Journal of Information Hiding and Multimedia Signal Processing ©2022 ISSN 2073-4212
Ubiquitous International Volume **13**, Number **4**, December **2022**

A Noise Robust Convolutional Neural Network by using Noise Removal Techniques

Chi-Yuan Lin*

Graduate Institute of Electronics Engineering
National Taiwan University, Taiwan
*Corresponding author: tyler.lapril@gmail.com

Rong-San Lin

Dept. of Computer Science and Information Engineering
Southern Taiwan University of Science and Technology, Taiwan
rslin@stust.edu.tw

Received October 2022; revised November 2022

Abstract. *To increase the error tolerance when recognizing images, there must be some methods for image filtering in the recognition network. Modern GPUs are very powerful at processing computer graphics and image processing. Compare to central processing units (CPUs), their parallel structure makes GPUs more efficient to process large blocks of data. Computer Vision (CV) has been used in many fields with the help of GPU. State-of-the-art computational power that uses Neural Networks (NNs) has made the most impressive advancement in the field of CV. However, neural network classifiers can be deceived by adding perturbations or noise. In this paper, a method to reduce the noise interference is proposed so that the hit rate of the neural network will be better when encountering pictures with noise. Our method can directly deploy into unmodified off-the-shelf NNs models and enhance the tolerance of noisy images. This experiment result shows that the Median filter is good at removing Salt-and-pepper noise, and this proposed can increase about 26% error tolerance.*
Keywords: Fault Tolerance, Reliability, Neural Network, Computer Vision

1. **Introduction.** Inspired by recent success in machine learning, NNs have been widely adopted in many applications, such as computer vision, speech recognition, and natural language processing, which have exhibited impressive performance in these tasks. In computer vision, NNs have a great impact on self-driving cars. The speed of identifying road conditions is a matter of life and death; also, it will affect the company's goodwill quite large. Another example is the Face ID, as the Face ID has become a basic piece of equipment used on mobile phones, Zhang et al. [1] suggested a new face tracking framework based on detection, tracking and prediction. Phone security is more and more important since some personal information is stored in it. To get the information, hackers may use some techniques to deceive face recognition. Recent studies show that NN-based image classifiers can be fooled by adversarial examples, and will cause the trained model to misclassify. In [2], Xu et al. had propose a residual separable convolutional neural network to improve the low accuracy in facial expression recognition.

Practical methods such as adversarial training can be extremely effective in causing misclassification. Adversarial training is the easiest and most brute-force way to defend against these attacks. It pretends to be the attacker, generates some adversarial examples

against the network, and then explicitly trains the model to not be fooled by them. On the defensive side, there is a model being trained that makes attackers difficult to discover adversarial input tweaks that lead to incorrect categorization.

This study is interested in the reliability of the convolution neural network with noise effects. We design a noise generator and then use several adversarial noise algorithms that can against the perturbations in images as illustration in Fig 1. Our main goal is to decrease these error classification caused by the noise. First, use Modified National Institute of Standards and Technology (MNIST) dataset, which do not contain noises, to train a NN model. Then, use noise generators on the MNIST dataset to produce two types of noise images. Secondly, in order to enhance the fault tolerant ability of NN models, our purpose is trying to restore the images using filters and input these images to the already trained model for predictions. It is noticed that the proposed method does not need altering the NN models. Finally, analyze the prediction of the result.

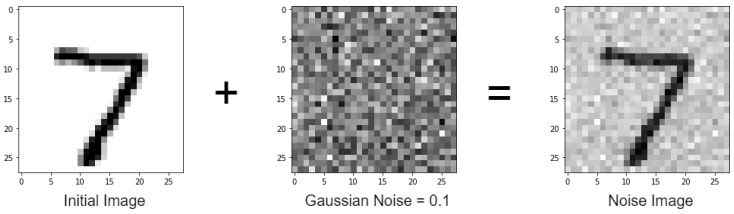


FIGURE 1. Schematic drawing of the image with perturbation

2. **Related works.** In many recent studies, deep neural network (DNN) was demonstrated to be deceived by the adversarial example [28], which is basically a small perturbation of original input. Therefore, the machine learning security is getting more and more important in the security community. In the very first, many researchers have focused on some traditional classifiers with powerful attack techniques [3, 5, 13, 29, 30, 31, 33]. These works have been proposed in order to better understand the effect of perturbation. In the same time, powerful defense techniques were accordingly proposed in existing literature or articles [6-10, 14, 21, 23] to against these adversarial attacks. Unfortunately, current state-of-the-art DNNs cannot suitable for these attack and defense techniques since these techniques cannot be directly applied to DNN-based classifiers. Hence, researchers have started paying attention to the security of DNNs. With the growing of studies on security, some attacks techniques on DNNs have been developed in [16, 18, 22, 24, 27, 32, 37]. Szegedy et al., [37] found that several state-of-the-art machine learning models are vulnerable to adversarial examples. The learning models may classify those correct examples into incorrect ones due to the existence of noises, which are slightly difference from original example. Moreover, Szegedy et al., also mentioned that because of the nonlinearity of deep neural networks, adversarial examples will target the fundamental blind spots of training models. This means although there are many different models using variety of algorithms and architectures, misclassification still happened in the same adversarial examples. Furthermore, improving the robustness of deep networks with a straightforward defense technique through retraining models by adding possible adversarial examples is a new way to overcome the adversarial attacks [16, 24, 25, 32, 34, 35].

Goodfellow et al.,[16] showed that the radial basis activation function is more likely to against adversarial examples. However, implementation of radial basis activation will cost huge effort like modifying the current architecture or a lot of knowledge of adversarial examples. At the same time, they also demonstrated that adversarial training would

result in regularization but in fact, general regularization strategies such as dropout, pre-training, and model averaging can only reduce a model's vulnerability by a certain level, and sometimes it even has no effect. Besides, Gu and Rigazio [19] provide a training strategy to improve the robustness of DNNs based on the structure of adversarial examples. Also, a solution called deep contractive network, a model with a new end-to-end training procedure that includes a smoothness penalty has been proposed to fix networks when again being attacked by new adversarial examples with smaller distortion. Although this work can increase the network robustness to adversarial examples without a significant performance penalty, which however limits the capacity of deep contractive networks compared to traditional DNNs and the cost of studying adversarial example is not suitable for the current design. Chalupka et al., [11] have proposed a manipulator function which uses causal reasoning to improve on the boundaries of a standard, correlational classifier. It, however, did not figure out a practical solution to tolerant noises.

Because retraining the target models would cost too much effort and once other new adversarial examples were encountered the training model is going to fail. We have to figure out an innovative way to help DNNs overcome adversarial attacks rather than just pre-trained a specified model. Papernot et al., [36] improved DNN's resilience to adversarial samples by investigating the use of distillation, which is a technique for reducing the dimensionality of DNN. Although the overhead of distillation is very low and it is easy to implement, the impact on other DNN models remains a question. Which mean this method is not suitable for all general DNN models. [38] This study presents a method based on deep learning to detect contraband in X-ray images.

Some studies have showed interests in how to detect the adversarial examples. Xu et al., [39] proposed a Feature Squeezing method to detect adversarial examples in advance. But this work needs numerous pre-generated adversarial examples to achieve high performance. Hence, it still remains a big challenge for those unknown attacks. Similarly, Grosse et al., [17] use statistical method to detect adversarial examples by estimating the data distribution of adversarial examples and benign examples. These two works are not good enough for arbitrary cases since both of them need large number of adversarial examples. Some studies [4, 15, 20] propose pre-trained models to achieve high performance using binary model or principle component analysis (PCA) techniques, even directly train the detector; but these all need lot of effort to build the pre-trained model, and they do not achieve the same result while facing unexpected attacks. In articles [12], [26], they both introduced a novel way to distinguish the examples. Then do prediction based on the categorization of example, adversarial or benign. However, these methods will cost too much computation resources and time.

For the reasons above, an innovative way was proposed to filter the noise of the images in order to reduce misrecognition, and then choose the convolution neural network (CNN) as our main classifier to distinguish different digits. On the other hand, when choosing the NNs model for CV identification, often the hardest part is to find the right discriminator for the job. Different discriminators are suited for different types of data and different problems. Our goal is to recognize road signs, pedestrians, and emergency accidents in an effective and efficient way. Moreover, when facing some secure problems, we do not want the trained model to be fooled by the perturbations to cause misclassification.

3. **Proposed method.** This study used Google Tensorflow to implement the CNN, as shown in Fig. 2. Each input image is a 1-D vector of 784 features (28×28 pixels), the convolution layer is made up of 16 filters, so that, it will produce 16 images with different characteristic feature maps. The downsampling layer is to shrink the 28×28 pixels into 14×14 pixels for reducing the training time and processing data, it also can solve the

overfitting problem. Following are two convolution layers and two downsampling layers that build up the flatten layer. The flatten layer reshapes the 36 2-D 7×7 vectors into 1 1-D 1764 vectors as the input data for neural. The hidden layer contains 128 neurons and the output layer is built up of 10 neurons corresponding to digits 0 to 9.

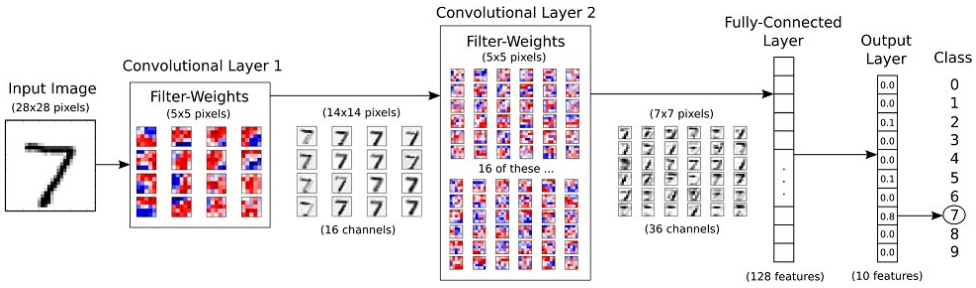


FIGURE 2. Schematic drawing of the Neural Network Architecture

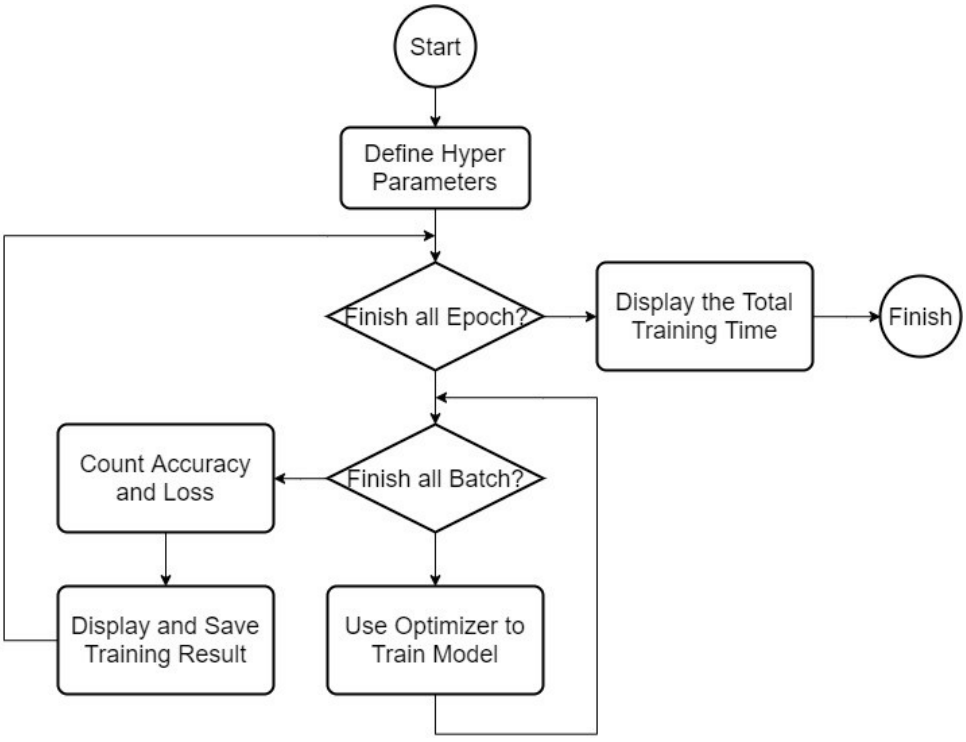


FIGURE 3. Training flowchart of the Convolutional Neural Network

Figure 3. is the CNN training process, our training data contained 55000 images, each batch has 100 images, the total batch is 550 batches and was executed for 10, 20,and 30 epochs. We use "Adam Optimizer" as our optimizer and "reduce mean" as our loss function. There are still lots of optimizer and loss functions that can be chosen to train the network model. For different kinds of identification objects, the training model must be customized to aim for better results.

3.1. **Noise.** Image noise is a random variation of brightness or color information in figures and is usually an aspect of electronic noise. Noise can be produced by the image sensor and circuitry of a scanner or digital camera. Noises in this paper are: i) Gaussian noise, which is statistical noise having a probability density function equal to that of the normal

distribution. ii) Salt-and-pepper noise, which is also known as impulse noise. This noise can be caused by sharp and sudden disturbances in the image signal, as shown in Fig. 4.

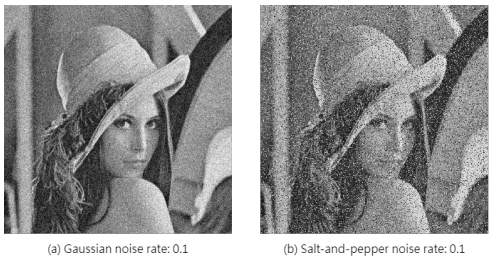


FIGURE 4. Image with different types of noise and rate

3.2. **Noise removal filter.** Following are the noise reduction filters that were used in our experiment. Noise cleaning can use neighbor-hood spatial coherence or neighborhood pixel value homogeneity. The basic idea in our method is "Box filtering", algorithm 1, which involves replacing each pixel of an image with the average in a box. Another noise reduction technique is "Median Filtering", algorithm 2. Median filter calculates the median of all pixels under the kernel window and replaces the central pixel with this median value. It is highly effective in removing Salt-and-pepper noise.

Algorithm 1 Box Filtering

```
for each test image do
    Sum = Add all pixel value in the filter;
    Result = Divide the Sum by filter size;
    Replace all pixel value in the filter with the Result;
end for
```

Algorithm 2 Median Filtering

```
for each test image do
    List = Sort the pixel values in the filter;
    Med = Select the median in the List;
    Replace all pixel value in the filter with Med;
end for
```

3.3. **Noise removal CNN.** Based on the traditional CNN, by adding the noise removal technique to reduce the noise affecting our trained model, there is no need to retrain the neural network, and we don't need to have background knowledge about CNN. It is quite easy to alter the filter for a different kind of noise and increase the tolerance ability of CNN quite significantly. In Fig. 5, two kinds of noise will be added to test images. Following, in Fig. 5 (c), these images with noises will pass through noise filters. However, in Fig. 5 (b) there is no noise filters. Finally, compare two predicted results with/without the noise removal filter.

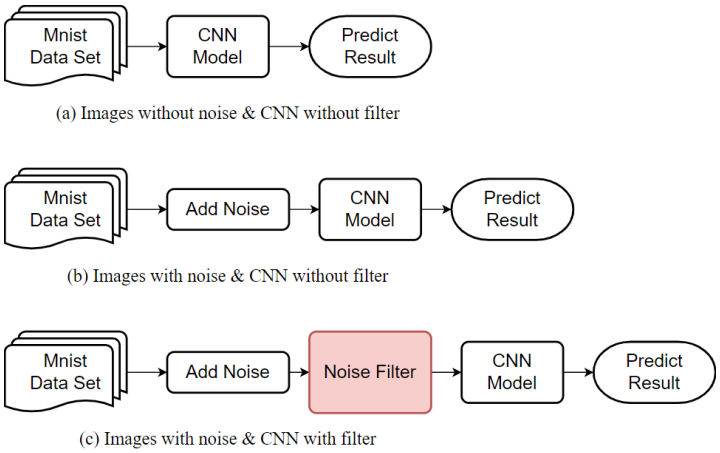


FIGURE 5. Flowchart of the CNN predicting

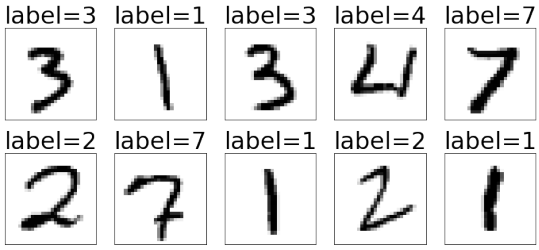
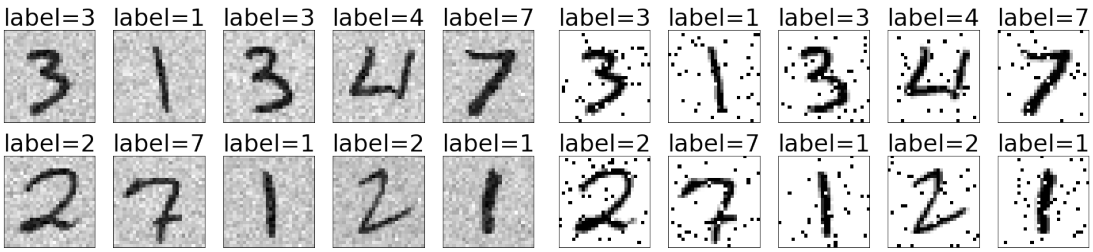


FIGURE 6. MNIST handwritten digits database

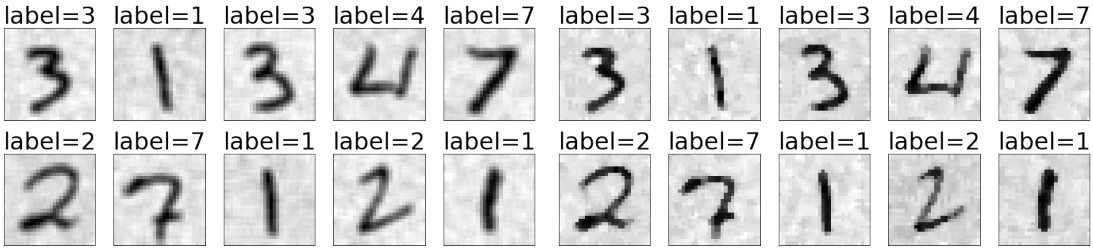


(A) With Gaussian noise (B) With Salt-and-pepper noise

FIGURE 7. Digits with different types of noise

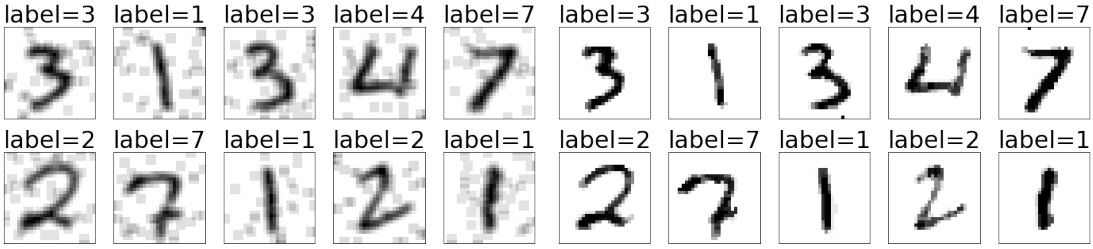
4. **Experimental result.** The MNIST database of handwritten digits has a training set of 60,000 images test set of 10,000 images. It is a good database for people who want to try learning techniques and pattern recognition methods on real-world data. Figure 6 shows the original digits images. Figure 7 (A) and (B) show the digits images with adding Gaussian noise and Salt-and-pepper noise. Figure 8 (A)-(D) show the digits images after noise reduction. In Fig. 8 are blurring than the initial images as well as in Fig. 7, but the impact of noise is less. Thus, CNN can classify the digits more accurately, without modifying the model. We applied our methodology on Google colab platform with GPU. Colaboratory is a free Jupyter notebook environment provided by Google. You can use free GPUs and TPUs to solve all these issues on it. The batch size of CNN is 100, the learning rate is 0.0001 and the dropout probability is 0.2.

Since the noise reduction will cause the images to become blurry, when the noise is slight, we don't recommend using the filter. Furthermore, we have figured that the Median



(A) Digits after noise reduction: Gaussian noise and Box filter

(B) Digits after noise reduction: Gaussian noise and Median filter



(C) Digits after noise reduction: Salt-and-pepper noise and Box filter

(D) Digits after noise reduction: Salt-and-pepper noise and Median filter

FIGURE 8. Digits after noise reduction

TABLE 1. Experiment result: Gaussian with epoch=10

Train epochs=10	No Noise	Gaussian		
Amplitude	-	0.1	0.2	0.3
No filter	258	302	412	483
Box filter	-	297	384	472
Median filter	-	303	390	477

TABLE 2. Experiment result: Salt-and-pepper with epoch=10

Train Epochs=10	No Noise	Salt-and-pepper		
Amplitude	-	0.1	0.2	0.3
No filter	258	509	1263	2607
Box filter	-	531	1165	2381
Median filter	-	336	427	705

filter is good at removing Salt-and-pepper noise. Table 1 and Table 2 show the predicted results for adding Gaussian noise and Salt-and-pepper noise with 10 training epochs. In Table 3 and Table 4, are 30 training epochs. The value in the tables are error predict numbers, which total testing images number is 10000. As the training epochs get higher, the accuracy of CNN gets higher, and the effect of noise also becomes higher. Therefore, the error tolerance can increase about by 26%, relative to that of only uses CNN model.

To conclude, compared with the other defense techniques, our method provides two advantages. First, it is adaptive, we can change the filter for different kinds of noise or perturbation attacks. Second, the method is not attack-specific and thus holds great promise for unknown attacks. The method can be directly integrated with a trained model, without retraining or modifying the model. The recognition error tolerance also can be improved obviously in our experimental result.

Table 3. Experiment result: Gaussian with epoch=30

Train Epochs=30	No Noise	Gaussian		
Amplitude	-	0.1	0.2	0.3
No filter	124	147	204	374
Box filter	-	140	250	338
Median filter	-	153	203	367

Table 4. Experiment result: Salt-and-pepper with epoch=30

Train Epochs=30	No Noise	Salt-and-pepper		
Amplitude	-	0.1	0.2	0.3
No filter	124	375	1180	2692
Box filter	-	325	905	1985
Median filter	-	219	305	542

5. **Conclusions.** This paper presents a straightforward method by adding filters before the neural network model so that the perturbations can be regarded as a kind of noise interference, and be effectively removed. Our scheme does not need to retrain or modifying the neural network models; also it can against distinct noise with various filters.

The experiments show that our method can promote reliability of the CNN, the next step is tried to detect the perturbations category. If perturbations can be classified in the early stage, different filter may be chosen to remove the perturbations.

REFERENCES

[1] Y.-J. Zhang, K. Zhang and Z.-M. Lu, Face tracking based on convolutional neural network and kernel correlation filter, *Journal of Network Intelligence*, vol. 6, no. 2, pp. 247-254, May 2021.

[2] X. Xu, J. Cui, X. Chen, and C.-L. Chen, A Facial Expression Recognition Method based on Residual Separable Convolutional Neural Network, *Journal of Network Intelligence*, vol. 7, no. 1, pp. 59-69, Feb 2022.

[3] B. Marco, N. Blaine, S. Russell, et al., Can machine learning be secure? *Proc. of the 2006 ACM Symposium on Information, Computer and unications Security, ASIACCS '06*, New York, USA, pp.16–25, 2006.

[4] A. N. Bhagoji, D. Cullina, P. Mittal, Dimensionality reduction as a defense against evasion attacks on machine learning classifiers, arXiv rint arXiv:1704.02654, 2(1).

[5] B. Battista, C. Igino, M. Davide, et al., Evasion attacks against machine learning at test time, *Lecture Notes in Computer Science*, pp.387–402, 2013.

[6] B. Battista, F. Giorgio, and R. Fabio, Multiple classifier systems for adversarial classification tasks, *MCS 2009. Lecture Notes in Computer nce*, vol. 5519, Springer, Berlin, pp.132–141, June 2009.

[7] B. Battista, F. Giorgio, R. Fabio, Multiple classifier systems for robust classifier design in adversarial environments, *Int. Journal of ine Learning and Cybernetics*, pp.27–41, Dec. 2010,1(1).

[8] B. Battista, F. Giorgio, R. Fabio, Multiple classifier systems under attack, *MCS 2010, Lecture Notes in Computer Science*, Springer, Berlin.5997, pp.74–83.

[9] B. Michael, K. Christian, S. Tobias, Static prediction games for adversarial learning problems, *J. Mach. Learn. Res.*, pp.2617–2654, September 2012.

[10] B. Michael, S. Tobias, Stackelberg games for adversarial prediction problems, *proc. of the 17th ACM SIGKDD international conference on ledge discovery and data mining*, pp.547–555, August 2011.

[11] C. Krzysztof, P. Pietro, E. Frederick, Visual causal feature learning, *Proc. of the thirty-first conference on uncertainty in artificial lligence*, pp.181–190, July 2015.

[12] F. Reuben, R.C. Ryan, S. Saurabh, et al., Detecting adversarial samples from artifacts, https://doi.org/10.48550/arXiv.1703.00410, 2017.

[13] F. Matt, J. Somesh, R. Thomas, Model inversion attacks that exploit confidence information and basic countermeasures, *Proc. of the 22nd SIGSAC Conf. on Computer and Communications Security*, pp.1322–1333, October 2015.

[14] G. Amir, R. Sam, Nightmare at test time: Robust learning by feature deletion, *Proc. of the 23rd Int. Conf. on Machine Learning, ICML '06*, York, NY, USA, pp.353–360, 2006.
[15] G. Zhitao, W. Wenlu, W.S. Ku, Adversarial and clean data are not twins, arXiv preprint arXiv:1704.04960,2017, https://doi.org/10.48550/v.1704.04960.
[16] I. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, arXiv 1412.6572, 2014.
[17] K. Grosse, P. Manoharan, N. Papernot, et al., On the (statistical) detection of adversarial examples, http://arxiv.org/abs/.06280, 2017.
[18] K. Grosse, N. Papernot, P. Manoharan, et al., Adversarial examples for malware detection, *Computer Security – ESORICS 2017 Cham* ,Springer Int. Publishing, pp.62–79.
[19] S. Gu, L. Rigazio, Towards deep neural network architectures robust to adversarial examples, *Int. Conf. on Learning Representations shop*, 2014.
[20] D. Hendrycks, and K. Gimpel, Early methods for detecting adversarial images, *Int. Conf. on Learning Representations Workshop*, 2017.
[21] A. Kantchelian, S. Afroz, L. Huang, et al., Approaches to adversarial drift, *Proc. 2013 ACM Workshop on Artificial Intelligence and security*, New York, USA, pp.99-110, Nov. 2013.
[22] C.Kereliuk, L. B. Sturm, and J. Larsen, Deep learning and music adversaries, *IEEE Transactions on Multimedia*, vol. 17, pp.2059-2071, Nov. 2015.
[23] A. Kołcz, and C.H. Teo, Feature weighting for improved classifier robustness, *Conf. on Email and Anti-Spam*, Mountain View, CA, USA, 2009.
[24] A. Kurakin, I. Goodfellow, and S. Bengio, Adversarial examples in the physical world, *Int. Conf. on Learning Representations*, July.
[25] A. Kurakin, I. Goodfellow, and S. Bengio, Adversarial machine learning at scale, *Int. Conf. on Learning Representations*, Toulon, ce, Nov. 2016.
[26] X. Li, and F. Li, Adversarial examples detection in deep networks with convolutional filter statistics, *IEEE Int. Conf. on Computer on, Venice*, Italy, pp.5775-5783, Dec. 2017.
[27] B. Liang, H. Li, M. Su, et al., Deep text classification can be fooled, *Proc. Int. Joint Conf. on Artificial Intelligence*, pp.4208-4215, July 2018.
[28] B. Liang, H. Li, M. Su, et al., Detecting adversarial image examples in deep neural networks with adaptive noise reduction, *IEEE transactions on Dependable and Secure Computing*, vol. 18, no. 1, pp.72-85, 2018.
[29] B. Liang, M. Su, W. You, et al., Cracking classifiers for evasion: A case study on the google's phishing pages filter, *Proc. Int. on World Wide Web*, Republic and Canton of Geneva, pp.345–356, 2016.
[30] D. Lowd, and C. Meek, Adversarial learning, *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery in Data Mining*, New York, USA, pp.641-647, Aug.
[31] M. Davide, C. Igino, and G. Giorgio, Looking at the bag is not enough to find the bomb: An evasion of structural methods for malicious files detection, *Proc. ACM SIGSAC Symposium on Information, Computer and Communications Security*, New York, NY, USA, pp.119–130, May 2013.
[32] M.D. Seyed-Mohsen, F. Alhussein, F. Pascal, Deep-fool: a simple and accurate method to fool deep neural networks, *Proc. of the IEEE on Computer Vision and Pattern Recognition*, pp.2574-2582, 2016.
[33] B. Nelson, M. Barreno, F.C. Jack, et al., Exploiting machine learning to subvert your spam filter, *Proc. Usenix Workshop on e-Scale Exploits and Emergent Threats*, USA, pp.1-9, Apr. 2008.
[34] N. Blaine, B. Marco, J.C. Fuching, et al., Exploiting machine learning to subvert your spam filter, *Proc. of the 1st Usenix Workshop on e-Scale Exploits and Emergent Threats*, no. 7, pp.1–9, April 2008.
[35] Z.-M. Chan, C.-Y. Lau and K.-F. Thang, Visual Speech Recognition of Lips Images Using Convolutional Neural Network in VGG-M Model, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 11, no. 3, pp. 116- 125, Sep.tember 2020.
[36] P. Nicolas, M. Patrick, W. Xi, et al., Distillation as a defense to adversarial perturbations against deep neural networks, *IEEE symposium on Security and Privacy*, pp.582-597, 2016.
[37] C. Szegedy, L. Wei, J. Yangqing, et al., Going deeper with convolutions, *Proc. of the IEEE Conf. on Computer Vision and Pattern recognition*, pp.1-9, 2015.
[38] H. Chen and Z. M. Lu, Contraband detection based on deep learning, Journal of Information Hiding and Multimedia Signal Processing, Vol. 13, No. 3, pp. 165-177, September 2022.

[39] X. Weilin, E. David, and Q. Yanjun, Feature squeezing: Detecting adversarial examples in deep neural networks, *Proc. Network and Distributed Systems Security Symposium*, San Diego, CA, USA, pp.18-21, Feb. 2018.