Audio Adversarial Examples Generation with Recurrent Neural Networks*

Kuei-Huan Chang†, Po-Hao Huang†, Honggang Yu‡, Yier Jin‡, and Ting-Chi Wang†
†Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
‡Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA
ax2082197@gmail.com, kevin841006kg@gmail.com, honggang.yu@ufl.edu, yier.jin@ece.ufl.edu, tcwang@cs.nthu.edu.tw

***Abstract*—Previous methods of performing adversarial attacks against speech recognition systems often treat this problem as a solely optimization problem and require iterative updates to generate optimal solutions. Although they can achieve high success rate, the process is too computational heavy even with the help of GPU. In this paper, we introduce a new type of real-time adversarial attack methodology, which applies Recurrent Neural Networks (RNN) with a two-step training process to generate adversarial examples targeting a Keyword Spotting (KWS) system. We extend our attack to physical world by adding extra constraints in order to eliminate the distortions in real world. In the experiment, we launch a real-time adversarial attack on the KWS system both in digital and physical world. The experimental results of digital world show that the execution time of our attack is more than 400 times faster than the state-of-the-art attack (i.e., C&W attack) with the comparable attack success rate. In physical world, after adding extra constraints, the perturbation becomes more robust such that the average attack success rate increases from 40.3% to 84.3%.**

I. Introduction

Deep Neural Networks (DNNs) have achieved great advances in some safety-critical scenarios, such as image classification, audio recognition and natural language processing [1]–[7]. However, studies have shown that these existing DNNs are vulnerable to adversarial attacks. Specifically, an adversary can easily fool a neural network to output incorrect results by adding the particular perturbation to the original inputs. Most of existing works on this topic mainly focus on the area of image recognition [8]–[17]. For example, Carlini *et al.* [11] introduced a new type of adversarial attack, so-called C&W attacks, against the deep neural neural networks with high success rate. Their goals are to craft the malicious images that can lead the target DNNs to misclassify their inputs but keep imperceptible to human eyes as the perturbations are tiny.

In the area of audio recognition, the goal of an adversary is to add imperceptible noises to the audio so that these audio can misclassify the automatic speech recognition (ASR) systems. Recent research has demonstrated that the malicious audio samples generated by the adversaries can also manipulate the transcription results of the ASR systems like keyword spotting (KWS) systems. Alzantot *et al.* [18] performed a black-box adversarial attack which crafts the adversarial audio by adding small background noises without having to know the detailed knowledge of the underlying model. Taori *et al.* [19] proposed a new black-box attack method to fool ASR systems with high success rate by combining the approaches of both genetic algorithm and gradient estimation. Zhang *et al.* [20] utilized the property of audio circuits to launch a completely inaudible attack on several popular speech recognition systems, like Siri, Google Now and Alexa. Since then, several follow-up works have been proposed to improve the previous adversarial attack methodologies [21]–[23]. Although the recent adversarial attacks have made significant progresses, it is difficult for them to be practically launched in real-world systems due to the following limitations: (1) The effectiveness of previous adversarial attacks on the ASR systems falls dramatically when facing complex physical world with environment-induced variables, such as background noises and reverberation. (2) Current recording equipments like microphone are often utilized to receive audio signal in real world and can automatically remove noise from the received audio signal by cutting out all but the audible frequency of sound. In this case, the audio perturbation generated by previous attack methods is often viewed as random noise and is easily diminished by such a recording equipment while its frequency goes beyond the range of audio frequency. (3) Current adversarial attacks on the ASR systems often take a large amount of time to generate a satisfactory adversarial example, which makes it difficult to be used in a real-time attack scenario. Many of the previous methods required many iterations of calculation and iterative update to transform an input into single adversarial example, showing that there is no simple mapping between them. Although they can achieve high success rate, the process is too computation heavy even with the help of GPU.

In this work, we propose a novel adversarial attack method against the popular ASR system (i.e., KWS) in both the digital and physical world. We assume an adversary who targets the KWS system has the full knowledge about the KWS system, such as exact training data, architectures, parameters, etc. The main contributions of this paper are summarized as follows:

- We introduce a new adversarial attack method that adopts the Recurrent Neural Networks (RNNs) for generating adversarial samples in real time. These generated samples can easily mislead the KWS system to output the target labels even under the over-the-air condition.
- We combine the RNNs pre-training and the RNNs fine-tuning in a way that is efficient and effective for crafting more robust audio adversarial examples. The proposed two-step training process helps us to speed up the parameter optimization for crafting the imperceptible perturbations that would be added to an original audio.
- The experimental results demonstrate that our attack method can achieve higher success rate and faster generation speed of malicious audio in both digital world and physical world when compared to previous schemes.

II. Related Works

A. Recurrent Neural Networks

RNNs are powerful models that have shown promising results in many sequential data prediction tasks, like video key-frame tagging and machine translation [24], [25]. Given a sequence $X = \{x^{(t)}, y^{(t)}\}_t \in R^{(D,K)\times T}$, where $x \in R^D$, $y \in R^K$ and T is the horizon of sequence X. RNNs can make the accurate predictions at the current time point t by effectively combining the previous

*This work was partially supported by the Ministry of Science and Technology of Taiwan under Grant No. MOST 108-2218-E-007-031

978-1-7281-4123-7/20/$31.00 © 2020 IEEE

input data with the current data. Currently, there are many well-known variations of RNNs, like vanilla RNNs, bidirectional RNNs, recursive RNNs and long short-term memory (LSTM) [26], which allow us to solve the problems of time sequence that traditional deep neural networks have difficulties to deal with.

In this paper, we use LSTM as our RNN cell to generate the adversarial audio examples for the target KWS systems. The LSTM, which basically consists of input gate, output gate and forget gate, is well-suited for the handling of data that involves with time or order (such as audio or video).

B. Keyword Spotting System

In this paper, we choose the KWS system introduced in [27] as our target model. The KWS system is often used to enable speech-based user interactions on intelligent appliances. The predefined keywords can be easily retrieved by this system from an audio dataset with high accuracy. The advantage of the KWS system is that a user can apply voice to operate the target device with the KWS system. Specifically, If a user speaks particular voice demands, the KWS system would receive these commands and then switch the device from one mode to another mode. Besides, unlike other speech recognition systems, the size of the KWS system is generally small enough to be widely applied to various embedded devices, such as mobile phones and vehicle-mounted electronics. The KWS system recognizes users' voice commands by running different types of deep neural networks, for example, the depthwise separable convolutional neural network (DS-CNN). In this paper, our goal is to apply the RNN model to generate the targeted audio adversarial examples which are usually imperceptible to human listeners.

C. Audio Adversarial Attacks

Alzantot *et al.* [18] use the genetic algorithm to generate adversarial examples for a KWS system. In order to launch the adversarial attacks against a speech to text system in black-box setting, Taori *et al.* [19] improve the attack algorithm in [18] by adding momentum mutation and using gradient estimation to speed up the process of convergence to final results. However, they require extremely long run time and large perturbations to craft adversarial examples, which makes it difficult to be used in a real-time attack scenario. Yuan *et al.* [22] find the mapping between hidden Markov model (HMM) state transition and the probability density function of the acoustic model, and then use a gradient-based approach to search for the minimum amount of perturbations to the probability density function identifier sequence of the original input such that it can be misclassified by the speech to text system. Different from the aforementioned methods, Zhang *et al.* [20] use a modulation technique to integrate the commands into the original audio so that the crafted audio examples can be effectively recognized by the ASR system. However, the perturbation modulated on a high carrier frequency can be easily filtered out by a low pass filter. Abdullah *et al.* [28] utilize some psychoacoustics techniques like time domain inversion and time scaling to make the audio dramatically change in time domain but still remain the same frequency domain feature, so only the model can correctly interpret messages, while the adversarial example will sound totally different to the original one, and hence will be easily noticed by human.

Besides, there are works that extend the digital world attack to physical world. To make the perturbation more robust against ASR systems in physical world, Zhang *et al.* [20] add random noise to enhance the perturbation against background noise. Yuan *et al.* [22] claim that the electronic noise from both the speaker and receiver is a variable for physical world attack, and it is quite different from case to case. Hence, they use random noise to simulate the electronic noise, making the adversarial example robust enough for different speakers and receivers. Yakura *et al.* [23] extend [21] by introducing impulse responses to the generation process of adversarial examples. They collect various datasets of impulse responses, which can make the adversarial example more robust to handle reverberations in complex physical environments. However, all the methods mentioned above take a long time to generate the adversarial examples by iteratively optimizing the objective function, which means that the proposed attack methods cannot be exploited in real-time applications. Since the datasets used for the adversarial examples generation cannot contain impulse responses of the physical environment, the attack is often ineffective under specific physical environments. In order to solve this problem, we propose an attack method based on the combination of some physical constraints in the objective function, such that the generated adversarial examples can be played over-the-air.

III. Methodology

A. Problem Formulation

Given an audio x, a target label t and the model of KWS system with h classification results $f: R^n \rightarrow T$, where n is the dimension of x and $T = \{i | 1 \leq i \leq h, i \in N\}$, our goal is to find a minimal perturbation δ for x with $f(x) \neq t$ and make the RNN model misclassify $x+\delta$ as label t:

$$\begin{aligned} minimize \|\delta\| \\ s.t. \quad f(x) = l, \\ f(x+\delta) = t, \\ l \neq t \end{aligned} \tag{1}$$

B. Digital World Adversarial Attack

1) Feature extraction: We build our RNN model as the structure in Fig. 1. The short-time Fourier transform (STFT) [29] is a popular feature extraction technique for audio signal. It applies Fourier transform to a sliding window as it moves over time and shows the frequency and phase features inside the sliding window. In this way, it not only performs feature extraction to the complicate data, but also significantly reduces the input size of the RNN model.

2) RNN structure: After the STFT layer, we split the data into real and imaginary part before feeding them into two separate stacked RNNs.

3) RNN pre-training: To speed up the training process, we let our RNN model mimic the perturbations generated by iterative fast gradient sign method (iFGSM) [30]. We define our RNN model as g_θ, where θ denotes the parameters for the RNN model. Given the training dataset X, for each input $x \in X$, with label l, we generate an adversarial example x^t_{adv} with label t by iFGSM, and then we subtract the original input x from the adversarial example x^t_{adv} to get the perturbation generated by iFGSM, $\delta^{x,t}_{adv}$, which will be the pre-training label for our RNN model. Finally we let the RNN model learn about how to generate the adversarial perturbations by minimizing the following loss function:

$$\underset{\theta}{minimize} \sum_{x \in X} \left\| g_\theta(x) - \delta^{x,t}_{adv} \right\| \tag{2}$$

We try to find the best θ which can minimize the total difference between the perturbations generated by our RNN model and iFGSM for all training data.

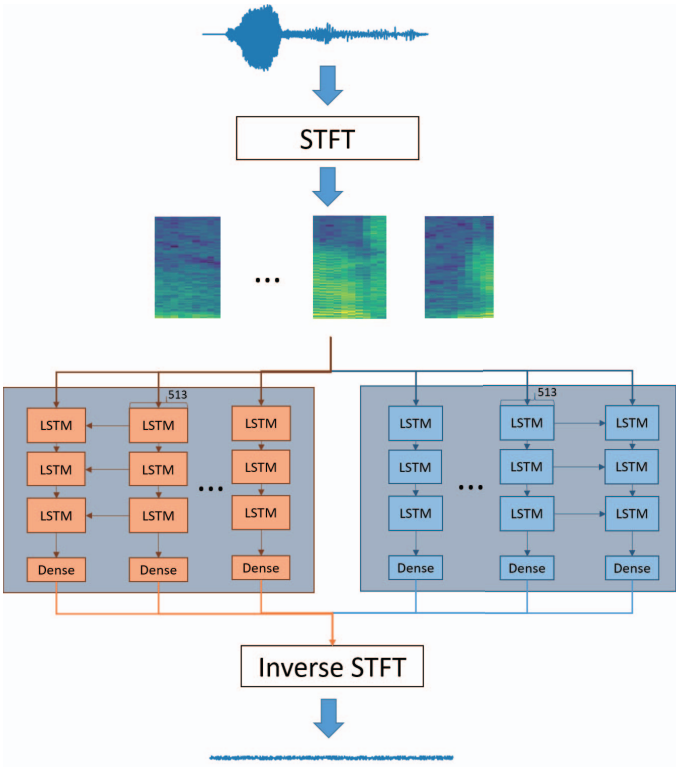


Figure 1: Procedure of perturbation generation with recurrent neural networks.

4) RNN fine-tuning: We further improve our RNN model by directly optimizing against the KWS network. In this step, we relax the constraint in Eq. (1) and set our objective function as follows:

$$\begin{aligned} &\underset{\theta}{minimize} \sum_{x \in X} (\|g_\theta(x)\| + c \cdot loss(x')) \\ &where\ x' = x + g_\theta(x) \end{aligned} \tag{3}$$

Here x denotes a training audio, $\|g_\theta(x)\|$ denotes the L_2 distance between the clean audio x and the perturbed audio x', $loss(x')$ reflects how good the adversarial example x' is, and $c > 0$ is a regularization parameter which controls how important $loss(x')$ is. The $loss(x')$ is defined below,

$$loss(x') = (\underset{i \neq t}{max}(L(x')_i) - L(x')_t + R)^+ \tag{4}$$

where $L(x')_i / L(x')_t$ denotes the logit score of label i/t when inferring x' on the KWS system. R is a non-negative constant which controls the confidence score of the target label t. If we increase R, our RNN model will tend to create the result with higher confidence score.

C. Robust Physical World Perturbation

In this subsection, we describe how to add some constraints on our objective function to make our adversarial example created by our RNN model propagate through physical world conditions such as background noise, reverberation, and recording equipment, which are also mentioned in [23]. The pipeline of the proposed audio adversarial attack method in physical world is shown in Fig. 2.

In recording equipment, the frequency response of a microphone falls within the range between lowest and highest frequencies. Generally, the frequency response of the microphone is ranging from around 80 Hz to 15 kHz, which would be a good choice for human voice since 20 Hz to 20 kHz is an audible range for humans. In other words, if the frequency of perturbation is outside the audible range, the perturbation will be filtered out. In order to get rid of this scenario, we use a band-pass filter to limit the frequency range of the perturbation.

In the physical environment which contains background noise, we use Gaussian white noise [31] to simulate the background noise. The advantage of Gaussian white noise is that it can mimic the circumstances of some random processes that appear in nature. As a result, Gaussian white noise becomes widely used in signal processing or audio engineering. The simulated audio after undergoing band-pass filter and adding Gaussian white noise is denoted below:

$$x' = x + Band(g_\theta(x)) + noise \tag{5}$$

where x denotes a training audio, $g_\theta(x)$ denotes the perturbation generated by the RNN model, $Band(g_\theta(x))$ denotes the band-pass filter which cuts off the frequency of $g_\theta(x)$ beyond the audible range, and $noise$ denotes the Gaussian white noise generated by Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$. The amplitude of the noise depends on the applied environment, and x' denotes an adversarial example.

When a sound is reflected by objects, the reverberation will be created. It is also one of the factors that influence the result of speech recognition. In order to reproduce the reverberation of the sound, we add impulse responses to the training audio to enhance the robustness. By convolving the audio with the impulse response of the physical environment, we can easily simulate the reverberation of the audio in the given physical environment. However, it is hard to get a real impulse response from the experimental room directly. Hence, we use a room impulse response simulator to simulate the impulse response. We first generate a 3D simulated room by configuring our real world experimental room dimension, the sound absorption coefficient of walls, and the maximum number of reflections. Secondly, we decide the positions of the source speaker and target recorder and then generate the room impulse response accordingly. At last, we make convolution of these impulse responses and the training audio.

Moreover, we want the adversarial example to be able to attack through the entire room no matter the locations of a recorder and speaker are. Thus, we divide the simulated room into various grids, and place the speaker and recorder in different grids and generate the impulse response independently. We reformulate the physical world objective function based on Eq (3) as follows:

$$\begin{aligned} &\underset{\theta}{minimize} \sum_{x \in X} \sum_{i \in I} (\|g_\theta(x)\| + loss(x')) \\ &where\ x' = T(x + Band(g_\theta(x)), i) + noise \\ &T(x, i) = x * i \end{aligned} \tag{6}$$

Here i denotes the simulated room impulse response, I denotes the impulse response dataset which is generated by the room impulse response simulator based on our testing environment, and function T denotes the convolution operation. Since the computation time by calculating the summation of all the convolutions with impulse responses for single input x is too high, we only convolve the input with one impulse response which is chosen randomly from I for each training iteration.

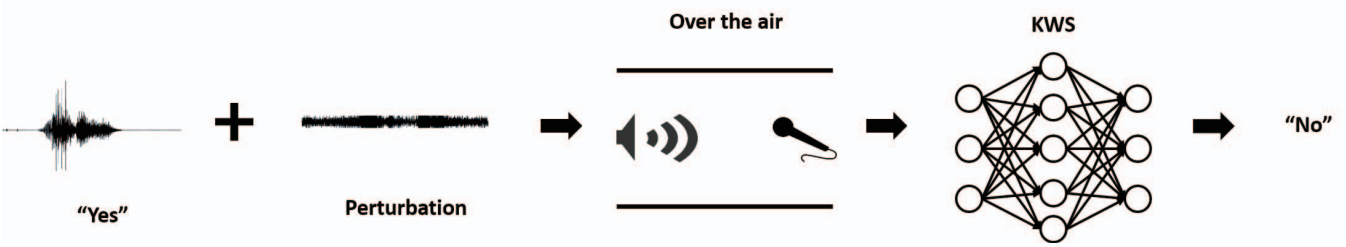


Figure 2: Overview of the proposed audio adversarial attack method in physical world. The generated adversarial examples need to be propagated through the air before the inference stage of the KWS system.

IV. Experimental Results

A. Experiment Setup

In the experiment, we train a depthwise separable convolutional neural network (DS-CNN) in the KWS system with 94.6% recognition rate as the target model [27]. We evaluate the experimental results on Google common voice data set [32], containing ten different basic commands. For each target t, we gather all other nine classes of audio in [32] and use 85% of the audio for training and 15% for testing. We train an RNN model targeting one of the ten classes in [32] as the procedure described in Section III-B. Similar to the previous attack method in [23], we also utilize the signal-to-noise ratio (SNR) as one of the evaluation matrics. All experiments are carried out on a server with an Intel i7-8086K 4GHz CPU with 16GB RAM and two NVIDIA GeForce RTX 2080Ti GPUs.

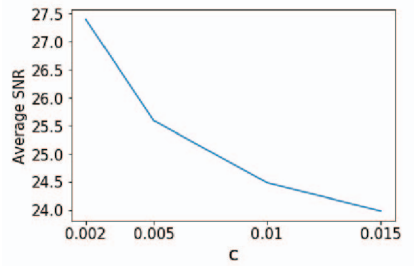
B. Digital World

We set the hyper-parameters as follows: The frame length, frame step and fast Fourier Transform size of STFT are 800, 200, 513, respectively. We implement our work with Tensorflow, and train our model using the Adam optimizer as the learning rate is set to 0.001. The Adam optimizer realizes the benefits of both Momentum and AdaGrad, achieving smoother update and faster convergence compared to other stochastic optimization methods.

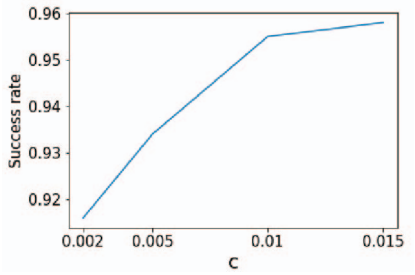
To get the most robustness out of our method, we need to find the value of the parameter c in Eq (3) for the best trade-off between attack success rate and average SNR in the loss function. We test the models with c equal to 0.015, 0.01, 0.005 and 0.002. For training our RNN models, we train them until the loss converges in the pre-training step and then train up to 30 epochs in the fine-tuning step. The empirical results in Fig. 3 show that the model with c equal to 0.01 makes the best trade-off between SNR and success rate. So we let the value of c equal to 0.01 for all the experiments in digital world.

Table I: Results generated by two different models

label	LSTM			Bidirectional		
	SR	SNR	time	SR	SNR	time
down	0.96	25.06	0.097	0.96	20.908	0.124
go	0.95	24.74	0.096	0.93	23.266	0.123
left	0.96	23.385	0.096	0.93	23.69	0.124
no	0.95	21.021	0.096	0.92	21.97	0.124
off	0.95	24.521	0.096	0.95	20.842	0.123
on	0.95	23.669	0.096	0.92	21.97	0.124
right	0.96	28.003	0.096	0.95	22.898	0.124
stop	0.96	25.7	0.096	0.95	22.99	0.124
up	0.96	24.231	0.096	0.95	22.767	0.124
yes	0.95	24.548	0.095	0.96	22.17	0.124
average	0.955	24.488	0.096	0.94	22.347	0.124



(a) average SNR with different c



(b) success rate with different c

Figure 3: Illustration of the attack results with different values of c.

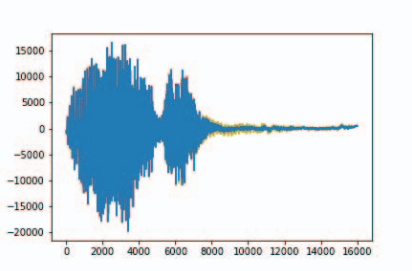
In Table I, we compare the performance of two different RNN structures. The LSTM model contains 3 layers of LSTM cells followed by a dense layer and the bidirectional model contains 2 layers of forward-pass LSTM cells, 2 layers of backward-pass LSTM cells with a dense layer at the end. SR denotes the attack success rate among the testing set, and time denotes the execution time for the corresponding method to generate a single adversarial example. The LSTM model outperforms the bidirectional model in SNR, success rate and execution time. In view of this, we choose the LSTM model to be our RNN structure for the rest of the experiments.

We compare our attack method with FGSM [9] and C&W [11] attacks on the same test dataset, and the results are shown in Table II. For the FGSM attack, we generate the adversarial example by calculating the partial derivative of the loss function with respect to the input data. The amplitude of perturbations for all data points is a hyper-parameter in FGSM attack, and hence to make a fair comparison, we set it to 0.0001 such that the resulting average SNR of FGSM attack is similar to ours. The FGSM attack only achieves 11.3% success rate, showing that transforming an audio to adversarial one is too complex to generate by calculating partial derivative only once.

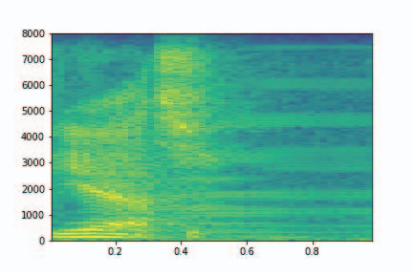
Next we compare our method with C&W attack. For C&W attack, we perform 9 steps of binary search and run 800 learning epochs

Table II: Comparison among three different attacks in digital world

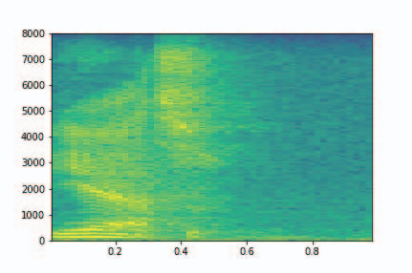
label	Our method			FGSM			C&W		
	SR	SNR	time	SR	SNR	time	SR	SNR	time
down	0.960	25.06	0.097	0.090	25.750	0.015	0.979	23.712	44.971
go	0.950	24.74	0.096	0.125	26.743	0.011	0.963	27.508	43.664
left	0.960	23.385	0.096	0.105	25.444	0.011	0.968	24.942	43.706
no	0.950	21.021	0.096	0.084	26.673	0.015	0.973	25.093	44.659
off	0.950	24.521	0.096	0.069	27.357	0.015	0.979	26.065	43.634
on	0.950	23.669	0.096	0.129	26.637	0.015	0.962	26.935	43.647
right	0.960	28.003	0.096	0.185	27.393	0.015	0.976	29.249	43.727
stop	0.960	25.7	0.096	0.095	26.402	0.015	0.988	25.180	43.742
up	0.960	24.231	0.096	0.116	26.233	0.015	0.971	26.816	43.645
yes	0.950	24.548	0.095	0.131	27.079	0.015	0.979	27.703	43.710
average	0.955	24.488	0.096	0.113	26.571	0.014	0.974	26.320	43.910



(a) Waveform comparison for the clean audio (blue), the reduced amplitude (red) and the added amplitude (yellow)



(b) STFT features for clean example



(c) STFT features for adversarial example

Figure 4: Illustration of an attack results with average SNR equal to 24.5

using Adam optimizer with learning rate equal to 0.01. C&W attack conducts many iterations of searching and optimizing to find the optimal solution and optimal hyper-parameter. As a result, the quality of C&W attack is slightly better than ours, but at the cost of long processing time. Our attack can achieve 95.5% success rate and 24.488 SNR on average and generate a single adversarial example within a tenth of a second. Also we visualize the waveform and the STFT features of our attack on an input in Fig. 4.

Table III: Comparison between our attack method with and without physical world constraints

label	with physical constraints		without physical constraints	
	success rate		success rate	
	0.5-meter	4-meter	0.5-meter	4-meter
down	0.752	0.645	0.333	0.134
go	0.753	0.673	0.404	0.116
left	0.880	0.836	0.513	0.322
no	0.866	0.660	0.388	0.333
off	0.883	0.866	0.415	0.106
on	0.840	0.700	0.582	0.178
right	0.892	0.867	0.483	0.387
stop	0.876	0.776	0.197	0.138
up	0.862	0.85	0.360	0.275
yes	0.823	0.786	0.360	0.133
average	0.843	0.766	0.403	0.212

C. Physical World

In the physical world scenario, we play and record the adversarial examples with a speaker (DELL AX210) and a microphone (Sony ECM-PCV80U). The place for the experiment is a meeting room (7m*7m*3m) with concrete walls, and the background noise is 32 dB. We randomly select 100 audio from the testing set for each label. The c and number of epochs are set to 0.015 and 50, respectively. If the frequency range of the band-pass filter is too small, the magnitude of perturbation will become too high. In contrast, a large frequency range will make the perturbation undermined with some microphones. As a result, we tried various ranges of frequencies and found that 1k Hz to 8k Hz is better because of less perturbation and higher robustness.

In order to simulate the impulse responses, we use [33] which provides an intuitive python object-oriented interface to quickly construct different simulation scenarios involving multiple sound sources and microphones in 3D rooms. We divide the simulated room into 9 grids, and place the speaker and recorder in different grids. By fixing the speaker in one grid, the recorder can be placed in one of the other 8 grids to produce 8 kinds of simulated impulse responses. Then we acquire 72 kinds of simulated impulse responses and add them to the training process. We set the distance between the speaker and recorder to 0.5 and 4 meters, respectively to test the robustness against different distances. The experimental results of our attack are shown in Table III.

We can see that the average attack success rate without any physical constraints is 40.3% in 0.5-meter and 21.2% in 4-meter scenario. In contrast, after adding the physical constraints by using the band-pass filter, white Guassian noise and simulated impulse responses, the average attack success rate is increased to 84.3% in 0.5-meter scenario. The success rate of our method is 76.6% in 4-meter scenario and drops only 7.7% when compared to 0.5-meter scenario. These results suggest that the adversarial examples generated by our attack method can successfully propagate through

the environment with background noise and reverberation. Besides, the frequency of the perturbation falls into the audible range which cannot be filtered out by the recording equipment. By using an impulse room simulator, we can make our adversarial example more robust against the physical environment. Furthermore, We can save more cost and time because we do not need to record the impulse response directly in the physical world.

V. Discussion

We introduce a powerful audio adversarial attack that can easily fool the KWS system to give targeted incorrect decisions in both digital world and physical world, demonstrating that our attack can be applied to evaluate the robustness of DNN based audio classifiers. We perform a recurrent neural network as shown in Fig. 1 for crafting malicious perturbations via combining short-time Fourier transform (STFT) and LSTM cells that can produce very similar, high confidence audio for all classes by integrating RNN pre-training algorithm and RNN fine-tuning algorithm into the training process. The results of the experiments demonstrate that our audio adversarial attack on the KWS system requires less computation time and achieves higher success rate compared to previous schemes due to our novel design of network architecture and feature extraction of the RNNs.

Our attack method still has some limitations. One major limitation is that we launch the adversarial audio attack against the target model in white-box setting, which means that an attacker has full access to the target model, such as network structures, parameters, training data, etc. In the future, we will mainly focus on conducting a black-box adversarial audio attack methodology in both digital world and physical world. Specifically, we plan to design a new optimization algorithm for generating more robust malicious audio, which can achieve higher success rate and transferability at lower distortions when compared to the previous works of adversarial attack. In the experiment, we also find that a few crafted audio can be easily noticed by human listeners due to the large perturbation added by an attacker, meaning the designed objective function in Section III can be further improved to generate more inconspicuous audio.

VI. Conclusion

In this paper, we propose using an RNNs model to generate audio adversarial examples against the KWS system. In digital world, the results of experiments show that our method can generate adversarial examples within one second while achieving high success rate and fair amount of perturbations. Besides, we also extend our work to physical world. We successfully mislead the KWS system to wrong keyword decisions in physical world conditions and make the perturbation more robust against different distances of the speaker and recorder in the physical environment. Future research directions include enhancing the transferability of malicious audio and designing effective defense mechanisms against adversarial attacks.

References

[1] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 1695–1699.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, June 2017.

[3] L. Logeswaran and H. Lee, "An efficient framework for learning sentence representations," *CoRR*, vol. abs/1803.02893, 2018.

[4] T. Afouras, J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Deep audio-visual speech recognition," *CoRR*, vol. abs/1809.02108, 2018.

[5] C. Chiu, T. N. Sainath, Y. Wu, and R. Prabhavalkar, "State-of-the-art speech recognition with sequence-to-sequence models," *CoRR*, vol. abs/1712.01769, 2018.

[6] Y. Chung, W. Weng, S. Tong, and J. Glass, "Towards unsupervised speech-to-text translation," *CoRR*, vol. abs/1811.01307, 2018.

[7] Z. Zhao, P. Zheng, S. Xu, and X. Wu, "Object detection with deep learning: A review," *CoRR*, vol. abs/1807.05511, 2019.

[8] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proc. International Conference on Learning Representations (ICLR)*, 2014.

[9] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *CoRR*, vol. abs/1412.6572, 2015.

[10] Rey, A. Wiyatno, and Xu, "Maximal jacobian-based saliency map attack," *CoRR*, vol. abs/1808.0794, 2018.

[11] N. Carlini and D. A. Wagner, "Towards evaluating the robustness of neural networks," *CoRR*, vol. abs/1608.04644, 2016.

[12] P. Zhao, K. Xu, S. Liu, Y. Wang, and X. Lin, "Admm attack: an enhanced adversarial attack for deep neural networks with undetectable distortions," in *Proc. Asia and South Pacific Design Automation Conference (ASP-DAC)*, january 2019, pp. 538–543.

[13] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," *CoRR*, vol. abs/1511.04599, 2016.

[14] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *CoRR*, vol. abs/1710.08864, 2019.

[15] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," *CoRR*, vol. abs/1610.08401, 2017.

[16] C. Moustapha, A. Yossi, N. Natalia, and K. Joseph, "Houdini: fooling deep structured prediction models," *CoRR*, vol. abs/1707.05373, 2017.

[17] Yanpei, X. Liu, C. Chen, D. Liu, and Song, "Delving into transferable adversarial examples and black-box attacks," *CoRR*, vol. abs/1611.02770, 2017.

[18] M. Alzantot, B. Balaji, and M. B. Srivastava, "Did you hear that? adversarial examples against automatic speech recognition," *CoRR*, vol. abs/1801.00554, 2018.

[19] R. Taori, A. Kamsetty, B. Chu, and N. Vemuri, "Targeted adversarial examples for black box audio systems," *CoRR*, vol. abs/1805.07820, 2018.

[20] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "Dolphin atack: inaudible voice commands," *CoRR*, vol. abs/1708.09537, 2017.

[21] N. Carlini and D. A. Wagner, "Audio adversarial examples: targeted attacks on speech-to-text," *CoRR*, vol. abs/1801.01944, 2018.

[22] X. Yuan, Y. Chen, Y. Zhao, Y. Long, X. Liu, K. Chen, S. Zhang, H. Huang, X. Wang, and C. A. Gunter, "Commandersong: a systematic approach for practical adversarial voice recognition," *CoRR*, vol. abs/1801.08535, 2018.

[23] H. Yakura and J. Sakuma, "Robust audio adversarial example for a physical attack," *CoRR*, vol. abs/1810.11793, 2018.

[24] G. Sreenu and M. A. Saleem Durai, "Intelligent video surveillance: a review through deep learning techniques for crowd analysis," *Journal of Big Data*, June 2019.

[25] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *CoRR*, vol. abs/1406.1078, 2014.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, November 1997.

[27] Y. Zhang, N. Suda, L. Lai, and V. Chandra, "Hello edge: keyword spotting on microcontrollers," *CoRR*, vol. abs/1711.07128, 2017.

[28] H. Abdullah, W. Garcia, C. Peeters, P. Traynor, K. R. B. Butler, and J. Wilson, "Practical hidden voice attacks against speech and speaker recognition systems," *CoRR*, vol. abs/1904.05734, 2019.

[29] "Short-time fourier transform," https://en.wikipedia.org/wiki/Short-time_Fourier_transform.

[30] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *CoRR*, vol. abs/1607.02533, 2016.

[31] "White gaussian noise," https://en.wikipedia.org/wiki/Additive_white_Gaussian_noise.

[32] "Speech commands dataset," https://research.googleblog.com/2017/08/launching-speech-commands-dataset.html.

[33] R. Scheibler, E. Bezzam, and I. Dokmanic, "Pyroomacoustics: a python package for audio room simulations and array processing algorithms," *CoRR*, vol. abs/1710.04196, 2017.