IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 8, 2021, accepted September 21, 2021, date of publication September 27, 2021,
date of current version October 5, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3115764*

# Robust Adversarial Attack Against Explainable Deep Classification Models Based on Adversarial Images With Different Patch Sizes and Perturbation Ratios

**THI-THU-HUONG LE[1,2], HYOEUN KANG[3], AND HOWON KIM[3], (Member, IEEE)**

[1]IoT Research Center, Pusan National University, Busan 609735, South Korea
[2]Faculty of Information Technology, Hung Yen University of Technology and Education, Hung Yen 16000, Vietnam
[3]School of Computer Science and Engineering, Pusan National University, Busan 609735, South Korea

Corresponding authors: Thi-Thu-Huong Le (lehuong7885@gmail.com) and Howon Kim (howonkim@pusan.ac.kr)

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant through the Korean Government [Ministry of Science and ICT (MSIT)] (Regional Strategic Industry Convergence Security Core Talent Training Business) under Grant 2019-0-01343, and in part by the Technology Innovation Program (Developed Onestop Smart Factory Integrated Package For Machine Robot Industry) through the Ministry of Trade, Industry and Energy (MOTIE), South Korea, under Grant 20004625.

**ABSTRACT** In recent years, adversarial attack methods have been deceived rather easily on deep neural networks (DNNs). In practice, adversarial patches cause misclassification that can be extremely effective. However, many existing adversarial patches are used for attacking DNNs, and only a few of them apply to both the DNN and its explanation model. In this paper, we present different adversarial patches that misguide the prediction of DNN models and change the cause of prediction results of interpretation models, such as gradient-weighted class activation mapping. The proposed adversarial patches have appropriate location and perturbation ratios, which comprise visible or less visible adversarial patches. In addition, image patches within small arrays are localized without covering or overlapping with any of the main objects in a natural image. In particular, we generate two adversarial patches that cover only 3% and 1.5% of the pixels in the original image, while they do not cover the main objects in the natural image. Our experiments are performed using four pre-trained DNN models and the ImageNet dataset. We also examine the inaccurate results of the interpretation models through mask and heatmap visualization. The proposed adversarial attack method could be a reference for developing robust network interpretation models that are more reliable for the decision-making process of pre-trained DNN models.

**INDEX TERMS** AI security, explainable AI (XAI), gradient-weighted class activation mapping (Grad-CAM), adversarial patch, image classification, pre-trained model.

## I. INTRODUCTION

The have become state-of-the-art models compared to traditional methods in the image recognition field and even obtained human-like results [1]. Nevertheless, noise on original images easily makes DNN models misclassify by generating adversarial images as shown in previous studies [2]–[5].

To generate adversarial images, an excellent concept is adding a small amount of pixel perturbation into a natural image as human imperceptibility. Such modification can cause deception of the classification model in predicting a different class using an adversarial image. However, previous methods did not focus on minimal modification, but modified a large number of pixels such that they may be perceptible to human eyes. For example, for adversarial images generated with the Jacobian-based saliency map approach [5], 4% perturbation of the total number of pixels is conducted and can be visible to the human eye. Hence, an expert can easily recognize abnormal noise, which is generated by adversarial large-pixel perturbation. In contrast, an attack on DNN models by modifying only one pixel on an image is proposed in the research study presented in [6].

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaochun Cheng.

The method was based on generating one-pixel adversarial perturbations using differential evolutions to create low-cost adversarial attacks against DNNs.

Recently, explainable artificial intelligence (XAI) has become a trend in AI research because it contains reliable interpretation models that explain the underlying decisions of machine learning and deep learning models. For instance, several research studies [7]–[9] have focused on describing a local explanation of the models' outputs for a given input [10]. Meanwhile, the explanation model and adversarial learning have a relationship between them [11], [12]. Therefore, XAI is also used to defend the AI model [13]. However, recent studies proposed several attack methods showing that some XAI models have also been easily attacked. Some examples are the input gradient [14], meaningful perturbation [15], fooling network interpretation [16], adversarial model manipulation [17], deceiving the local interpretable model-agnostic explanations (LIME), and Shapley additive explanations (SHAPs) [18].

One of the most well-known interpretation algorithms in DNN-based image classification task is the gradient-weighted class activation mapping (Grad-CAM) that performs well and outperforms state-of-the-art interpretation algorithms used in [9], [19]. Hence, we choose the Grad-CAM algorithm to mislead the explanation decision of pre-trained DNN models upon the proposed attack model. However, the challenge for misguiding an interpretable model is different for Grad-CAM due to the different architectures of pre-trained models. Each pre-trained DNN classification model has a different quality of Grad-CAM on the image. Figure 1 shows the results of Grad-CAM on two examples of image classification using four interpreted classification models.

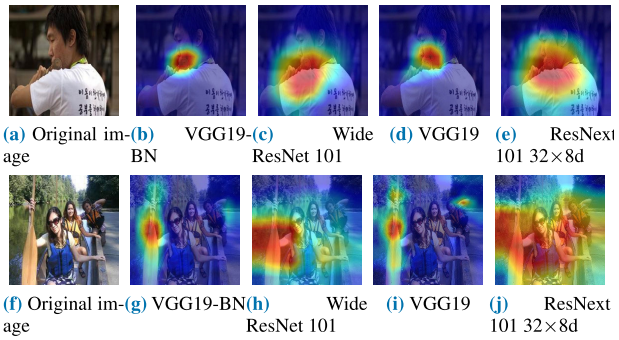



**FIGURE 1. Grad-CAM results on the two examples of pre-trained classification models.**

The two issues in adversarial attack research methods are (1) the generation of adversarial examples using noise that is indistinguishable to the human eye and covers the entire image [2], [3] and (2) visible noise that covers noteworthy feature of the main object in the natural image; for example, a face identification task has noise due to the existence of glasses with a specific pattern around a person's eyes [20]. Hence, in this study, we examine cases of visible or less visible noise localized to small areas of the image, such as a bounding box with up to 3% or 1.5% of the pixels, which do not cover the main objects of the image.

In this study, we create an adversarial attack algorithm that deceives the interpretation network, such as Grad-CAM and different architectures of classification networks. Our main contributions are as follows:

- We propose a robust adversarial image patch (AIP) by analyzing and determining its important factors, i.e., effective location, size, and perturbation ratio with different features from the adversarial patch in [16].
- We propose a general framework and algorithm for adversarial Grad-CAM, along with two types of the pre-trained DNN model architectures (i.e., feature module and no feature module). Additionally, we create two scenarios: (1) deceiving pre-trained model and making a heatmap of Grad-CAM on only AIP with full perturbation ratio and (2) deceiving the pre-trained model and Grad-CAM while highlighting both the main object and AIP with a part of perturbation ratio.
- We perform four experiments related to our proposed method on the ILSVRC image dataset. Two different types of pre-trained models are used (i.e., feature and no feature layer). Specially, we examined our proposed method on two pre-trained models: Visual Geometry Group 19-Batch Normalization (VGG19-BN) and Wide Residual Networks (Wide ResNet 101). Another two pre-trained models, i.e., Visual Geometry Group 19 (VGG19) and Residual Network (ResNext 101 $32 \times 8$d), are used for testing our method.
- We explain the Grad-CAM misinterpreted results using mask and heatmaps from Grad-CAM results to assess the results obtained using our method.

The remainder of this paper is structured as follows. Section II describes the related work. Section III presents the background of our proposed method. The proposed method is described in detail in Section IV. Section V presents the results and discussion, and finally Section VI concludes the paper.

## II. RELATED WORK

An adversarial example (AE) is a small instance in which intentional feature perturbations cause machine learning or deep learning models to make an incorrect prediction [21]. Later, Goodfellow *et al.* [3] proposed Fast Gradient Sign Method (FGSM) to improve AE with only one iteration of optimization.

Recent studies show that a DNN classification model is vulnerable to adversarial examples in different applications, e.g., AE against DNN-based network intrusion detection system (IDS) [22], DNN-based privacy leakage for Internet of Things (IoT)-based invisible AE [23], attack DNN-based wireless communication system [24], attack for medical image classification [25], [26], and so on. These results imply that the target model is attacked to reduce accuracy performance regardless of a white-box model or a black-box model.

### A. ADVERSARIAL PATCHES AGAINST DNN MODELS

An adversarial patch (AP) has been introduced first in the study presented in [27]. We can add an AP in any figure and scene, among others. In recent years, the AP is widely used against DNN-based applications. Recently, some researchers proposed several types of AP, such as DPatch, AP on attacking person detection, and IPatch. In particular, Liu *et al.* [28] and Zhao *et al.* [29] proposed attacking object detection using DPatch. DPatch is a black-box AP with a small patch in the input image. It can perform attacks against mainstream modern detectors, such as two-stage detector faster region-convolutional neural network and one stage detector you only look once (YOLO). Thys *et al.* [30] proposed APs to attack person detection, and the proposed method was successful in hiding people from a person detector. IPatch was a remote AP used in [31]. This patch could generate new scenes and impact other semantic models, such as object detectors.

### B. ADVERSARIAL METHODS AGAINST INTERPRETABLE DNN MODELS

Previously, researchers have concentrated on attacking interpretation models, and especially pre-trained DNN models. In [14], the proposed method focused on misleading the adversarial interpretability of DNN using input gradient. In [15], a deceiving interpretable model using meaningful perturbation was proposed. In addition, misguiding NN interpretation via adversarial model manipulation is proposed in [17]. This method has modified the model parameters; however, the adversary might not modify the model parameters in a practical setting. The researchers in [16] proposed a deceiving method for network interpretation in image classification by modifying only the pixels in a small image area without adjusting the model. However, the fooling success rate (FSR) of AP attacked results in some cases is not really high because the heatmap results are not highlighted resolutely or incorrectly in the AP target.

In this paper, we use an AP with a small area, and the reasons are explained as follows. First, Grad-CAM is based on extracting the last convolution layer (class activation mapping) that contains the important feature of an object or image to make the DNN's decision. The Grad-CAM results are highlighted by a heatmap with a determined mask. Hence, to mislead Grad-CAM on DNN models, we should make Grad-CAM highlight on the fixed target location that we want to deceive. Second, we control the settings using AP, where the adversary modifies the network interpretation and prediction through manipulating only a small region of the original image. Hence, the AP is suitable for fooling the Grad-CAM interpretation as well as the classification models. A consistent perturbation ratio was found, which made the AP invisible to ensure not losing the attack effect.

Recent work in [32] proposed a Wasserstein generative adversarial network (WGAN), which is a training framework to denoise blurriness to generate clean images. On the one hand, other researchers [33], [34] proposed their approaches against adversarial attacks on image and camera applications, respectively. These approaches are different viewpoints against adversarial attacks on DNN-based interpretation models. On the other hand, Veeraiah *et al.* [35] suggested a trust-based energy-efficient navigation in Mobile ad hoc networks (MANETs) that selects the best jumps in advancing the routing in securing MANETs. Other work such as [36] proposed DNN and Gaussian filtering for accurate magnetic resonance image super-resolution in the stationary wavelet domain. Nevertheless, these approaches protect the network and image domain. Otherwise, our scope is to mislead the interpretable pre-trained DNN model using the original image.

## III. BACKGROUND

### A. PRE-TRAINED DNN MODELS FOR IMAGE CLASSIFICATION

One of the major factors for the rapid advances in computer vision research is pre-trained models. Rather than developing everything from scratch, researchers can use these state-of-the-art models as a convenience. Pre-trained DNN models are neural networks trained on large benchmark datasets such as ImageNet. These models are used as target models for classification tasks and bring great benefits in developing open-source models for the deep learning community.

The major issue in training a model is to classify images into 1,000 separate object categories. We come across these 1,000 image categories in our day-to-day lives; they represent cars, cats, dogs, humans, and so on. Through transfer learning, the pre-trained network models can strongly generalize to images outside the ImageNet dataset and then transfer the learning of pre-trained models into our specific problems. The issue is how to determine the correct weights for the network through multiple forward and backward iterations. Indeed, we can directly use the architecture and weights of pre-trained models previously trained on large datasets. Then, the learning can be applied to our problem.

Recently, pre-trained models have been built using different libraries, such as Keras, TensorFlow, and PyTorch. Researchers used the ImageNet dataset to build these models because of the large image data size (1.2 million images). Pre-trained models for image classification on ImageNet have two main architectures. One has a feature module, and the other one has no feature module. Figure 2 shows the architecture of the pre-trained DNN models with the two types of architectures. In Figure 2, the pre-trained models have a feature module that consists of convolution block, max-pooling, and fully connected layers. Moreover, the pre-trained models with no feature module consist of convolution block layers, max-pooling layer, layer 1, layer 2, layer 3, layer 4, and a fully connected (FC) layer.

In this study, we selected two pre-trained models with a feature module and two pre-trained models with no feature module provided by the Torchvision library

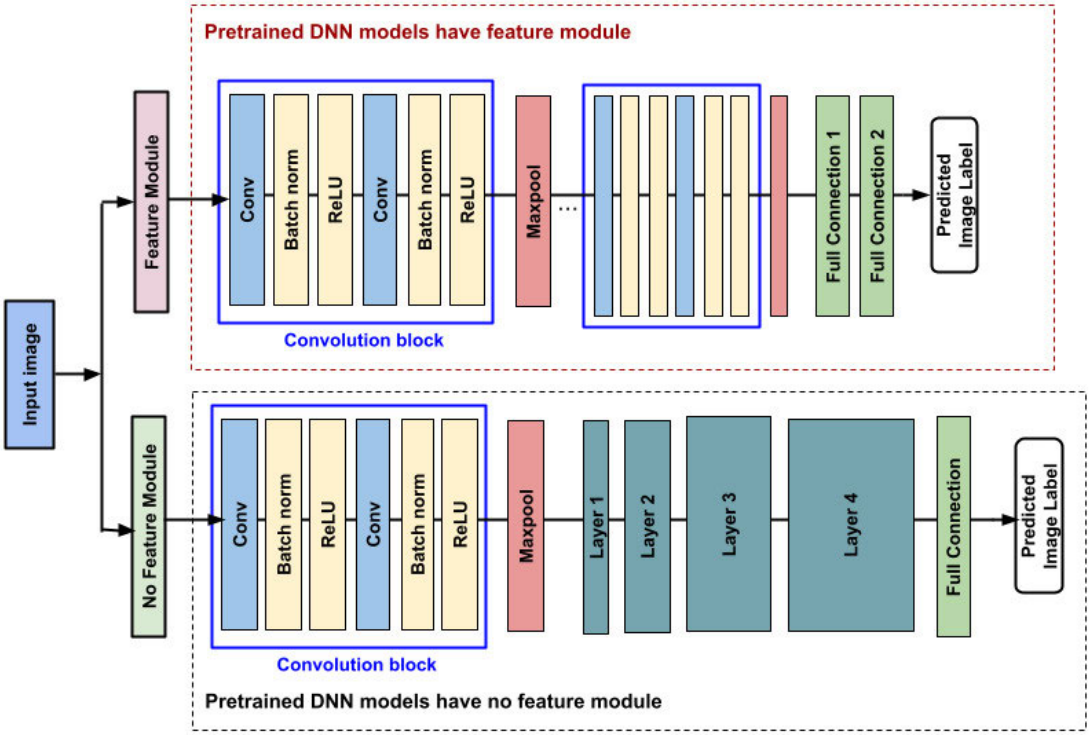



**FIGURE 2.** Architecture of pre-trained DNN models with and without feature module.

for our experiments. VGG19 with Batch normalization (VGG19-BN) and VGG19 are representatives selected as pre-trained models with a feature module. Two pre-trained models were selected with no feature module, such as Wide ResNet 101 and ResNet 101 (32 × 8d).

### B. GRAD-CAM EXPLANATION OF PRE-TRAINED DNNs MODELS

Class activation mapping (CAM) is a useful tool for explaining DNN models (such as the CNN model). It is based on replacing the fully connected layer attached to the convolution layer of the pre-trained model using global average pooling (GAP) and then by performing fine-tuning. CAM is possible to know which part of the image the neural network saw and make a judgment with a specific label. Despite the advantages, CAM has inherent disadvantages. Full connected layer (FC) must be replaced with GAP, which can use only the convolutional layer just before GAP, and the weight information of the dense layer behind the GAP is required. Hence, it is necessary to go through the process of fine-tuning or re-training. Due to this problem, it is not easy to apply CAM to CNNs that perform various purposes, such as visual question answer (VQA) or captioning in addition to object detection. The general idea behind Grad-CAM is similar to CAM. To understand which parts of an input image are important for a classification task, Grad-CAM uses the feature maps produced by the last convolution of pre-trained DNN models.

We first assume that we have some feature map $FM_1$, $FM_2, \ldots, FM_i$ that are weighted to create the final heatmap. Feature maps were weighted using alpha values that are based on gradients in Grad-CAM. Therefore, we can measure by gradients by using any neural network layer that does not require a particular architecture. The output of Grad-CAM is a class discrimination and localization map, e.g., a heatmap where the important feature part corresponds to a particular class. Figure 4 shows the concept of Grad-CAM with two types of architecture for pre-trained DNN models.

We have the score for class $c$ ($y^c$), which is the output for class c before the softmax function. Grad-CAM was applied to a neural network that has finished training. The weights of the neural network are fixed. We feed an image into the network to calculate the Grad-CAM heatmap for that image for a selected class of interest. Grad-CAM [9] has three steps:

- *Step 1:* Compute gradient. The gradient of $y^c$ with respect to feature map activation $FM^k$ of a convolution layer is

$$w^k = \frac{\partial y^c}{\partial FM^k} \tag{1}$$

- *Step 2:* Calculate alpha by average gradients. Apply the GAP for the gradients over the width dimension ($i$) and the height dimension ($j$) to obtain neuron importance weights $g_k^c$ calculated as follows:

$$g_c^k = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial FM^k} \tag{2}$$

where a number of pixels in the feature map $Z$ satisfies the equation $Z = \sum_i \sum_j 1$; and the average gradient $g$

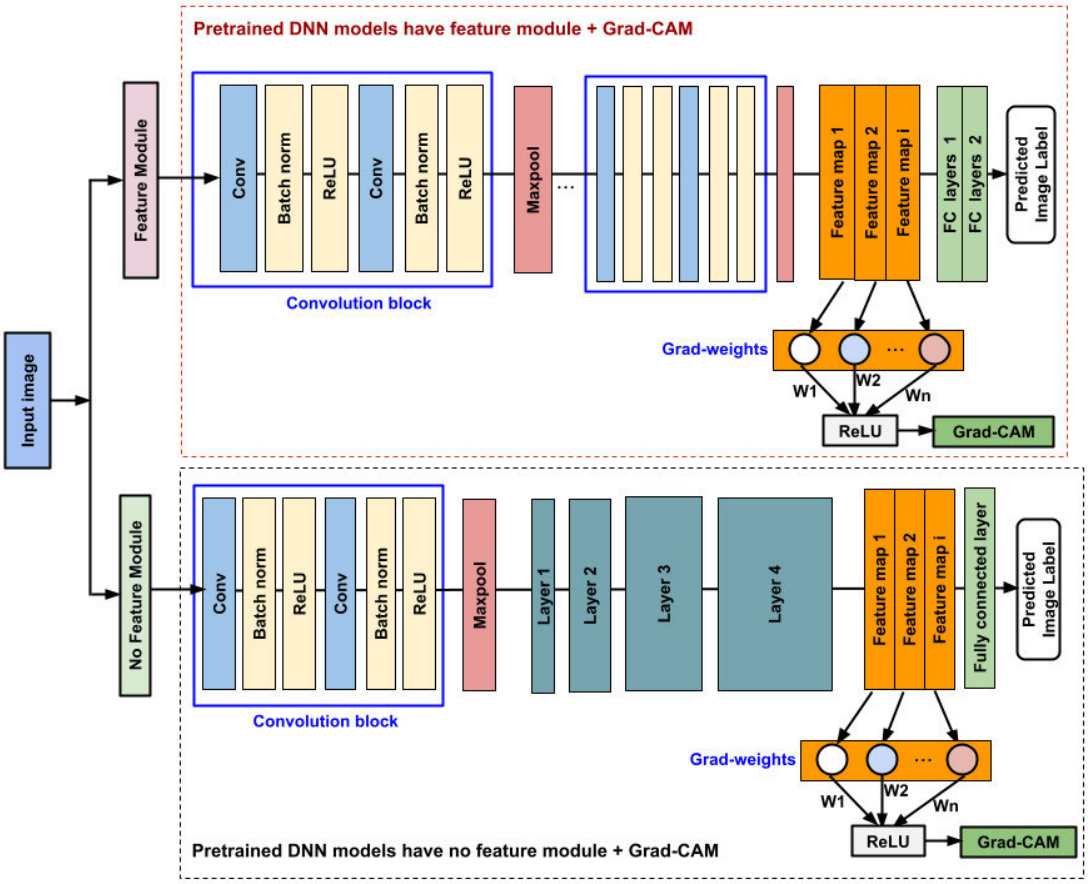



**FIGURE 3.** Grad-CAM interpretation of pre-trained DNN models.

for classes $c$ and feature map $k$ is going to be used in the next step as a weight applied to the feature map $FM^k$.

- *Step 3:* Calculate final Grad-CAM heatmap ($H$). Performing a weighted combination of the feature map activation $FM^k$ where the weights are the $\alpha_k^c$ just calculated:

$$H_g^c = ReLU \sum_k g_k^c FM^k \tag{3}$$

where the heatmap color is calculated using *applyColorMap* function in cv2 with COLORMAP_JET.

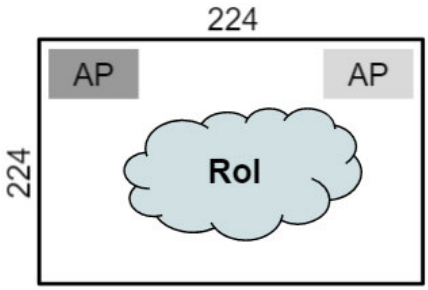



**FIGURE 4.** ROI and AP localized at the top-left corner and top-right corner in an image size of 224 × 224.

### C. ADVERSARIAL PATCHES AND LOCATION

Most adversarial patches of deep learning-based image classifiers use noise that does not cover the entire image. We must consider a region of interest (RoI) of the image to avoid overlapping APs on the main features of the image. RoI is an important portion of an image that contains the main object(s) that we want to filter or perform other operations. For example, we define a RoI by creating a binary mask that is the same size as an image to process pixels that define the RoI set to 1 and all other pixels set to 0.

The AP is not RoI, such as a top-left or top-right corner. We locate the patch on the top-left or top-right corner of the image without overlap with the main objects of interest. We assume that the input image size is 224 × 224 and the patch sizes are 64 × 64 and 32 × 32, which occupy almost 3% and 1.5% of the image area, respectively.

The AP size is a predetermined factor that could affect the effectiveness of the patch. There is a tradeoff between smaller patches that are harder to detect and defend, while larger patches provide a better attacking effect. In our experiment, we produced two different sizes of AP, namely, 32 × 32 and 64 × 64 to test the efficiency of their attacks. In this manner, we can better understand the relationship between AP size and its attacking effects to find the minimum size of a patch for a meaningful attack.

## IV. THE PROPOSED METHOD

In this work, we have built our method upon the Grad-CAM interpretation method and APs. Figure 5 shows an AP attached to an image with an input label as the "elephant" example. The pre-trained model has misclassification with the classified label "bird." Further, Grad-CAM interpretation results are misguided by highlighting the main feature of the misclassification result.

The heatmap has highlighted the patch quite strongly, disclosing the cause of the attack. The adversary attacks only the patch area, and the patch is the cause of the final misclassification toward the target category

The general proposed framework for deceiving Grad-CAM of the pre-trained DNN models is depicted in Figure 5. Its three main components are initialization of adversarial image patches, adversarial Grad-CAM attack, and explanation of Grad-CAM results. The proposed method is processed through three main components as well as three main steps described as follows:

- First, we need to create AIP in two cases from the original input image. The first case is AIP at a top-right location with patch size 64 × 64 and full perturbation ratio (i.e., 100%). The second case is AIP at the top-right location with patch size 32 × 32 and deduced perturbation ratio by 20%.
- Next, these patches will pull in pre-trained DNN models to extract feature maps with the last layer used to compute the heatmap of Grad-CAM in the second component. We will adjust and update gradient weights based on loss update. Then, we can find the final best AIP to fool the pre-trained model successfully, and Grad-CAM can attack the image.
- Finally, we can explain the Grad-CAM attacked results by generating mask and heatmap from fooling those results.

The proposed algorithm 1 generated an AP following the standard adversarial noise generation setup. In particular, we explored the generated localized APs as visible or less visible with noise to a single image in the first setup. We assume access to a pre-trained model ($pM$) that assigns adversarial image patch ($aiP$), Grad-CAM image perturbation ($giP$), mask fooled explanation ($mfE$), and heatmap fooled explanation ($hfE$) to the original input images ($oiI$). We computed the gradient total loss based on the Grad-CAM loss measurement. We seek $aiP$ that is calculated by the network based on perturbation ratio $P$, image, and gradient total loss. In other words, the $aiP$ comprises the original image with additive noise ($N$). This causes an optimization problem, i.e., seeking and adjusting a gradient total loss value to find the suitable $aiP$. We can find the gradient total loss using a stochastic gradient-based algorithm. We want noise $N$ to be limited to a small area over the image $oiI$ and to replace this area rather than be added to it. This is achieved by setting a mask perturbation value $P$ to be 1, if the patch size $pS$ is 64 × 64, or 0.2 if the patch size $pS$ is 32 × 32, and considering the noised image as $aiP$ to be

$$(1 - P) \odot oiI + P \odot N \tag{4}$$

where $\odot$ is element-wise multiplication.

To attach and hide $aiP$ in the network interpretation from the final prediction, we supplemented the loss function for optimizing until the heatmap of Grad-CAM interpretation is highlighted at the patch location. Hence, from the $aiP$, we optimized loss using the following equation:

$$argmin\Big[\sum (G(aiP) \odot P) + \alpha \times loss(aiP; y_t)\Big] \tag{5}$$

where $y_t$ is the target output and $\alpha$ is the hyper-parameter learning rate to handle the effect of two-loss terms, such as cross loss and total loss. G is the interpretation (heatmap), defined as the weighted sum of activations of the convolution layer discarding the negative values. In Eq. 5, there are two-loss components. The first loss is Grad-CAM loss, which computes the loss for the patch location pixels in the Grad-CAM tensor. For a 224 × 224 image, the AP sizes are 64 × 64 and 32 × 32. The second one is cross-entropy (CE) loss. We added CE loss if the target category is not the top predicted category.

In a pre-trained DNN model for image classification, we fed the original image to this network and obtained the final output decision. Further, we can explain the model decision result by generating a heatmap for the convolution layer to highlight the regions of the image that cause the predicted output of the network model as follows:

$$G = \sum \sum \frac{\partial(output\_Adv)}{\partial(feature\_Adv)} \tag{6}$$

where $output\_Adv$ is the predicted result of pre-trained DNN model, $pM$; and $feature\_Adv$ is the feature adversarial which is extracted from $pM$ with original input image, $oiI$.

In particular, we extracted the output of the last layer from the pre-trained models (line 12). Then, we computed the gradient of the loss corresponding to the last layer for the image adversarial (line 13). We also computed the gradient weighted class activation for perturbed images (line 14). Thereafter, we computed Grad-CAM for perturbed images (line 15). In lines 16–19, we computed the loss for the patch location pixels in Grad-CAM tensors. If the patch size is 64 × 64, the distribution is a ratio of 4:4, otherwise, if the patch size is 32 × 32, the distribution is a ratio of 2:2. If the target category is not the top predicted category, we calculated the CE loss (line 21). We minimized both Grad-CAM loss and CE loss (line 22) and computed the gradient of the total loss concerning the perturbed image (line 23). Lines 24–25 perform gradient ascent using a gradient of total loss with a learning rate of 0.05. From lines 29–32, we calculate GradCAM using AIP to visualize the Grad-CAM image perturbation result (g$iP$):

First, we must calculate Grad-CAM as mask adversarial ($mask\_Adv$) as follows:

$$mask\_Adv = G(aiP) \tag{7}$$

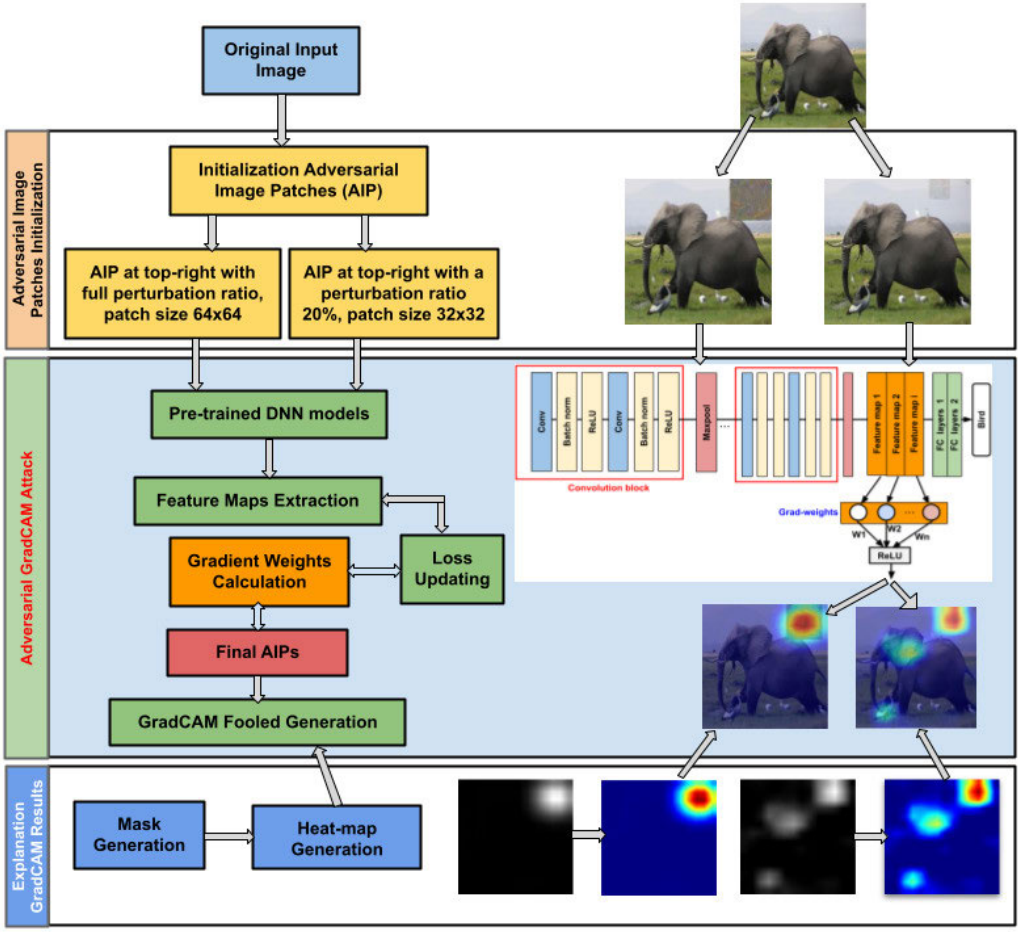



**FIGURE 5.** Proposed framework for deceiving Grad-CAM pre-trained models.

where the $G$ function is calculated based on the gradient and features with input is the $aiP$ following Eq. 6.

Then, we use $mask\_Adv$ to calculate $giP$:

$$giP = mask\_Adv + aiP \tag{8}$$

Subsequently, we interpreted the fooled Grad-CAM result through a mask fooled explanation ($mfE$) and heatmap fooled explanation ($hfE$). In lines 34–36, we explain the Grad-CAM fooled results; that is, the mask and heatmap using transpose ($T$) of mask adversarial following will provide the equations below:

$$mfE = \frac{mask\_Adv^T - min(mask\_Adv^T)}{max(mask\_Adv^T)} \tag{9}$$

$$hfE = ReLU(\sum \alpha \times mask\_Adv) \tag{10}$$

## V. EXPERIMENTAL RESULTS AND DISCUSSION

### A. EXPERIMENTS SETUP

#### 1) DATASET

We performed our experiments using ImageNet ILSVRC2012 [37] with different patch sizes and noise ratios. The images were resized into $224 \times 224$, and the noise square patches have the sizes of $64 \times 64$ and $32 \times 32$ (approximately 3% and 1.5% of the image pixels, respectively). We considered choosing a patch location around the corners, especially at the top-left or top-right corner, because these places do not cover the original image's main object(s). We made APs with noise until the desired confidence is reached or the loss in 1,000 iterations and a learning rate of 0.05 is minimized.

#### 2) PRE-TRAINED MODELS

We used the provided PyTorch pre-trained VGG19 [38], VGG19-BN [38], Wide ResNet 101 [39], and RestNext 101 ($32 \times 8$d) [40] trained on ImageNet. We used VGG19 and VGG19-BN as pre-trained DNN models that have feature module structure. Further, we used Wide ResNet 101 and ResNext 101 ($32 \times 8$d) that has no feature module structure. To generate the Grad-CAM value, we must extract the target layer of the pre-trained model. Table 1 shows four pre-trained models on the ImageNet dataset along with their module and target layer name information.

#### 3) LOSS FUNCTION

As mentioned in Section IV, we have two loss parts; the Grad-CAM loss and CE loss are described as follows:

**Algorithm 1** Fooling Interpretable Pre-Trained DNN Classification Models and Explanation

**input :** Original Input Image ($oiI$),
Pre-trained Models ($pM$),
Top-right Position ($trP$),
Patch Size ($pS$),
Perturbation Ratio ($pR$)
**output:** Adversarial Image Patch ($aiP$),
Grad-CAM Image Perturbation ($giP$),
Mask Fooled Explanation ($mfE$),
Heatmap Fooled Explanation ($hfE$)

Preprocess original image
  $img \leftarrow resize(oiI, (224, 224, 3))$
  $img\_T \leftarrow tensor(img)$
Initialize perturbation on image
  $trP \leftarrow (150, 0)$
  $P \leftarrow torch\_Zero(img)$
  $P[0, trP + pS] \leftarrow pR$
  $interation\_Fool \leftarrow 1000$
  $\alpha \leftarrow 0.05$
Create adversarial image patch
**while** $i < interation\_Fool$ **do**
  $feature\_Adv, output\_Adv \leftarrow pM(img\_T)$
  $g\_Adv \leftarrow \sum\sum \frac{\partial output\_Adv}{\partial feature\_Adv}$
  $g\_Weight \leftarrow g\_Adv \times feature\_Adv$
  $G \leftarrow ReLU(\sum g\_Weight)$
  **if** $pS \leftarrow 64$ **then**
    $g\_Loss \leftarrow \sum(G[0:4, 0:4])/16$
  **else**
    $g\_Loss \leftarrow \sum(G[0:2, 0:2])/16$
  **end**
  $ce\_Loss \leftarrow crossLoss(output\_Adv, y\_target)$
  $total\_Loss \leftarrow G(g\_Loss + \alpha \times ce\_Loss)$
  $g\_total\_Loss \leftarrow G(total\_Loss, img\_T)$
  $N \leftarrow N - \alpha \times g\_total\_Loss$
  $aiP \leftarrow (1 - P) \times img + P \times N$
  $i \leftarrow i\%10$
**end**
Calculate Grad-CAM image perturbation
  $mask\_Adv \leftarrow G(aiP)$
  $hM \leftarrow applyColorMap(255 \times mask\_Adv)/255$
  $giP \leftarrow hM + aiP$
  $giP \leftarrow giP/max(giP)$
Explain Grad-CAM fooled results
  $mfE \leftarrow transpose(mask\_Adv)$
  $mfE \leftarrow \frac{mfE - min(mfE)}{max(mfE)}$
  $hfE \leftarrow applyColorMap(255 \times mask\_Adv)/255$
return $(aiP, giP, mfE, hfE)$

- The CE loss equation was used for optimizing loss of fooling pre-trained DNN models:

$$ce\_Loss = -\sum_{i}^{C} y\_target_i \times log(output\_Adv_i) \quad (11)$$

where $C$ is categorical output.

**TABLE 1. Pre-trained models with module information and corresponding target layer name.**

| Pre-trained Model | Module | Target Layer Name |
|---|---|---|
| VGG19 | features | 35 |
| VGG19-BN | features | 51 |
| Wide ResNet 101 | layer4 | 2 |
| ResNext 101 32×8d | layer4 | 2 |

(a) AIP at top-right (b) Attacked (c) AIP at top-right (d) Attacked

**FIGURE 6. Grad-CAM attacked results on AP at top-right corner of VGG19-BN with full perturbation.**

(a) AIP at top-right (b) Attacked (c) AIP at top-right (d) Attacked

**FIGURE 7. Grad-CAM attacked results on AP at top-right corner of Wide ResNet 101 with full perturbation.**

(a) AIP at top-right (b) Attacked (c) AIP at top-right (d) Attacked

**FIGURE 8. Grad-CAM attacked results on AP at top-right corner with size 32 × 32) of VGG19-BN with 20% perturbation.**

(a) AP at top-right (b) Attacked (c) AP at top-right (d) Attacked

**FIGURE 9. Grad-CAM attacked results on AP at top-right corner with size 32 × 32) of Wide ResNet 101 with 20% perturbation.**

- The Grad-CAM loss equation was used for optimizing loss of fooling interpretation model:

$$g\_total\_Loss = G(feature\_Adv + \alpha \times ce\_Loss) \quad (12)$$

where $G$ is Grad-CAM function calculation.

### B. EXPERIMENTS RESULTS

In this section, we perform four experiments. The first experiment was performed by creating an AIP at the top-right location with the size of 64 × 64 and full perturbation ratio on two pre-trained models: VGG19-BN (feature module) and Wide ResNet 101 (no feature module). The next experiment was performed by creating AIP at top-right location with the

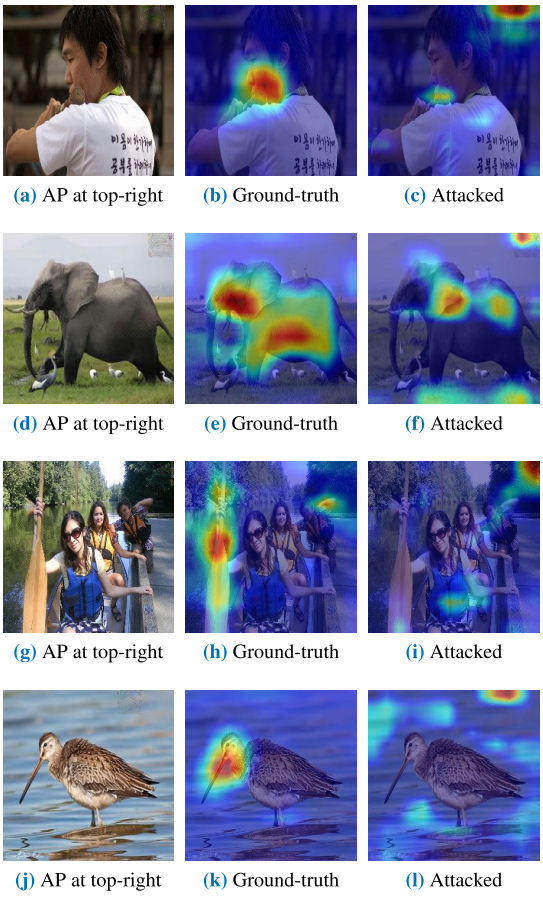



**FIGURE 10.** Grad-CAM attacked results using AIP at top-right corner with size 32 × 32 and 20% perturbation ratio on VGG19.

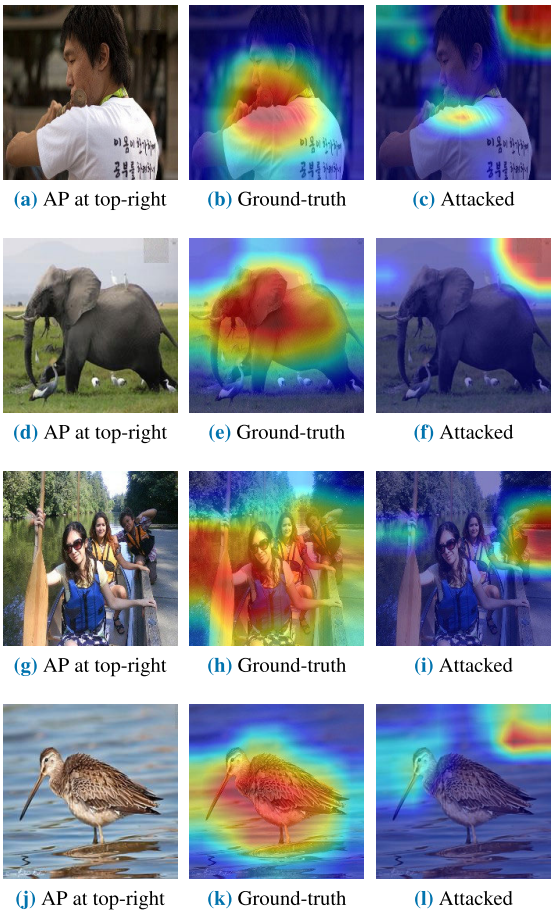



**FIGURE 11.** Grad-CAM attacked results using AIP at top-right corner with size 32 × 32 and 20% perturbation ratio on ResNext 101 (32 × 8d).

size of 32 × 32 and by reducing perturbation ratio by 20% to deceive the two pre-trained models, namely, VGG19-BN and Wide ResNet 101, along with interpretable Grad-CAM of these models. The third experiment validated our proposed method by deceiving two other representative pre-trained models: VGG19 (feature module) and Resnex101 32 × 8d (no feature module). Experiment 4 explains how the Grad-CAM attacked the results when four pre-trained models are used.

- *Experiment 1:* In this experiment, we used two trained models, namely, VGG19-BN and Wide ResNet 101, to attack both classification and interpretable results. We generated AIP at the top-right corner of the image with size 64 × 64 and full perturbation ratio 100%. Figures 6 and 7 show the fooling classified and interpreted Grad-CAM results on VGG19-BN and Wide ResNet 101, respectively.
  In summary, the interpretable Grad-CAM of two pre-trained models was completely deceived, and heatmaps were highlighted at the AIP target (Figs. 6b and 6d (VGG19- BN) and Figs. 7b and 7d (Wide Resnet 101)). However, these Grad-CAM results do not remain in the part of interpretable on the main object. Hence, we must adjust features of AIP that can fool Grad-CAM while keeping a part of Grad-CAM results on the main object.
- *Experiment 2:* In summary, the interpretable Grad-CAM of two pre-trained models are fully fooled with heatmaps that are highlighted at AIP target (Figs. 8b and 8d (VGG19-BN) and Figs. 9b and 9d (Wide Resnet 101)). However, these Grad-CAM results do not remain in the interpretation part of the main object. Hence, we must adjust the features of AIP that can fool Grad-CAM while keeping a part of Grad-CAM results on the main object. The main features of AIP that can fool Grad-CAM results are adversarial patch size and its perturbation ratio.
  In addition, we still attacked the Grad-CAM successfully even though we reduced the AIP size and perturbation ratio. It makes AIP less visible in the image, and it is not easy to discriminate against adversarial attacks to the human eye. However, the heatmap of Grad-CAM highlights a part of the main object. Additionally, the accuracy of the pre-trained DNN classification model is high even now.
- *Experiment 3:* To validate our proposed algorithm, we test the proposed attack model on two representative

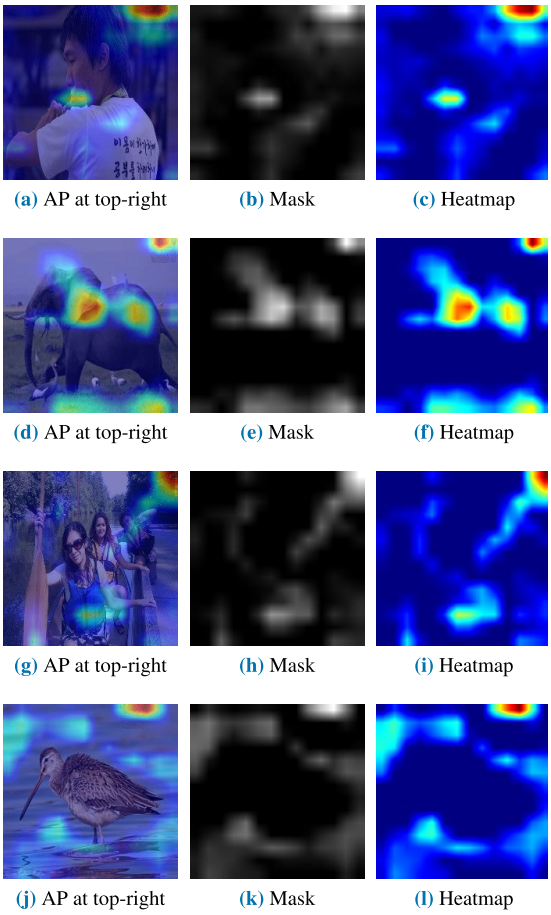



**FIGURE 12.** Explanation results on AP at top-right corner with size 32 × 32 and 20% perturbation on VGG19.

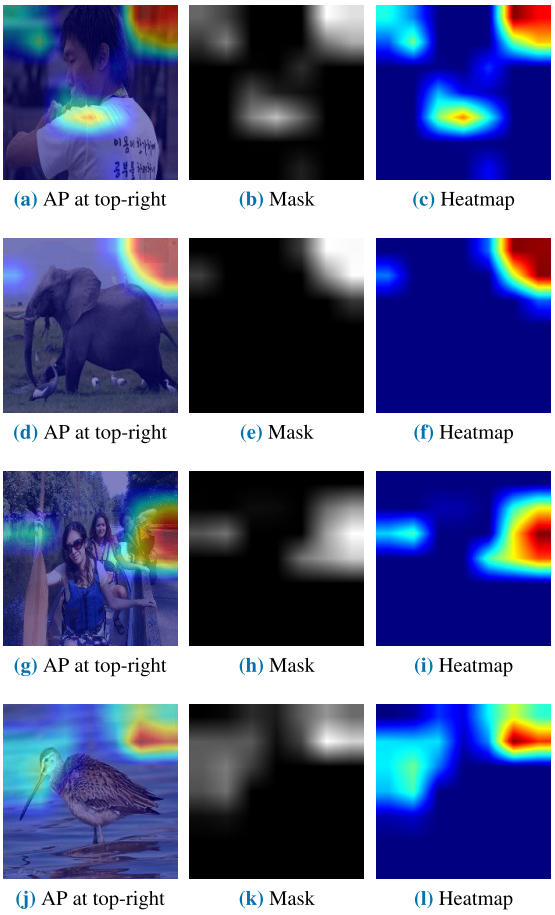



**FIGURE 13.** Explanation results on AP at top-right corner with size 32 × 32 and 20% perturbation on ResNext 101 (32 × 8d).

pre-trained models, including feature module and no feature module. For the feature module, we chose the VGG19 model as one family of VGG19-BN, and in the case of no feature module, we chose ResNext 101 (32 × 8d) model. Figures 10 and 11 show the Grad-CAM attacked with adversarial image at the top-right AIP size 32 × 32 and perturbation ratio 20% on two pre-trained models. In comparison to the ground truth of Grad-CAM results from two pre-trained classification models, the attacked Grad-CAM results show that both pre-trained models have been attacked with highlighted heatmap at the top-left corner with the AIP target.

In summary, the Grad-CAM attacked results show that the proposed fooling method can highlight the main object's part and top-right AP location target.

- *Experiment 4:* In this experiment, we explained our fooling Grad-CAM results based on calculating and visualizing the mask and heatmap of Grad-CAM attacked image based on XAI. Figures 12 and 13 present the explanation results using mask and heatmap corresponding to Grad-CAM attacked results with the AIP size of 32 × 32 and 20% perturbation ratio.

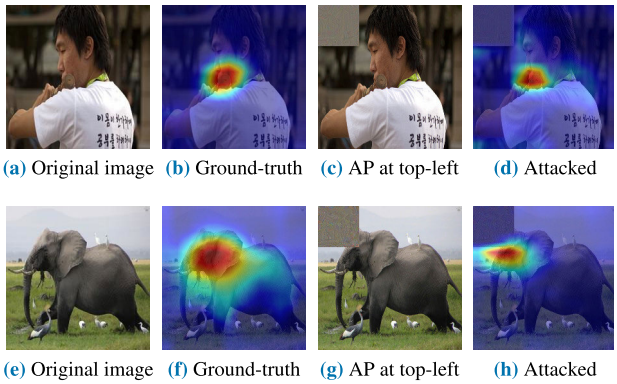
In summary, we obtained the fooling Grad-CAM results on two cases: more visible AIP (with the size of 64 × 64 and 100% perturbation ratio) and less visible AIP (with the size of 32 × 32 and 20% perturbation) on four pre-trained DNN models: VGG19-BN, Wide ResNet 101, VGG19, and ResNext 101 32 × 8d.

### C. DISCUSSION

#### 1) WHY NOT TOP-LEFT LOCALIZED ADVERSARIAL IMAGE PATCH

This section provides evidence to illustrate that AIP with top-left localization fails to fool several images from several pre-trained models when we applied previous adversarial attack methods [16]. For example, to pre-train VGG19-BN, Figure 14 shows the Grad-CAM attacked results are unsuccessful because the heatmap is only highlighted in the main object and not highlighted at the AIP target (shown in Figures 14d and 14h).

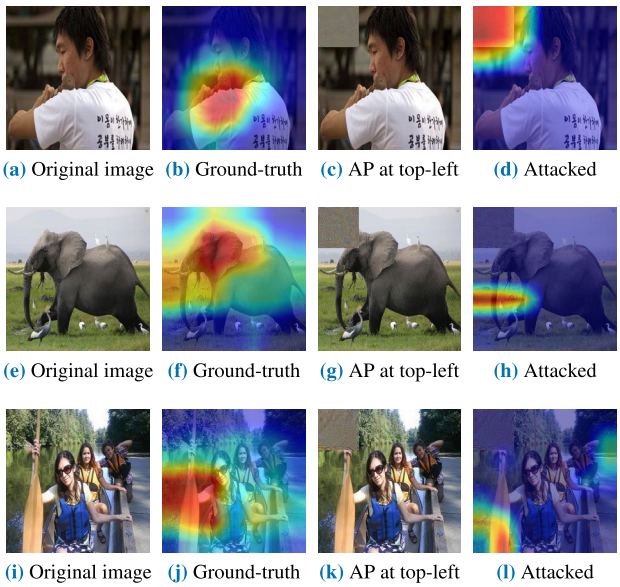
In addition, although the pre-trained model has feature modules, such as the VGG19-BN model, AIP cannot fool Grad-CAM of VGG19-BN at a top-left location with a patch size of 64 × 64, and a full perturbation ratio. For pre-trained models that have no feature modules such as Wide ResNet 101, the AP at the top-left location with 100% perturbation is also unsuccessful to fool Grad-CAM (shown in Figure 15).



(a) Original image (b) Ground-truth (c) AP at top-left (d) Attacked

(e) Original image (f) Ground-truth (g) AP at top-left (h) Attacked

**FIGURE 14. Examples of unsuccessful implementation for fooling Grad-CAM of pre-trained VGG19-BN model when applied [16].**

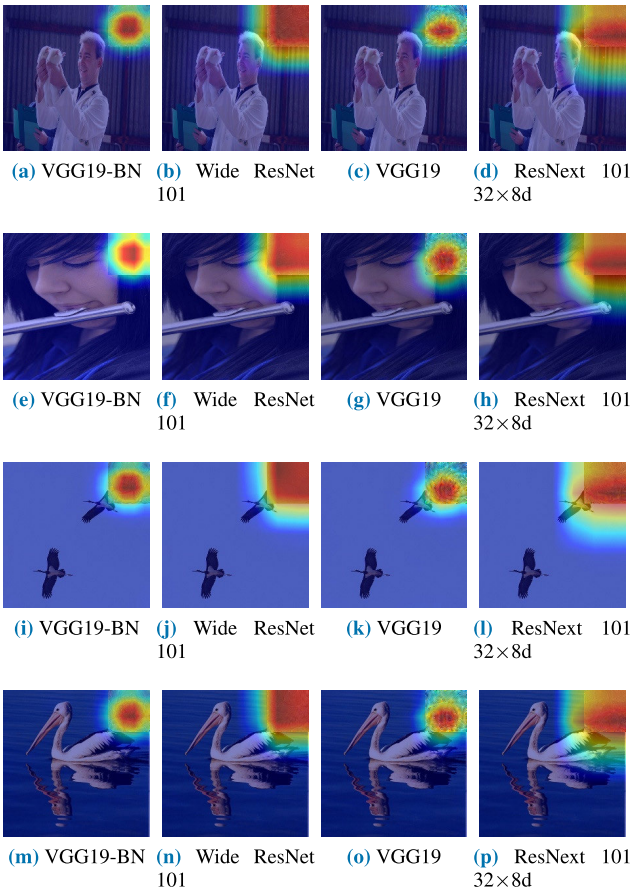


(a) Original image (b) Ground-truth (c) AP at top-left (d) Attacked

(e) Original image (f) Ground-truth (g) AP at top-left (h) Attacked

(i) Original image (j) Ground-truth (k) AP at top-left (l) Attacked

**FIGURE 15. Examples of unsuccessful implementation for fooling Grad-CAM of pre-trained Wide ResNet 101 model when applied [16].**



(a) VGG19-BN (b) Wide ResNet 101 (c) VGG19 (d) ResNext 101 32×8d

(e) VGG19-BN (f) Wide ResNet 101 (g) VGG19 (h) ResNext 101 32×8d

(i) VGG19-BN (j) Wide ResNet 101 (k) VGG19 (l) ResNext 101 32×8d

(m) VGG19-BN (n) Wide ResNet 101 (o) VGG19 (p) ResNext 101 32×8d

**FIGURE 16. Grad-CAM attacked of four trained models with 100% perturbation.**

Hence, to deceive Grad-CAM completely and make highlights at the AIP location only, we created an AIP with the size of 64 × 64 and 100% perturbation ratio. Figure 16 shows the Grad-CAM attacked results only at the AIP target of four pre-trained models.

In another case for fooling part of the Grad-CAM and keeping part of the correct Grad-CAM result from the pre-trained model explanation, we use AIP with the size of 32×32 and 20% perturbation ratio. Figure 17 shows the result of fooling part of four explained trained models.

In summary, depending on the purpose of attacking Grad-CAM interpretable, we can generate and adjust the AIP with certain size and perturbation ratio. If we create and use AIP with a size of 64 × 64 and a full perturbation ratio, AIP is more visible and easier to recognize APs with the naked eyes. However, the Grad-CAM attacked obtained a full-on AIP target. If we create and use AIP with a size of 32 × 32 and a perturbation ratio of 20%, AIP is less visible and hard to recognize APs with the naked eyes. However, the Grad-CAM attacked was obtained on both parts: a part of the remaining main object and a part of the AIP target.

### 2) EVALUATION PROPOSED METHOD VIA LOSS MEASUREMENT

To evaluate our proposed method, we measured two types of loss: Grad-CAM loss and CE loss. Figure 18 shows the Grad-CAM loss and CE loss of the proposed method on four pre-trained models in two cases. The first case is with the AIP size of 64 × 64 and a full perturbation ratio. In this case, the Grad-CAM loss (Fig. 18a) and CE loss (Fig. 18b) have their error value minimized. The second case is with the AIP size of 32 × 32 and reduced perturbation ratio by 20%. In this case, the Grad-CAM loss (Fig. 18c) and CE loss (Fig. 18d) are less accurate in fooling than the first case.

In conclusion, if we generate a top-right AIP with a size of 64 × 64 and full perturbation, attacking both DNN classification models and Grad-CAM interpretation is more accurate, but the drawback of AIP is more visible. In the case of creating a less visible AIP, we must adjust and reduce the patch size and perturbation to 32 × 32 and 20%, respectively.

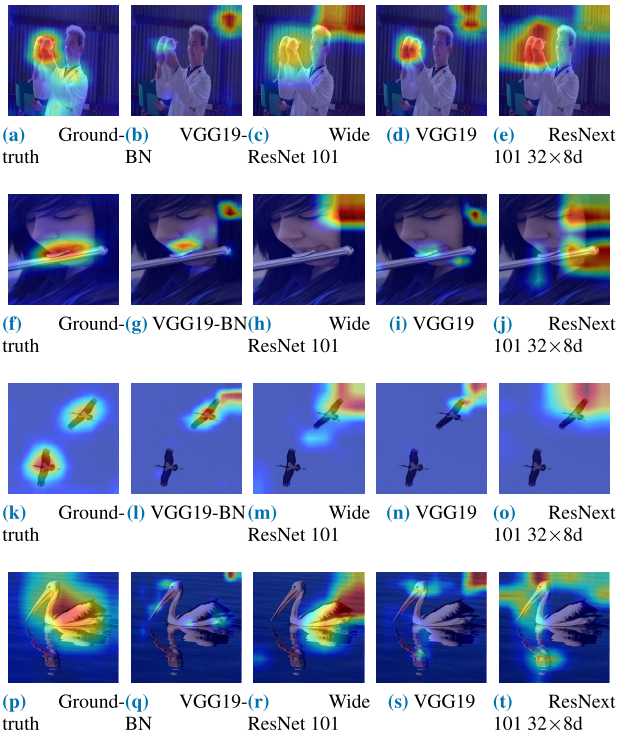
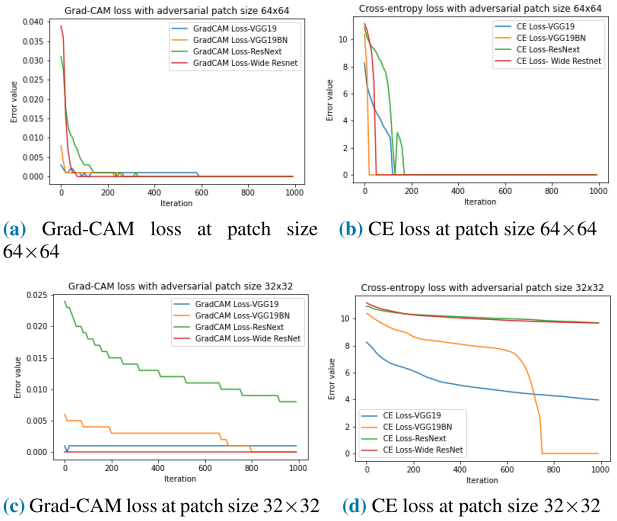



**FIGURE 17.** Grad-CAM attacked of four trained models with 20% perturbation.




(a) Grad-CAM loss at patch size 64×64
(b) CE loss at patch size 64×64
(c) Grad-CAM loss at patch size 32×32
(d) CE loss at patch size 32×32

**FIGURE 18.** Grad-CAM loss and CE loss with adversarial patch sizes 64 × 64 and 32 × 32.

However, we obtained less FSR (fooling success rate) on both pre-trained classifications and Grad-CAM interpretation.

## VI. CONCLUSION

In this paper, we proposed an adversarial algorithm to deceive both pre-trained classifications and Grad-CAM interpretation. The obtained results show that it is possible to learn visible and less visible AIP covering only 3% and 1.5% of pixels in an image. Further, localized adversarial patches at the top-right, along with different perturbation ratios, cause misclassification with high fooling success rates. Therefore, we introduce adversarial patches (small areas (3% and 1.5%) with restricted perturbation ratios of 100% and 20% respectively), which fool both the DNN classification models and their explainable model by the Grad-CAM algorithm. In summary, we successfully designed two cases attacking with different adversarial patches. The first AIP with the size of 64 × 64 and full perturbation ratio obtains the highlighted interpretation at the top-right, and the AIP is localized with a high fooling accuracy rate. In this manner, the Grad-CAM interpretation algorithm highlights the evident cause of the wrong prediction corresponding to misclassification results. In the second AIP with the size of 32 × 32 and perturbation ratio of 20%, the highlighted interpretation is obtained not only at the top-right AIP but also as part of the highlight is kept for the prediction with less fooling accuracy rate. However, this case provides a less visible AIP attached to the image. Moreover, either CE loss or the Grad-CAM loss of the second AIP case is more than that our attack method affects various settings of localized AIP at the top-right based on different sizes and perturbation ratios for different goals in visible or invisible AIP to the original image. In future work, we could consider applying several approaches in defensive system (e.g, WGAN) to build a robust defend method, which against adversarial learning on interpretable models.

## REFERENCES

[1] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1701–1708.
[2] C. Szegedy, W. Zaremba, I. Sutskever, and J. Bruna, "Intriguing properties of neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2013, pp. 2–11.
[3] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–11.
[4] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2574–2582.
[5] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, Mar. 2016, pp. 372–387.
[6] J. Su, D. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 828–841, Oct. 2019, doi: 10.1109/TEVC.2019.2890858.
[7] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–8.
[8] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.
[9] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.
[10] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller, "How to explain individual classification decisions," *J. Mach. Learn. Res.*, vol. 11, no. 6, pp. 1803–1831, 2010.
[11] G. Fidel, R. Bitton, and A. Shabtai, "When explainability meets adversarial learning: Detecting adversarial examples using SHAP signatures," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8, doi: 10.1109/IJCNN48605.2020.9207637.
[12] A. Ignatiev, N. Narodytska, and J. Marques-Silva, "On relating explanations and adversarial examples," in *Proc. 33rd Conf. Neural Inf. Process. Syst. (NeurIPS)*, Vancouver, BC, Canada, 2019, pp. 15883–15893.

[13] L. Rieger and L. K. Hansen, "A simple defense against adversarial attacks on heatmap explanations," in *Proc. Workshop Hum. Interpretability Mach. Learn. (WHI)*, 2020, pp. 1–22.
[14] A. S. Ros and F. Doshi-Velez, "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients," in *Proc. Assoc. Advancement Artif. Intell.*, 2017, pp. 1–10.
[15] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3449–3457, doi: 10.1109/ICCV.2017.371.
[16] A. Subramanya, V. Pillai, and H. Pirsiavash, "Fooling network interpretation in image classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2020–2029, doi: 10.1109/ICCV.2019.00211.
[17] J. Heo, S. Joo, and T. Moon, "Fooling neural network interpretations via adversarial model manipulation," in *Proc. 33rd Conf. Neural Inf. Process. Syst. (NeurIPS)*, Vancouver, BC, Canada, 2019, pp. 2925–2936.
[18] D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju, "Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods," in *Proc. AAAI/ACM Conf. AI, Ethics, Soc.*, Feb. 2020, pp. 180–186, doi: 10.1145/3375627.3375830.
[19] J. Adebayo, J. Gilmer, M. Muelly, I.J, Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," in *Proc. NeurIPS*, 2018, pp. 9525–9536.
[20] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 1528–1540.
[21] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, Apr. 2017, pp. 506–519, doi: 10.1145/3052973.3053009.
[22] K. Yang, J. Liu, C. Zhang, and Y. Fang, "Adversarial examples against the deep learning based network intrusion detection systems," in *Proc. MILCOM IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2018, pp. 559–564, doi: 10.1109/MILCOM.2018.8599759.
[23] X. Ding, S. Zhang, M. Song, X. Ding, and F. Li, "Toward invisible adversarial examples against DNN-based privacy leakage for Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 802–812, Jan. 2021, doi: 10.1109/JIOT.2020.3008232.
[24] A. Bahramali, M. Nasr, A. Houmansadr, D. Goeckel, and D. Towsley, "Robust adversarial attacks against DNN-based wireless communication systems," in *Proc. ICML Workshop Hum. Interpretability Mach. Learn. (WHI)*, 2021, pp. 1–12. [Online]. Available: https://arxiv.org/pdf/2102.00918v1.pdf
[25] X. Ma, Y. Niu, L. Gu, Y. Wang, Y. Zhao, J. Bailey, and F. Lu, "Understanding adversarial attacks on deep learning based medical image analysis systems," *Pattern Recognit.*, vol. 110, Feb. 2021, Art. no. 107332.
[26] H. Hirano, A. Minagi, and K. Takemoto, "Universal adversarial attacks on deep neural networks for medical image classification," *BMC Med. Imag.*, vol. 21, no. 1, pp. 1–13, Dec. 2021, doi: 10.1186/s12880-020-00530-y.
[27] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," in *Proc. NIPS Workshop*, 2017, pp. 1–6. [Online]. Available: https://arxiv.org/pdf/1712.09665.pdfs
[28] X. Liu, H. Yang, Z. Liu, L. Song, H. Li, and Y. Chen, "DPatch: An adversarial patch attack on object detectors," in *Proc. AAAI Workshop Artif. Intell. Saf. (SafeAI)*, 2019, pp. 1–8. [Online]. Available: https://arxiv.org/pdf/1806.02299.pdf
[29] Y. Zhao, H. Yan, and X. Wei, "Object hider: Adversarial patch attack against object detectors," 2020, *arXiv:2010.14974*. [Online]. Available: http://arxiv.org/abs/2010.14974
[30] S. Thys, W. V. Ranst, and T. Goedeme, "Fooling automated surveillance cameras: Adversarial patches to attack person detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 49–55.
[31] Y. Mirsky, "IPatch: A remote adversarial patch," 2021, *arXiv:2105.00113*. [Online]. Available: http://arxiv.org/abs/2105.00113
[32] S. Chen, D. Shi, M. Sadiq, and X. Cheng, "Image denoising with generative adversarial networks and its application to cell image enhancement," *IEEE Access*, vol. 8, pp. 82819–82831, 2020, doi: 10.1109/ACCESS.2020.2988284.
[33] L. Hui, Z. Bo, H. Linquan, G. Jiabao, and L. Yifan, "FoolChecker: A platform to evaluate the robustness of images against adversarial attacks," *Neurocomputing*, vol. 412, pp. 216–225, Oct. 2020, doi: 10.1016/j.neucom.2020.05.062.
[34] H. Lin, Y. Wo, Y. Wu, K. Meng, and G. Han, "Robust source camera identification against adversarial attacks," *Comput. Secur.*, vol. 100, Jan. 2021, Art. no. 102079, doi: 10.1016/j.cose.2020.102079.
[35] N. Veeraiah, O. I. Khalaf, C. V. P. R. Prasad, Y. Alotaibi, A. Alsufyani, S. A. Alghamdi, and N. Alsufyani, "Trust aware secure energy efficient hybrid protocol for MANET," *IEEE Access*, vol. 9, pp. 120996–121005, 2021, doi: 10.1109/ACCESS.2021.3108807.
[36] G. Suryanarayana, K. Chandran, O. I. Khalaf, Y. Alotaibi, A. Alsufyani, and S. A. Alghamdi, "Accurate magnetic resonance image super-resolution using deep networks and Gaussian filtering in the stationary wavelet domain," *IEEE Access*, vol. 9, pp. 71406–71417, 2021, doi: 10.1109/ACCESS.2021.3077611.
[37] R. Olga, D. Jia, S. Hao, K. Jonathan, S. Sanjeev, M. Sean, H. Zhiheng, K. Andrej, K. Aditya, B. Michael, C. B. Alexander, and F.-F. Li, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[38] S. Karen and Z. Andrew, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015, pp. 1–14.
[39] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 1–15.
[40] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1492–1500.

**THI-THU-HUONG LE** received the bachelor's degree from Hung Yen University of Technology and Education (HYUTE), Vietnam, the master's degree from Hanoi University of Science and Technology (HUST), Vietnam, and the Ph.D. degree from Pusan National University (PNU), South Korea, in 2020. She is currently a Postdoctoral Researcher with IoT Research Center, PNU. She has seven years of experience as a Lecturer with HYUTE. She has participated in machine learning projects, such as NILM, IDS, industry 4.0, and AI security. Her research interests include machine learning, deep learning, AI security, and signal processing.

**HYOEUN KANG** received the bachelor's degree from Pusan National University (PNU), South Korea, 2017, where she is currently pursuing the Ph.D. degree. She has participated in machine learning projects, such as NILM and intelligent gas turbine monitoring system. Her research interests include machine learning, deep learning, and digital twin.

**HOWON KIM** (Member, IEEE) received the bachelor's degree from KyungPook National University (KNU) and the Ph.D. degree from Pohang University of Science and Technology (POSTECH). He is currently a Professor with the Department of Electrical and Computer Engineering, the Chief of the Energy IoT ITRC, and the Chief of ISEC, Pusan National University. Before that, he worked with the Electronics and Telecommunications Research Institute (ETRI), as a Team Leader for ten years, since December 1998. He was the Visiting Chair of the Communication Security Group (COSY), Ruhr-University Bochum, Germany, as a Postdoctoral Researcher, from July 2003 to June 2004.