Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 9962972, 9 pages
https://doi.org/10.1155/2022/9962972

Research Article

Restricted-Area Adversarial Example Attack for Image Captioning Model

Hyun Kwon[1] **and SungHwan Kim**[2]

[1]*Department of Artificial Intelligence and Data Science, Korea Military Academy, Seoul, Republic of Korea*
[2]*Department of Applied Statistics, Konkuk University, Seoul, Republic of Korea*

Correspondence should be addressed to SungHwan Kim; shkim1213@konkuk.ac.kr

Received 16 November 2021; Revised 3 May 2022; Accepted 23 May 2022; Published 7 July 2022

Academic Editor: Cristiano Jacques Miosso

Deep neural networks provide good performance in the fields of image recognition, speech recognition, and text recognition. For example, recurrent neural networks are used by image captioning models to generate text after an image recognition step, thereby providing captions for the images. The image captioning model first extracts features from the image and generates a representation vector; it then generates the text for the image captions by using the recursive neural network. This model has a weakness, however: it is vulnerable to adversarial examples. In this paper, we propose a method for generating restricted adversarial examples that target image captioning models. By adding a minimal amount of noise just to a specific area of an original sample image, the proposed method creates an adversarial example that remains correctly recognizable to humans yet is misinterpreted by the target model. We evaluated the method's performance through experiments with the MS COCO dataset and using TensorFlow as the machine learning library. The results show that the proposed method generates a restricted adversarial example that is misinterpreted by the target model while minimizing its distortion from the original sample.

1. Introduction

Deep neural networks [1] provide good performance on tasks of image recognition [2, 3], speech recognition [4], text recognition [5], and data generation [6]. Recently, deep neural networks have also demonstrated good performance for image captioning [7], in which text is generated that explains a given image.

However, such image captioning models are vulnerable to adversarial examples [8–13]. An adversarial example is a sample created by adding noise to an original sample in such a way that it is incorrectly classified by the target model and yet remains correctly recognizable to humans. Adversarial examples cause an image recognition model to provide erroneous results. Research on adversarial examples in the context of image captioning models is ongoing.

Previous studies of adversarial examples targeting image captioning models have generated adversarial examples by adding adversarial noise to the entire image. In certain circumstances, however, it may be advantageous to create an adversarial example by adding adversarial noise just to a specific region of an image, for example, by attaching a sticker to the image. Then, in a situation in which there is limited opportunity to add noise to the entire image, it may still be possible to attack by applying a sticker or the like, adding noise just to a specific area of the image.

In this paper, we propose a method for generating restricted adversarial examples targeting image captioning models. The method adds a small amount of noise just to a specific area of an image, creating an adversarial example that is correctly recognized by humans but misclassified by the target model. The contributions of this paper are as follows. First, we propose a method for generating a restricted adversarial example targeting an image captioning model. The underlying principle and the steps of the proposed method are systematically explained. Second, we analyze the attack success rate, distortion, and classification results for adversarial examples generated by the proposed method. Third, we report the performance of the proposed method based on the results of experiments in which the ResNet model [14] was targeted and MS COCO [15] was used as an image dataset.

The remainder of the paper is structured as follows. In Section 2, studies related to the proposed method are reviewed. Section 3 explains the proposed method. Section 4 describes the experiments and presents the results. Section 5 provides further discussion of the proposed method. Finally, Section 6 concludes the paper.

2. Related Work

2.1. Image Captioning Model. A variety of studies [16–18] are being conducted on image-related models that use deep learning technology, such as the image captioning model, whose characteristics we describe here. The image captioning model can describe each object in an image by applying a long short-term memory (LSTM) model [19] and an attention mechanism that can solve the problems of recurrent neural networks (RNNs) [20]. In this model, the input image is encoded in 512 dimensions using a CNN and is then used as the input to the LSTM to generate sentences. The model is defined as

$$\Theta^* = \operatorname{argmax}_{\Theta} \sum_{I,S} \log p(S|I, X ; \Theta), \tag{1}$$

where Θ is the overall parameter of the LSTM model, I is the image, X is a list of objects extracted from the image, and S is the correct answer sentence. The chain rule is applied to process the variable-length sentence S as follows:

$$\log p(S|I, X ; \Theta) = \sum_{t=0}^{N} \log p(S_t|I, S_0, S_1, \cdots, S_{t-1} ; \Theta). \tag{2}$$

Equation (2) should be optimized using (S, I) pairs during training. It also uses CNNs to represent images. Currently, this model is the one most widely used in image processing problems and object recognition problems. Yolo9000 [21] is used to extract object recognition information.

For training, the LSTM model calculates each word generated by the word and image generated by log $p(S_t|I, S_0, \cdots, S_{t-1} ; \Theta)$. After the model is trained on the words, m_{t-1}, which is the output of the LSTM at $t-1$, is used as an input to the LSTM at t:

$$\begin{gathered} z_{-1} = \mathrm{CNN}(I), \\ z_t = W_e S_t, t \in \{0, \cdots, N-1\}, \\ p_{t+1} = \mathrm{LSTM}(z_t), t \in \{0, \cdots, N-1\}, \end{gathered} \tag{3}$$

where each word S_t is expressed as a one-hot vector. S_0 is a special character that marks the beginning. In this equation, image information generated by the CNN and words expressed in word embedding W_e are mapped to the same space. The image is entered once at $t = -1$. The sentence S^* created in this manner is subjected to attention and words generated by object extraction:

$$S = S^* \cdot X. \tag{4}$$

During training, words are generated by object extraction in various ways. Nouns are extracted from the correct answer sentence, and it is assumed that they are the result of object extraction. Therefore, attention is applied by extracting these nouns and nouns in the output sentence of the LSTM. A multihot vector is generated using the object-extracted noun and the noun in the LSTM output sentence. That is, the length of the vector is the same as the dictionary size, and if the noun exists, it is expressed as 1, and if it does not exist, it is expressed as 0. Cosine similarity is calculated using the multihot vector of the object-extracted noun and the multihot vector of the noun in the LSTM output sentence and is applied as a loss function. Calculating multihot vectors through cosine similarity [22] has the same effect as attention; this means that more weight is given to objects that appear both among the objects in the generated sentence and among the objects in the correct caption. The loss function is expressed as the sum of the negative likelihoods of the correct words at each step:

$$L(I, S, X) = -\sum_{t=1}^{N} \log p_t(S_t). \tag{5}$$

The loss function given by Equation (5) is minimized for all parameters of LSTM by inputting image information, word embedding information, and object-extracted word information by using the CNN.

2.2. Adversarial Examples. An adversarial example, first proposed by Szegedy et al. [8], is a sample created by adding a minimal amount of noise to an original sample in such a way that it is recognized as the original class by humans but is incorrectly classified by the target model.

There are three methods for adding noise to create the adversarial sample: L_0 [23], L_2 [24], and L_∞ [25]. In all three methods, the smaller the number, the more closely the adversarial example resembles the original sample. The L_2 method is the one that was used in this study.

Methods for generating adversarial examples for use in a white-box attack include the fast gradient sign method (FGSM) [26], DeepFool [27], and the Carlini and Wagner (CW) [28] method. These methods find the optimal adversarial example by reducing the gradient so that the probability value for a specific target class increases after the gradient is calculated based on the result value for each class corresponding to the value input to the target model. White-box attacks have a success rate of nearly 100% because in this type of attack, all information about the target model is known. The CW method generates an adversarial example with a high attack success rate and minimal distortion by specifically controlling these two characteristics. Much of the research, however, is focused on black-box attacks rather than white-box attacks. Black-box attack methods include the transfer attack, universal, substitute network, and decision boundary methods. The transfer attack [29] exploits the characteristic that an adversarial example generated by a random model can be effective against other models as well. It has a high attack success rate in the image field. The universal method [30] produces an adversarial example that is incorrectly classified by the model by adding a certain amount of noise to an arbitrary original sample. In this method, a relatively large amount of distortion is applied in the form of adversarial noise, which typically causes the input sample to be misclassified. The substitute network method [31] operates by creating a model similar to the target model through the use of multiple queries against the black-box model. It can then attack the black-box model after generating an adversarial example that is misclassified by the similar model. The decision boundary method [32] can be used when only the result produced by the black-box model for a given input value is known. This method applies distortion to the image just until it ceases to be recognized as the original class while maintaining the similarity between the adversarial image and the original image. It is not a mathematical method, and it cannot be said that it produces the optimal amount of noise, but it can be used to generate adversarial examples relatively easily. HopSkipJumpAttack [33] is an improved version of the decision boundary method, with proven mathematical convergence. After finding the decision boundary as a binary in the decision boundary, the method moves the boundary by epsilon in the vertical direction and then finds the adversarial example in the direction of finding it again by a binary search in the direction toward the original sample. By repeating this process several times, it is possible to generate an adversarial example that is misclassified and is only minimally distorted from the original sample. The method proposed in this paper generates a restricted adversarial example that is misclassified by the target model under the assumption of a white-box attack.

3. Proposed Scheme

3.1. Overview. The proposed method assumes a white-box scenario, in which information is available from the target model. The model provides the probability values for the result corresponding to the input case used when generating an adversarial example. Using this information, adversarial noise is added to an original image to increase the probability of its being incorrectly classified by the target model.

In the proposed method, the loss is divided into distortion loss and attack loss, unlike the loss in the existing method. Whereas conventional methods create the adversarial example by adding adversarial noise to all pixels of the original sample, the proposed method adds noise only to pixels in a specific area of the original sample. This approach allows the possibility of inducing misclassification by attaching a small sticker to a specific area of the original sample. If the adversarial noise is added to an area that is not obvious, the adversarial noise will also be less easily discerned, which is an advantage in terms of perceptibility to humans. In summary, the proposed method differs from other methods in that it adds adversarial noise only to a specific area, and it has the advantage of being able to execute an attack simply by placing a sticker on a sample.

3.2. Description of the Method. The proposed method generates a restricted adversarial example that is perceived as normal by humans but is incorrectly classified by the model. Figure 1 provides an overview of the method. As shown in the figure, a transformer receives the original sample and the original class as inputs and generates a transformed example by adding a small amount of noise to a specific part of the original sample. The transformer provides the transformed example to the target model and receives feedback on it. Through the feedback, the transformer obtains the loss function value and updates the transformed example. By repeating this process, the method adds a minimum amount of noise to a specific part of the original sample, thereby generating a restricted adversarial example that is perceived as normal by humans but is incorrectly classified by the target model.

The purpose of the proposed method is to create a restricted adversarial example. For this study, the transformer presented in [28, 34] was modified as follows:

$$x^* = x + \delta. \tag{6}$$

δ is a specific noise and modifies only the pixels in the specific area chosen by the attacker; pixels in areas other than this restricted area are set to a fixed value of zero. The captioning model D accepts x^* as the input value and provides the loss result to the transformer. At each iteration, the transformer repeats the above procedure to generate a restricted adversarial example while minimizing the total loss loss_T, which is defined as

$$\text{loss}_T = \text{loss}_{\text{distortion}} + c \cdot \text{loss}_{\text{attack}}, \tag{7}$$

where $\text{loss}_{\text{distortion}}$ is the distortion component of the loss function, $\text{loss}_{\text{attack}}$ is the classification loss function of D, and c is the weight value for model D and has an initial value of 1. $\text{loss}_{\text{distortion}}$ is the distortion distance between the transformed example x^* and the original sample x:

$$\text{loss}_{\text{distortion}} = |\delta|. \tag{8}$$

The distortion loss function is $|x^* - x| = |\delta|$ with L_0. los s_{attack} should be minimized:

$$\text{loss}_{\text{attack}} = g(x^*), \tag{9}$$

where $g(k) = Z(k)_{\text{org}} - \max \{Z(k)_j : j \neq \text{org}\}$, in which org is the original class and $Z(\cdot)$ [28, 35] exhibits the probabilities for the class predictions by the image captioning model D. D predicts the probability of the incorrect class to be higher than that of the original class by optimally minimizing $\text{loss}_{\text{attack}}$. Some discrete pixels can be selected even if the restricted pixel region is represented by the shape of $[[s_{11} : e_{11},\ s_{12} : e_{12}],\ [s_{21} : e_{21},\ s_{22} : e_{22}],\ [s_{31} : e_{31},\ s_{32} : e_{32}]]$ $(0 \leq s_{ij} \leq \text{width of sample}\,;\ 0 \leq e_{ij} \leq \text{height of sample}\,;\ 1 \leq i, j \leq 3)$.

4. Experimental Setup and Results

In this section, we present the performance analysis of the adversarial examples generated by the proposed method

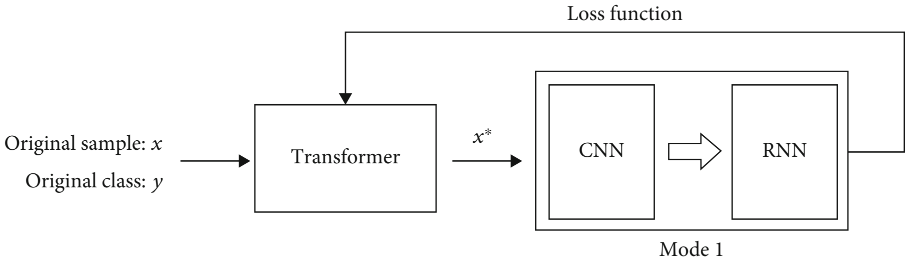


Figure 1: Overview of the proposed method.

for the image captioning model. The TensorFlow [36] machine learning library was used as the experimental environment, and an Intel(R) i5-7100 3.90 GHz computer was used as the server.

4.1. Experimental Setup. MS COCO [15] was used as the dataset for the experiment. This dataset was created for the purpose of performing computer vision tasks such as object detection, segmentation, and keypoint detection. It has 80 object classes, more than 1.5 million object instances, and 164,062 image samples. Of the 164,062 image samples, we used 82,783 for training, 40,504 for the validation, and 40,775 for the test.

A CNN–RNN model [7] was used as the target model for the experiment. Image feature extraction was performed using the CNN model, and caption generation was performed using the RNN model. The CNN model used ResNet 101 (Table 1), and the RNN model was an LSTM with embedding of size 256 and 512 dimensions. The learning rate was 0.001, and the number of epochs was 5.

In generating the restricted adversarial example, Adam [37] was used as the optimization algorithm. Each restricted adversarial example was generated by adding a minimal amount of noise to the lower right part of an original sample, corresponding to 1/16 of its total area. The learning rate was set to 0.005, and the number of repetitions was set to 1000. The performance was analyzed on 1000 adversarial examples randomly generated in this manner.

4.2. Experimental Results. Table 2 shows examples of original image samples and their corresponding adversarial examples generated by the proposed method. It can be seen that the proposed adversarial examples are nearly identical to their corresponding original samples. This is because they are created by applying a minimal amount of distortion to the original sample and are designed to be correctly recognized by humans but incorrectly classified by the target model.

Table 3 shows the images and their top three captioning results for an original sample, the baseline adversarial example, and the proposed adversarial example. The baseline adversarial example was generated by applying the fast gradient sign method (FGSM) as the baseline method. It can be seen that the baseline adversarial example was generated by adding noise throughout the original sample. The restricted adversarial example was generated by adding a minimal amount of noise to the lower right part of the original sample, corresponding to 1/16 of its total area. As can be seen, the restricted adversarial example is nearly identical to the original sample according to human perception. However, the caption interpretations for the original sample, the baseline adversarial example, and the proposed adversarial example differed. The target model correctly captions the original sample to fit the image but misinterprets the proposed adversarial example and captions it inappropriately.

Table 1: Architecture of CNN model. The max pooling layer is 3 × 3, and the stride is 2.

Layer	Output shape	Input shape
Conv1	112 × 112	7 × 7, 64, stride 2
Conv2_x	56 × 56	$\begin{bmatrix} 1\times 1, 64 \\ 3\times 3, 64 \\ 1\times 1{,}256 \end{bmatrix} \times 3$
Conv3_x	28 × 28	$\begin{bmatrix} 1\times 1{,}128 \\ 3\times 3{,}128 \\ 1\times 1{,}512 \end{bmatrix} \times 4$
Conv4_x	14 × 14	$\begin{bmatrix} 1\times 1{,}256 \\ 3\times 3{,}256 \\ 1\times 1, 1024 \end{bmatrix} \times 23$
Conv5_x	7 × 7	$\begin{bmatrix} 1\times 1{,}512 \\ 3\times 3{,}512 \\ 1\times 1, 2048 \end{bmatrix} \times 3$
FC	1 × 1	1000

Table 4 shows a comparison of BLUE scores for the original sample, the baseline adversarial example, and the proposed adversarial example. Here again, the baseline adversarial example was generated using FGSM. The BLUE score is an evaluation index for machine translations. As shown in the following formula, the unigram accuracy, bigram accuracy, or n-gram accuracy is obtained for a machine-translated sentence (by comparing it with the sentence as correctly translated), the geometric mean is taken, and then, a brevity penalty is applied if the sentence is short:

$$\begin{aligned} \text{BLUE} - n = \min &\left\{1, \exp\left(1 - \frac{\text{reference}_{\text{length}}}{\text{output}_{\text{length}}}\right)\right\} \\ &\times \exp\left(\frac{1}{n}\sum_{i=1}^{n} \log\left(\text{precision}_{\text{i}}\right)\right). \end{aligned} \tag{10}$$

Table 2: Examples of original image samples and their corresponding adversarial examples, generated by the proposed method.

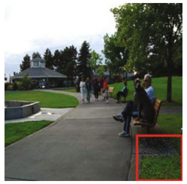
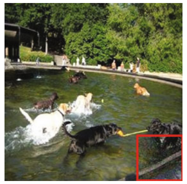
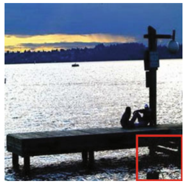
Original			
Proposed			

Table 3: Example images and captions for an original sample, the baseline adversarial example, and the proposed adversarial example.

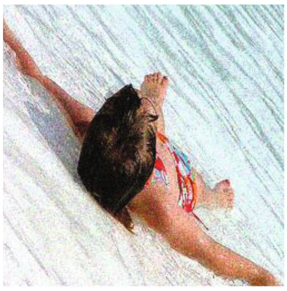
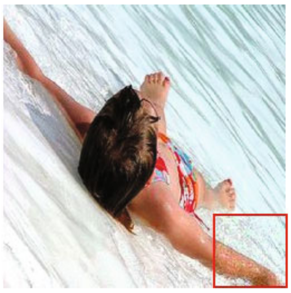
Description	Original sample	Baseline	Proposed
Image			
Caption1	"A young girl is lying in the sand, while ocean water is surrounding her"	"A young boy is jumping in the air"	"A young boy is jumping in the air"
Caption2	"A girl is stretched out in shallow water"	"A boy is stretched out in shallow water"	"A boy is stretched out in shallow water"
Caption3	"A girl wearing a red and multi-colored bikini is laying on her back in shallow water"	"A boy wearing a blue shirt is laying on her back"	"A boy wearing a blue shirt is laying on her back"

Table 4: Comparison of BLUE scores for an original sample, the baseline adversarial example, and the proposed adversarial example.

Description	Original sample	Baseline	Proposed
BLUE1	0.894	0.291	0.321
BLUE2	0.849	0.112	0.125
BLUE3	0.823	0.121	0.153
BLUE4	0.746	0.061	0.072

It can be seen in Table 4 that the BLUE score of the original sample is higher than those of the baseline adversarial example and proposed adversarial example. This is because the original sample is correctly recognized and is given a caption that has high accuracy, whereas the proposed adversarial example and the baseline adversarial example are misinterpreted and given incorrect captions that differ from that for the original sample.

The restricted adversarial example can be positioned anywhere on the image. The reason we have located it at the lower right is because it is easy for an attacker to apply an adversarial example by attaching a sticker to a corner of the image. If the proposed method is applied to an area other than the lower right, however, its performance remains the same. Table 5 shows possible positions for the restricted area: top left, top right, bottom right, bottom left, and center. It can be seen in the table that the proposed adversarial example is misinterpreted regardless of its position.

Figure 2 shows the attack success rate of the restricted adversarial example according to the size of the restricted area. From the figure, it can be seen that the attack success rate increases as the size of the restricted area increases. When the size of the restricted area is 1/16 of the total image area, the restricted adversarial example has an attack success rate of 100%.

5. Discussion

The proposed method creates a restricted adversarial example that is misclassified by the target model but poses no problem for human recognition. The target model generates a representation vector by extracting features from the original image and then generates a particular word through the use of a recursive neural network. The proposed method generates an untargeted adversarial example that causes the target model to misinterpret it and thus produce an arbitrary caption for it instead of the original caption. The experiments with the proposed adversarial example show that certain words in the caption provided by the target model for the proposed adversarial example differ from those provided for the original sample.

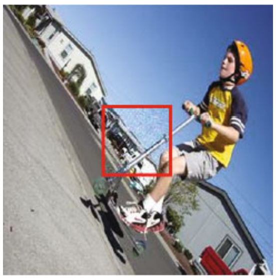
In addition, the correlation between the interpreted caption and the original caption was examined using the BLUE score as an evaluation index, and it was found that the BLUE score of the caption for the proposed adversarial example was lower than that for the original sample. This demonstrates that the proposed adversarial example is mistakenly classified and given an arbitrary caption that is different from the caption for the original image.

The proposed adversarial example is seen to be similar to the original sample in terms of human perception; this is because it is created by applying a minimal amount of distortion to the original sample. If the proposed method is used in a military scenario to generate an adversarial example by adding the optimal amount of noise to a particular image, the modified image can be misinterpreted by the enemy's recognition model. In the healthcare field, patient CT images can be used to generate an adversarial example using the proposed method, leading to misinterpretation. Therefore, an adversarial example generated by the proposed method would pose a serious threat because of the vulnerabilities of the image captioning model.

We applied the proposed method to a second dataset, Flickr 8K [38]. Table 6 shows example images and captions for an original sample from the Flickr 8K dataset and the corresponding proposed adversarial example.

As can be seen, the proposed adversarial example is misinterpreted and given a different caption from that for

TABLE 5: Possible positions of the restricted area for the restricted adversarial example: top left, top right, bottom right, bottom left, and center.

		
Caption1	"The girls are playing soccer"	
Caption2	"Two women in soccer uniforms playing soccer"	
Caption3	"Two young women on different teams are playing soccer on a field"	

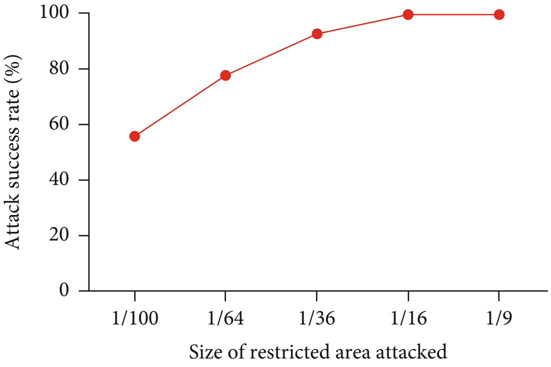
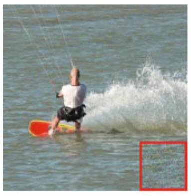


FIGURE 2: Attack success rate of the restricted adversarial example according to the size of the restricted area.

TABLE 6: Example images and captions for an original sample from the Flickr 8K dataset and the corresponding proposed adversarial example.

Description	Original sample	Proposed
Image		
Caption1	"A man holding onto ropes while boogie boarding"	"A boy in yellow is riding a scooter on the street."
Caption2	"A man holds onto ropes and is pulled through the water on his ski"	"A man on a scooter."
Caption3	"A man rides a wakeboard attached to a parachute"	"A boy wearing a helmet rides a scooter."

the original sample. This demonstrates that the method is applicable to the Flickr 8K dataset as well as to the MS COCO dataset.

6. Conclusion

In this paper, we have proposed a method for generating restricted adversarial examples for image captioning models. This method adds noise just to a specific area of the entire image, creating an adversarial example that is correctly recognized by humans but is misclassified by the target model. The experimental results demonstrate that the proposed method generates an adversarial example that is similar to the original sample in terms of human perception and yet is misclassified by the target model.

In future studies, this research can be extended to include other image datasets and to apply to the voice and text domains. In addition, the adversarial example could be generated using a generative adversarial net [39]. Finally, it would be interesting to investigate methods of defense against the proposed method.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the AI R&D Center of Korea Military Academy, the Hwarang-Dae Research Institute of Korea Military Academy, and the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2020R1C1C1A01005229, NRF-2021R1A4A5032622, and 2021R1I1A1A01040308).

References

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

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[3] G. Cheng, Y. Yao, S. Li et al., "DISNet: towards improving separability for remote sensing image change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2022.

[4] G. Hinton, L. Deng, D. Yu et al., "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[5] R. Collobert and J. Weston, "A unified architecture for natural language processing: deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*, pp. 160–167, 2008.

[6] H. Kwon, H. Yoon, and K.-W. Park, "Robust captcha image generation enhanced with adversarial example methods," *IEICE Transactions on Information and Systems*, vol. E103.D, no. 4, pp. 879–882, 2020.

[7] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: lessons learned from the 2015 mscoco image captioning challenge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 652–663, 2017.

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

[9] H. Kwon, H. Yoon, and D. Choi, "Restricted evasion attack: generation of restricted-area adversarial example," *IEEE Access*, vol. 7, pp. 60908–60919, 2019.

[10] H. Kwon and J.-W. Baek, "Adv-plate attack: adversarially perturbed plate for license plate recognition system," *Journal of Sensors*, vol. 2021, 10 pages, 2021.

[11] H. Kwon and J. Jeong, "AdvU-Net: generating adversarial example based on medical image and targeting U-Net model," *Journal of Sensors*, vol. 2022, 13 pages, 2022.

[12] H. Kwon, "MedicalGuard: U-Net model robust against adversarially perturbed images," *Security and Communication Networks*, vol. 2021, 8 pages, 2021.

[13] H. Kwon and J. Lee, "AdvGuard: fortifying deep neural networks against optimized adversarial example attack," *IEEE Access*, 2020.

[14] R. U. Khan, X. Zhang, R. Kumar, and E. O. Aboagye, "Evaluating the performance of resnet model based on image recognition," in *Proceedings of the 2018 International Conference on Computing and Artificial Intelligence*, pp. 86–90, 2018.

[15] T. Y. Lin, M. Maire, S. Belongie et al., "Microsoft coco: common objects in context," in *European Conference on Computer Vision*, pp. 740–755, Springer, 2014.

[16] G. Cheng, X. Sun, K. Li, L. Guo, and J. Han, "Perturbation-seeking generative adversarial networks: a defense framework for remote sensing image scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2022.

[17] X. Qian, X. Cheng, G. Cheng, X. Yao, and L. Jiang, "Two-stream encoder GAN with progressive training for co-saliency detection," *IEEE Signal Processing Letters*, vol. 28, pp. 180–184, 2021.

[18] G. Cheng, L. Cai, C. Lang et al., "SPNet: Siamese-prototype network for few-shot remote sensing image scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2022.

[19] M. J. I. Razin, M. A. Karim, M. Mridha, S. R. Rifat, and T. Alam, "A long short-term memory (LSTM) model for business sentiment analysis based on recurrent neural network," in *Sustainable Communication Networks and Application*, pp. 1–15, Springer, 2021.

[20] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, article 132306, 2020.

[21] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, 2017.

[22] P. Xia, L. Zhang, and F. Li, "Learning similarity with cosine similarity ensemble," *Information Sciences*, vol. 307, pp. 39–52, 2015.

[23] F. Zuo, B. Yang, X. Li, and Q. Zeng, "Exploiting the inherent limitation of l0 adversarial examples," in *22nd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2019)*, pp. 293–307, 2019.

[24] J. Rony, L. G. Hafemann, L. S. Oliveira, I. B. Ayed, R. Sabourin, and E. Granger, "Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4322–4330, 2019.

[25] L. Xie, Y. Wang, J.-L. Yin, and X. Liu, "Robust single-step adversarial training with regularizer," 2021, https://arxiv.org/abs/2102.03381.

[26] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, https://arxiv.org/abs/1412.6572.

[27] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2574–2582, 2016.

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

[29] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," in *5th International Conference on Learning Representations (ICLR 2017)*, 2017.

[30] S. M. Moosavi Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[31] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, 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, 2017.

[32] C. Finlay, A.-A. Pooladian, and A. Oberman, "The logbarrier adversarial attack: making effective use of decision boundary information," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4862–4870, 2019.

[33] J. Chen, M. I. Jordan, and M. J. Wainwright, "Hopskipjumpattack: a query-efficient decision-based attack," in *2020 ieee symposium on security and privacy (sp)*, pp. 1277–1294, 2020.

[34] H. Kwon, Y. Kim, K.-W. Park, H. Yoon, and D. Choi, "Friend-safe evasion attack: an adversarial example that is correctly recognized by a friendly classifier," *Computers & Security*, vol. 78, pp. 380–397, 2018.

[35] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, 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, Saarbruecken, Germany, 2016.

[36] M. Abadi, P. Barham, J. Chen et al., "Tensorflow: a system for large-scale machine learning," *OSDI*, vol. 16, pp. 265–283, 2016.

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

[38] M. Hodosh, P. Young, and J. Hockenmaier, "Framing image description as a ranking task: data, models and evaluation metrics," *Journal of Artificial Intelligence Research*, vol. 47, pp. 853–899, 2013.

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