Hindawi
Journal of Sensors
Volume 2024, Article ID 1124598, 9 pages
https://doi.org/10.1155/2024/1124598

Research Article

Evasion Attacks on Deep Learning-Based Helicopter Recognition Systems

Jun Lee,[1] Taewan Kim,[2] Seungho Bang,[3] Sehong Oh,[4] and Hyun Kwon[4]

[1]*Department of Game Software, Hoseo University, Asan-si 31066, Republic of Korea*
[2]*Chief Directorate of Strategic Planning, R.O.K Joint Chief of Staff, Seoul 04383, Republic of Korea*
[3]*Hanwha Aerospace, Seoul 07345, Republic of Korea*
[4]*Department of Artificial Intelligence and Data Science, Korea Military Academy, Seoul 01805, Republic of Korea*

Correspondence should be addressed to Hyun Kwon; hkwon.cs@gmail.com

Received 29 November 2023; Revised 11 February 2024; Accepted 29 February 2024; Published 22 March 2024

Academic Editor: Rajkishor Kumar

Identifying objects in surveillance and reconnaissance systems with the human eye can be challenging, underscoring the growing importance of employing deep learning models for the recognition of enemy weapon systems. These systems, leveraging deep neural networks known for their strong performance in image recognition and classification, are currently under extensive research. However, it is crucial to acknowledge that surveillance and reconnaissance systems utilizing deep neural networks are susceptible to vulnerabilities posed by adversarial examples. While prior adversarial example research has mainly utilized publicly available internet data, there has been a significant absence of studies concerning adversarial attacks on data and models specific to real military scenarios. In this paper, we introduce an adversarial example designed for a binary classifier tasked with recognizing helicopters. Our approach generates an adversarial example that is misclassified by the model, despite appearing unproblematic to the human eye. To conduct our experiments, we gathered real attack and transport helicopters and employed TensorFlow as the machine learning library of choice. Our experimental findings demonstrate that the average attack success rate of the proposed method is 81.9%. Additionally, when epsilon is 0.4, the attack success rate is 90.1%. Before epsilon reaches 0.4, the attack success rate increases rapidly, and then we can see that epsilon increases little by little thereafter.

1. Introduction

In a military context, determining the classification of an adversary's weapon system holds significant importance when strategizing for an operation. This is due to the fact that by successfully identifying the enemy's weapon system in advance, it becomes possible to gain insight into their intentions and subsequently devise effective countermeasures for the benefit of friendly forces.

In the military, a variety of surveillance systems are employed for the detection and classification of weapon systems. These surveillance systems gather information encompassing video footage, audio data, and signals, which is then subject to analysis through human interpretation, typically performed by surveillance officers or analysts. Nonetheless, there are limitations to visually identifying and categorizing weapon systems such as tanks and helicopters, especially when they are maneuvering at high speeds in military scenarios. Furthermore, it is anticipated that relying solely on human-operated surveillance systems will become increasingly constrained in situations where troop numbers are diminishing, and the precise monitoring of weapon systems becomes paramount. Hence, there is a pressing need for automatic weapon system identification technology in the military, which can detect and classify objects employing machine learning methodologies, thereby reducing reliance on human visual feedback.

Numerous research endeavors are presently underway to discern objects within image data procured via surveillance systems. Among these efforts, deep neural networks have exhibited noteworthy proficiency in the task of identifying weapon systems via image classification. However, it is worth noting that deep neural networks [1] are susceptible to vulnerabilities posed by adversarial example attacks [2–7].

Adversarial examples involve injecting slight perturbations into the original data, imperceptible to the human eye but sufficient to cause misclassification by the model. Consequently, video data integrated with adversarial examples could potentially lead to misinterpretations by friendly classifier deep neural networks.

Nonetheless, prior research on adversarial examples has largely overlooked the examination of adversarial attacks on complex imagery related to real-world military scenarios involving weapon systems. This paper addresses this gap by conducting a comprehensive investigation into adversarial example attacks, specifically targeting deep neural networks responsible for classifying attack helicopters and transport helicopters—critical components of military operations.

In this paper, we conducted an analysis of the adversarial example technique applied to a model designed for classifying different types of helicopter models. This method involves the generation of adversarial examples by introducing minimal noise to the original samples used by the helicopter classification model, resulting in misclassifications by the targeted model. The contributions of this paper can be summarized as follows: First, we introduced an adversarial example approach tailored to helicopter classification models relevant to military scenarios, elucidating the method's structure and underlying principles. Second, we performed diverse image analyses on adversarial examples and conducted an in-depth examination of classification probability values. Third, we gathered an actual military dataset and evaluated the method's performance. Furthermore, we verified the efficacy of adversarial examples when applied to models responsible for classifying real-world helicopters.

The remainder of this paper is structured as follows: Section 2 provides an introduction to the target model and reviews previous research pertaining to adversarial examples. Section 3 presents the methodology employed in generating adversarial examples. Section 4 encompasses the experimental procedures and evaluation. Section 6 engages in discussions concerning the implications of adversarial examples. Lastly, Section 7 offers concluding remarks.

2. Related Work

This section describes related studies on models used in helicopters that are target models and related studies on adversarial examples.

2.1. Contents on the Convolutional Neural Network. The classification model for helicopter types utilizes a convolutional neural network (CNN) as its foundation, as referenced in prior studies [8, 9]. CNN is a model that enhances performance by extracting specific image features through a modified architecture within a deep neural network. First, CNN distinguishes itself from traditional neural networks in terms of processing speed. Unlike conventional neural networks, which experience exponential increases in processing time as the parameter size grows, CNN mitigates computational demands by not connecting all perceptrons, allowing for faster learning. Second, image data are typically 3D, comprising height, width, and color components. The key difference lies in whether spatial information within image data are harnessed during the learning process. Traditional neural networks vectorize 3D data into 1D format for input. This transformation results in the loss of spatial information that closely relates to the data, including color. Conversely, CNN retains the spatial characteristics of image data throughout its layers, enabling the utilization of spatial information. Consequently, CNN, characterized by reduced computational complexity, swift performance, and the ability to consider spatial attributes of image data, has gained widespread adoption in image classification methodologies.

2.2. Content about Adversarial Examples. The concept of adversarial examples was initially introduced by Szegedy et al. [2]. Adversarial examples are samples that introduce minimal perturbations to original data, rendering them indistinguishable to humans but causing misclassification by deep learning models. The fundamental approach to generating adversarial examples involves iteratively updating the minimal perturbations through multiple queries to the target model, ultimately producing a sample that induces model misclassification with the smallest possible perturbation. To quantify the minimal perturbation, various metrics such as L_1, L_2, and L_∞ are used to measure the difference between the adversarial example and the original sample. Therefore, a prerequisite for an adversarial example is that it must introduce the smallest perturbation to the original sample while satisfying the condition for model misclassification. The minimal perturbation criterion ensures that the noise remains imperceptible to the human eye. In the case of typical color images, this noise characteristic is challenging for humans to discern. The condition that triggers model misclassification typically involves a point located outside the decision boundary of the original class. As a result, adversarial examples are generated outside this decision boundary while minimizing distortion relative to the original samples. There are different perspectives for categorizing adversarial examples, with some studies emphasizing the distortion between original samples, while others focus on adversarial examples that are misclassified concerning the model's decision boundary. Classification of adversarial examples can also be based on the level of information about the targeted model and the intended goal of the attack, with this method categorizing them accordingly.

2.2.1. Information for the Target Model. Adversarial examples are categorized based on the level of information available about the target model, resulting in two main divisions: white box attacks [10–14] and black box attacks [15–18]. A white box attack occurs when the attacker possesses complete knowledge about the target model. This encompasses awareness of the model's architecture, parameters, and the probability values associated with its outputs. Conversely, a black box attack is executed without any prior information about the target model. In the case of black box attacks, some studies consider the model's output probability values as a black box, attempting to infer these values for specific input data. Generally, if the attacker can ascertain the probability values for each class with black box access, it becomes relatively easy to generate adversarial examples, making the assumption of unavailable probability values a more challenging scenario. Practically speaking, black box attacks are somewhat closer to real-world conditions compared to white box attacks. Since white box attacks can achieve nearly 100% success rates in the realm of image classification, research efforts have shifted toward developing effective black box attack strategies. Within black box attacks, several techniques have emerged, including universal perturbation [19, 20], transfer attacks [21–23], and substitute network methods [24]. First, the universal perturbation method, although introducing strong noise, aims to mislead the attacker into classifying the data as the desired target class by adding specific noise to all original data. This technique utilizes noise that maximizes the gradient loss function across multiple models, adding noise to original samples to elevate the probability of another class appearing in the final softmax layer of deep learning models, thereby crafting adversarial examples susceptible to attack in a general context. Second, transfer attacks involve creating adversarial examples from one model that exhibit some degree of attack efficacy against an unknown model. By enhancing and diversifying the ensemble adversarial examples, which originally targeted local models, these attacks achieve higher success rates against any other model. Transfer attacks consistently yield high success rates, often stemming from the observation that models optimized for specific data tend to exhibit similar decision boundaries. Lastly, the substitute network method leverages the fact that when the target model operates as a black box, a closely resembling substitute network is first created. Adversarial examples generated for this substitute network subsequently exhibit some attack effect against the black box model. In related research, it was demonstrated that a similar model could be constructed for MNIST through 200 queries, illustrating the practical attack potential against real-world image machine learning services.

2.2.2. Purpose of Recognition of Adversarial Example. Adversarial examples can be categorized into targeted attacks [10] and untargeted attacks [25], depending on the attacker's objectives. Targeted adversarial examples are crafted to be misclassified as a specific class predetermined by the attacker, while untargeted adversarial examples are designed to be misclassified as any random incorrect class, deviating from the original class. Generally, untargeted adversarial examples are considered easier to generate, featuring less distortion, while targeted adversarial examples represent a more sophisticated form of attack, as they aim for misclassification into a class specified by the attacker. In most cases, the research on adversarial examples follows a sequence, starting with investigations into untargeted adversarial examples before delving into targeted adversarial examples once a sufficient body of research results has been accumulated.

3. Proposed Scheme

An adversarial example is created by introducing slight perturbations to test data, specifically targeting a pretrained model. As illustrated in Figure 1, these adversarial examples are generated by adding minimal noise to the original data, taking into account the classification scores produced by the target model.

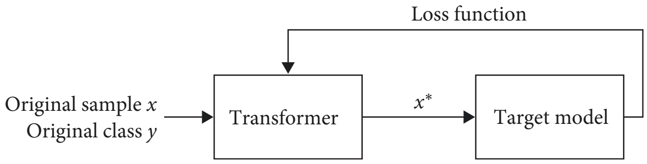


Figure 1: Overview of the proposed scheme.

This methodology can be expressed mathematically as follows: The operation function of the local model, denoted as M_L, is represented as $f_L(x)$. The local model M_L is trained using the original training dataset. Given the pretrained local model M_L, the original training data $x \in X$, their corresponding class labels $y \in Y$, and the target class labels $y \in Y$, we solve an optimization problem to create a targeted adversarial example x:

$$x : \underset{x}{\operatorname{argmin}}\ L(x, x)\ \text{s.t.}\ f_{\mathrm{L}}(x) = y^*, \tag{1}$$

where $L(\cdot)$ represents a distance metric between the original sample x and the transformed example x. The notation $\operatorname{argmin}_x F(x)$ signifies that $F(x)$ is minimized with respect to the value of x. The function $f_L(\cdot)$ is the local model's classification function, determining the input's class label. To generate these x examples, each adversarial example is produced using the fast gradient sign method [26].

The proposed method generates x^* using the L_∞ norm with the following equation:

$$x^* = x + \varepsilon \cdot \operatorname{sign}\left(\nabla_{\mathrm{loss}_{F,y}}(x)\right), \tag{2}$$

where y represents the original class, and F denotes the model's operation function. In this process, the gradient of the loss with respect to the input x is calculated, and the result is used to update x based on the ε value, resulting in the creation of x^*. Despite its simplicity, this method demonstrates strong performance.

4. Experimental Setup and Results

We demonstrate the effectiveness of generating adversarial examples for military models specialized in classifying helicopters through experimentation. This section outlines the experimental configuration used to assess the performance.

4.1. Datasets. The dataset was compiled using publicly available helicopter image data from the Internet. It included images of both attack helicopters and transport helicopters. Specifically, the attack helicopter used was the Hughes AH-64 model known as the Apache, while the transport helicopter was the Sikorsky Airlines UH-60 (S-70A) model referred to as the Black Hawk. Each helicopter's dataset comprised a total of 1,000 images, with 500 images collected evenly for each type of aircraft. Out of these, 400 images per category were allocated for training purposes, while the remaining 100 images

Table 1: Model architecture.

Layer type	Shape
Conv. with ReLU	(224, 224, 64)
Max pooling	(112, 112, 64)
Conv. with ReLU	(112, 112, 128)
Max pooling	(2, 2)
Conv. with ReLU	(56, 56, 128)
Max pooling	(2, 2)
Conv. with ReLU	(28, 28, 128)
Max pooling	(2, 2)
FC with ReLU	(100,352)
FC with ReLU	(1,024)
FC with ReLU	(256)
FC with ReLU	(256)
FC with ReLU	(64)
FC with ReLU	(16)
Softmax	(2)

Conv. means convolutional layer. FC indicates the fully connected layer.

(equivalent to 20% of the total image data) were reserved for testing and evaluation.

4.2. Model Configuration. The target model was constructed using a CNN. Table 1 illustrates the architecture of the CNN model employed for helicopter-type classification. Comprising a total of 17 layers, excluding the input layer, it encompasses four convolutional layers for feature extraction from image data, four max-pooling layers to filter out less relevant features, and one layer responsible for converting input information into 1D format. Additionally, the model incorporates a flatten layer, five fully connected dense layers for input–output connections, and three dropout layers, applied to specific perceptrons to mitigate overfitting. Table 2 shows the model parameters. This model was trained using 800 training data samples, achieving an impressive classification accuracy of 98.9% on 200 test data samples.

4.3. Generation on Adversarial Examples. Within the methodology, we generated 1,000 adversarial examples for each approach. These adversarial examples were crafted with the aim of causing misclassification into a class distinct from the original class, and we employed the Adam optimization algorithm [27] during the optimization process.

5. Experimental Results

The term "attack success rate" denotes the percentage at which an adversarial example is erroneously classified by the model as the specific target class chosen by the attacker. For instance, if 93 out of 100 adversarial examples are classified as the attacker's chosen class, the attack success rate would be 93%.

Figure 2 provides illustrations of original samples and their corresponding adversarial examples. In the figure, the original sample is correctly identified as an Apache by the model. However, the adversarial example, which introduces minimal noise to the original sample, is misclassified by the model as a Black Hawk. Remarkably, this adversarial example, although correctly recognized as an Apache by humans, is erroneously classified as a Black Hawk by the model.

Table 2: Model parameters.

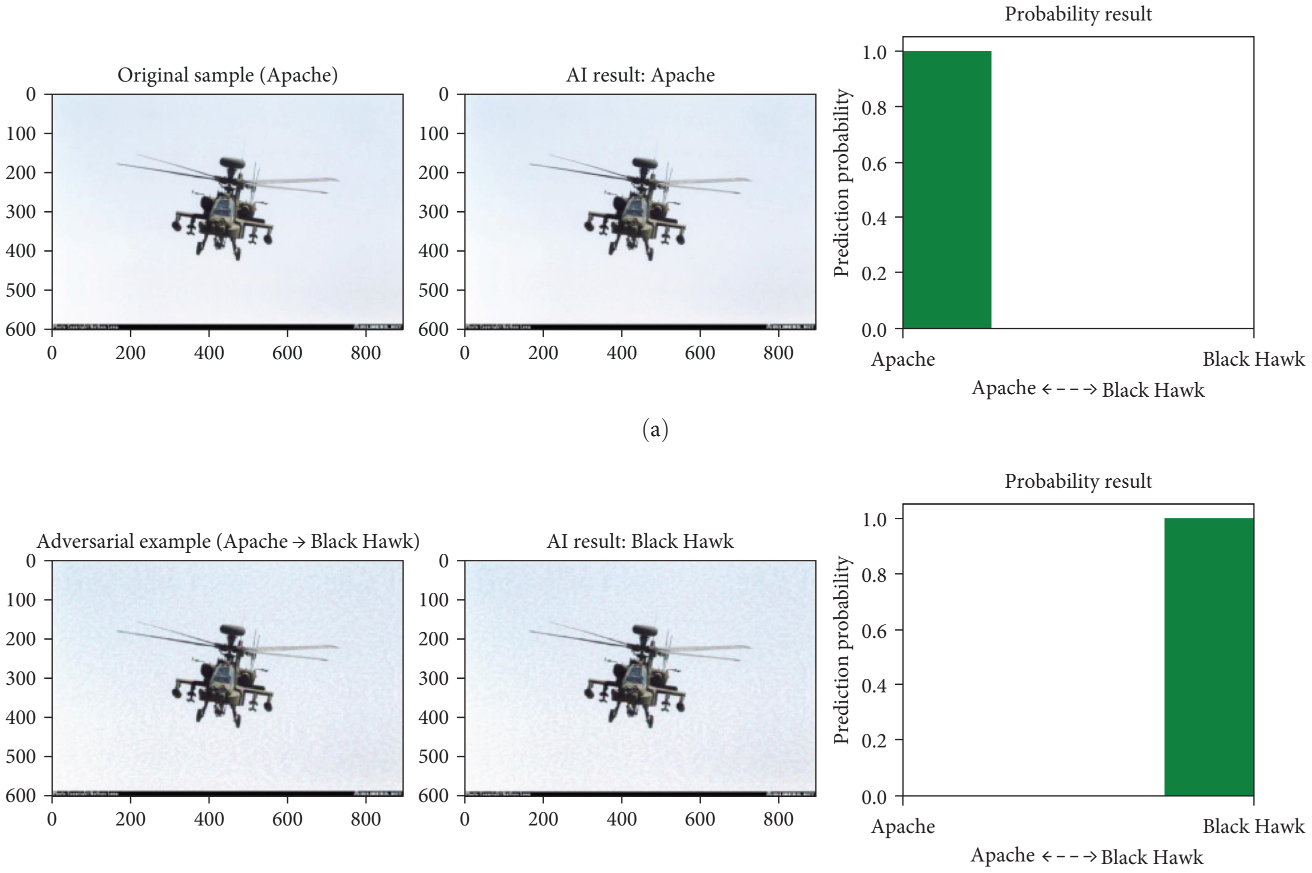
Parameter	Values
Optimizer	Adam [27]
Learning rate	0.1
Momentum	0.9
Delay rate	—
Dropout [28]	0.5
Batch size	128
Epochs	20

Moving on to Figure 3, it showcases examples of adversarial examples generated with varying epsilon values ranging from 0.1 to 0.9. The figure visually demonstrates that as the epsilon value increases, the level of distortion gradually intensifies. Nonetheless, it is important to note that adversarial examples manage to introduce adversarial noise without compromising the original sample's content.

Figure 4 provides insights into the attack success rate of adversarial examples in relation to the epsilon value. As depicted in the figure, as epsilon increases, the attack success rate of the adversarial example also rises. From epsilon 0.1–0.9, the average attack success rate of the proposed method is 81.9%. Additionally, when epsilon is 0.4, the attack success rate is 90.1%. Before epsilon reaches 0.4, the attack success rate increases rapidly, and then we can see that epsilon increases little by little thereafter.

Additionally, according to the reviewer's comment, we compared attack possibilities and performance by applying styless [29], styless-mi [30], styless-mi-ti [31], and styless-mi-ti-di [32] methods. The styless method was applied as a comparison method. The styless method is a method of performing a transfer attack by adding an inject style layer to the model and modifying the style of the original sample. This method has the advantage of being able to attack by applying several types of styles.

Figure 5 shows examples of adversarial samples generated by the proposed method, styless, styless-mi, styless-mi-ti, and styless-mi-ti-di methods. The proposed method shows the adversarial example generated after setting epsilon to 0.4. In the figure, we can see that for each method, adversarial samples are generated by adding minimal noise to the original samples. Table 3 shows the attack success rate of adversarial samples generated by the proposed method and the styless, styless-mi, styless-mi-ti, and styless-mi-ti-di methods. The proposed method shows an attack success rate of 90.1% when epsilon is 0.4. From the table, the styless-mi-ti and styless-mi-di methods have a higher attack success rate than the proposed method. However, in Figure 5, we can see that relatively more noise is reflected in the original sample in the styless-mi-ti and styless-mi-di methods. Additionally, the proposed method can be adjusted to increase the attack success rate by increasing the epsilon value. Therefore, it can be



(a)

(b)

Figure 2: An example of the original sample and the adversarial example: (a) original sample and (b) adversarial example.

seen that there is a trade-off between the noise added to the original sample and the attack success rate.

6. Discussion

In this section, we address the assumptions, advantages, attack considerations, applications, and limitations and future work of the proposed method.

6.1. Assumption. The method operates under the assumption that the attacker is conducting a white box attack on the target model. In this context, the attacker is required to possess knowledge about the model's architecture, parameters, and classification scores. This information is crucial for generating adversarial examples, as it is necessary to be aware of the classification scores associated with each class.

6.2. Advantage and Contributions of the Proposed Method. The advantage of this paper is that it directly constructed a helicopter dataset related to the military. Helicopter data published on the Internet were collected, and each image was labeled and verified by a professional soldier. Additionally, the proposed method is a study applying adversarial examples to military images. In existing research, adversarial example studies using military images were insufficient. In that respect, it is meaningful as a study on security and trust related to artificial intelligence models in the defense field. Lastly, image analysis of adversarial examples was performed by presenting the attack success rate and degree of image distortion according to epsilon.

The contributions of the proposed method include data aspects, helicopter classification model construction, and adversarial example generation for the helicopter classification model. In terms of data, we conducted experiments by constructing a dataset of military-used helicopter copters. We believe that building datasets in the field of artificial intelligence is also meaningful research and has contributions. This is because building a dataset and benchmarking, it is recognized by major academic societies and journals and published as a paper. Therefore, the contribution of the proposed method is that helicopter copters used in military affairs were collected by soldiers with expertise. Second, in terms of helicopter classification model, a CNN was constructed and trained to construct a military helicopter model. Third, we proposed adversarial sample generation for the helicopter copter model using the fast gradient sign method. The proposed method calculates the model's loss function for the input image and then adds adversarial noise to the input image in a direction that increases the value of the loss function. This method is a simple but effective adversarial sample that can cause misidentification of the target model.

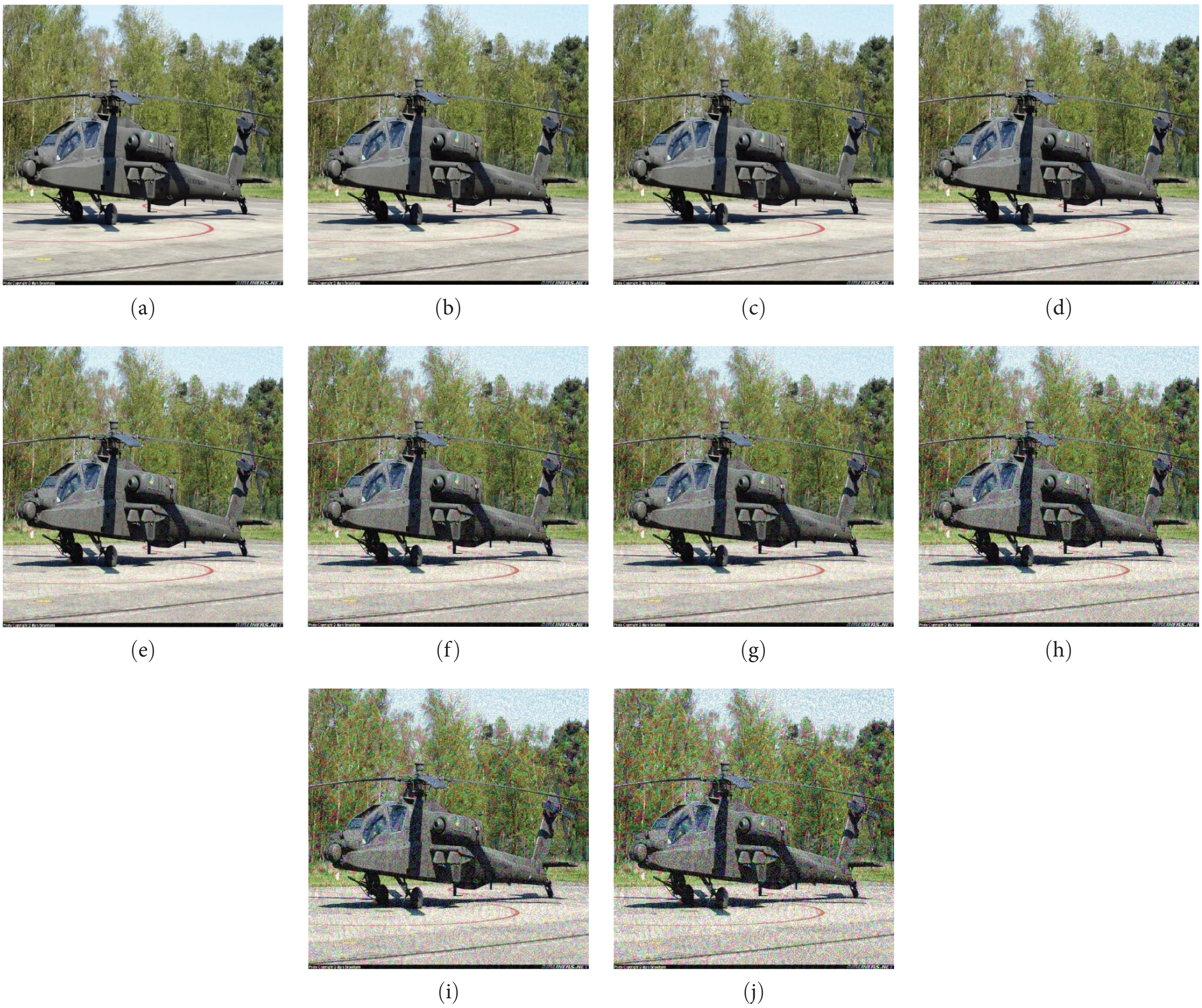


Figure 3: An example of adversarial examples according to epsilon ε: (a) original, (b) 0.1, (c) 0.2, (d) 0.3, (e) 0.4, (f) 0.5, (g) 0.6, (h) 0.7, (i) 0.8, and (j) 0.9.

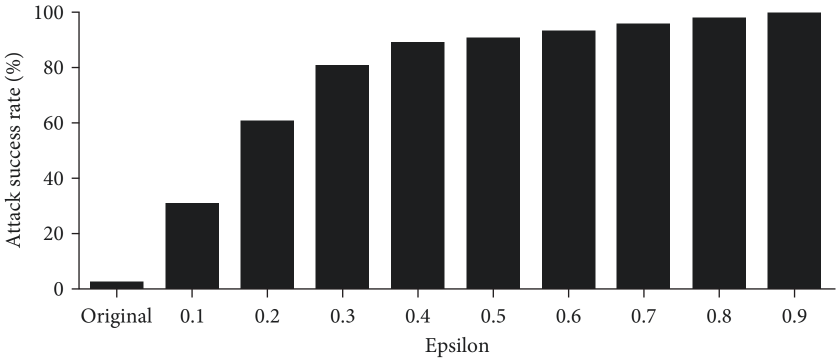


Figure 4: The attack success rate of the adversarial examples according to epsilon ε.

6.3. Attack Considerations. The target model employed in this study serves as a binary classifier, distinguishing between mobile helicopters and transport helicopters. Our focus in this paper has been on generating adversarial examples specifically tailored for a binary classifier model. However, it is worth noting that generating adversarial examples capable of causing misclassification by the binary classifier proved to be more challenging. Despite employing the iterative fast

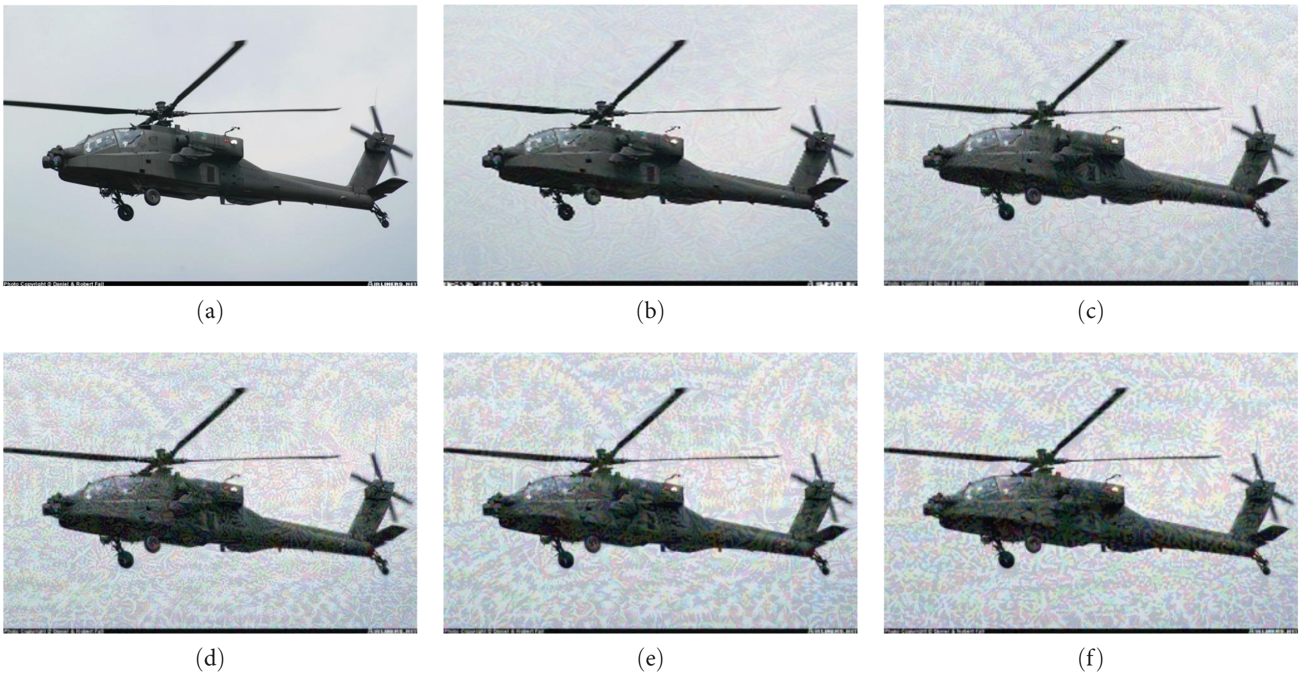

(a) (b) (c) (d) (e) (f)

Figure 5: An sampling of the adversarial example for the (a) original, (b) proposed, (c) styless, (d) styless-mi, (e) styless-mi-ti, and (f) styless-mi-ti-di methods.

Table 3: Comparison of the proposed method, styless, styless-mi, styless-mi-ti, and styless-mi-ti-di methods.

Description	Proposed (%)	Styless (%)	Styless-mi (%)	Styless-mi-ti (%)	Styless-mi-di (%)
Attack success rate	90.1	89.7	89.9	90.6	91.2

gradient sign method for attacking the binary classifier model, the results indicated that the attack was not particularly effective. Nonetheless, it is important to emphasize that even for a binary classifier model, successful attacks can still be executed by generating adversarial examples through the fast gradient sign method.

In addition, it is meaningful that a binary classifier was created by learning a target model by collecting helicopter datasets related to military conditions. It is not a dataset published on the Internet, but a dataset that is being applied in an actual military situation is directly collected and a model is developed. And it seems that there is a contribution point compared to other papers in the creation of adversarial examples for these models.

In terms of evaluation metrics, we used attack success rate as the evaluation metric in this paper. The attack success rate is the number of successful attacks divided by the total number of test data. For example, if out of 200 test data, 20 samples failed the attack, which means the attack success rate is 180/200, indicating a 90% success rate.

6.4. Applications. This technique holds the potential for military applications involving camouflage through the use of adversarial examples. It specifically generates adversarial examples for helicopters, which are accurately identified by humans but misclassified by the model. The applicability of this approach extends beyond helicopter classification models to encompass tank classification models and other deep learning models related to military operations. Consequently, it can be deployed to enhance camouflage systems for military assets like helicopters and tanks, thereby reducing the risk of misclassification by the model.

6.5. Limitations and Future Work. The proposed method is not scoped to apply adversarial examples in physical environments. In this study, the proposed method is to classify images when provided, using a deep learning model in a computer environment. Therefore, we did not apply the process of extracting and recognizing images from the real environment. In order to do this in a physical environment, adversarial patching methods, camera viewing angles, weather effects, etc., are complexly considered, so it is beyond the scope of this research. As a future study, camouflaging adversarial examples to military helicopters in real environments would be an interesting research topic.

In this approach, adversarial examples were successfully generated for a binary classifier model targeting helicopter types. However, generating adversarial examples for a multi-classifier model for helicopter types proved to be a more challenging task. In future research endeavors, the focus will shift toward developing methods for attacking models that classify various military equipment, such as tanks and self-propelled artillery, as well as exploring strategies for handling multiple classifiers in such scenarios.

7. Conclusion

In this paper, we have devised an adversarial example for a military helicopter copter classification model. Our approach yielded an adversarial example that was correctly identified by humans but misclassified by the model trained on real helicopter copter dataset. In the proposed method, adversarial examples are created by adding adversarial noise in a direction that increases the value of the loss function, which represents the difference between the predicted value of the target model for the image to which adversarial noise was added to the original sample. The experimental results demonstrated that the average attack success rate of the proposed method is 81.9%. Additionally, when epsilon is 0.4, the attack success rate is 90.1%. Before epsilon reaches 0.4, the attack success rate increases rapidly, and then we can see that epsilon increases little by little thereafter.

Future research endeavors will encompass evaluating the effectiveness of this approach with diverse image datasets [33] such as MNIST, CIFAR10, and ImageNet. Additionally, an intriguing avenue of exploration involves the creation of various adversarial examples utilizing generative adversarial networks [34]. Lastly, research on ensemble-type defense methods for the proposed method will be an interesting research topic in future research.

Data Availability

The data used to support the findings of this study will be available from the corresponding author upon request after acceptance.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the 2024 research fund of Korea Military Academy (Hwarangdae Research Institute) and the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2021R1I1A1A01040308).

References

[1] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[2] C. Szegedy, W. Zaremba, I. Sutskever et al., "Intriguing properties of neural networks," in *International Conference on Learning Representations*, ICLR, 2014.

[3] P. McDaniel, N. Papernot, and Z. B. Celik, "Machine learning in adversarial settings," *IEEE Security & Privacy*, vol. 14, no. 3, pp. 68–72, 2016.

[4] B. Biggio, G. Fumera, and F. Roli, "Security evaluation of pattern classifiers under attack," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 4, pp. 984–996, 2014.

[5] D. Meng and H. Chen, "Magnet: a two-pronged defense against adversarial examples," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 135–147, ACM, 2017.

[6] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: attacks and defenses," in *International Conference on Learning Representations (ICLR)*, ICLR, 2018.

[7] T. Strauss, M. Hanselmann, A. Junginger, and H. Ulmer, "Ensemble methods as a defense to adversarial perturbations against deep neural networks," arXiv preprint arXiv: 1709.03423, 2017.

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[9] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, IEEE, 2017.

[10] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, IEEE Computer Society, 2017.

[11] N. Papernot, P. McDaniel, S. Jha, Z. B. C. M. Fredrikson, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 372–387, IEEE, Saarbruecken, Germany, 2016.

[12] B. Yang, H. Zhang, Z. Li, Y. Zhang, K. Xu, and J. Wang, "Adversarial example generation with adabelief optimizer and crop invariance," *Applied Intelligence*, vol. 53, no. 2, pp. 2332–2347, 2023.

[13] M. K. Puttagunta, S. Ravi, and C. N. K. Babu, "Adversarial examples: attacks and defences on medical deep learning systems," *Multimedia Tools and Applications*, vol. 82, pp. 33773–33809, 2023.

[14] R. Singhal, M. Soni, S. Bhatt, M. Khorasiya, and D. C. Jinwala, "Enhancing robustness of malware detection model against white box adversarial attacks," in *International Conference on Distributed Computing and Intelligent Technology*, pp. 181–196, Springer, 2023.

[15] N. Narodytska and S. Kasiviswanathan, "Simple black-box adversarial attacks on deep neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1310–1318, IEEE, Honolulu, HI, USA, 2017.

[16] H. Kwon, Y. Kim, K.-W. Park, H. Yoon, and D. Choi, "Advanced ensemble adversarial example on unknown deep neural network classifiers," *IEICE Transactions on Information and Systems*, vol. E101.D, no. 10, pp. 2485–2500, 2018.

[17] Y. Bai, Y. Wang, Y. Zeng, Y. Jiang, and S.-T. Xia, "Query efficient black-box adversarial attack on deep neural networks," *Pattern Recognition*, vol. 133, Article ID 109037, 2023.

[18] T. Wu, T. Luo, and D. C. Wunsch II, "Black-box attack using adversarial examples: a new method of improving transferability," *World Scientific Annual Review of Artificial Intelligence*, vol. 1, Article ID 2250005, 2023.

[19] S. M. M. Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVF, number EPFL-CONF-226156, 2017.

[20] M. Macas, C. Wu, and W. Fuertes, "Adversarial examples: a survey of attacks and defenses in deep learning-enabled cybersecurity systems," *Expert Systems with Applications*, vol. 238, no. Part E, Article ID 122223, 2023.

[21] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "The space of transferable adversarial examples," arXiv preprint arXiv: 1704.03453, 2017.

[22] Z. Wang, H. Yang, Y. Feng et al., "Towards transferable targeted adversarial examples," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20534–20543, IEEE, Vancouver, BC, Canada, 2023.

[23] B. Chen, J. Yin, S. Chen, B. Chen, and X. Liu, "An adaptive model ensemble adversarial attack for boosting adversarial transferability," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4489–4498, IEEE, 2023.

[24] N. Papernot, P. McDaniel, I. Goodfellow, Z. B. C. Somesh Jha, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pp. 506–519, ACM, 2017.

[25] H. Kwon, Y. Kim, H. Yoon, and D. Choi, "Random untargeted adversarial example on deep neural network," *Symmetry*, vol. 10, no. 12, Article ID 738, 2018.

[26] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, ICLR, 2015.

[27] D. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *The International Conference on Learning Representations (ICLR)*, ICLR, 2015.

[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[29] K. Liang and B. Xiao, "Styless: boosting the transferability of adversarial examples," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8163–8172, CVF, 2023.

[30] Y. Dong, F. Liao, T. Pang et al., "Boosting adversarial attacks with momentum," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9185–9193, IEEE, Salt Lake City, UT, USA, 2018.

[31] Y. Dong, T. Pang, H. Su, and Jun Zhu, "Evading defenses to transferable adversarial examples by translation-invariant attacks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4312–4321, IEEE, Beach, CA, USA, 2019.

[32] C. Xie, Z. Zhang, Y. Zhou et al., "Improving transferability of adversarial examples with input diversity," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2730–2739, IEEE, Long Beach, CA, USA, 2019.

[33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: a large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, IEEE, Miami, FL, USA, 2009.

[34] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.