem of decrease in classification accuracy because it does not require any modification of the classification model, such as that required by the statistics-based detection model. Furthermore, the proposed system is entirely independent because it detects adversarial examples based on image feature characteristics that adversarial examples have in common,



(a) (b) (c)

Figure 2: Example of an adversarial attack. (a) The original image was classified as "fox" [17]. (b) The adversarial example was classified as "cow." (c) Visualization of the difference between the original image and the adversarial example.

Table 1: Adversarial attack block mechanism comparison.

	Adversarial noise reduction [9]	Robust model [10]	Statistics-based detection [11]	Proposed system
Adversarial example detection accuracy	Low	High	Low	High
Classification accuracy decrease	High	Low	No	No
Input image modification	O	X	X	X
Independence	O	X	X	O
Versatility	O	X	O	O

regardless of the object class that the classification model can classify. The low detection accuracy problem of the independent models is solved by adopting a deep neural network (DNN) as the detection method. Unlike the statistics-based detection model that uses only one feature, the proposed system uses various image features. In addition, the proposed system can use a hidden correlation between image features and adversarial examples by training the image feature characteristics using the DNN model. The DNN model can represent the high-dimensional data with encoded low-dimensional data using multilayer perceptrons. In this process, the DNN model can find hidden correlations that are hard to understand with traditional statistical mechanisms such as regression and clustering. Thus, the proposed system has a higher detection accuracy than the existing independent detection model. Moreover, the proposed system is versatile and independent. The performance of the proposed system is discussed in detail in Section 4.

3. Proposed System

The proposed system consists of an image feature extraction module and a DNN model (see Figure 3). The portion plotted with a long dashed line represents the processing flow of the model training stage. The portion plotted with the dashed-dotted line represents the processing flow of the detection stage. During the training stage, the proposed system receives an image dataset consisting of normal images and adversarial examples as a training dataset. The image feature extraction module extracts and quantifies image features such as image entropy, variance, edge density, edge noise, and DCT bias and reinforces the color composition from the image dataset. The DNN model receives the quantified image features and trains the image feature characteristics that adversarial examples have in common.

In the detection stage, the trained model judges the input images as normal or adversarial examples. The system extracts image features from the input image using the image feature extraction module and sends quantified image features as the input of the DNN model. The DNN model judges whether the input images are adversarial examples or not based on the trained characteristics of adversarial examples. The input image is judged as an adversarial example if the DNN model's decision result is greater than 0.5, and it is judged as the original image if the decision result is smaller than 0.5.

3.1. Key Image Features. The key feature of the proposed system is the edge noise that is commonly observed in adversarial examples. Figure 4 shows the edge extracted using the same algorithm and parameters for the adversarial example and original image. Figure 4(b) shows the visualization of the edge extracted from the original image (Figure 4(a)), and Figure 4(c) shows the visualization of the edge extracted from the adversarial example. The Canny edge algorithm [20] was used to extract the edges from the image. The Canny edge algorithm is a method of extracting an edge from an image based on a change in brightness. The Canny edge algorithm extracts the edge pixels by different brightness changes between pixels and finding value changes

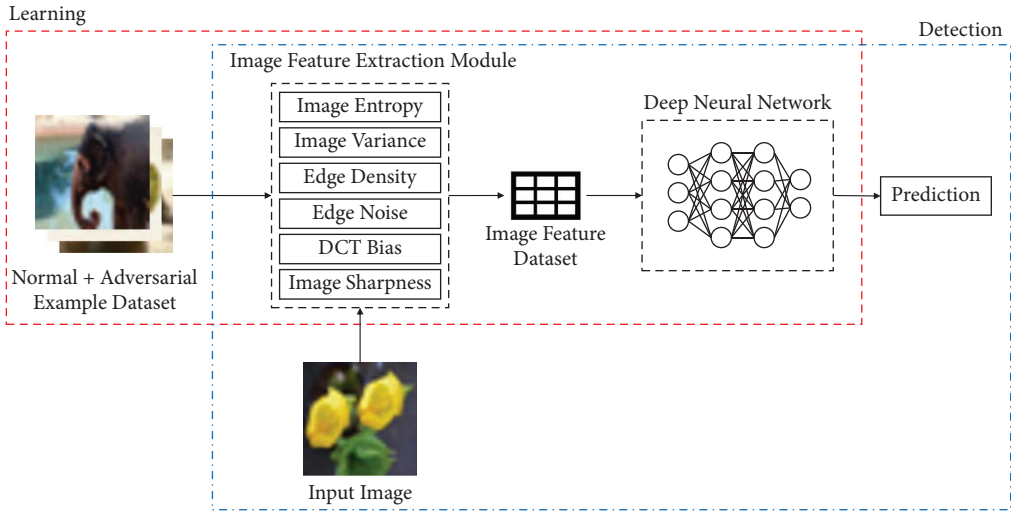


Figure 3: Proposed system structure.

(a) (b) (c)

Figure 4: Edge noise visualization. (a) Sample image [17]. (b) Edge extraction from the original image. (c) Edge extraction from the adversarial example.

rapidly. Thus, the threshold for a change in brightness must be set as a parameter. In the proposed system, the brightness change threshold th_l, th_u is automatically set based on the median brightness of the image. The brightness change threshold is calculated using Equation (1), where m is the median brightness value. The edge of the adversarial example appears irregularly broken, unlike the original image's edge, which appears as a continuous line similar to the visible contours. In addition, the edge of the adversarial example includes noises that are irrelevant to the visible contours.

$$\begin{aligned} th_l &= \max(0,\ (1-\sigma) * m), \\ th_u &= \min(255,\ (1+\sigma) * m). \end{aligned} \tag{1}$$

The degree of edge noise may differ depending on the density of the edge for each input image, image complexity, sharpness, and many other image features. Therefore, to detect adversarial examples accurately, it is necessary to determine an appropriate edge noise criterion for each input image by utilizing image characteristics other than edge noise. In the training of the proposed model, the edge component density, image sharpness, overall image dispersion, overall image entropy, and color composition ratio were used as training data along with edge noise. Detailed explanations and quantification methods for each image characteristic are described in detail in Section 3.2.

3.2. Training Dataset. Table 2 lists the image feature attributes used for training the adversarial example detection model. The independent variables are edge_noise, edge_density, dct_bias, variance, and entropy, and the dependent variable is_adv. The is_adv attribute indicates whether the image is an adversarial example. The dct_bias attribute has the highest correlation of −0.8172, and the edge noise and edge density have the correlations of 0.6974 and 0.5450, respectively. The edge_noise attribute is the ratio of the noise to the edge pixels. The edge_noise attribute represents the degree of edge noise. As explained in Section 3.1, adversarial examples contain more noise than normal images around the contours. The edge_density attribute is the ratio of edge

Table 2: Proposed system training dataset attributes.

Attribute	Datatype	Correlation	Explanation
edge_noise	Double	0.6974	Ratio of noise to the edge pixels
edge_density	Double	0.5450	Ratio of edge pixels to the total pixels of image
dct_bias	Double	−0.8172	DCT bias of the image
color_composition	Double (3)	0.1193	Percentage of each color channel (R, G, and B) of the image
Variance	Double	0.2620	Image variance of the image
Entropy	Double	0.1522	Image entropy of the image
is_adv	Boolean		Normal image/adversarial example

pixels to the total image pixels. The degree of edge noise may vary depending on the edge density. Thus, the edge_density attribute was selected as an attribute for the training dataset. The dct_bias attribute represents the sharpness of the image. The bias of the DCT coefficient was calculated using equation (2) to quantify the sharpness of the image. w and h are the width and height of the image, respectively. DCT$(i,\ j)$ is the DCT coefficient of the pixel in the i th row and j th column. Figure 5 shows the DCT coefficient (Figure 5(b)) of an adversarial example with low sharpness (Figure 5(a)) and the DCT coefficient (Figure 5(d)) of an adversarial example with high sharpness (Figure 5(c)). An adversarial example with a higher sharpness shows a higher DCT coefficient bias. The variance attribute is the variance of the pixel brightness value of the entire image, and the entropy attribute is the entropy of the pixel brightness value of the entire image. Both the variance and entropy attributes were used to quantify the complexity of the image.

$$R_i = \sum_{k=1}^{w} \mathrm{DCT}(i,k),$$

$$C_j = \sum_{k=1}^{h} \mathrm{DCT}(k,j), \tag{2}$$

$$\mathrm{bias}_{\mathrm{DCT}} = \sqrt{\left(\frac{h}{2} - \frac{\sum_{i=1}^{h} i * R_i}{\sum_{i=1}^{h} R_i}\right)^2 + \left(\frac{w}{2} - \frac{\sum_{i=1}^{w} i * C_i}{\sum_{i=1}^{h} C_i}\right)^2}.$$

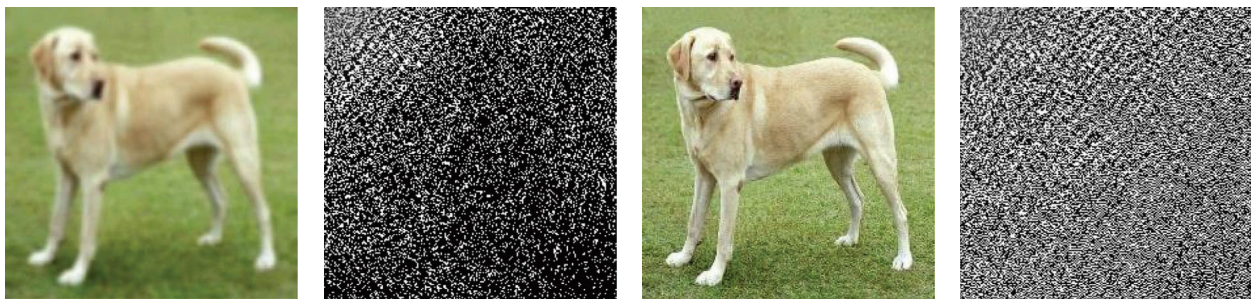
Figure 6 shows the processing flow of the edge noise attribute quantification. The Canny edge extraction algorithm and Gaussian filter were used to quantify the edge noise attributes. The flow is as follows: (a) Extract the edge from the image using the Canny edge extraction algorithm, and (b) apply a Gaussian filter. By applying a Gaussian filter, the area between the normal edge contour and the edge noise is merged. (c) Binarize the image by changing the pixel value of the pixel with a brightness above the threshold to 255 and the pixel value of the pixel that is not 0. In this case, the threshold value is determined based on the kernel size of the applied Gaussian filter. The threshold th is calculated using equation (3). This threshold is calculated based on the Gaussian function, where A is a normalizing coefficient that makes the sum of the filter elements equal to 1 and σ is the sigma parameter for the Gaussian filter, which represents the degree of blur. The increased pixel rate of the binarized image compared to that of the original extracted edge is used as the value of the edge_noise attribute. The increased pixel rate was calculated by dividing the pixel increment after the binarization step by the number of edge pixels. Figure 7 shows the edge noise of a normal image and an adversarial example using the edge_noise extraction algorithm. The edge noise of the adversarial example (see Figure 7(b)) appeared to be stronger than that of the normal image (see Figure 7(d)). The edge_noise value of the adversarial example (see Figure 7(a)) is 101.853, whereas that of the normal image (see Figure 7(c)) is 56.327.

$$\mathrm{th} = \mathrm{A} \cdot e^{-2/\sigma^2}. \tag{3}$$

The edge_density attribute is the ratio of the edge pixels to the total image pixels. The edge_density attribute is selected because the degree of edge noise can vary depending on the edge complexity. The variance attribute is the variance of the pixel brightness value of the image. The entropy attribute is the entropy of the pixel brightness value of the image. The entropy attribute was calculated using the Shannon algorithm [22]. The Shannon entropy of the data with a probability distribution P was calculated using Equation (4). The proposed model extracted image feature attributes from 26,666 images, trained the detection model with 18,666 data, and validated the model with 8,000 data.

$$H(P) = -\sum_{x} P(x) \log\ P(x). \tag{4}$$

3.3. Model Training. The correlation between each training data attribute and the adversarial example was analyzed to determine how each attribute affected the decision. The CfsSubsetEval algorithm provided by Weka [23], a data analysis tool, was used to analyze the correlation between each attribute and an adversarial example. The attributes showing a high degree of correlation with whether or not an adversarial example was found differed depending on the adversarial noise level of the adversarial example. In the adversarial example with very small adversarial noise at the level of $\varepsilon < 0.02$, the edge_noise and dct_bias characteristics showed the highest correlation in the decision. In the adversarial example with relatively large adversarial noise at the level of $\varepsilon > 0.05$, the edge_noise, dct_bias, and entropy characteristics showed the highest correlation with the decision. Unlike adversarial examples with small adversarial noise, edge noise attributes cannot be extracted normally on adversarial examples with large adversarial noise because of the noise. In other words, in an adversarial example with big



(a) (b) (c) (d)

Figure 5: DCT coefficient bias comparison: (a) adversarial example with low sharpness; (b) DCT coefficients of adversarial example with low sharpness; (c) adversarial example with high sharpness [21]; (d) DCT coefficients of adversarial example with high sharpness.

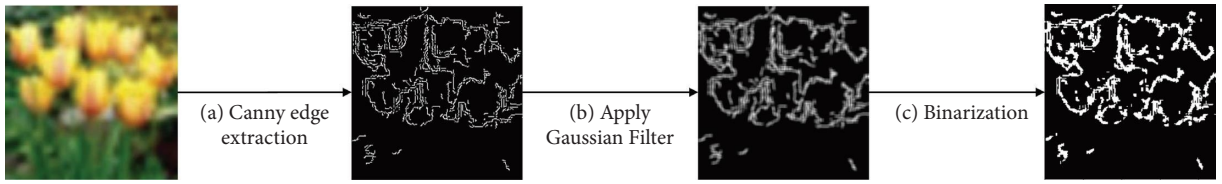
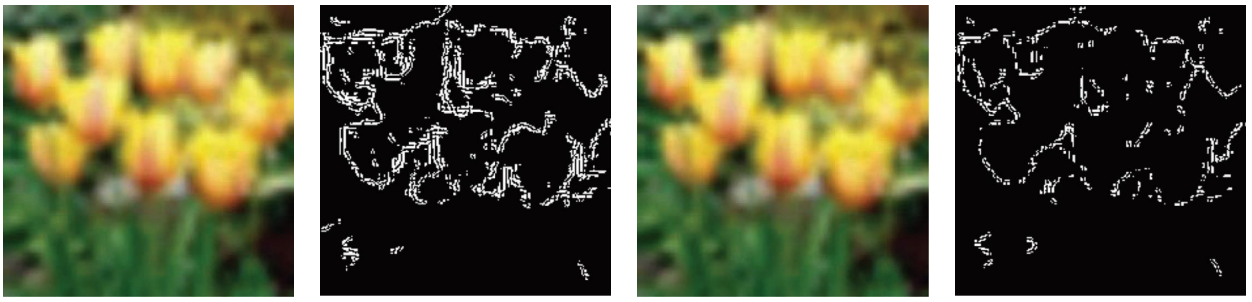


Figure 6: Edge noise quantification flow: original image [19]; (a) extract edge using Canny edge algorithm; (b) apply Gaussian filter to merge real contour and noise; (c) binarize to get filled area between contour and noise.



(a) (b) (c) (d)

Figure 7: Comparison of edge_noise extraction results in the adversarial example and normal image [19]. (a) Adversarial example. (b) Edge noise visualization of an adversarial example. (c) Normal image. (d) Edge noise visualization of the normal image.

adversarial noise, the dct_bias attribute acts as the main characteristic for determining whether or not it is an adversarial example, instead of the edge_noise characteristic. The entropy attribute acts as the comparison attribute for sharpness instead of the dct_bias attribute.

The proposed detection model is based on a DNN. Keras [24], a machine-learning framework, was used to train the proposed model. The DNN model has adjustable hyperparameters, such as layer composition and optimization functions, to achieve optimal performance. Table 3 compares the performance of each model using various hyperparameters to find the model that performs detection optimally. All of the hidden layers contain one dropout layer and one output layer, and the other layers are filled with a dense layer.

Table 4 lists the hyperparameters of the model determined through the experiment in Table 3. The proposed model consists of five hidden layers. Table 5 lists the types, number of nodes, activation functions, and dropout rate

Table 3: Hyperparameters for model performance.

Optimizer	# of hidden layers	Loss	Accuracy (%)
SGD	3	0.6843	52.94
	5	0.1172	97.83
	10	0.7012	52.68
Adagrad [25]	3	0.6816	53.28
	5	0.0661	98.23
	10	0.6844	61.06
Adam [26]	3	0.1185	97.20
	5	0.0362	99.89
	10	0.7144	47.32

(only for the dropout layer) of each layer. The hidden layers consist of four weight layers and one dropout layer. The dropout layer prevents detection accuracy loss because of overfitting. Adam was used as the optimization function. Adam optimizers have been widely used in DNN learning recently. Adam performs optimization considering the

Table 4: Model training hyperparameters.

Parameter	Value
# of hidden layers	5
Optimizer	Adam
Initial learning rate	0.01
Loss function	Binary cross-entropy

Table 5: Hidden layer composition.

No.	Type	# of nodes	Activation function/dropout rate
1	Dense	512	ELU
2	Dense	1024	ELU
3	Dropout	—	0.1
4	Dense	512	ELU
5	Dense	1	Sigmoid

momentum that changes according to the gradient, unlike the standard gradient descent (SGD) function, which performs optimization with a fixed optimization rate. Adam can avoid the problem of the optimization function deadlocking or not finding the lowest point. Binary cross-entropy was used as the loss function.

Figure 8 shows a screen capture of the implementation of the proposed system for detecting adversarial examples. Figure 8(a) is an adversarial example used for the experiment. The image should be classified as an airplane. However, the image was classified as a speedboat. Figure 8(b) shows the detection results of the image. Although the input image is an adversarial example with small adversarial noise that is difficult to distinguish from a normal image, the proposed system successfully detects the image as an adversarial example. Incentive transaction record-based user validation could drop the false alarm rate to a meaningful level.

4. Performance Evaluation

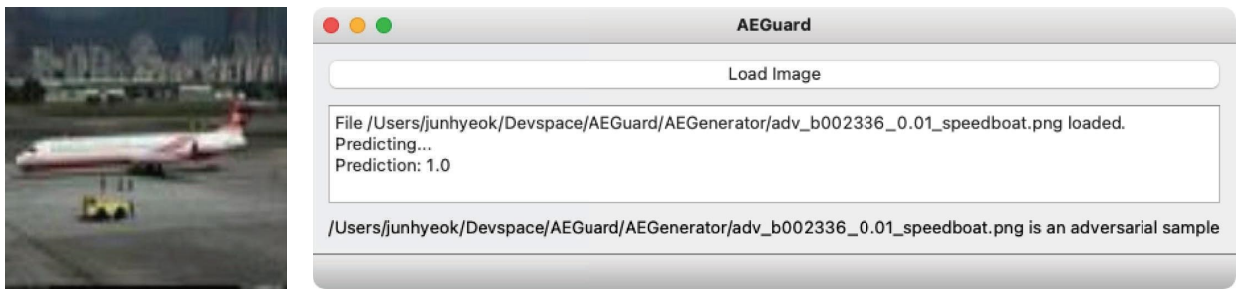
To evaluate the performance of the proposed model, we implemented the proposed robust [7] and statistical detection [8] models. Python3 was used to implement each of them, and Keras, a machine learning framework, was used to train the machine learning models. OpenCV [27], an image processing library, was used to extract image features. Each model used the CIFAR-10 dataset [14], an open dataset for image classification, as a training and verification dataset. Table 6 lists the system specifications for the experiments.

We implemented the robust model, the statistics-based detection model, and the proposed system to compare the detection accuracy of each mechanism according to the adversarial noise level of the adversarial example. The evaluation of performance was conducted with the FGSM adversarial examples generated with ε values of 0.01, 0.05, and 0.1. ε represents the degree of adversarial noise in an adversarial example. Smaller ε value means that the adversarial example is difficult to identify with the eye since it includes only small adversarial noise. The robust model showed 88.12% detection accuracy for the adversarial examples generated with an adversarial noise level of $\varepsilon = 0.01$. However, as the adversarial noise increased, the detection accuracy decreased, and this model showed a detection accuracy of 78.14% for the adversarial example generated with an adversarial noise level of $\varepsilon = 0.1$. This can be inferred because the robust model is resistant to adversarial examples by adding adversarial examples to the training data. Adding adversarial examples to the training dataset extends the classification criteria of the image feature characteristics to classify the image into a specific image class. However, the model cannot detect adversarial examples with large adversarial noise even with the extended classification criteria because the adversarial noise destroys the image feature characteristics of the original image.

The statistics-based detection model had a detection accuracy of 76.88% for adversarial examples generated with an adversarial noise level of $\varepsilon = 0.1$ (see Figure 9). However, for the adversarial example with a smaller adversarial noise of $\varepsilon = 0.01$, the detection accuracy dropped to 42.11%. This can be inferred because the statistics-based detection model detects adversarial examples based on image characteristics that differ between normal images and adversarial examples. The statistics-based detection model can easily detect adversarial examples with large adversarial noise because in this case, the distribution characteristics between normal images and adversarial examples are significantly different. However, the adversarial example with a small adversarial noise has a distribution similar to that of normal images. Thus, the statistics-based detection model has a relatively low detection accuracy for adversarial examples with small adversarial noise.

The proposed model has a detection accuracy of 99.51% in an adversarial example with an adversarial noise level of $\varepsilon = 0.01$. Furthermore, a detection accuracy of 99.91% was also shown for low-level adversarial examples with an adversarial noise level of $\varepsilon = 0.1$. Unlike the other two mechanisms, the proposed model consistently showed a high detection accuracy even when the adversarial noise level changes. This can be inferred because the proposed model considers the impact of various image features on the detection result, unlike the other mechanisms. The proposed model uses entropy and DCT bias as key image feature attributes to detect adversarial examples with large adversarial noise. For adversarial examples with small adversarial noise, the DCT bias and edge noise are used as key image feature attributes. Because the proposed system can flexibly choose the key image feature attribute according to the characteristics of the input image, the detection model can consistently detect various adversarial examples with high accuracy. In addition, the detection model can find the relation between image features and adversarial examples that cannot be found by simple regression analysis using the DNN model. Based on these characteristics, the proposed model shows consistently higher detection accuracy than the existing mechanisms.

The proposed system can detect the original images as adversarial examples. We calculate the false-positive rate of the proposed system on adversarial examples with various ε values (see Figure 10). On the adversarial examples with $\varepsilon = 0.1$, the proposed system shows 0.04% of false-positive



(a) (b)

Figure 8: Adversarial example detection using the proposed system. (a) Adversarial example used for the experiment [19]. (b) Detection result of the adversarial example.

Table 6: System specification for the experiment.

CPU	Intel core i7 2.8 GHz quad-core
RAM	32 GB DDR4
GPU	Nvidia GTX 1080 Ti
VRAM	11 GB GDDR5X

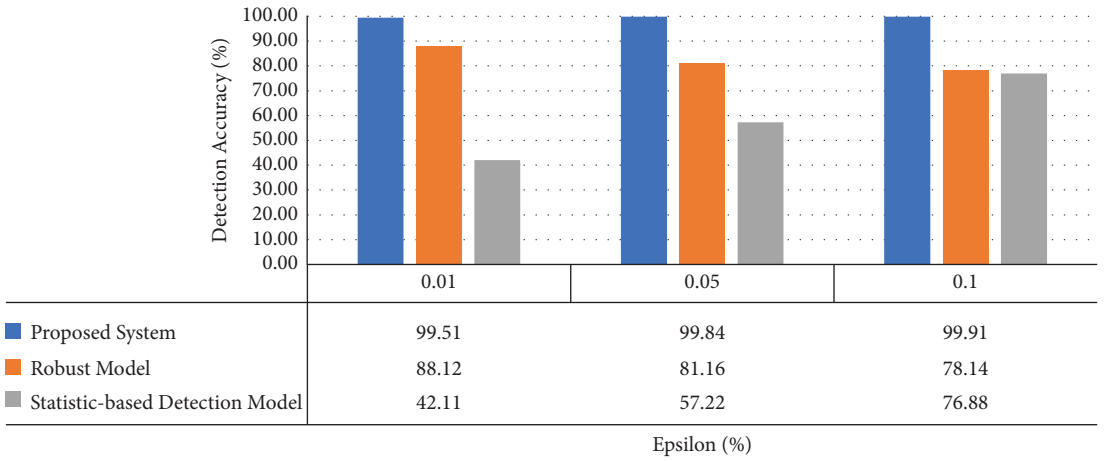


Figure 9: Comparison of the performance of adversarial attack block mechanisms.

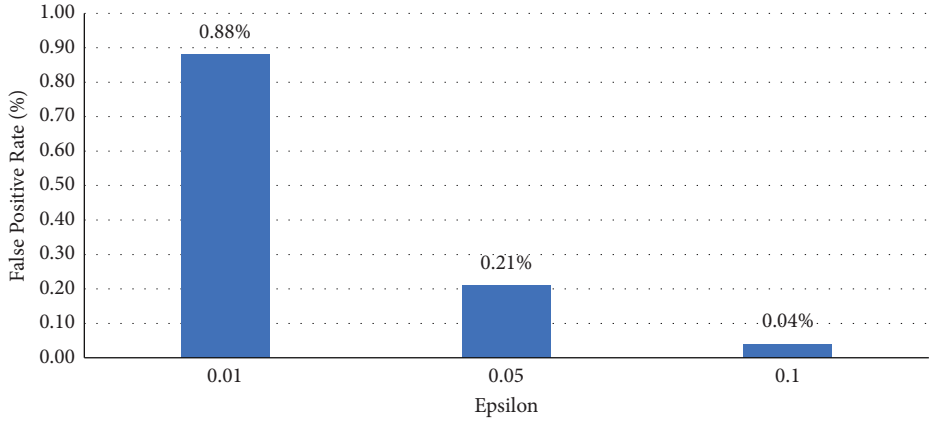


Figure 10: Comparison of false-positive rate of adversarial attack block mechanisms.

rate. However, in the adversarial examples with $\varepsilon = 0.01$, the proposed system shows 0.88% of false-positive rate, which is relatively higher. This result can be inferred because the adversarial examples with a lower ε value has less distinct the characteristics of the adversarial examples. However, since the proposed system has an accuracy of 99.51%, which is relatively higher than that of existing mechanisms, this level of false-positive rate is understandable.

5. Conclusion

In this paper, we proposed a mechanism to quantify various image features, such as edge noise and DCT bias, and detect adversarial examples based on image features using the DNN model. The DNN model is trained to detect the image feature characteristics that adversarial examples have in common. The detection model shows higher detection accuracy than the existing mechanisms because the model considers the impacts of various image features on the detection result. Moreover, the proposed detection model is completely independent of the classification model. Thus, common problems of existing mechanisms, such as a decrease in classification accuracy and a high training cost caused by the dependency of detection mechanisms, are solved.

The performance of the proposed model was evaluated by comparing its detection accuracy with that of the robust [7] and statistics-based detection [8] models. The feasibility of the proposed system was demonstrated by showing that the proposed detection model consistently achieves high accuracy for adversarial examples with various degrees of noise and sharpness. In future work, we going to adopt the proposed system to various image classification systems such as ResNet and AlexNet and conduct a performance evaluation for various adversarial example generating methods such as PGD and L-BFGS to improve the performance of the proposed system.

Data Availability

The data are available from the following link: https://github.com/junhyeok-dev/AEGuard.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (No. 2018R1A2B6009620).

References

[1] X. Feng, Y. Jiang, X. Yang, M. Du, and X. Li, "Computer Vision algorithms and hardware implementations: a survey," *Integration*, vol. 69, pp. 309–320, 2019.

[2] C. Badue, R. Guidolini, R. V. Carneiro et al., "Self-driving cars: a survey," *Expert Systems with Applications*, vol. 165, Article ID 113816, 2021.

[3] N. Bhadwal, V. Madaan, P. Agrawal, A. Shukla, and A. Kakran, "A. Smart border surveillance system using wireless sensor network and computer vision," in *Proceedings of the 2019 International Conference on Automation, Computational and Technology Management (ICACTM)*, pp. 183–190, London, UK, April 2019.

[4] A. M. Wyglinski, X. Huang, T. Padir, L. Lai, T. R. Eisenbarth, and K. Venkatasubramanian, "Security of autonomous systems employing embedded computing and sensors," *IEEE Micro*, vol. 33, no. 1, pp. 80–86, 2013.

[5] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: a survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018.

[6] Y. Deng, X. Zheng, T. Zhang, C. Chen, G. Lou, and M. Kim, "An analysis of adversarial attacks and defenses on autonomous driving models," in *Proceedings of the 2020 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 1–10, Austin, TX, USA, March 2020.

[7] I. Corona, G. Giacinto, and F. Roli, "Adversarial attacks against intrusion detection systems: taxonomy, solutions and open issues," *Information Sciences*, vol. 239, pp. 201–225, 2013.

[8] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial attacks and defences: a survey," 2018.

[9] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2019.

[10] U. Shaham, Y. Yamada, and S. Negahban, "Understanding adversarial training: increasing local stability of neural nets through robust optimization," 2016.

[11] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. McDaniel, "On the (statistical) detection of adversarial examples," vol. 13, 2017.

[12] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *IJEAST*, vol. 04, no. 12, pp. 310–316, 2020.

[13] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, "HotFlip: white-box adversarial examples for text classification," 2017.

[14] C. Guo, J. R. Gardner, Y. You, A. G. Wilson, and K. Q. Weinberger, "Simple black-box adversarial attacks," in *Proceedings of the International Conference on Machine Learning*, pp. 2484–2493, California, USA, June 2019.

[15] A. U. H. Qureshi, H. Larijani, M. Yousefi, A. Adeel, and N. Mtetwa, "An adversarial approach for intrusion detection systems using Jacobian saliency map attacks (JSMA) Algorithm," *Computers*, vol. 9, no. 3, p. 58, 2020.

[16] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2015.

[17] A. Krizhevsky, "The CIFAR-10 dataset,".

[18] Y. Zheng, Y. Lu, and S Velipasalar, "An effective adversarial attack on person Re-identification in video surveillance via

dispersion reduction," *IEEE Access*, vol. 8, pp. 183891–183902, 2020.

[19] M. Zhou, J. Wu, Y. Liu, S. Liu, and C. Zhu, "DaST: data-free substitute training for adversarial attacks," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Patten Recognition (CVPR)*, pp. 231–240, Seattle, WA, USA, June 2020.

[20] W. Rong, Z. Li, W. Zhang, and L. Sun, "An improved Canny edge detection algorithm," in *Proceedings of the 2014 IEEE International Conference on Mechatronics and Automation*, pp. 577–582, Tianjin, China, August 2014.

[21] Google, "Tensorflow example image," 2022.

[22] P. A. Bromiley, N. A. Thacker, and E. Bouhova-Thacker, "Shannon entropy, renyi entropy, and information," *Statistics and Inf. Series*, pp. 1–8, 2004.

[23] University of Waikato, "Weka 3," 2021.

[24] C. François, "Keras," 2021.

[25] A. Lydia and S. Francis, "Adagrad—an optimizer for stochastic gradient descent," *IJICS*, vol. 6, no. 5, pp. 565–568, 2019.

[26] P. K. Diederik and B. Jimmy, "Adam: a method for stochastic optimization," 2014.

[27] Intel, "OpenCV," 2021, https://opencv.org/.