A Cascade Methodology to Evaluate Black-Box Recognition Systems Based on a Copycat Algorithm

Dinh Cong Nguyen[1], Nhan Tam Le[2], Van Hoa Mai[3], Tuong Quan Nguyen[4], Van Quan Nguyen[4] and The Cuong Nguyen[1]

[1]*Hong Duc University, Thanh Hoa, Vietnam*

[2]*Microsoft, Ha Noi, Vietnam*

[3]*ThinkLABs JSC, Thanh Hoa, VietNam*

[4]*Ministry of Public Security, Vietnam*

Keywords: White-Box Attack, Black-Box Attack, Adversarial Image, Deep Learning, Convolutional Neural Network.

Abstract: With the significant advancements of deep learning (DL) and convolutional neural networks (CNNs), many complex systems in the field of computer vision (CV) have been effectively solved with promising performance, even equivalent to human capabilities. Images sophistically perturbed in order to cause accurately trained deep learning systems to misclassify have emerged as a significant challenge and major concern in application domains requiring high reliability. These samples are referred to as adversarial examples. Many studies apply white-box attack methods to create these adversarial images. However, white-box attacks might be impractical in real-world applications. In this paper, a cascade methodology is deployed in which the Copycat algorithm is utilized to replicate the behavior of a black-box model (known as an original model) by using a substitute model. The substitute model is employed to generate white-box perturbations, which are then used to evaluate the black-box models. The experiments are conducted with benchmark datasets as MNIST and CIFAR10 and a facial recognition system as a real use-case. The results show impressive outcomes, as the majority of the adversarial samples generated can significantly reduce the overall accuracy and reliability of facial recognition systems up to over 80%.

1 INTRODUCTION

Deep neural networks (DNNs), a branch of artificial intelligence (AI), have achieved remarkable achievements in recent years. DNNs have been applied in various fields such as object detection, object recognition, object classification, speech recognition, and natural language processing (NLP) (Bouwmans et al., 2019). Thanks to sophisticated architectural designs, powerful hardware capabilities, and abundant data sources, DNNs have demonstrated superior effectiveness compared to traditional methods. For examples, in image classification, convolutional neural networks (CNNs) can classify images with challenging contexts equivalent to humans (Jiao et al., 2019).

However, many studies have also pointed out that DNN-based intelligent systems also pose security risks (Serban et al., 2020). Specifically, for object classification/recognition, simply adding a sufficient amount of noise to an image can compel the model to produce incorrect results. These images containing such noise are commonly referred to as adversarial images. Many researches have focused on exploring and evaluating the robustness of DNN models through generated crafted samples, thereby, proposing defense mechanisms.

Research into the robustness of DNNs has predominantly centered on investigating the white-box approach (Serban et al., 2020). This approach assumes complete control and access to DNNs, facilitating the analysis of their performance. Through the use of the back-propagation method, which computes gradients of the output concerning the input of DNNs, it becomes feasible to determine the influence of altering pixel values on the loss function and the predicted image labels, as observed in image classification scenarios. However, the majority of practical systems do not expose their internal configurations as architecture and weights. This leads the attacks to become unfeasible.

Nguyen, D., Le, N., Mai, V., Nguyen, T., Nguyen, V. and Nguyen, T.
A Cascade Methodology to Evaluate Black-Box Recognition Systems Based on a Copycat Algorithm.
DOI: 10.5220/0012402500003660

In *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2024) - Volume 2: VISAPP*, pages 662-670
ISBN: 978-989-758-679-8; ISSN: 2184-4321

In this study, we explore the black-box attack scenario. This type of attack implies that the user can access the input and output of a DNN but not its internal models. Therefore, it is not reasonable to create white-box perturbations through back propagation. However, by using the Copycat algorithm(Correia-Silva et al., 2018) to generate a substitute model, the research has demonstrated that the perturbations created by this substitute model can be used to launch attacks that reduce the overall accuracy of the black-box model.

Our contributions of the paper are listed as:

- A new approach employing the cascade methodology for black-box attack methods is introduced. It allows to create the white-box perturbations to evaluate the black-box model without requiring any information from the original model.
- Experiments will be conducted to evaluate the effectiveness of the proposed methods on standard datasets and the proposed dataset.

The remainder of the paper is put in order as follows. Section 2 gives a brief overview of the adversarial attacks in literature. Section 3 details the proposed approach. The performance evaluation is discussed in section 4. Finally, section 5 concludes the paper and presents perspectives.

2 RELATED WORKS

2.1 Adversarial Attacks Based on White-Box Approaches

It is assumed that the white-box attacks to DNNs happen with a clear and precise knowledge of the targeted models. It means that model architectures and weights are opened to attackers. One of the baseline methods is the Fast gradient sign method (FGSM) (Goodfellow et al., 2014). It employs the sign of the gradient based on the back propagation in the white-box deep neural model in order to propose a minimum level of adversarial perturbation. The perturbation is embedded into an input image to generate an adversarial image. Given a model F with a loss function $L(F,x,y)$, x is an input image while y is the label of the input. The new adversarial image $x^* = x + \delta$ with δ can computed as Eq. 1.

$$\delta = \varepsilon \times sgn(\nabla_x L(F,x,y)) \tag{1}$$

The objective of this process is to force the CNN to classify x^* into other classes y' with $y' \neq y$ and $(y' = argmax(F(x^*)))$. Thus, x^* is created with a small value of δ. This guarantees that the changes cannot be distinguished by human (Nguyen et al., 2022).

Jacobian-based Saliency Map (JSMA) (Papernot et al., 2016b) focuses on modifying the pixels of an input image to create an adversarial example. It utilizes the concept of saliency maps, which highlight the most important pixels in influencing the model's decision. For each input image x, through the model function F, it generates the corresponding label with $F(x) = y$ using any CNNs. The goal of the JSMA algorithm is also to create an adversarial image x^* based on the a Jacobian-based saliency map that is very similar to x in order to misclassify the target label, such that $F(x^*) = y' \neq y$. The JSMA algorithm is based on a greedy search algorithm to find pairs of image pixels through the saliency map in order to modify these pixel pairs until the resulting image can cause misclassification by a CNN model. This is also a weakness of this algorithm as it is difficult to implement it on high resolution images due to the large search space.

DeepFool (Moosavi-Dezfooli et al., 2016) employs the attack strategy of using the minimum level of noise to introduce into the input image in order to cause misclassification. To achieve this, DeepFool calculates the distance of the data point to the decision boundary of the classifier. The data point is then replaced in such a way that the distance changes until the data point is misclassified.

Carlini & Wagner (CW) (Carlini and Wagner, 2017) is also an optimization-based method used to generate adversarial examples that can fool DNNs. As other white-box approach, it aims to find the minimum perturbation required to misclassify an input image while ensuring the perturbations are imperceptible. CW is as a gradient-descent based method. However, it differs from other methods where they have only tried to estimate the minimum level of noises embedded into input images. CW defines an objective function that incorporates two components: the first component encourages the perturbation to be small to ensure imperceptibility; and the second component encourages the misclassification of the image.

2.2 Adversarial Attacks Based on Black-Box Approaches

Black-box approaches to attacking (DNNs) refer to methods that aim to exploit vulnerabilities in the model without accessing its internal architecture or parameters. These approaches rely on the model's input-output behavior and make limited or no assumptions about its internal workings. Here are some

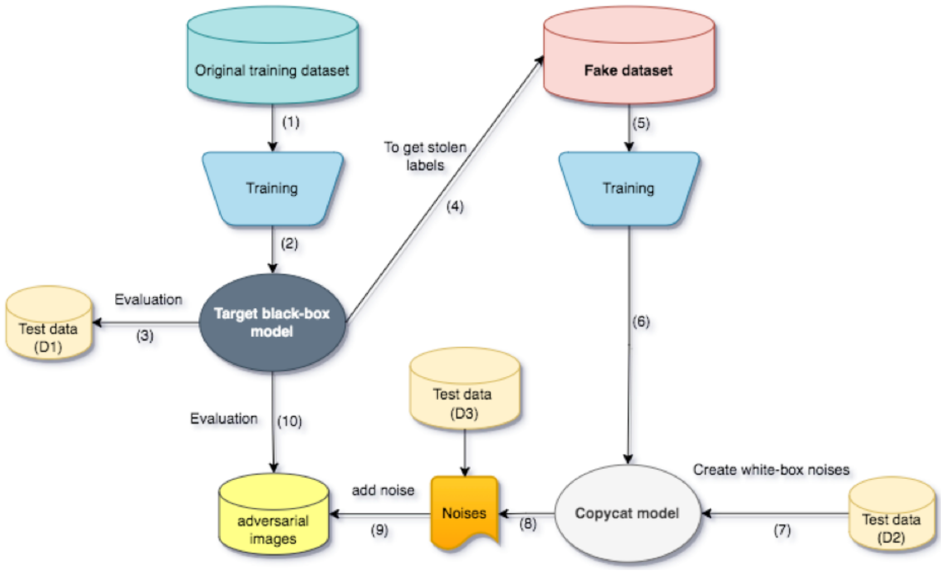

Figure 1: Our proposed system with four main steps.

common black-box attack methods.

Transferability and substitute models use the transferability of adversarial images. The images generated for one model can often fool another model trained on a similar task. The attacker generates adversarial examples using a substitute model with a known architecture and uses them to attack the target model (Moosavi-Dezfooli et al., 2017a; Moosavi-Dezfooli et al., 2017b; Papernot et al., 2016a; Papernot et al., 2017).

Query-Based Attacks make queries to the target model to gain information about its decision boundaries or gradients. The attacker crafts adversarial examples by perturbing the input and observing the model's responses. This information is then used to generate effective adversarial examples (Liu et al., 2016).

Zeroth-Order Optimization: In this approach, the attacker does not have access to gradients or other internal information of the target model. Instead, they rely on the model's input-output behavior and use optimization algorithms to find adversarial examples (Chen et al., 2017).

Black-box attacks are particularly challenging as they operate without complete knowledge of the target model. Attackers have to rely on limited information and make intelligent decisions to craft adversarial examples that can fool the model. These approaches mimic real-world scenarios where the attacker has limited access to the target model's internal details, making them practical and applicable in various scenarios.

Our research focuses on the use of substitute models. We leverage the advantages of the Copycat model (Correia-Silva et al., 2018) in replicating information about the behavior and knowledge of black-box models. After successfully performing the replication process and evaluating its effectiveness, we employ the replicated model as a white-box substitute model. Through this substitute model, we generate white-box adversarial perturbations using the back propagation.

3 PROPOSED APPROACH

3.1 Introduction

In context of using substitute models, we are constructing a proposal system as depicted in Figure 1. The system can be divided into three parts. The first part involves training and evaluating the target model. Naturally, this process is independent and confidential. The second part involves utilizing the Copycat technique to copy the target model. Finally, the substitute model is used to create adversarial images based on white-box attacks. These adversarial images are then applied in reverse to evaluate the accuracy and reliability of the black-box model. In the following sections, we will go deeper into specific parts for further discussion.

3.2 Training and Evaluating the Target Black-Box Model

During the training phase, we construct the scenarios of the target model. The confidential data used to train the model consists of images and labels. This corresponds to steps 1 and 2 in Figure 1. After the training process is completed, the accuracy and reliability of the black-box model will be evaluated in step 3 using the test dataset D1.

3.3 Copying the Target Black-Box Model

To carry out the process of copying the target model, some challenges that need to be addressed are: (1) selecting an appropriate substitute model architecture without any knowledge of the internal structure of the black-box target model; (2) generating suitable synthetic data through queries to the black-box model. This dataset will then be used to train the substitute model, aiming to provide the substitute model with knowledge and behavior similar to the target model as much as possible. Unlike the studies (Tramèr et al., 2016) that focused on copying machine learning (ML) models such as decision trees, logistic regression, and Support Vector Machine (SVM) (Shi et al., 2017) that trained a classification model to copy text classification models as Naive Bayes and SVM; authors (Papernot et al., 2016a) that used deep learning (DL) models to copy ML models such as SVM and K-nearest Neighbor (kNN); authors in (Papernot et al., 2017) that utilized DNNs to copy DNNs models but with small datasets like MNIST or GTSRB. In this research, we approach the task of copying CNNs for more complex classification problems with real-case datasets. To achieve this at minimal cost, we have adopted the approach of the Copycat method as proposed in (Correia-Silva et al., 2018). It should be noted that in this scenario, the training and testing processes of the substitute model are conducted by the attacker. It could be divided into two main parts (**A, B**) as discussed below.

A - Generating Fake Data

The fake dataset is a completely different dataset from the one used to train the original model. This dataset can consist of images from the same problem domain as the original model, or it can be generated from random natural images. In the first case, it is assumed that the attacker has access to images from the same problem domain (PD) that was used to train the original model. In the second case, it is assumed that the attacker has access to large publicly available datasets, which can include random natural images or images that do not belong to the problem domain of the original image dataset used to train the model (known as non problem domain - NPD).

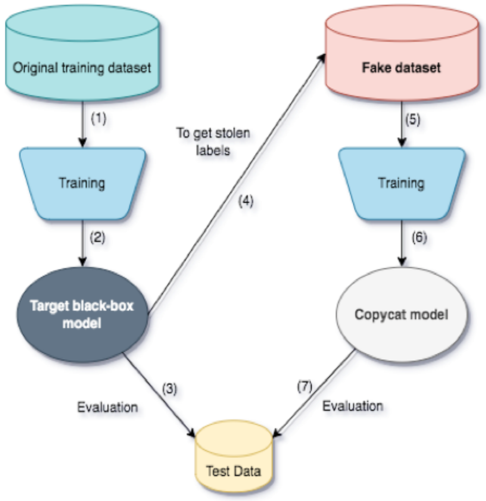

Figure 2: The copy process of the target black-box model.

To create the fake dataset, the attacker uses the original model itself to automatically label these datasets by querying images through the inference process of the model. The labels generated by the original model in this process are called stolen labels. The expected labeled fake dataset captures the general knowledge of the original model, allowing another models to be retrained and achieve performance close to the original model. In practical applications, it is not simple to obtain images from a specific domain. Therefore, the dataset within the same domain as the original dataset is usually smaller, while the dataset outside the original dataset is larger as it can be freely obtained from various sources, as presented in Figure 2.

B - Training the Copycat Model

After obtaining the fake dataset, the training of the Copycat model is initiated. At the beginning, the attacker selects an architecture for the Copycat model. It is worth noting here that the attacker is not aware of the architecture of the target model. However, this does not hinder the knowledge transfer to another model.

The output of the Copycat model is adjusted to match the data domain of the target model, which means the number of outputs must be processed to align with the number handled by the target model. It also emphasizes the importance of prioritizing the use of pretrained models that were trained on large datasets like ImageNet and are close to the domain of the target model. Finally, the training process of the Copycat model involves fine-tuning using smoothing techniques with the fake data generated in the previous step. The training process will stop until the

Copycat model can approximately perform with the target model.

3.4 Generating Adversarial Images

When the copying process is complete, we use this substitute model to craft adversarial samples. The substitute model is a white-box model. Therefore, we have full access to use it for crafting white-box samples. In this study, we only employ digital perturbations. Previous approaches have discussed little about the impact of these perturbations on real-world applications. The digital implementations are exploited based on the back propagation including FGSA, JSMA, DeepFool, and CW.

3.5 Evaluation

The goal of this process is to assess the impact of the adversarial samples generated from section 3.4 on black-box models. For clarity, we remind here that the generation of these adversarial images is entirely reliant on training and constructing substitute models without any intervention into the original models.

4 EXPERIMENTAL RESULTS

4.1 Experimental Setups

4.1.1 Datasets

In order to evaluate our approach, we employ here two popular datasets MNIST (LeCun, 1998), CIFAR10 (Krizhevsky et al., 2009), and our own dataset in face recognition application with a description of Table 1.

Table 1: Dataset description.

Dataset	Labels	Training images	Testing images
MNIST	10	60,000	50,000
CIFAR10	10	10,000	10,000
Our dataset	4	3,000	600

4.1.2 Copying the Black-Box Model with the Benchmark Datasets

To evaluate the attack process based on the Copycat approach, we utilized the CNN network model proposed in the research (Carlini and Wagner, 2017). Furthermore, we experimented with two training scenarios on the MNIST and CIFAR10 datasets using this model. This results in two cases of model corresponding to each dataset. These models are considered as the black-box models. The model architectures are presented in Table 2.

Table 2: Model architectures.

Layer Type	MNIST Model	CIFAR Model
Convolution + ReLU	3x3x32	3x3x64
Convolution + ReLU	3x3x32	3x3x64
Max Pooling	2x2	2x2
Convolution + ReLU	3x3x64	3x3x128
Convolution + ReLU	3x3x64	3x3x128
Max Pooling	2x2	2x2
Fully Connected + ReLU	200	256
Fully Connected + ReLU	200	256
Softmax	10	10

After completing the training process with two black-box models, we use the LeNet5 model to copy these models. Their hyper parameters are selected in Table 3.

- On the MINIST model we employ two strategies: (1) with the data of the PD from the MINIST, (2) with the data of the NPD from the CIFAR10.

- On the CIFAR10 model we we employ two strategies: (1) with the data of the PD from the CIFAR10, (2) with the data of the NPD from the ImageNet.

Table 3: Model hyper parameters.

	MNIST Model	CIFAR Model
Epochs	15	50
Batch size	128	128
Learning rate	1e-3	1e-3
Input image size	32x32x1	32x32x3
Optimization	Adam	Adam

4.1.3 Copying the Black-Box Model Our Dataset

We conducted a further experiment with this approach based on real data collected from facial recognition systems. In the our context, we assume the ResNet152 model (He et al., 2016) as the black-box model for the facial recognition. As the same protocol with the benchmark datasets. We have employed the VGG19 (Simonyan and Zisserman, 2014) to copy the black-box model with here two scenarios: (1) with the data of the PD from the our dataset, and (2) with the data of the NPD from the ImageNet. The hyper parameters have been chosen as outlined in Table 4.

Table 4: Model hyperparameters.

	VGG19 with the PD	VGG19 with the NPD
Epochs	50	50
Batch size	32	32
Learning rate	1e-3	1e-3
Input image size	224x224x3	224x224x3
Optimization	Adam	Adam

All experiments were implemented using Pytorch framework[1] with GPU NVIDIA GeForce RTX 3060, 12GB RAM.

[1] https://pytorch.org/docs/stable/index.html

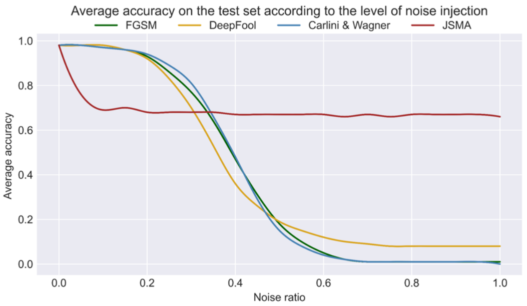

Figure 3: The average accuracy on the test set based on the level of noise injection in the MNIST with the PD.

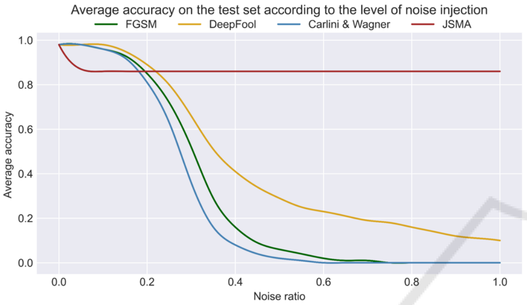

Figure 4: The average accuracy on the test set based on the level of noise injection in the MNIST with the NPD.

4.2 Results and Discussions

4.2.1 Black-Box Models Trained on the MNIST and CIFAR10 Datasets

In this experiment, a process of generating the Copycat models from a black-box model is evaluated. The original model (known as the black-box model) obtained 98.13% of accuracy on the MNIST dataset while the Copycat model was 98.08% of accuracy on PD and 90.61% on the NPD. In contrast, the copy process showed a poorer performance on the CIFAR10 dataset. The black-box accuracy was 81.23% while the Copycat model accuracy just reached 65.55% and 55.87% on the PD and NPD respectively, as shown in Table 5.

Table 5: The top 1 accuracy of the trained models with the black-box model and the Copycat model with the PD and NPD.

Models	MNIST	CIFAR10
The black-box model (CW)	98.13%	81.23%
The Copycat model - LeNet5 with PD	98.08%	65.55%
The Copycat model - LeNet5 with NPD	90.61%	55.87%

The main objective here is to create substitute models in such a way that these models possess knowledge similar to or closely resembling the target model. As the results, these substitute models will be utilized to generate various white-box perturbations as explained above as the FGSM, Deepfool, CW, and JSMA approaches. These forms of perturbations will be applied to input images to craft adversarial images. These adversarial images will then be employed to assess the robustness of the black-box model across the test datasets. Figure 3 illustrates the achieved results. With test data from the MNIST dataset, the PD Copycat model generates various corresponding perturbations at different thresholds. All 3 types of perturbations of the FGSM, CW, and Deepfool approaches demonstrate relatively similar attack capabilities in reducing the overall average accuracy of the black-box model. When the noise level is around 70%, the model almost entirely misclassifies the results with FGSM and CW while Deepfool decreases the model accuracy by only 90% despite applying noise to the entire dataset. In contrast, the JSMA shows the weakest attack effectiveness in reducing the model's accuracy. It achieves a maximum reduction of slightly over 30%.

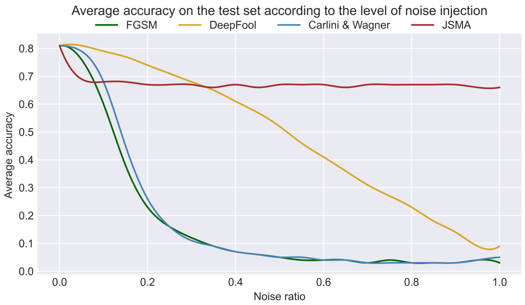

Figure 5: The average accuracy on the test set based on the level of noise injection in the CIFAR10 with the PD.

With the Copycat-NPD model on the MNIST dataset, the trend of reducing the accuracy of the original model corresponding to the perturbations has a small difference to the PD model. The CW is better than the FGSM with a small gap. However, with the JSMA, it still shows a poor performance. The significant decrease in the attack capability to lower the model's accuracy reaches only around 10% for the highest noise ratio, illustrated in Figure 4.

With the CIFAR10 dataset, the impact of the various perturbations created is similar for both the Copycat-PD and Copycat-NPD models. However, the FGSM and the CW exhibit a stronger attack capability compared to the other two types of perturbations. With only 20% noise introduced into the test images, they can significantly reduce the overall accuracy of the model down to about 22%, presented in Figures 5 and 6. Meanwhile, the attack potential of the JSMA appears to have reached saturation. Specifically, based on observations from both Figures 5 and 6, despite the substantial increase in noise

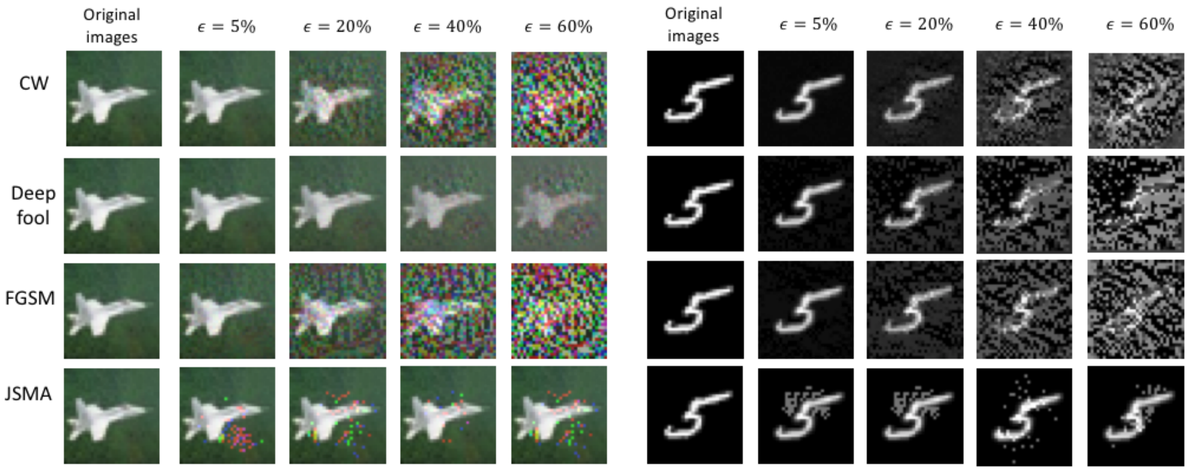

Figure 7: The effects of the FGSM, Deepfool, CW, and JSMA noise on the input images at various thresholds.

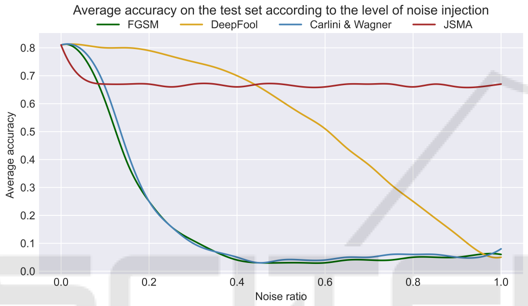

Figure 6: The average accuracy on the test set based on the level of noise injection in the CIFAR10 with NPD.

levels, the impact on the model remains largely unchanged compared to the 15% threshold of noise.

In order to gain a better understanding of how the noise impacts the input images, Figure 7 visualizes these effects graphically. It is easily noticeable that the images are significantly affected by these types of noise. For noise thresholds greater than 40%, almost all input images are heavily distorted. This is not feasible in real-world scenarios due to the high probability of being detected by the naked eye. Therefore, depending on specific applications, appropriate thresholds will be chosen. In the context of this study's observations with the MNIST and CIFAR10 datasets, 20% of noise threshold seems fitting with the CIFAR10 and 40% of noise threshold have to fix with the MNIST. This is because, at this point, the average model accuracy across the data domain is reduced to approximately 60% with the CIFAR10, while the visual representation of the images experiences small changes. Additionally, selecting an appropriate noise threshold also reduces the burden of the noise training process.

4.2.2 Black-Box Model Trained on the Our Given Datasets

To evaluate real-world systems, a facial recognition system is used. We employ a relatively large black-box model, ResNet152 (He et al., 2016). To fit into our testing scenarios, we have treated it as a black-box model without any interventions into the original model.

A substitute model used for replication is the VGG19 model, with two implementations known as the PD-VGG19, NPD-VGG19 models, utilizing the proposed dataset as well as the dataset from ImageNet respectively, given in Table 6.

Table 6: The top 1 accuracy of the trained models with the black-box model (ResNet152) and the Copycat model (VGG19).

Models	Top 1 accuracy (%)
The black-box model (ResNet152)	100%
The Copycat model - VGG19 with the PD	100%
The Copycat model - VGG19 with the NPD	93.75%

The accuracy of the original model using the ResNet152 architecture reaches 100% on the test dataset. Similarly, the Copycat model using VGG19 also achieves 100% accuracy on the PD. Meanwhile, the Copycat model using the NPD has an accuracy lower by 7%. The two substitute models, after copying the behavior of the original model, will be employed to generate adversarial samples. Note that in this scenario, we do not utilize the JSMA. The reason is that the JSMA demonstrates poor performance on the small MNIST and CIFAR10 datasets. Additionally, the JSMA relies on a greedy search approach. Thus, for images with larger dimensions, this search becomes impractical in reality.

It can be seen in Figure 8, with the substitute model using VGG19 on the PD, various types of noise

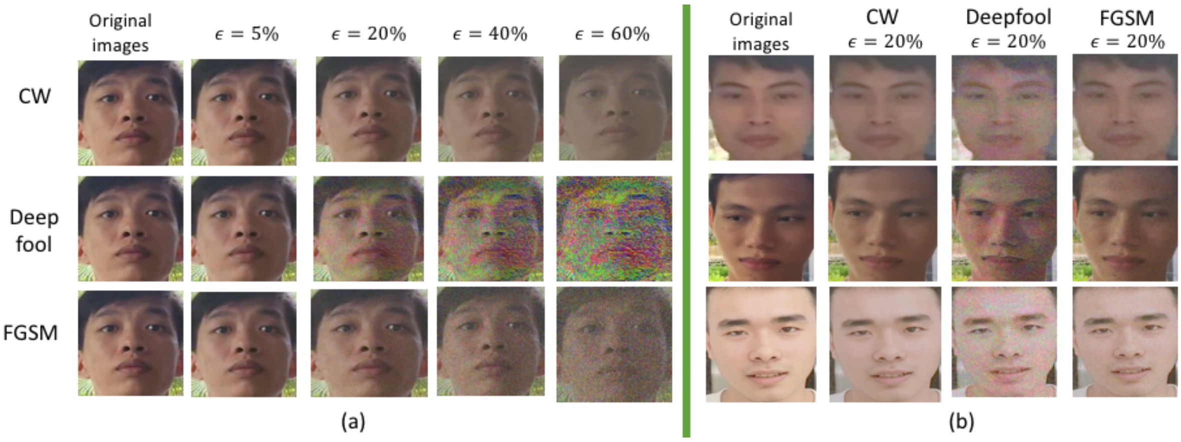

Figure 10: The impacts of the CW, Deepfool, and FGSM on the input images are given: (a) demonstrates the effects of different noise thresholds, and (b) depicts the impact of a 20% noise threshold on various images.

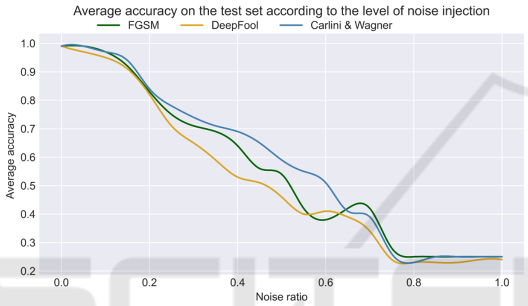

Figure 8: The average accuracy on the test set based on the level of noise injection in our dataset with PD.

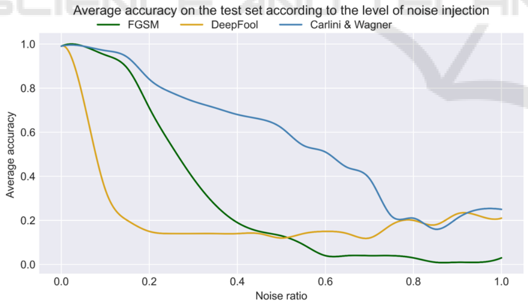

Figure 9: The average accuracy on the test set based on the level of noise injection in our dataset with NPD.

are generated at different thresholds. The created adversarial images are used to evaluate their impact on the original model, which tends to exhibit relatively similar trends. Observing the Figure 9, we can witness that the impact trends of the three types of noise as the FGSM, DeepFool, and CW are relatively similar. However, when the noise ratio is smaller than 50%, the DeepFool has a better effect in reducing the average accuracy of the model.

In contrast, with the VGG19 model on the NPD in Figure 9, the DeepFool noise demonstrates a clear superiority over the other two types of noise. At a noise ratio of approximately 20%, it has reduced the average accuracy of the system by about 80%. Figure 10 visually demonstrates the impact of various noise types on the facial dataset collected by our real-world deployed facial recognition application. It's evident that the attacking noises have the ability to significantly lower the model's accuracy, especially those with substantial changes in the input image, such as Deepfool. For noise densities exceeding 40%, the input images undergo considerable distortion. Similarly, as in the previously mentioned case, the proposed optimal noise threshold is around 20%. This threshold enables Deepfool to reduce the model's recognition accuracy to 80%, while remaining relatively inconspicuous to the naked eye.

5 CONCLUSIONS AND PERSPECTIVES

In this study, we take advantage of the Copycat method to generate substitute models with behaviors closely resembling the target models, aiming to create adversarial images for evaluating black-box models. Unlike previous black-box attack methods, this approach can replicate and generate adversarial samples for both the PD and NPD models. The generated adversarial samples are employed to attack black-box models, and the attack effectiveness is demonstrated to be relatively high using certain types of noise such as FGSM, CW and DeepFool.

In further research, we will continue to expand this approach to evaluate models that require high accuracy, such as license plate recognition systems in security. Another potential direction for research involves generating physical samples for attacks, targeting the verification of models like facial recognition systems at airports, terminals, and other locations.

REFERENCES

Bouwmans, T., Javed, S., Sultana, M., and Jung, S. K. (2019). Deep neural network concepts for background subtraction: A systematic review and comparative evaluation. *Neural Networks*, 117:8–66.

Carlini, N. and Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *SP*, pages 39–57. IEEE.

Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C.-J. (2017). Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *ACM*, pages 15–26.

Correia-Silva, J. R., Berriel, R. F., Badue, C., de Souza, A. F., and Oliveira-Santos, T. (2018). Copycat cnn: Stealing knowledge by persuading confession with random non-labeled data. In *IJCNN*, pages 1–8. IEEE.

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

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*, pages 770–778.

Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., and Qu, R. (2019). A survey of deep learning-based object detection. *IEEE Access*, 7:128837–128868.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

LeCun, Y. (1998). The mnist database of handwritten digits. *IEEE Signal Processing Magazine*, 29(6):141–142.

Liu, Y., Chen, X., Liu, C., and Song, D. (2016). Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*.

Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., and Frossard, P. (2017a). Universal adversarial perturbations. In *CVPR*, pages 1765–1773.

Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., Frossard, P., and Soatto, S. (2017b). Robustness of classifiers to universal perturbations: A geometric perspective. *arXiv preprint arXiv:1705.09554*.

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. (2016). Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*, pages 2574–2582.

Nguyen, D. C., Le, N. D., Nguyen, T. C., Nguyen, T. Q., and Nguyen, V. Q. (2022). An approach to evaluate the reliability of the face recognition process using adversarial samples generated by deep neural networks. In *ICISN 2022*, pages 237–245. Springer.

Papernot, N., McDaniel, P., and Goodfellow, I. (2016a). Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. (2017). Practical black-box attacks against machine learning. In *ACM ASIACCS*, pages 506–519.

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. (2016b). The limitations of deep learning in adversarial settings. In *EuroS&P*, pages 372–387. IEEE.

Serban, A., Poll, E., and Visser, J. (2020). Adversarial examples on object recognition: A comprehensive survey. *CSUR*, 53(3):1–38.

Shi, Y., Sagduyu, Y., and Grushin, A. (2017). How to steal a machine learning classifier with deep learning. In *HST*, pages 1–5. IEEE.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., and Ristenpart, T. (2016). Stealing machine learning models via prediction {APIs}. In *USENIX Security 16*, pages 601–618.