Generating Adversarial Examples in Audio Classification with Generative Adversarial Network

Qiang Zhang
Graduate School
Army Engineering University
Nanjing, Jiangsu, China
zq308297543@126.com

Jibin Yang
Command and Control Engineering College
Army Engineering University
Nanjing, Jiangsu, China
yjbice@sina.com

Xiongwei Zhang
Command and Control Engineering College
Army Engineering University
Nanjing, Jiangsu, China
xwzhang9898@163.com

Tieyong Cao
Command and Control Engineering College
Army Engineering University
Nanjing, Jiangsu, China
cty_ice@sina.com

2022 7th International Conference on Image, Vision and Computing (ICIVC) | 978-1-6654-6734-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICIVC55077.2022.9886154

***Abstract*—To improve the performance of acoustic adversarial examples, this paper proposes an adversarial generation model based on Generative Adversarial Network (GAN) for audio classification. By introducing the classification model into GAN, this paper proposes a general GAN framework to execute adversarial attacks for audio classification. Then we propose a Short-time Synthesis GAN-based (SSGAN) attack method, which can reduce the complexity of audio adversarial example generation, and further improve the generality and performance of the GAN-based audio adversarial example generation. Experiments on audio classification datasets such as UrbanSound8k and ESC50 show that compared with existing audio adversarial example generation methods, the proposed method generates adversarial examples with lower perceptibility, and has a higher attack success rate and attack efficiency for typical audio classification models.**

Keywords—generative adversarial network, adversarial example, audio classification

I. INTRODUCTION

With the widespread use of intelligent acoustic systems in industry, transportation and other infrastructures, the security of acoustic systems is gradually gaining attention. As one of the important applications, audio classification models have adopted many deep learning-based implementations [1]–[3]. However, it has been shown that deep learning-based acoustic models are vulnerable to attacks from adversarial examples [4]. Unlike the large number of improvements emerging in computer vision [5]–[12], the generation of adversarial examples in audio classification models has not been well addressed. Some typical methods such as FGSM [13], genetic algorithm [14], and particle swarm optimization algorithm [15] have been tried for acoustic adversarial example generation. The gradient-based iterative optimization adversarial example generation algorithms have also been proposed in [16] to attack audio classification models. However, these methods can only optimize perturbation for specific samples and generate adversarial examples with low efficiency. Furthermore, the adversarial examples are still easy to perceive by human.

Generative Adversarial Network (GAN), as a false sample generation technique, can not only effectively improve the recognition ability of discriminative networks, but also can be used for adversarial example attacks in image and speech processing [11], [17]. Generating adversarial examples using GAN is generic and can reduce the difficulty of applying adversarial examples. However, the generation of audio adversarial examples is still challenging due to the much larger search space of samples than images [13]. In this paper, we propose a Short-time Synthesis GAN-based (SSGAN) attack method to generate adversarial examples for audio classification. Using SSGAN, we can reduce the difficulty of adversarial example generation by optimizing the sample input scheme. The experimental results show that SSGAN has a strong attack capability on the state-of-the-art audio classification models, and the generated adversarial examples are almost indistinguishable from real samples. The contributions of this paper are summarized as follows.

First, to the best of our knowledge, it is the first time to integrate GAN and the victim classifier into one framework to generate adversarial examples in audio classification.

Second, we optimize the adversarial generation framework with the short-time synthesis method, which can reduce the complexity to generate adversarial examples, and further improve the generality and performance of the GAN-based audio adversarial examples generation.

Third, compared with the existing gradient-based iterative optimization algorithm, SSGAN greatly improves the efficiency of adversarial attacks and reduces the perceptibility of perturbations while maintaining a higher attack success rate, making the adversarial-training-based defense measures implementable.

The rest of this paper is organized as follows: Section II briefly reviews the related work. Section III introduces the basic model for audio adversarial example generation. Section IV details the proposed SSGAN attack method. Section V describes the experimental results. Section 6 concludes this work.

II. RELATED WORK

In this section, we review GAN and adversarial example generation in the field of audio classification.

A. Generative Adversarial Network

GAN [18] is composed of a generator and a discriminator, and its essential idea is a game between both components. GAN has been successfully used in many vision tasks, such as image generation, image-to-image conversion, and text-to-image

This work is partly supported by NSFC project under Grant 62071484.

synthesis. Recently, GAN has been gradually employed in adversarial example generation tasks. For example, [11] used GAN to generate adversarial examples of real images to attack image classification models. Reference [17] used one-dimensional convolutional GAN to generate fixed-length adversarial examples for short-time speech and music styles classification. However, generating fixed-length adversarial examples is not universal for other datasets containing samples of various lengths. As the first effort to utilize GAN to generate adversarial examples for audio classification, we generate adversarial examples using GAN with a two-dimensional convolutional structure, which is widely applied in audio classification [3].

B. Adversarial Examples for Audio Classification

The study of adversarial examples in audio classification started late, and only recently is reported practicing in urban soundscape classification tasks. Reference [16] uses the gradient-based iterative optimization algorithm to generate adversarial examples for UrbanSound8k dataset [19], the essential idea of which is derived from the iterative attack method to generate universal adversarial perturbations [12] in the image domain. This method seeks perturbations by solving complex optimization problems, and generates adversarial examples inefficiently. Furthermore, the energy of adversarial perturbations is usually large, making adversarial examples easily perceived.

III. Basic Model

Suppose a white-box undirected attack scenario, let X be the audio waveform space and Y be the label space. A trained audio classifier $f(\cdot)$ (hereinafter called the victim classifier $f(\cdot)$) judges the sample $x \in X$ as its true class, i.e., $f(x)=y$. The purpose of the adversarial attack on $f(\cdot)$ is to generate an adversarial example x_{adv} that can effectively deceive it, i.e., $f(x_{adv})=y^*$, $y^* \neq y$, where y^* is a reference label different from the true label y. While effectively deceiving $f(\cdot)$, x_{adv} should remain as similar as possible to x in human auditory perception, i.e., a normal person cannot distinguish between x_{adv} and x by hearing. Therefore, it is necessary to find a small perturbation δ which satisfies $x_{adv}=x+\delta$ and $f(x_{adv}) \neq y$. Thus, the adversarial example generation problem can be transformed into the following optimization problem.

$$\delta^*=\arg_\delta \min\, l(f(x+\delta), y^*),\ s.t.:\|\delta\|<\varepsilon \tag{1}$$

where $l(\cdot)$ is the loss function to evaluate the classification accuracy. $\|\cdot\|$ is the norm operator, e.g., l_1, l_2, …, l_∞. ε represents the maximum of allowed perturbation amplitude. δ^* is the trained small perturbation.

IV. Proposed Approach

Different from the construction of universal adversarial perturbations, a general GAN framework to generate different adversarial perturbations for different samples is proposed. In this section, we will detail the proposed SSGAN in terms of four aspects: the general framework, the network structure, the loss function, and the short-time adversarial example design method.

A. General Framework

Fig. 1 illustrates the proposed general framework. The perturbation $\delta=G(x)$ is generated using the generator G, and the adversarial example x_{adv} is obtained by adding x and δ.

$$x_{adv}=x+G(x) \tag{2}$$

Then, x_{adv} and x are fed into the discriminator D, which distinguishes them and drives G to generate δ that is imperceptible to D. After training, for a given input, the adversarial example can be efficiently generated using the obtained G^*. In contrast to the basic GAN [18], the victim classifier $f(\cdot)$ is combined with D to perform judgments in the framework. $f(\cdot)$ is involved in guiding G generating x_{adv}, but the parameters of $f(\cdot)$ are not updated.

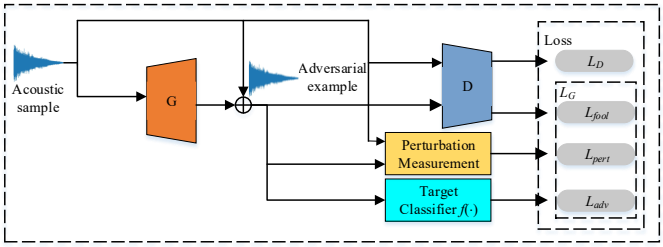


Fig. 1. General GAN framework for generating audio classification adversarial examples.

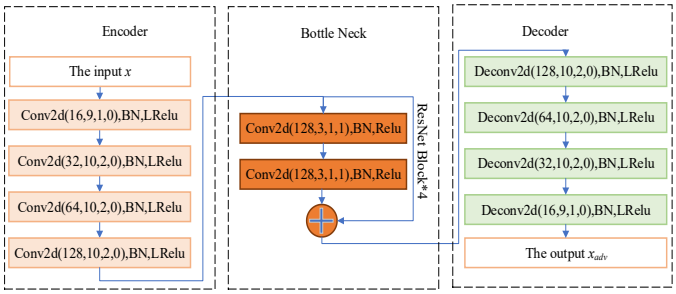


Fig. 2. The structure of the generator. The Conv2d(a,b,c,d)/Deconv2d(a,b,c,d) means the 2-dimensional convolutional/deconvolutional structure, where a denotes the number of channels, b denotes the kernel size, c denotes the stride size, d denotes the padding size.

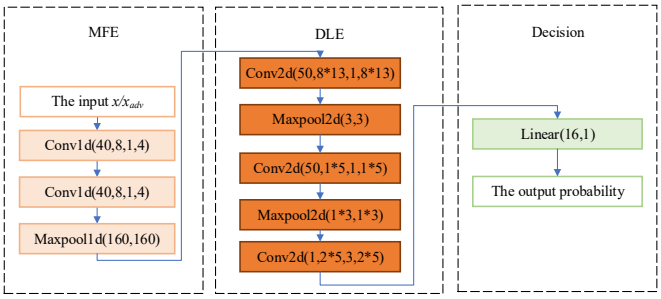


Fig. 3. The structure of the discriminator. The Conv1/2d(a,b,c,d) means the 1/2-dimensional convolutional structure, where a denotes the number of channels, b denotes the kernel size, c denotes the stride size, d denotes the padding size. The Maxpool1/2d(e, f) means the 1/2-dimensional maxpooling operation, where e and f denote the maxpool size.

B. Network Structure

In the proposed framework, G is implemented using a combined structure of codecs and residual modules. The

effectiveness of this structure has been verified in image transformation [20]. D is composed of three modules: Mel-like Feature Extracting (MFE), Deep Embedding Learning (DEL), and Decision, which follows the design idea of EnvNet [21]. The detailed network structures of G and D are shown in Fig. 2 and Fig. 3.

C. Loss Function

The loss function is a key factor in training GAN to get a generator that can deceive $f(\cdot)$. We carefully design the loss functions for G and D, respectively.

D loss: To enhance the ability of D to determine truth/falsehood about x/x_{adv}, we define L_D using the Mean Squared Error (MSE) loss.

$$L_D=-\mathrm{E}_x[(1-\mathrm{D}(x))^2]-\mathrm{E}_x[(\mathrm{D}(x_{adv}))^2] \tag{3}$$

where $\mathrm{E}(\cdot)$ denotes the mathematical expectation. $\mathrm{D}(x)$ and $\mathrm{D}(x_{adv})$ indicate the probability that x or x_{adv} is the real sample. The closer to 1 the probability is, the more likely the input is a real sample. The closer to 0 the probability is, the more likely the input is an adversarial example. D aims to correctly discriminate x and x_{adv}, so it expects that $\mathrm{D}(x)$ is close to 1 and $\mathrm{D}(x_{adv})$ close to 0.

G loss: G expects to generate δ that can effectively deceive D while being less susceptible to auditory perception, for which L_G is designed as follows.

$$L_G=L_{fool}+\alpha L_{adv}+\beta L_{pert} \tag{4}$$

where L_{fool} is the adversarial loss, which represents the ability of G to deceive D. L_{adv} is the victim classifier loss, which represents the ability of G to attack $f(\cdot)$. L_{pert} is the perturbation loss, which represents the magnitude of the perturbation. L_G combines the three losses with the expectation that x_{adv} can achieve higher attack ability, deception ability and lower perturbation energy. The hyperparameters α, β are the weights to balance the importance of the three losses.

1) Adversarial loss L_{fool}

To encourage G to improve the ability to mislead D, we use MSE loss to define L_{fool}.

$$L_{fool}=\mathrm{E}_x[(1-\mathrm{D}(x_{adv}))^2] \tag{5}$$

We can see that minimizing L_{fool} is equivalent to converging $\mathrm{D}(x_{adv})$ to 1, which essentially instructs x_{adv} to simulate the real audio data distribution.

2) Classifier loss L_{adv}

A primary goal of G is to deceive $f(\cdot)$, so L_{adv} should be able to measure the difference between the true label of x_{adv} and the reference label. Intuitively, we define L_{adv} using the predicted logical values $\boldsymbol{p}$ for x_{adv}.

$$L_{adv}=\mathrm{E}_x[\max(p_y-\max_{j\neq y}p_j+m,0)] \tag{6}$$

where $\boldsymbol{p}=f(x_{adv})=(p_1,\ldots,p_j,\ldots,p_C)$, p_j represents the probability that x_{adv} belongs to class j, C is the number of classes. We can see that minimizing L_{adv} encourages $f(\cdot)$ to classify x_{adv} as class $j\neq y$, and that p_j is at least m greater than p_y.

3) Perturbation loss L_{pert}

The goal of L_{pert} is to ensure that the constructed adversarial examples are similar to the real samples. We define it in terms of l_2 norm of the adversarial perturbation δ.

$$L_{pert}=\mathrm{E}_x[\|x_{adv}-x\|_2] \tag{7}$$

L_{pert} controls the perturbation energy and avoids the perturbation explosion, which means GAN tends to choose δ with smaller magnitude when training.

Finally, the trained generator G^* is obtained by solving the following min-max optimization problem with L_D, L_G combined.

$$\mathrm{G}^*=\arg\min_{\mathrm{G}}\max_{\mathrm{D}}(L_G+L_D) \tag{8}$$

D. Design of Short-Time Adversarial Examples and SSGAN

Due to the long duration of audio sample, the search space for adversarial perturbations is very large, which means it is extremely time-consuming to create adversarial examples by directly attacking the entire audio sample. Furthermore, training GAN requires a large amount of data with a consistent data length, but the sample durations are not consistent across audio datasets. To solve this problem, we propose a Short-time Input Processing (SIP) scheme, which allows us to focus on the generation of adversarial examples in short-time segments. We incorporate the SIP scheme into the GAN-based ESC adversarial example generation model, and propose the SSGAN method. SSGAN can generate adversarial segments of flexibly adjustable lengths and adapt to real input signals of different lengths. The details of the method are shown in Fig. 4.

During training, a batch of B audio segments of fixed t_1 seconds are randomly clipped from B real samples of various lengths. If there is a segment less than t_1 seconds, the section from the corresponding real sample will be padded into that segment to ensure that all segments have the same duration. Then these B segments are fed into GAN to obtain the corresponding t_1-second adversarial segments. During testing, considering a real sample with t_2 seconds, it will be clipped into j t_1-second segments without overlap. With these j segments as inputs, the trained G^* will generate the corresponding adversarial segments. These segments are then stitched together and restored to the adversarial example of the same length as the real audio signal.

V. Simulation Experiments and Analysis

This section first introduces the experimental setup. Then, we study the attack performance of SSGAN with audio segments of different lengths to determine the optimal short-time segment length. Further, we compare SSGAN with the gradient-based iterative optimization method to verify its effectiveness. All experiments are implemented on a workstation equipped with an NVIDIA 2080Ti GPU.

A. Experimental Setup

1) Dataset

To evaluate the effectiveness of SSGAN, we select ESC50 [22] and UrbanSound8k dataset for experiments. Details of the datasets are shown in Table I. During experiments, all data are resampled to 16kHz, mono channel, and each sample is normalized using its absolute maximum.

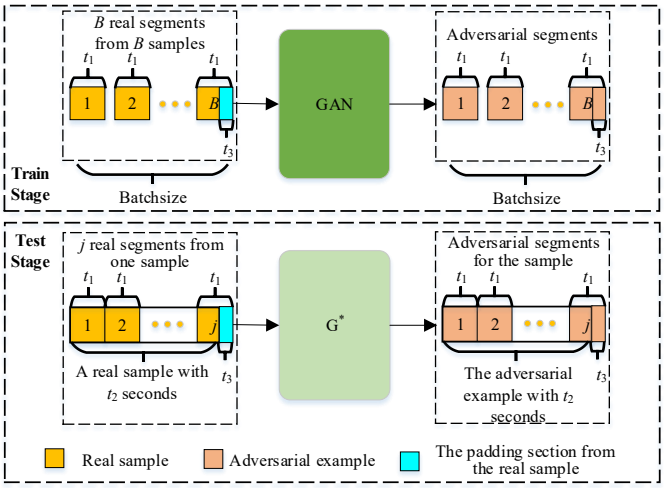


Fig. 4. SSGAN: GAN-based Short-time Synthesis adversarial example generation method.

2) Implementation details

a) Parameter setting: Our experiments show that larger β corresponds to higher perceptual similarity of x_{adv} to the real sample; larger α corresponds to higher attack success rate. To weight the attack success rate and the perceptual similarity, the weight hyperparameters α and β in (4) are set to 10 and 1, respectively. Empirically, the hyperparameter m in (6) is set to 10. We train GAN for 150 epochs with a batch size of 16 and Adam optimizer. The initial learning rate is 0.0001, and decreases by 0.5 times every 10 epochs.

b) Metrics: We use the attack success rate (ASR) to evaluate the attack performance. We cumulate the number ($N_{success}$) of adversarial examples successfully attacking $f(\cdot)$ while the corresponding original samples are accurately predicted, and difine N_{right} as the number of original samples correctly classified. ASR can be calculated as follows.

$$\mathrm{ASR}=N_{success}/N_{right} \tag{9}$$

To assess the similarity between the adversarial example and the real signal, we use the Signal-to-Noise Ratio (SNR) defined in [16].

$$\mathrm{SNR}(x_{adv})=20\log_{10}(\mathrm{P}(x)/\mathrm{P}(\delta)) \tag{10}$$

where $\mathrm{P}(\cdot)$ is the power of a signal $\boldsymbol{v}(v_1,\ldots,v_i,\ldots,v_n)$ of length n, defined as follows.

$$\mathrm{P}(v)=(\textstyle\sum_{i=1\sim n} v_i^2/n)^{1/2} \tag{11}$$

In general, SNR reflects the ratio of the average energy of the real signal and the perturbation noise. Obviously, the larger the SNR, the lower the possibility the adversarial example being perceived.

c) Training strategy: In our proposed framework, G and D are trained in an alternating manner, i.e., D is fixed when G is trained, and vice versa. We train D first and then G, so that D can have a reasonable ability to distinguish x_{adv} from x when training starts. We use soft labels and label flipping with a probability of 5% in the training of D.

B. Comparison of Attack Performance with Audio Segments of Different Lengths

On ESC50 dataset, we compare the attack effects of audio segments with different lengths using EnvNet as the victim model, which was replicated in Pytorch. The test accuracy of the reproduced victim model is 82.5%, which is higher than the accuracy of 81.3% in human hearing test [22] and much higher than the accuracy of 64% reported in the original paper. Considering that the sample length of ESC50 dataset is 5 seconds, we select four different lengths of audio segments to train GAN and launch segment synthesis attack experiments. The experimental results are shown in Table II, MSNR is the average SNR of the adversarial examples successfully attacking $f(\cdot)$. ACC is the accuracy of $f(\cdot)$ on the real samples. The bolded value in each column is the maximum of that column.

The comparison reveals that the attack is more effective when the segment length is 1.6 seconds. We analyze that it is because the differences between various audio classes can be better learned by GAN within 1.6 seconds. Then GAN can learn more appropriate perturbations and successfully implement the attack, so the ASR and MSNR are both higher. The audio segment with shorter length will reduce the discriminative property of samples and the network's ability to discriminate between various samples. So the difficulty of the attack increases accordingly. A longer segment length will increase the learning difficulty of the GAN. Compared with shorter segment lengths, the network trained with longer segments is easy to underfit, and the attack effect decreases.

C. Comparison with Gradient-Based Iterative Optimization Scheme

To further verify the effectiveness of SSGAN, we conduct attack experiments using EnvNetv2 [3] as the victim model on UrbanSound8k dataset. With 1.6 seconds audio clips as inputs, we train SSGAN and compare it with two gradient-based iterative optimization algorithms [16]. The results are shown in Table III.

The comparison reveals that our method has a higher ASR and a better SNR. The average time to generate an adversarial example that can successfully attack $f(\cdot)$ is much small (0.03 seconds). To intuitively analyze how close x_{adv} is to x, we compare the corresponding audio waveforms and spectrograms in Fig. 5. From Fig. 5(a)-(d), we can see that the adversarial example learned by SSGAN is close to the real sample in terms of waveforms as well as spectrograms. SSGAN tends to learn high frequency perturbations with low energy. Furthermore, Fig. 5(e) shows that the perturbation amplitude is much smaller. We believe that SSGAN learns the perturbation pattern for a successful attack through the segment synthesis attack.

TABLE I. DETAILS OF THE DATASETS USED FOR EXPERIMENTS

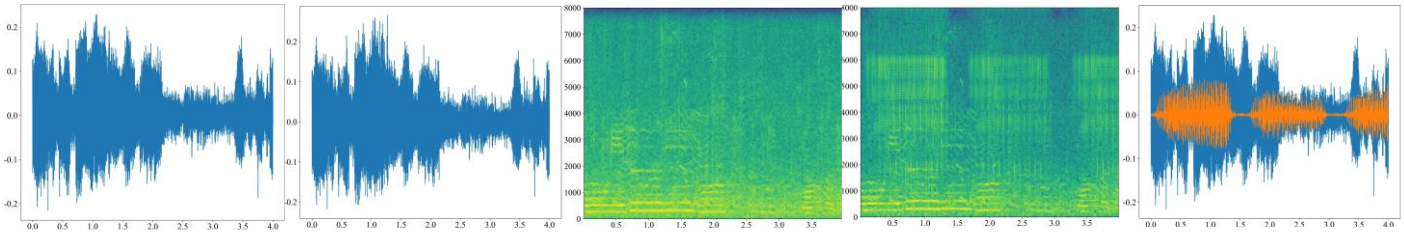
Dataset	Class	Sample	Train	Test	Time	Channel
ESC50	50	2000	1800	200	5s	1
UrbanSound8k	10	8732	7858	874	≤4s	2

TABLE II. ATTACK PERFORMANCE COMPARISON OF SSGAN WITH DIFFERENT LENGTHS OF AUDIO SEGMENTS ON ESC50 DATASET

Time (s)	Train			Test		
	ACC (%)	ASR (%)	MSNR (dB)	ACC (%)	ASR (%)	MSNR
0.8	94.8	98.3	31.2	76.5	97.7	31.0
1.6	**97.6**	**98.7**	**35.7**	**82.5**	**97.8**	**35.7**
3.2	95.3	98.6	34.2	74.5	97.4	34.0
4.0	92.0	97.9	35.1	74.0	97.4	35.0

TABLE III. COMPARISON OF ATTACK PERFORMANCE OF DIFFERENT METHODS ON URBANSOUND8K DATASET. "N/A" INDICATES THE LACK OF COMPARISON DATA

Methods	Train			Test			
	ACC (%)	ASR (%)	MSNR (dB)	ACC (%)	ASR (%)	MSNR (dB)	Time (s)
Iterative[16]	N/A	91.0	N/A	N/A	66.9	25.0	N/A
Penalty[16]	N/A	90.0	N/A	N/A	83.1	18.7	N/A
SSGAN	98.4	**96.7**	33.6	95.2	**96.9**	**33.9**	0.03



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

Fig. 5. Comparison of the real sample x with its corresponding adversarial example x_{adv}. From left to right: (a) and (b) are the waveforms of x and x_{adv}, respectively; (c) and (d) are the spectrograms of x and x_{adv}, respectively; in (e), the waveform of the perturbation between x and x_{adv} is orange, and the waveform of x is blue.

VI. CONCLUSION

In this paper, Generative Adversarial Network (GAN) is used for the first time to generate adversarial examples for audio classification. By introducing the victim classifier model into GAN, we propose a general GAN framework. Then SSGAN is proposed based on the framework. It uses short-time audio segments as inputs, which can reduce the complexity to generate adversarial examples, and further improve the generality and performance of the GAN-based audio adversarial examples generation. Preliminary attack experiments on typical audio datasets show its generalization performance partly, and that SSGAN generates adversarial examples with lower perceptibility and has higher attack success rate and attack efficiency on state-of-the-art audio classification models, compared with existing adversarial example generation methods for audio classification.

REFERENCES

[1] O. A, D. S, Z. H, S. K, V. O, and G. A, "Wavenet: A generative model for raw audio," *SSW*, vol. 2, 2016.

[2] A. Khamparia, D. Gupta, N. G. Nguyen, A. Khanna, B. Pandey, and P. Tiwari, "Sound classification using convolutional neural network and tensor deep stacking network," *IEEE Access*, vol. 7, pp. 7717–7727, 2019, doi: 10.1109/ACCESS.2018.2888882.

[3] T. H. Yuji Tokozume, Yoshitaka Ushiku, "Learning from between-class examples for deep sound recognition," 2018, [Online]. Available: https://openreview.net/forum?id=B1Gi6LeRZ.

[4] C. Szegedy *et al.*, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014, [Online]. Available: https://arxiv.org/pdf/1312.6199v4.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015, pp. 1–11.

[6] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *International Conference on Learning Representations*, 2019, pp. 1–14, doi: 10.1201/9781351251389-8.

[7] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE Symposium on Security and Privacy*, 2017, pp. 39–57, doi: 10.1109/SP.2017.49.

[8] N. Papernot, P. Mcdaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *IEEE European Symposium on Security and Privacy*, 2016, pp. 372–387, doi: 10.1109/EuroSP.2016.36.

[9] S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: a simple and accurate method to fool deep neural networks," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2574–2582, doi: 10.1109/CVPR.2016.282.

[10] W. Hu and Y. Tan, "Generating adversarial malware examples for black-box attacks based on GAN," *arXiv*, 2017, [Online]. Available: https://arxiv.org/pdf/1702.05983.pdf.

[11] C. Xiao, B. Li, J. Y. Zhu, W. He, M. Liu, and D. Song, "Generating adversarial examples with adversarial networks," in *International Joint Conference on Artificial Intelligence*, 2018, pp. 3905--3911, [Online]. Available: https://doi.org/10.24963/ijcai.2018/543.

[12] S. M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1765–1773, doi: 10.1109/CVPR.2017.17.

[13] N. Carlini and D. Wagner, "Audio adversarial examples: targeted attacks on speech-to-text," in *IEEE Symposium on Security and Privacy Workshops*, 2018, pp. 1–7, doi: 10.1109/SPW.2018.00009.

[14] M. Alzantot, B. Balaji, and M. Srivastava, "Did you hear that? adversarial examples against automatic speech recognition," *arXiv*, 2018, [Online]. Available: https://arxiv.org/pdf/1801.00554v1.

[15] T. Du, S. Ji, J. Li, Q. Gu, T. Wang, and R. Beyah, "SirenAttack: generating adversarial audio for end-to-end acoustic systems," in *ACM Asia Conference on Computer and Communications Security*, 2020, pp. 357–369.

[16] S. Abdoli, L. G. Hafemann, J. Rony, I. Ben Ayed, P. Cardinal, and A. L. Koerich, "Universal adversarial audio perturbations," *arXiv*, 2019, [Online]. Available: https://arxiv.org/pdf/1908.03173v5.

[17] D. Wang, L. Dong, R. Wang, D. Yan, and J. Wang, "Targeted speech adversarial example generation with generative adversarial network," *IEEE Access*, vol. 8, pp. 124503–124513, 2020, doi: 10.1109/ACCESS.2020.3006130.

[18] I. Goodfellow *et al.*, "Generative adversarial networks," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[19] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *ACM Conference on Multimedia*, 2014, pp. 1041–1044, doi: 10.1145/2647868.2655045.

[20] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*, 2016, pp. 694–711, doi: 10.1007/978-3-319-46475-6_43.

[21] T. H. Yuji Tokozume, "Learning environmental sounds with end-to-end convolutional neural network," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 2721–2725.

[22] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *ACM Multimedia Conference*, 2015, pp. 1015–1018, doi: 10.1145/2733373.2806390.