Partial Retraining Substitute Model for Query-Limited Black-Box Attacks

Hosung Park 1, Gwonsang Ryu 2 and Daeson Choi 3

1 Department of Cyber Security and Police, Busan University of Foreign Studies, Busan 46234, Korea; hspark0865@bufs.ac.kr
2 Department of Software Convergence, Graduate School of Soongsil University, Seoul 06978, Korea; gsryu@soongsil.ac.kr
3 Department of Software, Soongsil University, Seoul 06978, Korea
* Correspondence: sunchoi@ssu.ac.kr

Received: 16 September 2020; Accepted: 8 October 2020; Published: 14 October 2020

Abstract: Black-box attacks against deep neural network (DNN) classifiers are receiving increasing attention because they represent a more practical approach in the real world than white box attacks. In black-box environments, adversaries have limited knowledge regarding the target model. This makes it difficult to estimate gradients for crafting adversarial examples, such that powerful white-box algorithms cannot be directly applied to black-box attacks. Therefore, a well-known black-box attack strategy creates local DNNs, called substitute models, to emulate the target model. The adversaries then craft adversarial examples using the substitute models instead of the unknown target model. The substitute models repeat the query process and are trained by observing labels from the target model’s responses to queries. However, emulating a target model usually requires numerous queries because new DNNs are trained from the beginning. In this study, we propose a new training method for substitute models to minimize the number of queries. We consider the number of queries as an important factor for practical black-box attacks because real-world systems often restrict queries for security and financial purposes. To decrease the number of queries, the proposed method does not emulate the entire target model and only adjusts the partial classification boundary based on a current attack. Furthermore, it does not use queries in the pre-training phase and creates queries only in the retraining phase. The experimental results indicate that the proposed method is effective in terms of the number of queries and attack success ratio against MNIST, VGGFace2, and ImageNet classifiers in query-limited black-box environments. Further, we demonstrate a black-box attack against a commercial classifier, Google AutoML Vision.

Keywords: adversarial examples; black-box attack; substitute model; deep neural networks

1. Introduction

Deep neural network (DNN) classifiers have made significant progress in many domains such as image classification [1,2], voice recognition [3,4], malware detection [5,6], and natural language processing [7]. Despite their great success, recent studies have demonstrated that DNNs are vulnerable to well-designed input samples called adversarial examples [8,9]. Legitimate inputs are altered by adding small perturbations to force a trained classifier to misclassify them, while humans cannot perceive the alteration. In other words, the purpose of these attacks is to maximize misclassification while minimizing the perturbations. The adversarial examples represent a serious threat to DNN classifiers. For example, manipulated traffic signs can confuse autonomous vehicles [10,11] and adversarial voices can deceive automatic voice recognition models [12,13] such as Apple’s Siri and Amazon’s Alexa.
Early studies of adversarial examples use white-box environments, in which the adversaries can access the target model without restrictions or own the same model. They therefore focus on the attack success ratio and have little interest in how many times they must access the target model. In white-box attacks [8,9], adversaries have full knowledge regarding the target model, such as network architecture, training data, parameters, and classification results (predicted labels and probabilities). With this information, white-box attacks can calculate the gradients of the target model and craft precise adversarial examples based on the gradients. The success rates in this type of attack are almost 100%. Some studies [14,15] have demonstrated that defending DNNs from white-box attacks is extremely difficult.

In contrast, black-box attacks involve limited knowledge of the target model. In the real world, DNN classification services normally do not provide information regarding their systems to prevent security threats or for business reasons. They wish to prevent adversaries and competitors from obtaining the details of their models. The adversaries can access the target model only through exposed service interfaces, i.e., using queries. Therefore, white-box attack algorithms based on gradients cannot be directly applied, and the number of queries becomes an important issue.

A well-known black-box attack strategy creates a local DNN, called a substitute model [16,17], to emulate the target model. The adversaries then craft adversarial examples using the substitute model instead of the unknown target model. The substitute model is trained using a labeled dataset obtained by repeated queries made to the target model. For successful attacks, previous studies focus on generating a dataset composed of proper samples to efficiently train substitute models. However, emulating the target model usually requires a massive number of queries because training a new DNN from the beginning normally requires numerous training data. Moreover, these studies have to emulate the entire target model even for one-sample attacks. They attempt to build a substitute model as a master key for the target model with which adversaries can craft adversarial examples from any samples. In many cases, adversaries wish to perform one attack (from one class to another class) instead of multiple attacks (from classes to classes). For example, an adversary may attempt to deceive a face authentication system by pretending to be a legitimate user. In autonomous vehicle systems, misclassifying a stop sign as a speed limit sign is enough to lead to a traffic accident. Therefore, we argue that black-box attacks can be classified into one-sample attacks and any-sample attacks. Existing studies are suitable for any-sample attacks but are inefficient for one-sample attacks in terms of the number of queries required.

In this study, we propose a novel training method applied to the substitute model for one-sample attacks to minimize the number of queries. We consider the number of queries as one of the important factors for practical black-box attacks as well as the attack success ratio. Real-world systems often restrict the number of queries to prevent security threats and, at a minimum, many queries lead to waste of resources, such as time and finances, even if there is no limit placed on them. Google web services limits the number of queries as 25,000 per day and 10 per s and users have to pay for more queries. The Clarifai API costs a few dollars per 1000 queries after the first 2500 predictions. Amazon web services also provide API Gateway service that developers can restrict queries in many ways such as rate-limit, IP-level, user-level, and application-level throttling. Therefore, our purpose is to achieve a similar attack success ratio to that of the existing substitute-model attack using fewer queries. In black-box attacks with substitute models, the attack success ratio is proportional to the number of queries, because the more queries made implies more labeled training data. Therefore, substitute models are more likely to accurately represent the target model. Based on this characteristic, our goal is also represented by achieving a higher attack success ratio with the same number of queries.

The proposed method does not emulate the entire target model and adjusts only the partial classification boundary to focus on a current attack. It also does not generate queries from the beginning. That is, training the substitute model is divided into pre-training and retraining phases, and queries are used only in the retraining phase. We assume that the adversaries are aware of the type of target classifier, such as face, animal, or product classifiers, because each service normally has a specific purpose. In the
pre-training phase, the substitute model is trained in the usual manner according to the type of target classifier. Even though the pre-trained substitute model is different from the target model, it helps to reduce the number of queries needed to achieve a successful attack. This concept of transferability \cite{18,19} is based on the idea that an adversarial example modified for a single type of model is effective for other models that classify the same type of data. In this phase, adversaries do not query the target model, and the substitute model is trained using other data, which are different from the training data used to develop the target model. After pre-training, the proposed method crafts an adversarial example using the substitute model and sends it to the target model as a query. The substitute model is retrained using the classification results of the adversarial examples instead of benign data, which involves partial retraining only focusing on the current attack sample. Crafting adversarial examples and retraining the substitute model are iterated to increase the attack success ratio.

Our contributions are as follows.

- **Black-box attack with substitute model for one sample**: We argue that black-box attacks are classified into one-sample attacks and any-sample attacks according to the purpose of the attack. Previous studies focus on any-sample attacks, and thus, are inefficient for one-sample attacks. We propose a partial retraining method applied to the substitute model, which is suitable for one-sample attacks.

- **Label-only black-box environments**: We assume strong black-box environments in which adversaries can only observe labels of samples via the results obtained through queries. Some black-box attacks or gray-box attacks use more information, such as training data and probabilities of classifications along with labels. In particular, black-box attacks relying on gradient estimation definitely require probabilities of all classes (details in Section 2.1). The label-only setting, we consider, is a more severe environment for black-box attacks.

- **Query-efficient black-box attacks**: To decrease the number of queries, the proposed method does not emulate the entire target model and adjusts only the partial classification boundary related to a current attack sample to retrain the substitute model. Mini-batch retraining using a pool of adversarial examples is also useful for query reduction. We demonstrate that our query-efficient method achieves a higher attack success ratio in query-limited environments.

We evaluate the proposed method through experiments based on classifiers using the MNIST \cite{20}, VGGFace2 \cite{21}, and ImageNet \cite{22} datasets. The simulation results indicate that the proposed method achieves a higher attack success ratio in comparison with the previous substitute-model attack \cite{16} in query-limited environments based on the same upper bound for the number of queries. We further use our method to perform a targeted attack on Google AutoML Vision \cite{23}, demonstrating the applicability of the attack on commercial systems.

2. Related Work

Szegedy et al. \cite{8} demonstrated that DNN-based classifiers are vulnerable against adversarial examples. Earlier, many studies \cite{9,18,24–27} proposed attack algorithms generating adversarial examples in white-box environments. Black-box attacks are more challenging but also represent a more practical approach, as they reflect real world environments. They only use queries to obtain information from the target model, and according to each method, there is a difference in the definitions of responses, such as top-1 label only, top-k labels, and classification probabilities. Several studies have demonstrated domain-specific black-box attacks against certain systems, such as voice recognition \cite{12,13}, malware detection \cite{28,29}, and object detection systems \cite{30}. In this section, we focus on the existing black-box attacks against image classification models, which use either gradient estimation or substitute model techniques.
2.1. Gradient Estimation

Chen et al. [31] proposed a gradient estimation method for black-box attacks. They apply zeroth order optimization via pixel-by-pixel finite differences to estimate the gradient, and then construct adversarial examples using white-box attack algorithms based on the estimated gradient. According to their analysis, numerous queries are required to naively estimate the gradients with respect to all pixels; thus, they designed an iterative coordinate descent procedure for query reduction. Subsequently, they successfully attacked MNIST, CIFAR-10, and ImageNet classifiers. Narodytska & Kasiviswanathan [32] proposed a gradient estimation attack that constructs adversarial examples by perturbing a small fraction of pixels. They observed that adding a perturbation to a small set of pixels is often sufficient for successful attacks and selected the perturbations using a greedy local-search. They performed targeted black-box attacks on MNIST, CIFAR10, SVHN, STL10, and ImageNet classifiers.

These gradient estimation methods have clear advantages, i.e., fewer queries and precise attacks, when compared to substitute model techniques. However, they must use probabilities for all classes, implying non-label-only black-box environments that we refer to as a gray-box. Image classification services will provide top-1 labels, but additional information such as top-k labels and all probabilities of classes are not mandatory. If target services do not offer additional information (i.e., label-only black-box environments), these gradient estimation techniques are unusable. Ilyas et al. [33] introduced gradient estimation techniques applicable not only to gray-box environments but also to label-only black-box environments (providing top-k labels precisely). In a label-only setting, instead of classification probabilities, they estimate proxy scores using ranks of various adversarial examples obtained through random perturbations. However, this solution requires numerous queries ($2.7 \times 10^6$) to achieve a 90% attack success ratio on the ImageNet classifier, because the proxy scores are relatively imprecise and depend on the random perturbations.

2.2. Substitute Model

Another approach for black-box attacks is to create a substitute model, which is trained using a labeled dataset obtained through queries to the target model. The basic idea is to emulate the target model, and then, craft adversarial examples by applying white-box attack algorithms to the substitute model. Papernot et al. [16,17] successfully used this method to perform untargeted attacks on a commercial classifier, Google Cloud Prediction API (now Google AutoML Vision and Vision API), and a smaller dataset, MNIST. Liu et al. [19] demonstrated that targeted attacks are more difficult, especially on large-scale datasets such as ImageNet. They used an ensemble of multiple substitute models to overcome these issues and succeeded in attacks on the Clarifai API.

Unlike gradient estimation, attacks with substitute models assume that an adversary only receives a label corresponding to a sample image, i.e., a query, matching our definition of black-box environments. To maximize the number of successful attacks, they attempt to train the model to be as similar to the target model as possible. However, emulating the target model requires many queries. Training a new DNN from the beginning normally requires numerous training data, and furthermore, network architectures of recent DNNs are large scale and complex. Note that, unlike other methods, Liu et al.’s substitute model attack does not require any queries to the target model, but only 18% of attacks succeeded against Clarifai API using the ImageNet dataset.

3. Proposed Method

The environment we use is summarized as follows: targeted attack, one-sample attack, black-box environment (using only top-1 labels), and query-limited environment. The proposed method involves crafting adversarial examples and retraining a substitute model, as shown in Figure 1. The target model is an image classifier and all information is hidden in the black-box. The attacker can send a query (an image sample) and receive a classification result via a normal service interface. The substitute model is a local DNN that is pre-trained according to the type of target classifier. Note that the
details of the substitute model, such as network architecture, parameters, training data, and leaning
algorithm, are different from the target model because the attacker cannot access this information in
black-box attacks. The proposed method crafts an adversarial example using the substitute model and
sends it to the target model as a query and receives the classification result, that is, a label. If the label is
not the target class (if the attack fails), the substitute model is retrained using the adversarial example
labeled by the target model. This process is repeated until the attack is successful.

Figure 1. Overall process of the proposed method.

3.1. Pre-Training the Substitute Model

Before crafting adversarial examples, the proposed method prepares an initial substitute model.
We assume that adversaries are aware of the type of target classifier. Based on the type, the adversaries
can estimate the network architecture and training data of the substitute model. The substitute model
is trained using prepared training data. The initial substitute model is different from the target model;
however, there is a slight chance of deceiving the target model with the adversarial examples crafted
using the initial substitute model, as demonstrated by the concept of transferability [18,19]. Therefore,
retraining the substitute model is employed to increase the possibility of transferability. Note that
queries are not generated in this pre-training phase.

A simple but efficient idea is to exploit well-known pre-trained models in open source communities.
For example, Keras Applications [34] provide various well-known image classification models,
including ResNet [35], VGG [1], and Inception [36] trained with the ImageNet dataset. According to our
experimental results, a similar network architecture to that of the target model requires fewer queries for
black-box attacks. If the adversaries can discover the network architecture of the target model, a powerful
black-box attack method is possible. However, this is somewhat unlikely in a real-world scenario.

3.2. Crafting Adversarial Examples

After pre-training, the substitute model is used to craft adversarial examples. This involves
a white-box attack and exploits two previous well-known algorithms: Iterative fast gradient sign
method (I-FGSM) [25] and Carlini and Wagner (C&W)'s attack [24]. These two algorithms have been
extensively used in previous black-box attacks because they have distinct advantages respectively.
I-FGSM is simple and fastest algorithm and C&W provides highest attack success ratio among
white-box attack algorithms.

FGSM computes the gradient only once to generate an adversarial example. For a targeted attack
applied to a specific class, FGSM craft adversarial example as

$$x' \leftarrow x - \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y'))$$

(1)
where $x$ is the original input data, $x'$ is an adversarial example, $y'$ is the target class, $J$ is a loss function, and $\epsilon \in [0, 1]$ is a constant value used to constrain the perturbation.

I-FGSM extends the FGSM algorithm to craft more precise adversarial examples using iterations. I-FGSM generates a targeted adversarial example as

$$x'_0 = x$$

$$x'_{n+1} = \text{Clip}_{x,\epsilon} \{ x'_n - a \cdot \text{sign}(\nabla_{x'_n} J(\theta, x'_n, y')) \}$$

where $a$ is a small constant value. We set $a = \frac{\epsilon}{N}$ where $N$ is an upper bound of iteration. I-FGSM avoids making large changes to each pixel during each iteration by clipping pixel values. In other words, $\text{Clip}_{x,\epsilon}$ limits the change of the generated adversarial image in each iteration so that the result will be in the range of $[\max(0, x - \epsilon), \min(1, x + \epsilon)]$. Note that, typically, each pixel is normalized to be within $[0, 1]$.

C&W's attack is a high-performance targeted attack and can defeat defensive distillation [14]. They define generating a targeted adversarial example as optimization problem with a given input $x$ to determine perturbation $\delta$ as follows.

$$\text{minimize} \quad \| \delta \|_p + c \cdot f (x + \delta)$$

$$\text{such that} \quad x + \delta \in [0,1]^n$$

where $\| \delta \|_p$ is $l_p$ norm, $c$ is a constant to balance constraints, and an adversarial example $x' = x + \delta$. They also define seven objective functions, but one of the most effective functions, $f$, is obtained as

$$f(x') = \max (\max(Z(x'_i) : i \neq t) - Z(x'_t), -k)$$

where $Z$ denotes the softmax function (so $Z(x)$ are the logits), $k$ is a constant for controlling the confidence, and $t$ is a target class. Instead of $\delta$, they apply a new variable $w$ by $\delta = \frac{1}{2}(\tanh(w) + 1) - x_i$ and optimize over $w$ for iteration and smoothing of clipped gradient descent. Among three possible distance metrics, most effective $L_2$ norm is applied, and then, the optimization problem can be described by

$$\text{minimize} \quad \| \frac{1}{2}(\tanh(w) + 1) - x \|_2 + c \cdot f(\frac{1}{2}(\tanh(w) + 1))$$

3.3. Retraining the Substitute Model

The proposed method sends a query containing the crafted adversarial example to the target model. After classification by the target model, the proposed method receives the result, i.e., a label, in response to the query. If the attack fails, the labeled example becomes a new training sample for the substitute model. We then have to change the classification boundary before the next attack. Note that the substitute model classifies the sample as the target class because it is crafted based on the substitute model. In the retraining process, the proposed method iteratively inputs the sample into the substitute model until it classifies the sample as the received label. Because the substitute model is already trained, this retraining process may require a certain number of iterations. In this study, we ignore the training efficiency of the substitute model. For practical black-box attacks, we focus only on the number of queries and attack success ratio. After retraining, the proposed method crafts another adversarial example based on the retrained substitute model. Subsequently, crafting of an adversarial example and retraining the substitute model are repeated until the attack is successful or the number of iterations reaches a pre-set threshold.

Figure 2 indicates the differences between the previous substitute-model attack and proposed method.

The previous method attempts to emulate the entire target model. All boundaries of the substitute model (dotted lines) are moved toward the boundaries of the target model (solid lines). If the previous
method sends numerous queries to the target model, the boundaries may become similar to the boundaries of the target model. In contrast, the proposed method modifies only the partial boundary corresponding to the current attack based on one sample. In the current attack, the original is labeled as class 2 and the target is class 1. When the boundary is different from that of the target model and the adversarial example (a red dot) does not cross the solid line, the attack fails. After retraining the substitute model using the failure sample, the partial boundary is changed. This process is repeated until the red dot finally crosses the solid line.

![Figure 2. Boundary changes of substitute model](image)

Figure 2. Boundary changes of substitute model (a) previous training method and (b) proposed method.

Note that the substitute model generated by the previous method can be used for any samples and targeted attacks. In Figure 2, using the original sample (class 2), the previous method can generate adversarial examples targeting classes 0, 1, and 3, whereas the proposed method can target only class 1. However, it is assumed that in a real-world system the possible number of queries is limited. As previously mentioned, the previous method requires a large number of queries to train the substitute model. Under a limited-query condition, the attack success ratio of the proposed method is higher than that of the previous method, as demonstrated by the experiments conducted and presented later in this paper.

Algorithm 1 presents the pseudocode for partial retraining of the substitute model. This process repeats until the query limit (line 1) is reached, or when the attack succeeds (line 4–5). In each iteration, an adversarial example, $x'$, is crafted by white-box attack methods applied to the pre-trained substitute model, $F()$ (line 2). The adversary sends a query containing the adversarial example to the target model, $\tilde{F}()$, and receives a classification result, label $y$ (line 3). If the attack fails, the adversarial example is relabeled with the received label, $y$, (line 7) and the substitute model is retrained using the relabeled adversarial example (line 9). For efficient training, we construct a retraining data set, $S$, with adversarial examples from every iteration (line 8), and feed these to the substitute model as a mini-batch. The substitute model is retrained until every adversarial example is classified as a received (valid) label from the target model.
Algorithm 1: Partial retraining of substitute model.

**Input:** image sample $x$,  
  target class $y_t$,  
  target model $\tilde{F}(\cdot)$,  
  pre-trained substitute model $F(\cdot)$,  
  maximum number of queries $max_q$

**Output:** adversarial example $x'$

1. for $i = 1$ to $max_q$ do
   2. # white-box attack with substitute model
   3. $x' : F(x') = y_t$, $x' \leftarrow x + \delta$
   4. # query to target model $\rightarrow$ classification label $y$
   5. $y \leftarrow \tilde{F}(x')$
   6. if $y = y_t$ then
     7. break
   8. end if
   9. label($x'$) $\leftarrow y$
   10. add $x'$ to $S$
   11. # training substitute model on set $S$
   12. train($F(\cdot)$, $S$)
13. end for
14. return $x'$

4. Experimental Evaluation

4.1. Experimental Settings

The performance of the proposed method is compared with the results of previous substitute-model attack methods [17]. The experimental settings are crucial for simulating black-box environments and analyzing the performance of the proposed method.

4.1.1. Dataset

For the various experimental environments, we exploit three datasets: MNIST, VGGFace2, and ImageNet.

MNIST [20] is a handwritten digit dataset used to train a DNN. It comprises 60,000 training and 10,000 test images, which are classified into 10 classes (in digits from 0 to 9). The task associated with the dataset is to identify the digit corresponding to each image. Each $28 \times 28$ grayscale sample is encoded as a vector with pixel intensities in the interval $[0, 1]$ and obtained by reading the image pixel matrix row-wise.

VGGFace2 [21] is a face dataset containing 3.31 million images of 9131 subjects. This dataset is widely used for face recognition and authentication via machine learning. Blackbox attacks on VGGFace2 dataset are more complicated than on MNIST because each sample is larger and presented in color ($224 \times 224$ pixels $\times$ 3 RGB). We selected 10 subjects representing more than 500 images.

ImageNet [22] is large-scale dataset containing various types of images, consisting of more than 20,000 categories. Experiments on this dataset help to analyze the impact of the number of classes. We randomly selected 100 classes from this dataset. The images varied in resolution and we resized them to $256 \times 256$ using standard preprocessing techniques.

4.1.2. Target and Substitute Models

The target models are pre-trained using the three datasets. For each dataset, we construct one target model and various substitute models that use a different training dataset owing to the
black-box conditions. Therefore, each target model is trained by randomly selecting 50% of the images from a dataset. The substitute models randomly select 10% among the remaining images for training. The network architecture, parameters, and training data of the substitute models are necessarily different from the target model because the adversaries are not aware of any details regarding the target. To satisfy this condition, we use different well-known image classification models, such as VGG, ResNet, and Inception. Note that the number of classes is the same but distributions of training data are different in each model.

4.1.3. Attack Algorithms

The adversarial examples are crafted by a white-box attack algorithm based on the substitute model. We apply the I-FGSM [25] and C&W attack algorithms [24] because they are well-known and have been extensively used in previous black-box attacks.

4.2. Experimental Results (Mnist)

The performance metrics, attack success ratio and the number of queries, are average values obtained from multiple experiments. As mentioned previously, the attack success ratio is proportional to the number of queries, and thus, we have to analyze two metrics simultaneously. For each sample, the target classes were the other 9 classes. The maximum upper limit on the number of queries is 1000.

Table 1 presents the network architectures of the target model and substitute models using the following abbreviations: C: convolutional layer, M: max-pooling layer, F: fully connected layer with sigmoid function, S: softmax layer. Substitute models sub1–sub4 are deformed from the target model, and this test strategy is referred to as the previous method [17]. All substitute models are trained using different training data obtained from the target model. Sub1 uses the same architecture as the target model to evaluate the impact of different training data. In one experiment, e.g., sub2-C&W, the parameters of the previous method and proposed method are the same.

| Models | C  | C  | M  | C  | C  | M  | F  | F  | S  |
|--------|----|----|----|----|----|----|----|----|----|
| Target | 32 | 32 | $2 \times 2$ | 64 | 64 | $2 \times 2$ | 128 | 128 | 10 |
| Sub1   | 32 | 32 | $2 \times 2$ | 64 | 64 | $2 \times 2$ | 128 | 128 | 10 |
| Sub2   | 32 | 32 | $2 \times 2$ | 64 | 64 | $2 \times 2$ | 200 | 200 | 10 |
| Sub3   | 32 | 64 | $2 \times 2$ |    |    | 128 | 128 | 10 |
| Sub4   | 28 | 28 | $2 \times 2$ | 48 | 48 | $2 \times 2$ | 128 | 128 | 10 |

In Table 2, the attack success ratios of the proposed method are shown to be higher than those of the previous method. The proposed method does not perform queries when the substitute model is pre-trained. Furthermore, it performs only partial retraining using failed adversarial examples. In contrast, the previous method uses the query process for training the substitute model from the beginning and attempts to emulate the entire target model. These differences are the main reasons behind the results. Note that the attack success ratio may increase if the upper limit of the number of queries increases.

The results indicate that performance is significantly more affected by the network architecture than by the attack algorithm used. Furthermore, the differences in the results between the substitute models using the same attack algorithm are greater than those between the algorithms using the same substitute model. Sub1 achieves the highest success ratio and lowest number of queries because its architecture is the same as that of the target model (although the training data are different). Sub3, which has fewer layers in comparison with other models, achieves the worst results. I-FGSM is the faster attack algorithm, while C&W is slower but more precise; thus, the attack success ratio of C&W is higher in normal white-box environments. However, in our query-limited environments, I-FGSM provides slightly better performance than C&W owing to its faster attack.
Table 2. Black-box attack results (MNIST, query limit = 1000).

| Network Architecture | Attack Algorithm | Previous Method | Proposed Method |
|----------------------|------------------|-----------------|-----------------|
|                      |                  | % Succ. | # Query | % Succ. | # Query |
| Sub1                 | I-FGSM           | 58.2    | 529     | 96.3    | 285     |
| Sub1                 | C&W              | 56.8    | 694     | 97.2    | 294     |
| Sub2                 | I-FGSM           | 43.7    | 724     | 86.1    | 417     |
| Sub2                 | C&W              | 41.4    | 795     | 83.1    | 452     |
| Sub3                 | I-FGSM           | 32.3    | 815     | 67.8    | 692     |
| Sub3                 | C&W              | 29.4    | 883     | 68.6    | 784     |
| Sub4                 | I-FGSM           | 38.2    | 737     | 79.2    | 419     |
| Sub4                 | C&W              | 38.6    | 751     | 77      | 421     |

Note that if the attacker keeps attacking the target model with other samples, the previous method may achieve better results than the proposed method. The substitute model of the previous model can be more powerful as the number of sample attacks increases because it tries to emulate the entire target model. However, if the target model is query-limited, the previous method will have difficulty attacking the target even when considering any-sample attacks.

In Table 2, # query refers to the average number of queries of only successful attacks and failed attacks (using 1000 queries) are ignored. This metric cannot fully represent query efficiency, even though it can show general trends, because it depends on the attack success ratio. This little insufficiency remains even if # query includes the number of queries associated with failed attacks. To analyze query efficiency, it is better to compare the attack success ratio according to the upper bound of the number of queries. Figures 3 and 4 show query efficiency in this manner when the substitute models are Sub1 (best case) and Sub3 (worst case), respectively. Both methods stop iterating when the attack is successful or the number of queries reaches its upper bound. The success ratio of the proposed method is always higher than that of the previous method when the upper bound is the same, regardless of the attack algorithm. In other words, the proposed method requires fewer queries than the previous method to reach a certain attack success ratio. We also observe that the success ratio of the proposed method rapidly increases when the upper bound is approximately 400 in Figure 3, and 700 in Figure 4. In contrast, the points of rapid increase for the previous method are near 800 and 900, respectively. The network architecture of Sub3 is the most different from the target model, thus Sub3 requires more queries than the other architectures to adjust classification boundaries, i.e., to achieve successful attacks.

![Figure 3](image-url)  
**Figure 3.** Attack success ratio according to the upper bound of the number of queries (Sub1, MNIST).
4.3. Impact of Image Size (VggFace2)

The purpose of this section is to compare attack results when the target model is trained by small images (MNIST, Section 4.2) and bigger images (VGGFace2, Section 4.3). Each sample in the VGGFace2 dataset is higher-resolution than in MNIST and images are in color, meaning that black-box attacks on VGGFace2 dataset are more complicated. There are 10 classes (the same number as for MNIST) and, for each test image, the target classes are the other 9 classes. The network architecture of the target model is ResNet50 [35], and we apply the recent loss functions, SphereFace [37] and Ring Loss [38] to improve the classification accuracy of the target model. The ratio of the training data is 50% (approximately 270 images of each face) and they are flipped horizontally for data augmentation. Using the ResNet50 network architecture for the target model, the classification accuracy is approximately 99% after training.

We construct two substitute models whose network architectures are respectively VGG16 [1] and Inception [36]. The maximum upper limit of the number of queries is 10,000.

Table 3 shows that the proposed method achieves better performance on large-sized image classification models. The attack success ratios are much higher and the number of queries is lower in this query-limited environment. The main reasons are the partial retraining of the substitute models and pre-training them without queries as well. Large-size image classification requires large-scale networks as target models, which also makes the attack methods more complicated. In other words, substitute models require more training data to emulate either the entire target model or the partial decision boundary.

![Graph showing attack success ratio](image_url)

**Figure 4.** Attack success ratio according to the upper bound of the number of queries (Sub3, MNIST).

**Table 3.** Black-box attack results (VGGFace2, query limit = 10,000).

| Network Architecture | Attack Algorithm | Previous Method | Proposed Method |
|----------------------|------------------|-----------------|-----------------|
|                      |                  | % Succ. | # Query | % Succ. | # Query |
| VGG16                | I-FGSM           | 29.7%   | 6128    | 82.4%   | 5015    |
| VGG16                | C&W              | 25.1%   | 7351    | 79.7%   | 4942    |
| InceptionV3          | I-FGSM           | 40.1%   | 5986    | 84.1%   | 3914    |
| InceptionV3          | C&W              | 38.9%   | 7010    | 82.4%   | 4598    |

Figures 5 and 6 show query efficiency when the substitute models are VGG16 and InceptionV3, respectively. Similar to the results for MNIST, the success ratio of the proposed method is always higher than that of the previous method when the upper bound is the same. The success ratio of the proposed method rapidly increases when the upper bound approaches 5000 in both Figures 5 and 6. In contrast, the points of rapid increase for the previous method are near 10,000, the maximum upper bound. In both graphs, the proposed method decreases the number of queries by more than 50% compared with the previous method to achieve the same attack success ratio. For example, the proposed method
requires less than 5000 queries to reach a 30% attack success ratio while the previous method needs more than 10,000 queries, as shown in Figure 5.

Figure 5. Attack success ratio according to the upper bound of the number of queries (VGG16, VGGFace2).

Figure 6. Attack success ratio according to the upper bound of the number of queries (InceptionV3, VGGFace2).

4.4. Impact of the Number of Classes (Imagenet)

In this experiment, the image samples in ImageNet are similar in resolution to VGGFace2. The target model however is used to classify 100 classes to analyze the impact of the number of classes. We randomly choose 30 samples for each class, and each sample randomly chooses 10 target classes because of the high number of classes. To prepare the target model, we apply the ResNet50 model using Keras-application [34] after tuning it for 100-classes output. After training, the classification accuracy is approximately 93%. It is less meaningful to examine the results when the target model cannot classify the original image correctly. Therefore, we exclude samples classified incorrectly by the target model. We construct two substitute models and the network architectures are VGG16 [1] and Inception [36]. These substitute models are also pre-trained based on the Keras-application. The maximum upper limit of the number of queries is 10,000.

Table 4 indicates that the proposed method achieves better performance than the previous method when the target model contains many classes. Compared with Table 3, the attack success ratios of the previous method decrease by more than half. However, the attack success ratios of the proposed method are only slightly reduced, although the number of classes increased from 10 to 100 in this experiment. The numbers of queries also slightly increased. We believe these results occur because of the difference in strategies between wholly training and partially retraining.
substitute models. Many classes lead to complicated decision boundaries, thus substitute models have difficulty emulating the entire target model. The proposed method however focuses on the boundary between the original class and the target class. In other words, partial retraining is relatively immune to the number of classes. This characteristic can be a great advantage in black-box attacks.

Table 4. Black-box attack results (ImageNet, query limit = 10,000).

| Network Architecture | Attack Algorithm | Previous Method | Proposed Method |
|----------------------|------------------|-----------------|-----------------|
|                      | % Succ. | # Query | % Succ. | # Query |
| VGG16                | I-FGSM  | 10.2%  | 7450    | 77.8%  | 5648    |
| VGG16                | C&W     | 8.5%   | 7841    | 75.1%  | 5012    |
| InceptionV3          | I-FGSM  | 19.9%  | 6853    | 83.7%  | 4305    |
| InceptionV3          | C&W     | 15.5%  | 7581    | 79.4%  | 4916    |

Figures 7 and 8 show query efficiency when the substitute models are VGG16 and InceptionV3, respectively. Similar to the results for the other datasets, the success ratio of the proposed method is always higher than that of the previous method when the upper bound is the same. The success ratio of the proposed method rapidly increases when the upper bound approaches 6000 in Figure 7 and 5000 in Figure 8. In contrast, it is difficult to observe the points of rapid increase in the previous method because the attack success ratios are too low. As the number of classes increases, the gap between the two methods widens. In Figure 8, the proposed method requires 4000 and 5000 queries to reach 20% and 40% attack success ratios, respectively, while the previous method requires more than 10,000 queries to achieve the 20% attack success ratio.

![Figure 7](image)

**Figure 7.** Attack success ratio according to the upper bound of the number of queries (VGG16, ImageNet).

To mention time efficiency, we provide examples of required time to reach similar attack success ratio. In case of InceptionV3 substitute model using I-FGSM attack algorithm, the proposed method spends about 760 s (including 540 s for pre-training time) to reach 10% attack success ratio while the previous method spends about 980 s. In the same case, the proposed method spends about 830 s to reach 20% attack success ratio while the previous method spends about 1120 s. Even though the previous method has no pre-training phase, it needs more time for training substitute model because of many queries and data augmentation. We do not consider delay for query, and if query delay is included, the gap will be widened because the previous method needs much more queries.
4.5. Real-World Attack on Google AutoML Vision

We attack Google AutoML Vision (GAV) [23] to demonstrate the applicability of the proposed method. GAV provides custom models for image classification using an automated training process. If a user submits training data and labels, a trained model classifies images via the custom labels. The network architecture and parameters of the model are hidden. Prediction results include top-3 labels and probabilities; however, we use only a top-1 label according to our black-box environment. Based on the ImageNet dataset, the experimental settings are the same as those specified in Section 4.4 in order to compare the results between simulations and real attacks. ResNet50 is used as the target model in previous experiments and is added to the set of substitute models in this section because we do not know the target model's architecture.

Table 5 shows the attack results against GAV. The upper bound of the number of queries is limited to 10,000 to compare with the results in Table 4. The performance of the real-world attack is slightly worse than the simulation results. We cannot analyze the reasons because the information regarding the GAV model is hidden. We postulate that some possible reasons are unique network architecture, different data pre-processing, further treatments for improving classification accuracy, or defense techniques.

Table 5. Real-world attack results on Google AutoML Vision (ImageNet, query limit = 10,000).

| Network Architecture | Attack Algorithm | Previous Method % Succ. | # Query | Proposed Method % Succ. | # Query |
|----------------------|------------------|--------------------------|---------|--------------------------|---------|
| VGG16                | I-FGSM           | 9.3%                     | 7637    | 66.3%                    | 6236    |
| VGG16                | C&W              | 6.8%                     | 8913    | 60.6%                    | 5743    |
| InceptionV3          | I-FGSM           | 14.1%                    | 7495    | 69.1%                    | 5123    |
| InceptionV3          | C&W              | 11.2%                    | 8138    | 64.9%                    | 5564    |
| ResNet50             | I-FGSM           | 14.6%                    | 6812    | 72.3%                    | 4981    |
| ResNet50             | C&W              | 12.4%                    | 7448    | 67.2%                    | 5218    |

Figure 9 shows the query efficiency when the substitute model is ResNet50, which provided the best result. Note that the maximum upper bound of queries is 16,000. The attack success ratio reaches 95.84% with the ResNet50 substitute model, I-FGSM algorithm, and 16,000 queries. The proposed method requires less than 6000 queries to reach a 30% attack success ratio but the previous method needs more than 16,000 queries.
5. Discussion

5.1. Extendability

The proposed black-box attack is demonstrated on three image datasets, MNIST, VGGFace2, and ImageNet, to analyze the effects of the different datasets. The proposed method works well on the various datasets, that is, the performance is always much better than the previous method, although larger images lead to some performance degradation. Therefore, the proposed method is extendable within the image classification domain. Furthermore, we believe that it can be applied to other classification domains involving various datasets such as audio, video, and other numerical data. The proposed method requires no domain-specific process for image data such as pixel-by-pixel [31] or fraction of pixels [32] treatments. The adversary, however, must consider network architectures appropriate to the type of domain in the process of pre-training substitute models.

5.2. Practicality

The experimental results demonstrate that the proposed method always reaches a certain level of attack success ratio using less than a third of the number of queries required by the previous method. In other words, it is advantageous to apply the proposed method until three sample attacks are achieved, even though the attack scenario is not a one-sample attack scenario. The previous method requires more than four times the number of queries in some cases (see 10% attack success ratio in Figure 1) and in particular has difficulty succeeding when the upper bound of queries is low. In the case of ImageNet, the proposed method achieves a 10% attack with 3000 queries, where the previous method has difficulties until surpassing 3000 queries. This indicates that under severe conditions, our attack method has a better chance to succeed, regardless of whether one-sample attacks or any-sample attacks are used. We do not present the time required for attack but the more queries mean the more delay.

Previous works based on substitute models and the proposed method have a same limitation regarding practicality. In the experiments, the data sets used by the target model and the substitute model come from the same dataset, even though they are divided and not overlapped. For the real-world scenario, it is more practical that the attacker constructs a new dataset or use another existing dataset. This experimental limitation may come from the difficulty of constructing a new dataset which have similar type of the original dataset.

5.3. Performance Improvement

Through the experiments, we have found that network architecture is one of the key factors for a successful black-box attack. Both performance metrics: attack success ratio and number of queries, are affected by how similar the network architecture of the substitute model is to that of the target.
model (see Table 2). In image classification tasks, many studies have tried to find the best architecture to improve the classification accuracy. A few architectures consequently remain as the top choices, such as ResNet, Inception, and VGG. Therefore, we selected these architectures as substitute models in the black-box attack experiments. That is, we consider that many image classification services are highly likely to use these high-performance and ready-made architectures. However, if we infer the network architecture of a target model using mathematical, algorithmic, heuristic, or practical approaches, it is likely that improved performance and domain extension will be achieved. This approach will be considered in future work.

Similar to different network architectures, different preprocessing techniques also affect the number of queries. For example, the target model may resize images before using them as inputs. If the input size is different from that of the substitute model, this preprocessing can be a kind of feature squeezing [39] which is a defense technique against adversarial examples. However, even if feature squeezing is applied to the target model, adversary attacks can succeed through repeated attacks [40], but this leads to query increment. Preprocessing-independent attacks or inferring preprocessing of the target model can be issues for query-limited black-box attacks.

5.4. Countermeasures

Two types of defense strategies are reactive methods, which seek to detect adversarial examples, and proactive methods, which make the target model robust. No fully effective defense mechanism is known, but these techniques can cause attack performance degradation. As mentioned above, attacks can succeed through repeated attempts, but reactive defenses such as feature squeezing [39] and adversarial examples detection [15], lead to query increments. However, previous studies have shown that black-box attacks with substitute models can evade some proactive defenses: adversarial training [8,18] and defensive distillation [14]. Both defenses fundamentally assume white-box attacks and make it difficult for adversaries to calculate useful gradients. Attacks with substitute models do not use gradients and use only classification labels; thus, they are not affected as much by proactive defenses.

6. Conclusions and Future Work

This study proposes a new method for training the substitute model with the purpose of decreasing the number of queries. Because of query-limited services and systems present in the real world, minimizing the number of queries is an important factor in practical black-box attacks. Unlike previous substitute-model attacks, the proposed method does not emulate the entire target model and adjusts only a boundary needed for a one-sample attack by partial retraining of substitute models. The experimental results demonstrate that the proposed method dramatically decreases the number of queries to achieve a similar attack success ratio and that a higher success ratio is achieved when the target model is query-limited. Furthermore, the proposed method is relatively immune to the number of classification classes in the experiments. Finally, we succeed in crafting targeted adversarial examples against a commercial classifier, Google AutoML Vision.

The number of queries required is still too high, even though it is much lower than that of previous studies. We will attempt to further reduce it in future studies. Inferring the network architecture of the target model is a possible direction to accomplish this. We further intend to apply this idea to domains other than image classification.

Author Contributions: Conceptualization, H.P. and D.C.; Data curation, G.R.; Formal analysis, H.P. and G.R.; Funding acquisition, D.C.; Investigation, H.P. and G.R.; Methodology, H.P.; Project administration, D.C.; Software, H.P. and G.R.; Supervision, D.C.; Validation, H.P. and D.C.; Visualization, G.R.; Writing—original draft, H.P.; Writing—review & editing, D.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No.2020R1A2C1014813) and the research grant of the Kongju National University in 2020.
Conflicts of Interest: The authors declare no conflict of interest.

References

1. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.

2. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Neural Information Processing Systems Conference (NIPS), Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.

3. Hinton, G.; Deng, L.; Yu, D.; Dahl, G.E.; Mohamed, A.-R.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.N.; et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process. Mag.* 2012, 29, 82–97. [CrossRef]

4. Saon, G.; Kuo, H.K.J.; Rennie, S.; Picheny, M. The IBM 2015 English conversational telephone speech recognition system. *arXiv* 2015, arXiv:1505.05899.

5. Yuan, Z.; Lu, Y.; Wang, Z.; Xue, Y. Droid-sec: Deep learning in android malware detection. In Proceedings of the ACM Conference on SIGCOMM, Chicago, IL, USA, 17–22 August 2014; pp. 371–372.

6. Saxe, J.; Berlin, K. Deep neural network based malware detection using two dimensional binary program features. In Proceedings of the 10th International Conference on Malicious and Unwanted Software (MALWARE), Fajardo, PR, USA, 20–22 October 2015; pp. 11–20.

7. Collobert, R.; Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th International Conference on Machine Learning (ICML), Helsinki, Finland, 5–9 July 2008; pp. 160–167.

8. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. In Proceedings of the International Conference on Machine Learning (ICML), Beijing, China, 31 January 2014.

9. Zhang, F.; Chan, P.P.; Biggio, B.; Yeung, D.S.; Roli, F. Adversarial feature selection against evasion attacks. *IEEE Trans. Cybern.* 2015, 46, 766–777. [CrossRef] [PubMed]

10. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial examples in the physical world. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.

11. Evtimov, I.; Eykholt, K.; Fernandes, E.; Kohno, T.; Li, B.; Prakash, A.; Rahmati, A.; Song, D. Robust physical-world attacks on machine learning models. In Proceedings of the IEEE Conference Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.

12. Carlini, N.; Mishra, P.; Vaidya, T.; Zhang, Y.; Sherr, M.; Shields, C.; Wagner, D.; Zhou, W. Hidden voice commands. In Proceedings of the 25th USENIX Security Symposium, Vancouver, BC, Canada, 10–12 August 2016; pp. 513–530.

13. Zhang, G.; Yan, C.; Ji, X.; Zhang, T.; Zhang, T.; Xu, W. Dolphinattack: Inaudible voice commands. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; pp. 103–117.

14. Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In Proceedings of the IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016; pp. 582–597.

15. Meng, D.; Chen, H. Magnet: A two-pronged defense against adversarial examples. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; pp. 135–147.

16. Papernot, N.; McDaniel, P.; Goodfellow, I. Transferability in machine learning: From phenomena to black-box attacks using adversarial samples. *arXiv* 2016, arXiv:1605.07277.

17. Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z.B.; Swami, A. Practical black-box attacks against machine learning. In Proceedings of the ACM Asia Conference on Computer and Communications Security (ASIA-CCS), Abu Dhabi, UAE, 2–6 April 2017; pp. 506–519.

18. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
19. Liu, Y.; Chen, X.; Liu, C.; Song, D. Delving into transferable adversarial examples and black-box attacks. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.

20. Deng, L. The mnist database of handwritten digit images for machine learning research [best of the web]. IEEE Signal Process. Mag. 2012, 29, 141–142. [CrossRef]

21. Cao, Q.; Shen, L.; Xie, W.; Parkhi, O.M.; Zisserman, A. Vggface2: A dataset for recognising faces across pose and age. In Proceedings of the 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG), Xi’an, China, 15–19 May 2018; pp. 67–74.

22. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.

23. Google AutoML Vision. Available online: https://cloud.google.com/vision/ (accessed on 14 October 2020).

24. Carlini, N.; Wagner, D. Towards evaluating the robustness of neural networks. In Proceedings of the IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; pp. 39–57.

25. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial machine learning at scale. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.

26. Moosavi-Dezfooli, S.M.; Fawzi, A.; Frossard, P. Deepfool: A simple and accurate method to fool deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2574–2582.

27. Moosavi-Dezfooli, S.M.; Fawzi, A.; Fawzi, O.; Frossard, P. Universal adversarial perturbations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1765–1773.

28. Hu, W.; Tan, Y. Black-box attacks against RNN based malware detection algorithms. In Proceedings of the Workshops at AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 4–10 February 2017.

29. Xu, W.; Qi, Y.; Evans, D. Automatically evading classifiers. In Proceedings of the Network and Distributed Systems Symposium, San Diego, CA, USA, 21–24 February 2016.

30. Sharif, M.; Bhagavatula, S.; Bauer, L.; Reiter, M.K. A general framework for adversarial examples with objectives. ACM Trans. Priv. Secur. 2019, 22, 1–30. [CrossRef]

31. Chen, P.Y.; Zhang, H.; Sharma, Y.; Yi, J.; Hsieh, C.J. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (AISeC), New York, NY, USA, 11–14 November 2017; pp. 15–26.

32. Narodytska, N.; Kasiviswanathan, S.P. Simple black-box adversarial perturbations for deep networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.

33. Ilyas, A.; Engstrom, L.; Athalye, A.; Lin, J. Black-box adversarial attacks with limited queries and information. In Proceedings of the International Conference on Machine Learning (ICML), Stockholm, Sweden, 10–15 July 2018.

34. Keras Applications GitHub Website. Available online: https://github.com/keras-team/keras-applications (accessed on 14 October 2020).

35. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

36. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Workshops at AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.

37. Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; Song, L. Sphereface: Deep hypersphere embedding for face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 212–220.

38. Zheng, Y.; Pal, D.K.; Savvides, M. Ring loss: Convex feature normalization for face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 5089–5097.
39. Xu, W.; Evans, D.; Qi, Y. Feature squeezing: Detecting adversarial examples in deep neural networks. In Proceedings of the Network and Distributed Systems Security Symposium (NDSS), San Diego, CA, USA, 26 February–1 March 2017.

40. He, W.; Wei, J.; Chen, X.; Carlini, N.; Song, D. Adversarial example defense: Ensembles of weak defenses are not strong. In Proceedings of the 11th USENIX Workshop on Offensive Technologies (WOOT), Vancouver, BC, Canada, 14–15 August 2017.

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).