Research Article

Adv-Plate Attack: Adversarially Perturbed Plate for License Plate Recognition System

Hyun Kwon ¹ and Jang-Woon Baek ²

¹Department of Electrical Engineering, Korea Military Academy, Seoul 01819, Republic of Korea

²Department of Architecture Engineering, Kyung Hee University, Gyeonggi 17104, Republic of Korea

Correspondence should be addressed to Jang-Woon Baek; baekjw@khu.ac.kr

Received 20 August 2021; Accepted 15 September 2021; Published 1 November 2021

Academic Editor: Haibin Lv

Copyright © 2021 Hyun Kwon and Jang-Woon Baek. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Deep learning technology has been used to develop improved license plate recognition (LPR) systems. In particular, deep neural networks have brought significant improvements in the LPR system. However, deep neural networks are vulnerable to adversarial examples. In the existing LPR system, adversarial examples study specific spots that are easily identifiable by humans or require human feedback. In this paper, we propose a method of generating adversarial examples in the license plate, which has no human feedback and is difficult to identify by humans. In the proposed method, adversarial noise is added only to the license plate among the entire image to create an adversarial example that is erroneously recognized by the LPR system without being identified by humans. Experiments were performed using the baza silka dataset, and TensorFlow was used as the machine learning library. When epsilon is 0.6 for the first type, and alpha and the iteration of the second type are 0.4 and 1000, respectively, the adversarial examples generated by the first and second type generation methods are reduced to 20% and 15% accuracy in the LPR system.

1. Introduction

The license plate recognition (LPR) system [1] is used in various places as a system for recognizing letters or numbers on the license plate of a vehicle. For example, when a vehicle travels at speeds higher than the allowable speed on a highway, the LPR system can automatically recognize the vehicle number of the speeding vehicle and impose a penalty or charge on that vehicle. In addition, the LPR system is used to detect the number of vehicles that are violating traffic light laws. In terms of paying the usage fee on highways, the LPR system can automatically recognize the vehicle number and process the toll fee from that. In addition, when paying the parking fee at a mart or corporate building, the LPR system can automatically recognize the vehicle number and calculate the parking fee.

In the LPR system, it is easy to recognize the vehicle license plate when the license plate is positioned at the bottom front of the vehicle and has a pattern of letters or numbers on the license plate. A typical LPR system [2]

extracts corresponding features after matching vehicle specifications and vehicle license plates. After that, it recognizes the numbers or letters on the license plate using a recognition method such as a support vector machine [3]. However, the existing LPR system has a disadvantage in recognizing letters or numbers on a license plate, and it is inferior to the convolution neural network (CNN) [4] in terms of classification because it is affected by environmental factors. For better recognition of numbers or letters on the license plate, recognition [5] of the license plate using CNN provides good performance for image recognition and image classification. The CNN model improves license plate recognition by classifying images segmented by font size into numbers or letters in the last procedure of license plate recognition.

However, CNN models such as deep neural networks [6] are vulnerable to adversarial examples [7]. An adversarial example is a sample with a little noise added to the original data and is incorrectly classified by the CNN model, although it is not abnormal to humans. If such an adversarial example is applied to a license plate, despite seeming

correct to humans, an adversarial license plate can be incorrectly classified by the LPR system based on the CNN model. Existing methods [8–10] using adversarial examples applied to the vehicle license plate recognition system intentionally causes misrecognition by adding noise to a specific spot or changing the brightness of light on the license plate. However, these existing methods have limitations that can be easily identified by humans or require separate human feedback. Unlike the existing method, the proposed method creates an adversarial example on the license plate by adding small noise that is difficult for humans to discern, only to the license plate.

In this paper, we propose a method of applying adversarial examples to the license plate area in the LPR system. In this method, adversarial noise is added to the area corresponding to the license plate, which is normal for humans to see, but is incorrectly recognized by the LPR system. The contributions of this paper are as follows. First, we systematically explained the framework of the proposed method. In addition, we explained the principle of the proposed method and the generation procedure. Second, for the vehicle LPR system, the accuracy for images with adversarial noise and the difference between the original data and adversarial example were analyzed in various ways. Third, to verify the performance of the proposed method, the performance was checked using baza silka dataset [11]. In addition, the proposed method analyzed each of the overall procedures in detail in the LPR system.

The rest of this paper is organized as follows. In Section 2, related works on the proposed method were explained. In Section 3, the content of the proposed method is described. In Sections 4 and 5, the experimental environment and experimental results were explained. Section 6 describes the discussion of the proposed method, and finally, Section 7 consists of the conclusion of this paper.

2. Background and Related Work

This section describes the basic procedural principles of LPR, adversarial examples, and related studies on adversarial examples that attack LPR.

2.1. Basic Explanation of License Plate Recognition. The process of the license plate recognition (LPR) system consists of three steps. The first step is to locate the license plate on the vehicle over the entire image. The second step involves dividing a number or character for each word in the license plate. The third step is the process of recognizing and classifying the characters or numbers of each word. There are convolutional neural network (CNN) [4], support vector machines (SVMs) [3], character templates [12], and hybrid discriminative restricted Boltzmann machine (HDBM) [13] methods to recognize numbers or characters of each word. Among these, CNN performs better in image classification than other methods by automatically extracting features from images. Therefore, CNN is most widely used in LPR systems, and the proposed method generates adversarial examples for LPR systems that perform character classifiers based on these CNNs.

CNN has a structure of a multilayer feed forward neural network. It consists of convolution layers, pooling layers, and fully connected layers. The convolution layer is composed of convolution kernels (filters) and creates a feature map for the input image. Convolution kernels are composed of a weight combination structure using the backward propagation algorithm [14], and each image feature is extracted. The pooling layer [15] is a layer located behind the convolution layer and compresses information from the convolution layer and includes maximum pooling and L_2 pooling. The fully connected layers are the traditional neural network structure, which is connected last in the CNN structure and used for class prediction of the image.

The LPR system is used to recognize characters from 0 to 9 and A to Z in the license plate. In general, multiple separate CNNs are trained to recognize words or numbers. When function $f(\cdot)$ means CNN classifier for license plate character and X means input image, the output value of CNN provides vector probability value and is the corresponding value of each class such as $y = \text{argmax}_{i \in \{0, \dots, N-1\}} Z(i)$. Predicted label y shows the largest confidence value among the N classes.

2.2. Basic Description of Adversarial Examples. The adversarial example was first proposed by Szegedy et al. [16, 17]. Adversarial examples are samples that add minimal noise to the original data, so that there are no detectable problems to humans, but are misclassified by the model. These adversarial examples can be classified into four categories. First, adversarial examples are classified into targeted attacks [18] and untargeted attacks [19, 20] according to the purpose of the attack. A targeted attack is a sample in which the adversarial example is misclassified as a target class by the model. On the other hand, an untargeted attack is a sample that causes the adversarial example to be misclassified by the model as a wrong class rather than the original class. Untargeted attacks have the advantage of being able to attack with fewer repetitions and less distortion than targeted attacks. Second, adversarial examples are classified into white box attacks [21, 22] and black box attacks [23, 24] according to the information on the model. A white box attack is an attack in a situation where the attacker knows all the information about the model. The attacker knows all the information about the structure of the model, the parameters of the model, and the probability values at the output of the model. On the other hand, a black box attack is an attack in which the attacker does not have information about the model. The attacker can only receive results for input values to the model. Third, adversarial examples can be divided into L_1 [25], L_2 [26], and L_∞ [27] according to the method of giving distortion.

$$\sum_{i=0}^n \sqrt[p]{\|x_i - x_i^*\|^p}, \quad (1)$$

where x_i is the original i_{th} pixel and x_i^* is an adversarial example i_{th} pixel. Regardless of which of the three distortion methods is used, the smaller the value, the less the distortion

of the adversarial example is minimized from the distortion of the original sample. Fourth, as for methods of generating adversarial examples, there are fast-gradient sign method (FGSM) [28] and Iterative FGSM (I-FGSM) [29], DeepFool [30], and Carlini and Wagner methods [31]. These methods generate an adversarial example, through feedback of the result of the input into the model, with minimal distortion. First, the attacker provides a transformed example in which some noise is added to the original data as an input value to the model. Next, the model provides the attacker with a probability value corresponding to each class for the input value. The attacker creates a transformed example by adding a little bit of noise so that it is misclassified as the desired target class based on the feedback probability value and then uses it as an input value to the model. By repeating this process, adversarial examples that are misclassified by the model, with minimal noise, are generated. In this paper, we proposed a method of generating adversarial examples by modifying the FGSM and I-FGSM method to add a little of noise to only specific areas of the original image.

2.3. Related Studies of Adversarial Examples on License Plate. There are three representative methods of studying adversarial examples in the license plate. First, Qian et al. [8] proposed a spot evasion attack method for the LPR system. This proposed method has a practical advantage as a method of misrecognition by changing only a specific pixel in an image. However, the disadvantage is that certain pixels are excessively identified in human perception, and it is limited to Chinese vehicle data and the point that an appropriate spot must be found in the entire image. Second, Zha et al. [9] proposed the RoLMA method in the LPR system. This method uses light spots to misrecognize the license plate. However, this method's disadvantage is that it requires human feedback and a complicated procedure for an appropriate light generation location and is easily identifiable by a person due to the spot of light. Third, Rana and Madaan [10] proposed a method to generate an adversarial example of the entire image in a license plate recognition system. This method adds noise to the vehicle picture as a whole and adds overall noise to the vehicle environment as well as the license plate but has a disadvantage of being easily identified by a person using a method such as overlay. This method is somewhat less realistic because it does not modulate only the license plate, which is a specific area. The above existing method has limitations in that it is easy to identify by a person as a spot in a specific area or requires a separate person's feedback. In addition, there was a limitation of distorting or overlaying the entire picture which could be identified by humans. In this paper, we proposed a method that was unidentifiable to humans by adding distortion that is difficult to see on a specific license plate and is misclassified by a license recognition system.

3. Proposed Method

Figure 1 shows an overview of the proposed method. In the figure, the procedure of the proposed method preferentially detects the area for the license plate in the original sample.

A transformed example is created by adding a little bit of noise to only the detected license plate among the entire image. The generated transformed example is provided as an input value to the LPR system including CNN. CNN, which is the last step in the LPR system structure, provides the resulting probability value for the input value to the transformer as feedback. The transformer updates the adversarial noise, based on received feedback of the probability value of each class, and generates a transformed example so that the probability value of the target class is higher and then provides it as an input to the LPR system. By repeating this process, an adversarial example is created that is undetectable to humans but is misrecognized by LPR system as another license plate.

In terms of mathematical expression, let $f(\cdot)$ be the operation function in the LPR system. Given the original sample x as the input value, the following optimization problem is solved to create an adversarial example x^* .

$$\arg \max_{x^*} d(x, x^*) \text{ such that } f(x) \neq f(x^*), \quad (2)$$

where $d(\cdot)$ is a function that means the distance between the original sample and the adversarial example. Through this optimization problem, it is possible to create an adversarial example that is misclassified by the LPR system while minimizing the difference between the original sample and the adversarial example.

In order to generate adversarial examples, the proposed adversarial example was modified by modulating only the license plate area of the entire image. In the first type, the proposed method generates x^* through L_{∞} .

$$x^* = x + \epsilon \cdot \text{sign}(\nabla \text{loss}_{F,t}(g(x))), \quad (3)$$

where $g(\cdot)$ is a function that extracts only the license plate area among the entire image, F is an object function of the model, and t is a target class. In every iteration, the gradient is updated by from the original x , and x^* is found through optimization.

In the second type, the proposed method generates x^* as follows:

$$x^* = x_{i-1} - \text{clip}_{\epsilon}(\alpha \cdot \text{sign}(\nabla \text{loss}_{F,t}(g(x_{i-1})))), \quad (4)$$

where $g(\cdot)$ is a function that extracts only the license plate area among the entire image. In every step, a smaller amount, α , is updated and eventually clipped by the same value.

4. Experiment Setup

Through experiments, the proposed method generates adversarial examples that are correctly recognized by humans but misrecognized by the LPR system.

TensorFlow [32] was used as a machine learning library, and a Xeon E5-2609 1.7 GHz was the server used.

4.1. Datasets. As the license plate dataset, the dataset [11] was used for 500 shots taken using the Olympus Camedia

FIGURE 1: Overview of the proposed method.

```

Input: original sample  $x$ , epsilon  $\epsilon$ , alpha  $\alpha$ , iteration  $r$ 
First type for adversarial example generation:
 $s \leftarrow g(x)$  // license place area
 $\delta \leftarrow \text{sign}(\nabla \text{loss}_{F,t}(s))$ 
 $x^* \leftarrow x + \epsilon \cdot \delta$ 
return  $x^*$ 

Second type for adversarial example generation:
 $x_1^* \leftarrow x$ 
for  $i \leftarrow 1$  to  $r$  do
   $s \leftarrow g(x_i^*)$  // license place area
   $\delta \leftarrow \text{clip}_{\epsilon}(\alpha \cdot \text{sign}(\nabla \text{loss}_{F,t}(s)))$ 
   $x_i^* \leftarrow x_i^* - \delta$ 
end for
return  $x_r^*$ 

```

ALGORITHM 1: Adv-plate attack.

c-2040zoom digital camera. This data is a photo of various vehicles, such as cars, trucks, and buses, in different weather conditions, such as sunny, cloudy, rainy, and dark weather environments. In the dataset, the number of pixels in the image was (640,480,3), and the total number of pixels was 921,600. Of the total data, there were 400 training data and 100 test data that were randomly selected.

4.2. Pretraining of License Plate Recognition System. After extracting each character from the license plate, the LPR system recognized numbers or characters through CNN in the last step. The parameters and structure of the CNN model in the LPR system are specified in Tables 1 and 2. Adam [33] was used as the optimization algorithm of the CNN model. This system was trained in learning data with a 92% accuracy for the test data.

4.3. Generation of Adversarial Example. For the first-generation and second-generation, the proposed method generates adversarial examples by giving noise only to the license plate among the entire images. The first proposed method generated 100 adversarial examples by changing epsilon from 0.1 to 0.9. The second proposed method ran-

domly generated 100 adversarial examples by changing the alpha from 0.1 to 0.6 and using different iterations, namely, 50, 100, 500, and 1000.

5. Experimental Results

The accuracy is the match rate between the original characters on the license plate and the characters recognized by the LPR system. For example, if 92 of the 100 samples are recognized by the LPR system as the original characters, the accuracy is 92%. In the distortion, L_2 , which is the sum of the square root of each pixel difference from the original sample, is used as the distortion measure.

Figure 2 shows examples of images of the original sample and the adversarial example of the proposed method for vehicle data. The proposed method uses the first-generation method to give epsilon 0.4 and adds optimization noise only to the license plate to the generate adversarial example. In the figure, in terms of human perception, the proposed adversarial example has little difference from the original sample. As described above, the proposed method added specific noise only to the license plate from the original

TABLE 1: CNN parameters.

Parameter	Values
Learning rate	0.1
Momentum	0.9
Delay rate	10 (decay 0.0001)
Dropout	0.25
Batch size	128
Epochs	100

TABLE 2: CNN architecture.

Layer type	Shape
Convolution+ReLU	[3, 3, 64]
Convolution+ReLU	[3, 3, 64]
Max pooling	[2]
Convolution+ReLU	[3, 3, 128]
Convolution+ReLU	[3, 3, 128]
Max pooling	[2]
Convolution+ReLU	[3, 3, 256]
Convolution+ReLU	[3, 3, 256]
Max pooling	[2]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Max pooling	[2]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Max pooling	[2]
Fully connected+ReLU	[4096]
Fully connected+ReLU	[512]
Softmax	[36]

sample, so that there is no problem for human recognition, but is misrecognized by the LPR system.

Figure 3 shows the vehicle license plate recognition procedure for the original sample and proposed adversarial example in the LPR system and the classification score of the first word for the original sample and the proposed adversarial example by CNN. In the figure, the left side shows the results for the original sample, and the right side shows the results for the proposed adversarial example. The proposed method was set to epsilon 0.4 as the first type method. In Figures 3(a)–3(h), the LPR system first modulates the image through a grayscale for the input image. After that, in the step of adaptive threshold and find contours, the main part of the image is modulated with lines. At the stage of processing data, it has a procedure of processing rectangles for things that can be predicted as words. After that, the candidate group is selected as a rectangular area that potentially could be characters or numbers for consideration of the character size. After that, considering the order in which the contours are arranged, the rectangle for the candidate letter or number is selected. After that, the image is extracted for each square, and the letter is recognized using

CNN for each word. Figure 3(i) shows the classification score of the first letter for the original sample and the proposed adversarial example for the license plate. The CNN model can classify a total of 36 cases from 0 to 9 and A-Z for each character. In Figure 3(i), in the original sample, the 36th number corresponding to “Z” is 32.1, which was the highest number, suggesting the model correctly recognized the original sample. On the other hand, in the adversarial example sample, the 9th number corresponding to “8” was the highest number at 34.3, so the adversarial example was misrecognized as 8 by the model. In this way, the proposed method adds adversarial noise to the vehicle license plate, so that proposed adversarial samples can be erroneously recognized by the LPR system.

Figure 4 shows an example of an adversarial example generated using the first type of the proposed method according to epsilon. In the figure, as epsilon increases, more noise is added to the license plate. In terms of human perception, noise is added only to the license plate and not to the entire image, so it is similar to the original sample.

Figure 5 shows an example of an adversarial example generated by the second type of the proposed method according to the alpha value and iteration. In the figure, it can be seen that as the iteration and epsilon increase, more noise is added to the license plate. In terms of human recognition, the second proposed method produces an adversarial example that is almost indistinguishable to the original sample.

Figure 6 shows the accuracy of the adversarial examples generated by the first type of the proposed method according to epsilon. For each epsilon, 100 adversarial examples were randomly generated. In the figure, it can be seen that the accuracy of the adversarial example decreases as the value of epsilon increases. However, as epsilon increases, there is a trade-off that adds more noise to the license plate. As shown in the figure, epsilon’s decrease in accuracy is insignificant from 0.7 or higher. When epsilon’s value is about 0.4 to 0.6, the accuracy of the adversarial example is moderately low while minimizing distortion.

Figure 7 shows the accuracy of the adversarial examples generated by the second type of the proposed method according to the alpha value and iteration. As the alpha value increases, the accuracy of the adversarial example decreases. From an alpha value of 0.4, the reduction in accuracy of the adversarial example is insignificant. In addition, as the iteration increases, the accuracy of adversarial example decreases. However, compared to the alpha value, iteration has little effect on the accuracy reduction of adversarial examples.

6. Discussion

6.1. Assumption. The assumption of the proposed method is to attack the LPR system using CNN. The proposed method must know the confidence value as the output result for the input value in the LPR system. This is because the first and second types of the proposed method generate adversarial examples by adding optimal noise until a high probability of the wrong class allocation is obtained with confidence.

FIGURE 2: Examples of the original sample and the adversarial example of the proposed method for vehicle data.

FIGURE 3: The procedure for the original sample (left) and proposed adversarial example (right) in the LPR system and the classification score of the first word for the original sample and the proposed adversarial example by CNN.

FIGURE 4: An example of adversarial examples generated using the first proposed method according to epsilon.

6.2. Accuracy. For the first and second types of the proposed method, the accuracy of adversarial examples varies according to epsilon, iteration, and alpha values. In the first type, since the gradient ascent increases according to the amount of epsilon, as epsilon increases, enough adversarial noise to cross the model's decision boundary is added, and the attack success rate of adversarial examples increases. In the second type, as the iteration and alpha values increase, the amount of noise is increased step by step, and the optimal noise can be found through multiple iterations, thereby increasing the attack success rate of adversarial examples. However, if epsilon, iteration, and alpha values increase, the accuracy decreases due to the attack success rate of the adversarial example, but the added noise increases, so it can be identified by humans. Therefore, the proposed method needs to be adjusted according to the attacker's request to add appropriate adversarial noise to the original data.

6.3. Classification Scores. The proposed method generated adversarial examples by adding some noise to the area corresponding to the license plate in an image. In the classification score, the original sample has the highest score to the original class. However, the adversarial example shows that the wrong class has a high classification score. In this way, the proposed method generates an adversarial example with

a high classification score to the wrong class using minimal noise.

6.4. Adversarial Noise. In the proposed method, the attack effect of adversarial noise has two effects: reducing the performance of license plate detection by incorrect segmentation and preventing the license plate recognition from being properly performed. First, the effect of adversarial noise prevents segmentation on the license plate. In Figure 2, the original sample of the license plate was correctly segmented. However, in the case of the adversarial example, the result of mis-segmentation of the license plate due to the adversarial noise was obtained. Therefore, the proposed method can cause the license plate to be misidentified.

Second, the recognition of letters or numbers on the license plate is misidentified as wrong letters or numbers due to adversarial noise. This is because the proposed method generates adversarial examples by adding adversarial noise so that the class of the wrong character is higher when using feedback on the classification score. Adversarial examples with this noise are hardly discernible by humans but are misrecognized by the LPR system.

6.5. Applications. This proposed method can be applied in autonomous vehicles or military situations. In the case of

FIGURE 5: An example of an adversarial example generated by the second type of the proposed method according to the alpha value α and iteration.

FIGURE 6: The accuracy of the adversarial examples generated by the first type of the proposed method according to epsilon.

an autonomous vehicle, it is possible to create an adversarial example that misrecognizes a specific license plate by manipulating the license plate. In military situations, if the license plate is misrecognized, an enemy vehicle could be mistaken for an ally.

6.6. Limitation and Future Research. The proposed method should know the confidence value of the LPR system. If

the confidence value is not known in the LPR system, the creation of adversarial examples using the proposed method is limited. Therefore, generating adversarial examples with the proposed method without confidence value information is also an important research topic. In addition, the proposed method targets an LPR system based on the CNN model. If the LPR system is not based on the CNN model, the attack effect of the proposed method may be inferior.

Future studies could consider black box attacks with unknown confidence values, rather than just assuming a white box attack with known confidence values. They could use the substitute network method among the black box attack methods. This method creates a similar model by using multiple queries against the black box model. The adversarial example generated by the white box attack on the similar model generated above will be able to attack the target model, which is a black box model. Therefore, in the proposed method, a black box attack is possible using the substitute network method.

FIGURE 7: The accuracy of the adversarial examples generated by the second type of the proposed method according to the alpha value and iteration.

7. Conclusion

The proposed method generated adversarial examples by adding minimal noise only to the license plate in an entire image. Unlike the existing method, there is no human feedback and minimal noise. The proposed scheme can create an adversarial example that is erroneously recognized by the LPR system. When epsilon is 0.6 for the first type and when alpha and the iteration of the second type are 0.4 and 1000, respectively, the experimental results show that the adversarial examples generated by the first and the second type are degraded to 20% and 15% accuracy for the model, respectively.

In future research, the proposed method can be extended by using a generative adversarial net [34] model to generate adversarial examples. In addition, defense research [35] on the proposed method will be an interesting research topic. A defense technique [36, 37] of the proposed method would be to detect adversarial examples using a classification score. Because a specific pattern in the classification score exists in the adversarial example, a method to detect the adversarial example using this pattern could be considered in a future study.

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 work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2019R1F1A1059249) and the Basic Science

Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2021R1I1A1A01040308).

References

- [1] S.-L. Chang, L.-S. Chen, Y.-C. Chung, and S.-W. Chen, "Automatic license plate recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 1, pp. 42–53, 2004.
- [2] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic license plate recognition (alpr): a state-of-the-art review," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 2, pp. 311–325, 2012.
- [3] S. Suthaharan, *Support Vector Machine, in: Machine Learning Models and Algorithms for Big Data Classification*, Springer, 2016.
- [4] P. Kim, *Convolutional Neural Network, in: MATLAB Deep Learning*, Springer, 2017.
- [5] P. Shivakumara, D. Tang, M. Asadzadehkaljahi, T. Lu, U. Pal, and M. H. Anisi, "CNN-RNN based method for license plate recognition," *CAAI Transactions on Intelligence Technology*, vol. 3, no. 3, pp. 169–175, 2018.
- [6] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.
- [7] H. Kwon, *Medicalguard: U-Net Model Robust against Adversarially Perturbed Images*, Security and Communication Networks 2021, 2021.
- [8] Y.-g. Qian, D.-f. Ma, B. Wang et al., *Spot evasion Attacks: Adversarial Examples for License Plate Recognition Systems with Convolutional Neural Networks*, 2019, arXiv preprint arXiv:1911.00927.
- [9] M. Zha, G. Meng, C. Lin, Z. Zhou, and K. Chen, "Rolma: a practical adversarial attack against deep learning-based LPR systems," in *International Conference on Information Security and Cryptology*, pp. 101–117, Springer, 2019.
- [10] K. Rana and R. Madaan, "Evaluating effectiveness of adversarial examples on state of art license plate recognition models," in *2020 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 1–3, Arlington, VA, USA, 2020.
- [11] V. Srebric, *Postupci za Poboljšanje Kontrasta Sivih Slika*, 2003.
- [12] Y.-P. Huang, S.-Y. Lai, and W.-P. Chuang, "A template-based model for license plate recognition," in *IEEE International Conference on Networking, Sensing and Control*, 2004, pp. 737–742, Taipei, Taiwan, 2004.
- [13] H. Larochelle and Y. Bengio, "Classification using discriminative restricted Boltzmann machines," in *Proceedings of the 25th international conference on Machine learning*, pp. 536–543, 2008.
- [14] M. Cilimkovic, *Neural Networks and Back Propagation Algorithm*, Institute of Technology Blanchardstown, Blanchardstown Road North Dublin 15, 2015.
- [15] N. Murray and F. Perronnin, "Generalized max pooling," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2473–2480, Columbus, OH, USA, 2014.
- [16] C. Szegedy, W. Zaremba, I. Sutskever et al., "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014.
- [17] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against

- deep neural networks,” in *in: Security and Privacy (SP), 2016 IEEE Symposium on*, pp. 582–597, IEEE, 2016.
- [18] H. Kwon, Y. Kim, K.-W. Park, H. Yoon, and D. Choi, “Multi-targeted adversarial example in evasion attack on deep neural network,” *IEEE Access*, vol. 6, pp. 46084–46096, 2018.
- [19] A. Wu, Y. Han, Q. Zhang, and X. Kuang, “Untargeted adversarial attack via expanding the semantic gap,” in *in: 2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 514–519, Shanghai, China, 2019.
- [20] H. Kwon, Y. Kim, H. Yoon, and D. Choi, “Random untargeted adversarial example on deep neural network,” *Symmetry*, vol. 10, no. 12, p. 738, 2018.
- [21] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, *Hotflip: White-Box Adversarial Examples for Text Classification*, 2017, arXiv preprint arXiv:1712.06751.
- [22] C. Guo, J. R. Gardner, Y. You, A. G. Wilson, and K. Q. Weinberger, *Simple Black-Box Adversarial Attacks*, 2019, arXiv preprint arXiv:1905.07121.
- [23] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, *Black-Box Adversarial Attacks with Limited Queries and Information*, 2018, arXiv preprint arXiv:1804.08598.
- [24] N. Narodytska and S. Kasiviswanathan, “Simple black-box adversarial attacks on deep neural networks,” in *in: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1310–1318, Honolulu, HI, USA, 2017.
- [25] N. Liu, H. Yang, and X. Hu, “Adversarial detection with model interpretation,” in *in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1803–1811, 2018.
- [26] 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 *in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4322–4330, Long Beach, CA, USA, 2019.
- [27] D. Petrov and T. M. Hospedales, *Measuring the Transferability of Adversarial Examples*, 2019, arXiv preprint arXiv:1907.06291.
- [28] I. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in *in: International Conference on Learning Representations*, 2015.
- [29] A. Kurakin, I. J. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” in *5th International Conference on Learning Representations, {ICLR} 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*, OpenReview.net, 2017, <https://openreview.net/forum?id=HJGU3Rodl>.
- [30] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “Deepfool: a simple and accurate method to fool deep neural networks,” in *in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2574–2582, Las Vegas, NV, USA, 2016.
- [31] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in *in: Security and Privacy (SP), 2017 IEEE Symposium on*, IEEE, pp. 39–57, San Jose, CA, USA, 2017.
- [32] M. Abadi, P. Barham, J. Chen et al., “Tensorflow: a system for largescale machine learning,” in *in: OSDI*, vol. 16, pp. 265–283, 2016.
- [33] D. P. Kingma and J. Ba, “Adam: a method for stochastic optimization,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015, <http://arxiv.org/abs/1412.6980>.
- [34] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., “Generative adversarial nets,” *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.
- [35] H. Kwon and J. Lee, “Advguard: fortifying deep neural networks against optimized adversarial example attack,” *IEEE Access*, 2020.
- [36] H. Kwon, Y. Kim, H. Yoon, and D. Choi, “Classification score approach for detecting adversarial example in deep neural network,” *Multimedia Tools and Applications*, vol. 80, no. 7, pp. 10339–10360, 2021.
- [37] H. Kwon, “Detecting backdoor attacks via class difference in deep neural networks,” *IEEE Access*, vol. 8, pp. 191049–191056, 2020.