NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

ADVERSARIAL ATTACKS ON UNDERWATER SOUNDSCAPE CLASSIFICATION SYSTEMS

by

Jason A. Henry

June 2021

Thesis Advisor:	Marko Orescanin
Second Reader:	Joshua A. Kroll

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2021	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE
ADVERSARIAL ATTACKS ON UNDERWATER SOUNDSCAPE CLASSIFICATION SYSTEMS

5. FUNDING NUMBERS

6. AUTHOR(S) Jason A. Henry

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

Deep convolutional neural networks (CNN) are shown to be effective in underwater soundscape classification, providing the potential for increased automation and performance of contact detection systems on board ships and autonomous unmanned underwater vehicles (UUV). CNNs are known to be vulnerable to adversarial attacks that add a small perturbation to the input, causing a classifier to incorrectly classify the input example. A common method in audio classification is to transform source audio into spectrogram images to use as features for classification. We test several established image-based adversarial attack methods against an underwater soundscape classifier to demonstrate the vulnerability of a system reliant on spectrograms. Five methods successfully fooled the target classifier over 80% of the time with small ϵ. Additionally, this thesis introduces a novel, perceptually motivated, audio-based adversarial attack on audio classification systems. The attack modifies an existing attack generation scheme to include perceptually motivated penalty functions with the goal of reducing loudness of the adversarial noise, which reduces the perceptibility of the attack. Inclusion of perceptual metrics in the attack training reduces the relative loudness of generated perturbations by 4.5 dB for attacks against the underwater soundscape classifier and 8.7 dB for speech command classifier on average without impacting the success of the attack.

14. SUBJECT TERMS convolutional neural networks, CNN, adversarial, attacks, sound, audio, underwater, classification, generative, unmanned underwater vehicle, UUV	**15. NUMBER OF PAGES** 61
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

ADVERSARIAL ATTACKS ON UNDERWATER SOUNDSCAPE CLASSIFICATION SYSTEMS

Jason A. Henry
Ensign, United States Navy
BS, United States Naval Academy, 2020

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 2021

Approved by: Marko Orescanin
Advisor

Joshua A. Kroll
Second Reader

Gurminder Singh
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Deep convolutional neural networks (CNN) are shown to be effective in underwater soundscape classification, providing the potential for increased automation and performance of contact detection systems on board ships and autonomous unmanned underwater vehicles (UUV). CNNs are known to be vulnerable to adversarial attacks that add a small perturbation to the input, causing a classifier to incorrectly classify the input example. A common method in audio classification is to transform source audio into spectrogram images to use as features for classification. We test several established image-based adversarial attack methods against an underwater soundscape classifier to demonstrate the vulnerability of a system reliant on spectrograms. Five methods successfully fooled the target classifier over 80% of the time with small ϵ. Additionally, this thesis introduces a novel, perceptually motivated, audio-based adversarial attack on audio classification systems. The attack modifies an existing attack generation scheme to include perceptually motivated penalty functions with the goal of reducing loudness of the adversarial noise, which reduces the perceptibility of the attack. Inclusion of perceptual metrics in the attack training reduces the relative loudness of generated perturbations by 4.5 dB for attacks against the underwater soundscape classifier and 8.7 dB for speech command classifier on average without impacting the success of the attack.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

DoN	Department of the Navy
CNN	convolutional neural network
RNN	recurrent neural network
FAPG	fast audio adversarial perturbation generator
PESQ	perceptual evaluation of speech quality
STOI	short-time objective intelligibility
MSE	mean squared error
PMP	perceptually motivated penalty functions
ViSQOL	virtual speech quality objective listener

THIS PAGE INTENTIONALLY LEFT BLANK

Disclaimer

This material is based upon activities supported by the National Science Foundation under Agreement No 1565443. Any opinions, findings, and conclusions or recommendations expressed are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

Deep learning techniques are proving to be useful across a variety of domains. Deep convolutional neural network (CNN) architectures have improved our ability to perform tasks like image recognition and speech translation [1], [2]. The Department of the Navy (DoN) has shown in interest in exploring the benefits of machine learning and artificial intelligence in the modern combat environment [3], [4]. The deployment of CNN technologies to applications in the underwater acoustic environment is of significant interest to the DoN. Specifically, there are numerous activities in the DoN using CNN architectures for classification in the underwater acoustic environment. Other work focuses on classiying marine life and manmade vessels using underwater acoustic information [5], [6]. The ever-decreasing cost of computation and sensors creates an opportunity for the DoN to deploy arrays of unmanned sound monitors that leverage CNN to observe ocean traffic in areas of interest.

CNN architectures are quite powerful; however, it has been shown that many CNN architectures can be forced to incorrectly classify an input when a small, specifically crafted perturbation is added to that input. Attacks of this type were first shown in the image domain by Goodfellow et al. in 2015 [7]. Similar attacks were shown to exist on sound classification systems by creating perturbations in the audio domain by Carlini et al. in 2018 [8]. The existence of these adversarial examples have important implications for the creation and deployment of systems reliant on CNNs.

1.1 Research Objectives

CNNs have been mostly applied to solving problems in the computer vision applications specifically in image recognition tasks. To a much lesser extent, certain audio recognition tasks have also been able to successfully leverage the benefits of AI/ML technology. Adversarial examples can target networks in both of these domains. However, they are not well understood in the audio domain, mainly because of the lack of applications and the existence of the publicly available data sets. The applications of CNN to tasks in underwater

soundscape classification are largely unexplored. As a result, the existence of adversarial examples on classifiers in underwater soundscape classification has yet to be shown in a wide set of applications.

In underwater soundscape classification, classifiers often rely on generating spectrogram images for classification [9]. This means that there are two possible avenues for adversarial attacks on such classifiers: adversarial spectrogram images and adversarial audio waveforms. The goal of this research is to demonstrate the vulnerability of a CNN based underwater soundscape classifier to adversarial examples of both kinds. Other work conducting adversarial attacks in the audio domain often omit an in-depth discussion of the perceptual qualities of the generated audio attacks. We show, through a survey of existing image-based adversarial attack methods, that the differences between generated adversarial images and the original are imperceptibly different to a observer. Early tests from the audio-based attacks in this work suggest that audio generated by similar methods may be detectable by a human observer. We wish to generate adversarial audio examples in a manner that reduces perceptual differences between the adversarial audio and the original audio signal by incorporating perceptual metrics into the objective function. Finally, we aim to develop a modular framework that allows for the development and extension of this type of audio attack to deployment in a physical space.

1.2 Significant Contributions

The contribution of this work is three-fold. First, we evaluate the susceptibility of a sound classification system reliant on spectrogram images to known image-based attacks. Evaluation shows that several attacks, available in open source libraries, are capable of successfully attacking such a system with success rates nearing 100 percent for relatively small perturbations. Additionally, these attacks can produce adversarial spectrogram that are imperceptibly different from the original.

Next, we propose the first ever perceptually motivated, audio-based adversarial attack on audio classification systems. We do this by altering an existing attack scheme to include perceptually motivated penalty functions in the generation of waveform perturbations. Using a penalty functions that measures the loudness of the adversarial signal to the source signal, we show that we can achieve an average improvement of the generated adversarial signals

relative loudness by 4.5 dB and 8.7 dB for underwater sound data and speech command data, respectively. This improvement come with a negligible decease in the success rate of the generated attacks.

Finally, we provide a mathematical approach for creating additional perceptually motivated penalty functions. This framework allows for the simple implementation of additional penalty functions based on popular perceptual audio standards such as perceptual evaluation of speech quality (PESQ) and short-time objective intelligibility (STOI). Our framework enables further improvements in audio-based attacks, especially for problems related to speech classification.

1.3 Organization

We first provide an overview of adversarial attacks targeting CNN-based classification methods, with a focus on methods designed to attack audio classification systems in Chapter 2. Chapter 3 discusses our study evaluating the application of some commonly available image-based attacks on spectrogram-based audio classifiers. Specifically, we examine a set of methods available in an open source toolbox, Foolbox [10], targeting an underwater soundscape classification network [11]. In Chapter 4 we present our proposed audio attack generation method, adapted from the method developed by Xie et al. [12], including the development of perceptually motivated penalty functions. Chapter 5 presents an analysis of the performance of the proposed method's performance. Finally, Chapter 6 provides insight into potential future work and overall conclusions.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Previous Work

Over the last decade improvements of deep convolutional neural network (CNN) architectures by have led to improvements in tasks like image recognition and speech translation [1], [2]. CNNs work by leveraging convolutional filters inside of the network to detect patterns in the input signal while reducing the number of parameters required when compared to dense neural networks. Rather than focusing on the fundamentals of CNNs that an interested reader can find in many textbooks [13], we provide an overview of adversarial attacks on neural networks.

2.1 Defining an Adversarial Attack

An adversarial attack is the deployment of any method of generating an adversarial perturbation with the intent of compromising the target network.

Adversarial examples for a specific target classifier are benign input values that have been altered to raise a false response from a target classifier, while appearing unchanged to a human observer. In its simplest form an adversarial example $\bar{x}$ is the sum of the original signal x and some small perturbation δ as in equation 2.1.

$$\bar{x} = x + \delta \tag{2.1}$$

Several methods to arrive at a perturbation δ that satisfies the adversarial condition have been shown. Goodfellow et al. provides one such method known as the fast gradient sign method (FGSM) [7]. This method calculates a value of δ by finding the sign of the gradient of the cost function, C, used in training the target network, fixed at the current network parameters, θ. The method is captured in Equation 2.2, where ϵ is a constant chosen to constrain the size of the perturbation.

$$\delta = \epsilon \cdot sign\{\nabla C(\theta, x, y)\} \tag{2.2}$$

Another method to arrive at an adversarial perturbation for an image is known as the basic iterative method, as described by Kurakin et al. [14]. This method extends the FGSM by framing it as an iterative process. A clipping function is introduced in order to constrain the magnitude of each pixel value in the adversarial image. The basic iterative method (BIM) is given by:

$$X_0^{adv} = X,\ X_{N+1}^{adv} = Clip\Big\{X_N^{adv} + \delta\Big\}, \tag{2.3}$$

where δ is the result of Equation 2.2 for X_N^{adv}.

A limitation of the BIM is a lack of stopping criteria. BIM as originally presented, runs for a specified number of iterations. As a result, there is no guarantee as to the success of the method generating an perturbed image that raises a false response from the target classifier. Liiv and Strömberg introduce the stopping criteria $\hat{k}(\bar{x}) \neq \hat{k}(x)$, where $\hat{k}(.)$ gives the classification of an example by the neural network [15]. The authors go on to extend the BIM to four different equations that each implement a different norm as a constraint on the generated example. Additionally, the authors introduce another style of adversarial attack, DeepFool, which is based on the gradient of the target network's output instead of its learned parameters. Despite the different gradient calculation, this method largely follows the same iterative process in Equation 2.3 using the stopping criteria introduced by the authors.

2.2 Adversarial Audio Attacks

Many methods for generating adversarial examples have been focused on image classification networks. One reason for this is the areas of image classification and regression tasks has been the largest breeding ground for work utilizing and expanding CNN technology. However, some of the most interesting new user technologies rely on voice and speech recognition in order to provide services to end users (e.g., Apple's Siri, Amazon's Alexa, and Google Assistant). One example of a speech-to-text transcription neural network is DeepSpeech, introduced by Hannun et al. in 2014 [2]. DeepSpeech is a type of neural network known as a recurrent neural network (RNN). RNN are different from other neural networks by retaining activation values from previous example input in order to calculate the activation for the next example. Operation in this manner makes RNN well suited for op-

erating on streams of input data where maintaining temporal context can provide additional insight into the correct classification for a single input value, like speech-to-text translation. Using this technique the authors were able to achieve error rates lower than 5 comparable speech-to-text systems.

Comparable with how the success of neural network-based image classification systems encouraged the development of attacks against those systems, the success of DeepSpeech as a speech-to-text software encouraged the development of its own adversarial attacks. In 2018, Carlini and Wagner describe a technique to generate adversarial audio examples capable of fooling the DeepSpeech network [8]. The authors employ a method similar to image attacks described above. They frame the perturbation generation as a constrained optimization problem that minimizes the sum of the L-2 norm of the perturbation and a loss function to evaluate the closeness of the classification of the perturbed audio to the correct label. An additional constraint is placed on the generated audio perturbation to keep the intensity of the generated signal below some threshold. Intensity of a given signal, x, is measured in decibels as $dB(x) = \max_i 20 \cdot \log_{10}(x_i)$ [8]. In addition to the success rates of the proposed attacks the authors offer some insight to the intensity of the perturbation, δ, relative to the original signal, x, according to Equation 2.4.

$$dB_x(\delta) = dB(\delta) - dB(x) \tag{2.4}$$

For this relative intensity measurement, a more negative value equates to a lower intensity in the generated perturbation and therefore a greater chance that the addition of the perturbation to the original signal goes unnoticed by an observer. The attacks generated by the author's method were able to achieve a 100 percent success rate with the relative intensity, according to Equation 2.4, of -31 dB on average [8].

An interesting extension of this was conducted by Yakura and Sakuma in 2019 [16]. In their work, the authors adapt the methods introduced by [8] to conduct an attack in the physical space. This is in contrast to [8] where the crafted attacks dealt exclusively with the digital audio signals. Yakura and Sakuma increase the robustness of the generated adversarial examples such that they retain their adversarial properties when played through a speaker and then recorded by a separate microphone. The proposed method augments the perturbation generation process described in [8] by using combination of a band-pass filter

and the addition of white noise to the perturbation. By combining these two augmentation methods, adversarial signals were generated that were successfully played over the air to attack the target classification system. Generated signals from this method have a signal-to-noise ratio of -4.0 dB on average. The signal-to-noise ratio inverts the relationship between the original signal and the perturbation as given in 2.4. By this metric, a larger number suggests that the perturbation carries less energy in comparison to the original signal. The authors include a third augmentation method that accounts set of impulse responses gathered from a variety of potential deployment environments. This augmentation method is ineffective in generating adversarial examples capable of being played over the air unless combined with both of the previously mentioned metrics. In this case, the generated signals have a signal-to-noise ratio of 10.6 dB on average. The perturbations generated by this method carry more energy than the ones developed by Carlini and Wagner [8]; however, the authors of this work report that of 25 human observers polled most were unable to detect the perturbation through observation. Avoiding human detection while being deployable over the air makes this attack method arguably more effective than the methods that it extends.

2.3 Fast Adversarial Attacks

A limitation of the methods discussed in the previous section, is the speed at which adversarial examples can be generated. The iterative optimization process can require a significant amount of computation time in order to arrive at a solution. The authors of [8] stated that their method requires approximately one hour of computer time on a system with a single NVIDIA 1080Ti graphics processor. Xie et al. identified this weakness and propose an new attack generation scheme, fast audio adversarial perturbation generator (FAPG), with the goal of significantly reducing the time required to generate adversarial examples [12]. Using a generative neural network based on the Wave-U-Net architecture [17] the method is able to produce an adversarial example in a single forward pass of the network, greatly improving generation speeds.

A potential drawback of a generative neural network architecture for creating targeted adversarial attacks is that a typical generative architecture would a unique generator for each target class. This problem stems from the way a generative network is trained to target a specific class. A network's target class is controlled through the loss function used to train the network parameters. With that loss function being adjusted to train the model to

output perturbations that look create adversarial examples that look like the target class. To circumvent this limitation and reduce memory required to support targeted attacks for all possible target classes, the authors modify the Wave-U-Net architecture to include some number of swappable parameters. For n target classes, there would be n different sets of parameters that could be swapped into the generative network. The training procedure then includes a periodic swapping of these parameters and a matching adjustment to the loss function so that the model was training to learn the target class associated with that set of parameters. The authors choose the "bottom" layer in the Wave-U-Net architecture, that is the layer where the input has been reduced in size to the maximum extent of the network. In theory, the swappable layers could be at any point in the network; however, the selected layer makes intuitive sense as it is the layer immediately before the network begins upsampling the input back to the original input shape. Said another way, the activations from this layer directly seed the generative process in the network.

The FAPG method was tested across three types of audio recognition tasks: speech command recognition, speaker recognition, and environmental sound classification. The quality of the generated examples were evaluated on three metrics: fooling rate, success rate, and a distortion metric. Fooling rate is the rate at which the target network incorrectly classifies an adversarial example. Success rate differs from fooling rate by evaluating a targeted attack on its ability to force the target network to classify an example as the selected target class. Finally, the authors give a distortion metric to measure the intensity of the generated perturbation relative to the original signal in decibels. While slightly different in presentation, this metric is the same as the one used in [8].

$$D(x_i, \delta_t) = 20log_{10}\frac{max(\delta_t)}{max(x_i)} \tag{2.5}$$

Across all three audio classification tasks the authors report success rates over 90 percent. Additionally, these generated samples have a distortion level between -30 dB and -18 dB. The distortion metrics used by Xie et al. is based on the maximum intensity of a single sample from an audio waveform. This means that this metric is not comprehensive for the entirety of the signal. A property that creates certain edge cases where a signal may appear of a better quality than it actually is; however, it is sufficient to determine a ballpark relationship between the magnitude of the perturbations relative to the original signals.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Image-Based Attacks

A common method for audio classification is to process the input audio by translating it from a time series signal to frequency space to create a spectrogram image. Classification is then performed on these spectrogram images instead of the raw audio input. This audio processing method is used by the classifier developed by Pfau and serves as the target classifier for this study [11]. Knowing the target network processes audio signal into the frequency domain creates a potential attack vector for fooling the system. By accessing the system just between the data processing of the target system and the classification network, perturbed audio examples can be supplied to the classifier to elicit incorrect classifications.

3.1 Underwater Sound Dataset and Target Network

The underwater sound dataset and target neural network used throughout this work were developed by Pfau [11]. The dataset consists of 30-second audio clips stored as wav files with a 4 kHz sample rate. The audio was recorded at Thirty Mile Bank off of the coast of Southern California between December 2012 and April 2013. Pfau goes on to assign each 30 second clip to a class based on the presence of a ship during that clip according to an ontology introduced by Santos-Domínguez [18]. The ontology groups ships into four classes based on the relative tonnage of each vessel, leaving a fifth class for the presence of no ship.

Class	Example Vessel Types
A	Fishing Vessel, Tug, Towing Vessel
B	Pleasure Craft, Sailboat, Pilot
C	Passenger Ship, Cruise Ship
D	Tanker, Container Ship, Military Ship, Bulk Carrier
E	No ship present

Table 3.1. Example vessels for each class.
Adapted from [11].

Presence of a specific class of ship was determined by comparing the time stamps of a sound clip with corresponding shipboard Automatic Identification System information and broadcast Maritime Mobile Service Identity numbers or International Maritime Organization numbers.

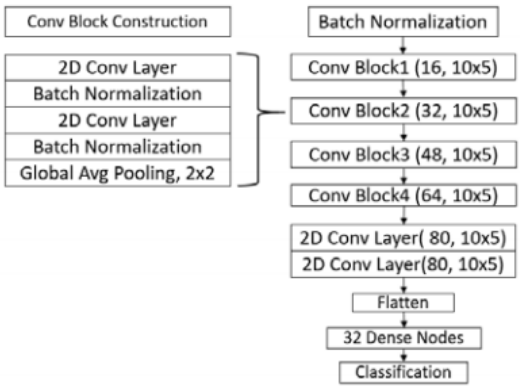

Figure 3.1. Structure of target classification network.
Source: [11].

Input audio is processed into spectrogram images by using a Mel-filter bank to create spectrograms on a scale similar to the human auditory range. The network processes the input images through a series of convolutional blocks composed of individual convolutional layers followed by batch normalization layers. The performance of the version of the target network on the test dataset is show in Table 4.1. A more detailed explanation and evaluation of the target network and dataset construction can be found in [11].

	Precision	Recall	F1-Score
class A	0.67	0.86	0.75
class B	0.71	0.88	0.78
class C	0.58	0.97	0.73
class D	0.96	0.72	0.82
class E	0.68	0.99	0.81
accuracy	-	-	0.80

Table 3.2. Precision, Recall, and F1-Score by target network.

3.2 Foolbox Survey

To evaluate the susceptibility of the target network to adversarial attack, we conduct a survey of known attack methods. The tested methods are available in an open source python library Foolbox [10]. Methods discussed in Section 2.1 are implemented with several gradient descent methods and norms constraining the adversarial perturbations. Foolbox also implements attack methods that differ in approach to those discussed previously. Examples of these attacks include the Carlini-Wagner Attack [19], the Inversion Attack [20], the Boundary Attack [21], and others.

Foolbox specifies two approaches for conducting attacks: misclassification and targeted misclassification. The misclassification approach uses each Foolbox attack method to perform an untargeted attack. This means that for a given example spectrogram a successful attack is any perturbed version of the example for which classifier classifies the adversarial example as anything other than the example's true label. The targeted misclassification approach will perform a targeted attack. That is to say, for any example provided the attack method will attempt to find a perturbation that makes the adversarial spectrogram be classified as a given class.

The survey is conducted by evaluating each available attack implemented by Foolbox at different values of ϵ and the default parameters associated with that attack. All attacks are conducted with the misclassification criteria, such that they are creating untargeted attacks. Each attack method is tested on the 7264 test examples in the underwater sound data set.

3.3 Survey Results

Our survey considers each attack made available by Foolbox. For consistency between the evaluated attacks, we exclude attacks that require any additional, attack-specific parameters from the survey. Of the 44 attacks implemented in Foolbox, 29 were able to evaluated on the underwater sound dataset. Attacks are evaluated on their ability to create successful adversarial examples from the examples available in the test set. Success rates from the top N attacks are shown in Figure 3.2.

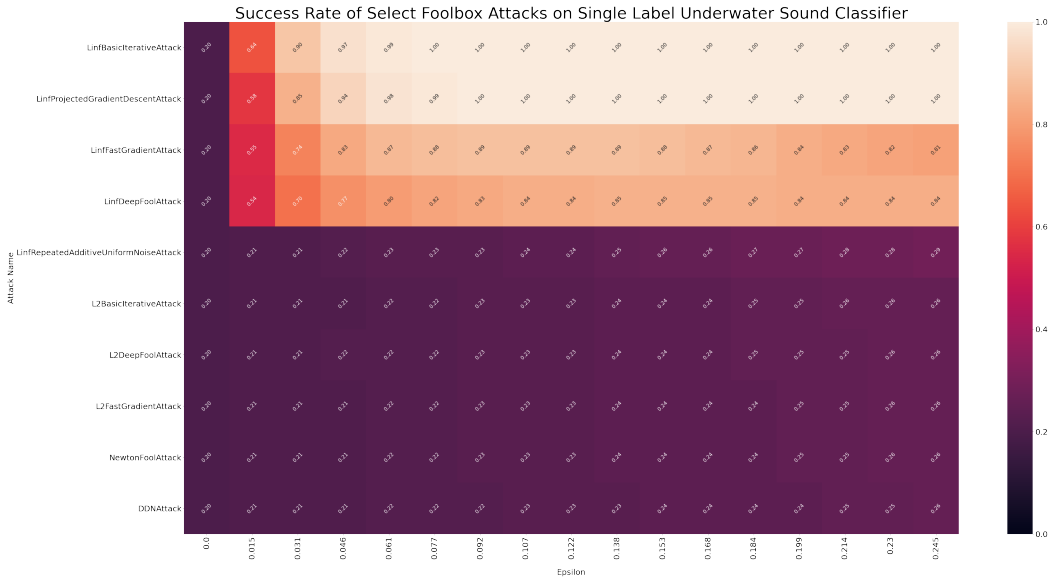

Figure 3.2. Heatmap of top performing Foolbox attack success rates for listed epsilon values.

The results of the survey show that the classifier is susceptible to a variety of the attacks implemented in Foolbox. For the attacks tested, all demonstrate some level of success in attacking the target classifier. Among the best performing attacks are those that rely on the L_∞ distance metric. We hypothesize that the L_∞ distance metric allows for larger perturbations per pixel when compared to other distance metrics such as L_1 or L_2, due to the metric constraining the size of only the largest pixel value in the perturbation. The best performing attacks begin to reach 100 percent success rates for fairly small values of ϵ, around 0.071. Examples of successful attacks generated using methods implemented in Foolbox are given in Figure 3.3.

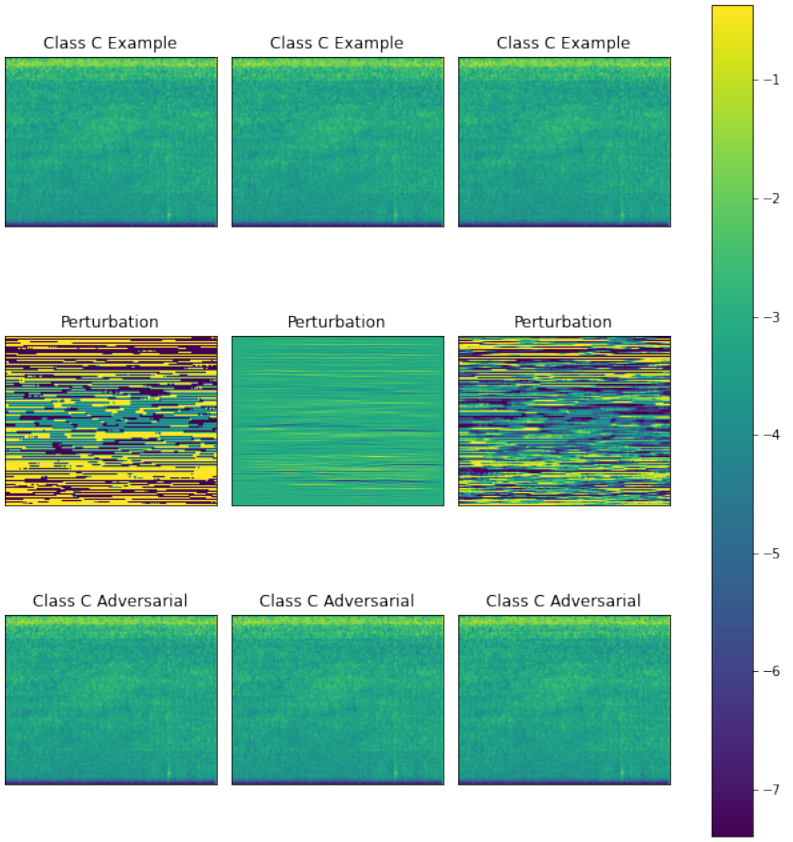

Figure 3.3. Examples of L_∞ Deepfool attack (left), L_2 Basic Iterative attack (center), and L_∞ Basic Iterative attack (right) on a Class C spectrogram image with $\epsilon = 0.06$.

The generated adversarial images demonstrate the property of being imperceptibly different from the original spectrograms. This property makes these attacks more likely to fool a human observer, in addition to the target network. The quality of being imperceptible to an observer makes the attack more likely to remain undetected, when compared to an attack that can be perceived by an observer.

3.4 Attack Scenario

Deploying this type of image-based attack to affect a real sound classification system poses some challenges. The first of which is the digital nature of this attack. In order to provide a perturbed spectrogram as input to the classifier, the system would need to be intercepted

between the audio processing step and the input to the neural network. Secondly, many of the Foolbox attacks require the parameters of the network to be available in order to construct an attack. To overcome these obstacles, the attack would need to have full access to the inner workings of the sound classification system. Assuming that the attacker is able to gain access to the system, he or she would be able to use one of these image based attacks to distort spectrogram images at will. The attack can use this to hide the presence of one of his or her underwater assets or to mislead the system by making it report a contact where there is none. The attack also has the advantage of being inconspicuous. An attacker with access to the system, could choose to take the system offline completely, alerting his or her adversary to the system failure. Alternatively, by injecting adversarial images into the network, the attacker is able to maintain a persistent advantage by selectively choosing when the system will malfunction and for how long. While this type of attack is theoretically possible, the requirement to have full access to the system is not plausible in real situation. In the scenario where an attacker has gained access to the system, the attacker could deploy simpler methods of attack to achieve the same effect as with injecting adversarial examples into the network. One example would be for the attack to manipulate the output of the network directly. That is to say, regardless of the input to the classifier, an attacker with complete access to the system is able to arbitrarily change the classification reported by the network.

CHAPTER 4:
Audio Attack Methodology

Audio-based attacks on input features in time domain of sound classification system have two distinct advantages over image based attacks. First, constructing adversarial audio attacks does not require any assumptions to be made about how the classifier processes audio input. Specifically, it does not rely on the classifier using spectrogram images as input features. This enables the attack generation to occur in a black-box fashion. Secondly, an adversarial audio input can be extended into an attack deployed in a physical space. For this type of classification system, an adversarial image would need to be provided to the target classifier digitally, circumventing the input processing done by the system. In contrast, adversarial audio can be provided as raw input to be provided by the system. An attack of this type, if it is sufficiently robust, can be deployed such that the adversarial audio is played through the sound transmission medium to be detected by the system in the physical space [16].

Adversarial audio attacks can be categorized into two broad types: iterative optimization [8], [16] or through a generative model [12]. We focus on the latter approach because a generative model will allow for faster audio generation times at deployment. Increased audio generation speeds make this style of attack more well-suited to a real-time attack in a physical space. A draw back of generative models for audio signals is that there are often significant artifacts in the audio signal due to the generation process [9]. Audio artifacts reduce the effectiveness of these adversarial attacks because a human observer is able to identify artificially crafted signals by the presence of artifacts. To combat this we propose a novel approach for reducing the impact of audio artifacts by introducing penalty functions based on perceptual audio metrics. Including these functions in training forces the generative model to produce audio signals that are perceptually similar to the source signals, thereby reducing the perceived audio artifacts.

4.1 Training Scheme

Commonly adversarial audio is generated by treating each example as an individual optimization problem [8], [16]. This approach has been shown to generate high quality ad-

versarial examples that are difficult for humans to distinguish from benign examples. As pointed out by Xie et al. [12], these methods are often too slow to deploy an attack in real time. To combat this they developed an attack generation method using a generative neural network to construct audio perturbations. This approach reduces generation time for single adversarial examples compared to the iterative optimization approach by leveraging offline learning. At generation time, only a single pass through the generative network is required to create an adversarial example. For this reason, we adapted a training method based on the work of Xie et al. for the present study. Our approach is Algorithm 1, below:

Algorithm 1: Audio Attack Training Procedure

Result: Trained audio perturbation generator: G(.).

Input: training data $X = \{x_1, ..., x_n\}$, labels: $Y = \{y_1, ...y_n\}$, target classifier: $F(.)$, penalty functions: $P = \{P_0..P_k\}$

for *each training epoch* **do**

  for *number of steps* **do**

    $x \leftarrow$ minibatch of m samples from X

    $y \leftarrow$ minibatch of m labels from Y

    $x' \leftarrow x + G(x)$

    $y' \leftarrow F(x')$

    $loss = \frac{1}{k+1}\big(- \frac{1}{m} \sum_{i=0}^{m} y_i \cdot log(1 - \hat{y_i}) + (1 - y_i) \cdot log(\hat{y_i})\big) + \frac{1}{k+1} \sum_{i=0}^{k} P_i(x, x')$

    $G(.) \leftarrow backprop(loss)$

  end

end

The implemented training scheme differs from that in [12] in two significant ways: the objective function is changed to train the generative network to preform an untargeted attack and the generative network is simplified to have fixed embeddings. The two changes are related. Training for an untargeted attack is conducted in part to simplify the learning process for the generator. To facilitate a targeted attack, additional parameters would need to be added to the network architecture. The additional parameters could take the form of multiple output layers, swappable embeddings [12], or as completely separate neural networks. Intuitively, removing the requirement for the additional parameters reduces the training time required for the model to learn how to create adversarial perturbations. Additionally, it is sufficient to show the vulnerability of the target network to adversarial examples and how to improve the quality of those adversarial examples through an untargeted attack. The original architecture had swappable embeddings to facilitate the training of the generative network to construct

targeted attacks. Since the study does not desire a targeted attack on the network, there is no longer the need for the swappable embeddings in the generative network.

The generative network is trained in a manner similar to how Generative Adversarial Networks (GAN) are trained [22]. In GAN training, two neural networks are being trained simultaneously. One of these networks is the generator and the other is the discriminator. The job of the generator is to generate realistic examples that look like the training data, such that generated examples are able to fool the discriminator network. The job of the discriminator is to learn how to distinguish between examples that were a part of the training dataset and those created by the generator. An overview of training method is provided by Figure 4.1.

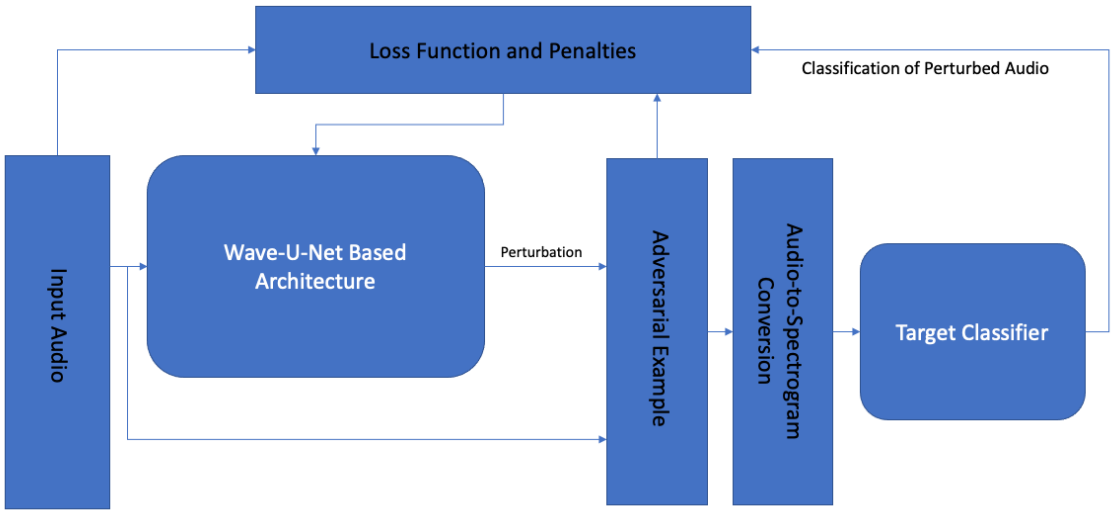

Figure 4.1. Overview of audio perturbation generation training scheme.

Our training scheme, defined in Algorithm 1, exclusively trains the generative network with the goal of create audio perturbations. The target classifier plays the role of the discriminator. Since, the discriminator is not being trained in our attack scheme we are able to treat the target classifier as a black-box. In other words, our attack scheme is black-box, independent of the parameters of the target network. A more general definition of a black-box attack would prohibit any knowledge of the classification system beyond the input and output, not only the parameters of the target classifier. In implementation, our attack does not conform to this latter definition. We implement an "Audio-to-Spectrogram" conversion module to replace the normal input pipeline of the classification system. This module performs all of

the input processing that is performed to translate audio into spectrograms that the target classifier uses as input features. In reality this module is not required to perform the attack. We could have written each adversarial example to a file and provided them to the input pipeline of the target classifier instead. Training in this method would be very slow as each epoch would require multiple reads and writes of data from memory. This module is implemented in order to improve the training time of the generative architecture by removing the need for these read and write operations. Since the conversion model is optional, the form of the attack satisfies the general definition of a black-box attack as well, even if our implementation does not.

4.2 Generative Architecture

Wave-U-Net was developed by Stoller et al. for the task of audio source separation [17]. The architecture functions by passing an input audio source through a series of n downsampling blocks, followed by a series of n corresponding upsampling blocks. Each Downsample block, D_i, performs a 1-D convolution with filter size F_d followed by a decimation. As defined by the authors of the orignal paper, the decimation is performed by discarding every other sample in the time domain. An Upsampling block, U_i concatenate the features from the output of D_i and then performs a simple upsample of these features. This is followed by 1-D convolution with filter size. F_u. This process reduces the size of the feature space significantly through downsampling, and then returns it to the original input dimensionality through the matching upsampling steps. Operating in this fashion creates a generative network the is seeded by the original input signal. This approach is in contrast to using noise as the input to the generative network directly. In effect, the downsampling process creates a specifically crafted noise vector that is used as input to the generative portion of the architecture. This property makes the architecture a natural choice for generating adversarial audio samples, which is the task of taking an arbitrary audio signal and generating a new audio signal to add to the original to perform an adversarial attack. A diagram of the Wave-U-Net architecture is shown in Figure 4.2. The original architecture output k audio signals as the output of an audio source separation task. This architecture differs from the original architecture by only having a single output channel as multiple output channels are not needed for the task of generating adversarial examples.

The original paper specifies values for F_d, F_u, and n as 15, 5, and 12 respectively. The audio

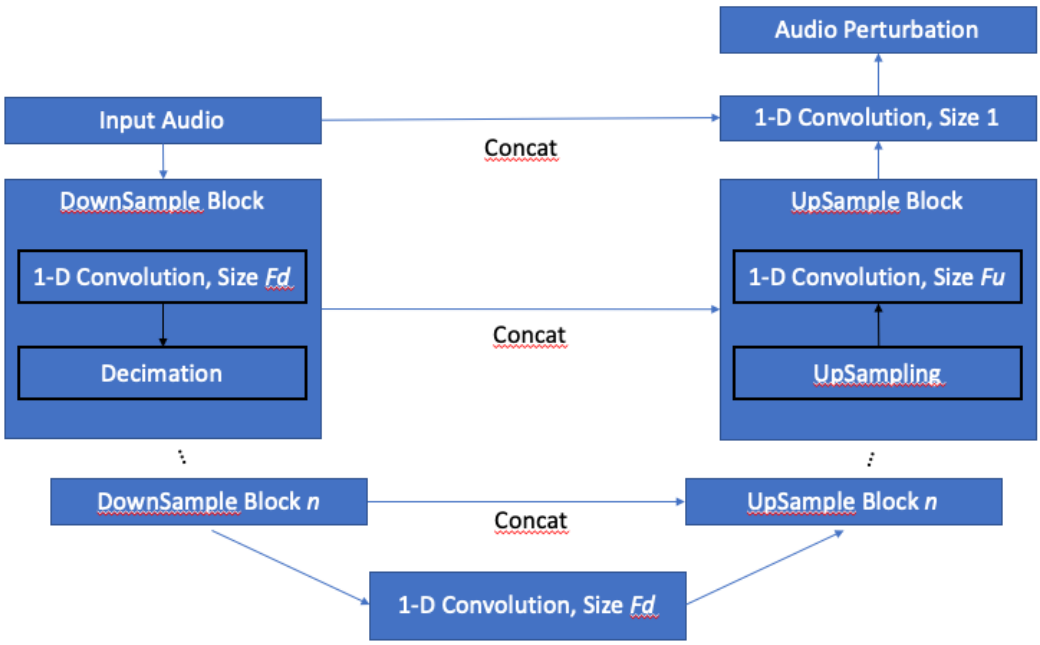

Figure 4.2. Modified Wave-U-Net architecture.

signals used in [17] and the ones used in this work have a different number of samples per signal(16384 and 120000). To account for this difference the parameters used by the authors were changed to allow for proper functioning of the network, while maintaining as many of the original parameters as possible. As a result, the depth of the network needed to be limited. The values for F_d, F_u, and n used in this experiment are 15, 5, and 6.

4.3 Choice of Loss Function

As indicated in section 4.1, the generative network is being trained to generate an untargeted audio perturbation for a given input sample. To accomplish this, the network needs to be encouraged to produce audio signals that when added to the input audio produce an incorrect classification from the target network. This can be expressed for use in optimization in terms of the Cross Entropy loss function that is often used in training neural networks:

$$CrossEntropy(y, \hat{y}) = -\frac{1}{N}\sum_{i=0}^{N} y_i \cdot log(\hat{y}_i) + (1 - y_i) \cdot log(1 - \hat{y}_i). \tag{4.1}$$

This loss function is small when the true label, y, is similar to the label predicted by the neural network, $\hat{y}$. Upon inspection this captures the an opposite behavior to what is desired in the loss function to train an untargeted attack. That is, the desired loss function should be small when y and $\hat{y}$ are different and large when the two values are similar. The desired loss function can be achieved by substituting $1 - \hat{y}$ for $\hat{y}$ in equation 4.1:

$$CrossEntropy(y, 1 - \hat{y}) = -\frac{1}{N}\sum_{i=0}^{N} y_i \cdot log\Big((1 - \hat{y}_i)\Big) + (1 - y_i) \cdot log\Big(1 - (1 - \hat{y}_i)\Big). \quad (4.2)$$

Which, when simplified becomes one component of the training objective for our proposed attack method.

$$UntargetedLoss(y, \hat{y}) = -\frac{1}{N}\sum_{i=0}^{N} y_i \cdot log(1 - \hat{y}_i) + (1 - y_i) \cdot log(\hat{y}_i) \quad (4.3)$$

A similar derivation can be used to arrive at a loss function to train a generative network to construct a targeted attack on the target network. From equation 4.1, the true label y is replaced with a fixed set of labels y^*, a set of labels where each individual label is the label belonging to the class being targeted.

$$TargetedLoss(y^*, \hat{y}) = -\frac{1}{N}\sum_{i=0}^{N} y_i^* \cdot log(\hat{y}_i) + (1 - y_i^*) \cdot log(1 - \hat{y}_i) \quad (4.4)$$

For example, in an attack targeting class A, y^* becomes an appropriately sized matrix consisting of labels matching class A. The behavior of this function is to be small when all predicted labels are similar to the label for class A, indicating that all examples fooled the target classifier into predicting the examples to be class A, the definition of a successful targeted attack.

Attacks using this targeted loss function are explored in [12]. This work does not go on to evaluate the effectiveness attacks created using this loss function, instead choosing to focus

on untargeted attacks and the impact of perceptually motivated penalty functions. Equation 4.4 is included here for completeness.

4.4 Perceptually Motivated Penalty Functions

Penalty functions are functions that evaluate some additional constraint to be placed on an optimization problem. In training a neural network, penalty functions are often used to regularize the weights of the network [13]. Alternatively, a penalty function may be used to constrain the output of the network more directly. We will be taking the latter approach. As a result, the objective function for the neural network becomes a sum of the loss function and one or more of these penalty functions.

$$J(\cdot) = Loss(\cdot) + P(\cdot) \tag{4.5}$$

Xie et al. define an objective function of this form for their FAPG attack [12]. In their function they place an additional constraint on the generated adversarial signal in the form of an L2-norm, or mean squared error (MSE), between the adversarial example and original audio signal.

$$J(y*, \hat{y}, x, \hat{x}) = -\frac{1}{N}\sum_{i=0}^{N} y_i^* \cdot log(\hat{y_i}) + (1 - y_i^*) \cdot log(1 - \hat{y_i}) + \beta \cdot \|x - \hat{x}\|_2 \tag{4.6}$$

Experimentation with the MSE penalty function in an untargeted attack lead to the creation of audio signals that sounded artificial (e.g., sound had audible artifacts). This artificial quality of the generated sounds could lead to an observer identifying an adversarial example as such. It would be nice if the network could be trained to generate sounds that sounded more natural to a human observer. To accomplish this, we propose the inclusion of perceptual quality metrics as penalty functions into the training of the generative network. Perceptual quality metrics, such as loudness, STOI, and PESQ, are all metrics by which audio signals can be evaluated and scored based on ways that a human would perceive that audio signal.

In order to define a form for perceptually motivated penalty functions (PMP), we begin by defining a difference function, Δ, for a given perceptual metric, Q, over the original

and adversarial signals x and $\hat{x}$. In certain cases, a perceptual metric may be inherently relative (e.g., PESQ). Defining a difference relationship for relative metrics does not make any sense because they are defined in a way to capture the difference between some signal and a reference signal. As a result the difference function, Δ, for a relative metric is simply the result of that metric.

With a difference function established, we can develop a function to penalize any relative difference in the chosen perceptual metric by following the steps outlined in [23]. We proceed by defining an inequality using the relative function and a desired threshold parameter, α. This inequality establishes the constraint that the measured perceptual difference between the original and adversarial signal should be no larger than α.

$$\Delta_Q(x, \hat{x}) \leq \alpha \tag{4.7}$$

Which, can be rewritten as:

$$\Delta_Q(x, \hat{x}) - \alpha \leq 0. \tag{4.8}$$

From this inequality, we would like to define a function that is 0 when this inequality is false. That is when the perceptual difference between the adversarial signal and original signal is less than the chosen threshold there should be no penalty. Otherwise, this function should return some positive value to penalize the difference on the metric Q. Since, the difference relationship is defined to be greater than 0 in the cases we want to penalize, a maximum function can be used to define a penalty function of order n,

$$P_Q(x, \hat{x}) = \max\left[0,\ \Delta_Q(x, \hat{x}) - \alpha\right]^n \tag{4.9}$$

Equation 4.9 represents the most general form of a PMP. To demonstrate the use of a PMP in training an adversarial attack we define a penalty using the integrated loudness of the audio signals. The integrated loudness calculation is performed using the standard described by ITU-R BS.1770-4 [24]. The loudness of a signal over a time interval, T, is defined as:

$$LKFS = -0.691 + 10\log_{10}\sum_i G_i \cdot z_i. \tag{4.10}$$

G_i are the weighting coefficients for the audio channels and z_i is the power of the input signal according to

$$z_i = \frac{1}{T} \int_0^T y_i \, dt \tag{4.11}$$

The numerical implementation used is available in an open-source python toolbox, pyloudnorm [25].

As the integrated loudness is an absolute metric, we must first define a difference relationship between the original and adversarial signals.

$$\Delta_L(x, \hat{x}) = LKFS(\hat{x}) - LKFS(x). \tag{4.12}$$

Equation 4.12 will be a positive value when the adversarial signal is louder than the original signal, a property that we would like to penalize by the inclusion of a PMP. Next, we set $\alpha = 0$. This means that we penalize the adversarial signal for being any louder than the source signal. Finally, we decided on a quadratic PMP to match the order of the penalty function used in [12]. This gives our loudness PMP as:

$$P_L(x, \hat{x}) = \max\left(0, \ \Delta_L(x, \hat{x})\right)^2. \tag{4.13}$$

4.5 Experimental Setup

To evaluate the effectiveness of the proposed training scheme and PMP we will train several Wave-U-Net generators using the training scheme described in Algorithm 1. We will test the training scheme on two different datasets. The first is the same dataset and target classifier discussed in Section 3.1. In addition we will test the attack generation scheme on a classifier trained on a subset of Google's Speech Command Dataset [26]. Our subset consists of utterances of the digits zero through nine and will be referred to as the SC09 dataset. Each sample in the SC09 dataset is one second long and saved at a 16 kHz sample rate. For the SC09 dataset, a classifier is trained using the same architecture as described in the

TensorFlow speech recognition tutorial [27]. The architecture given in Tensorflow's tutorial is shown in Figure 4.3 and the training curves for our implementation of this architecture are given in Figure 4.4.

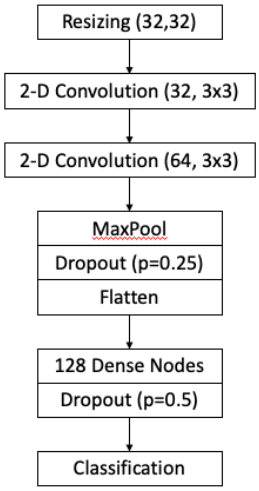

Figure 4.3. Structure of SC09 target classification network.

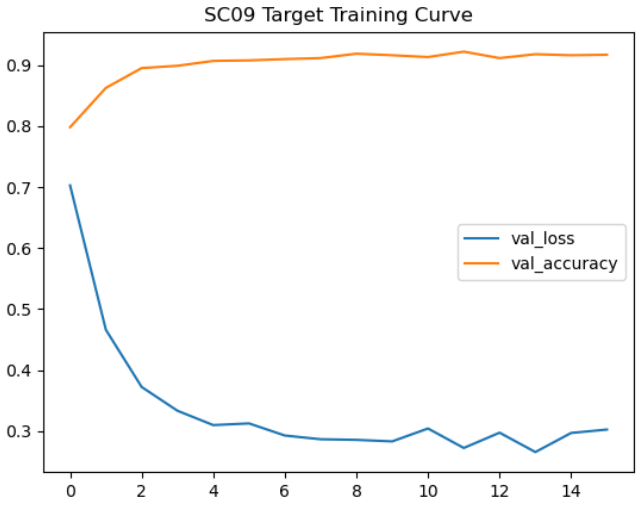

Figure 4.4. SC09 target model training curves.

Despite its simplicity the target architecture is able to achieve a fairly high degree of accuracy

	Precision	Recall	F1-Score
Zero	0.93	0.95	0.94
One	0.94	0.92	0.93
Two	0.88	0.90	0.89
Three	0.97	0.92	0.95
Four	0.92	0.96	0.94
Five	0.89	0.91	0.90
Six	0.96	0.93	0.94
Seven	0.94	0.94	0.94
Eight	0.93	0.93	0.93
Nine	0.92	0.92	0.92
accuracy	-	-	0.93

Table 4.1. Precision, Recall, and F1-Score by target network on SC09 classifier.

across all 10 classes, Table 4.1. The target classifier achieves an overall accuracy of 93% on the test examples and the lowest F1-score for any class is 0.89.

The additional architecture and dataset are chosen to be able to offer more insights to the perceptual differences between generated signals as a human observer and to test the method against publicly available data and architecture. Verifying measured perceptual improvements on the underwater soundscape data is difficult because to a human observer this data is substantially similar to noise. Instead, by including a test of speech data we are able to manually inspect generated examples for perceptual differences because we know what speech is meant to sound like. Furthermore, both the dataset and code to create the model are publicly available enabling reproduction of these tests.

For each dataset, two generative networks will be trained. One of these networks will be trained using an objective function that utilizes a MSE penalty function like the attacks conducted in [12].

$$J_{MSE}(y, \hat{y}, x, \hat{x}) = \frac{1}{2}\Big(- \frac{1}{N}\sum_{i=0}^{N} y_i \cdot log(1 - \hat{y}_i) + (1 - y_i) \cdot log(\hat{y}_i) + \|x - \hat{x}\|_2 \Big) \quad (4.14)$$

The other will be trained with an objective function incorporating the PMP defined in equation 4.13.

$$J_L(y, \hat{y}, x, \hat{x}) = \frac{1}{2}\Big(- \frac{1}{N} \sum_{i=0}^{N} y_i \cdot log(1 - \hat{y}_i) + (1 - y_i) \cdot log(\hat{y}_i) + \max\left(0,\ \Delta_L(x, \hat{x})\right)^2 \Big) \quad (4.15)$$

All four networks will be trained for a total of 20 epochs with a batch size of 64. The network is optimized using the Adam optimizer with a learning rate 0.0001, and decay rates $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The iteration of each network that performs the best on its objective function over a separate set of validation data will be used for analysis in Chapter 5.

CHAPTER 5:
Audio Attack Results

5.1 Evaluation Criteria

We will be evaluating the success of the generated adversarial attacks on three criteria: success rate, a distortion metric, and relative loudness. The success rate of an attack is the percentage of generated examples that successfully fool the target classifier. The distortion metric used will be calculated using the same equation used by Xie et al. [12] shown in equation 2.5. Finally, we will evaluate the relative loudness of the generated perturbation in comparison to the source signal. Given that the loudness of a signal is given in decibels, the following equation describes a ratio between the source signal and the perturbation in terms of loudness, similar to the signal-to-noise ratio.

$$RelativeLoudness(x, \delta) = loudness(x) - loudness(\delta) \tag{5.1}$$

An important aspect of this metric is that it is a comparison of the perturbation and the original signal as opposed to the adversarial signal and the original signal as is done with the difference function shown in Equation 4.12. If we were to compare the adversarial signal to the original signal in this manner we would likely see them to be of a similar loudness, as this is the behavior encouraged by minimizing Equation 4.15 during the training process. Instead, we compare the loudness of the original signal to the perturbation to arrive at a metric that is similar in formulation as a signal-to-noise ratio. For this metric, a large value indicates a perturbation that is much quieter than source signal, a characteristic that is desired in constructing an adversarial attack.

5.2 Underwater Soundscape Attacks

Attacks on the underwater soundscape classifier show a high rate of success using both a MSE penalty function and the loudness-based PMP accross all of the test examples, Table 5.1. The generative model trained with our PMP performed significantly worse on class

B examples. When trained with an MSE penalty function the attack success rate on class B is 0.828, while only being 0.148 with the PMP. This was the least successful class for from training with both of the tested objective function. Class B is also the least represented class in the dataset. Of the 58,290 examples present in the training dataset, only 1,031 are class B examples. Being the least represented class it makes sense the the generative model performs the worst on these class. Since class B is such a small percentage of the total number of examples, the poor performance on class B does not have a significant impact on the success rate when measured as a fraction of all examples.

Class	MSE Penalty	Loudness Penalty	**Delta**	Truncated MSE Penalty	Truncated Loudness Penalty	**Delta**
Class A	0.998	0.998	**0.0**	0.998	0.998	**0.0**
Class B	0.828	0.148	**-0.680**	0.828	0.148	**-0.680**
Class C	0.998	0.860	**-0.138**	0.998	0.860	**-0.138**
Class D	1.00	0.998	**-0.002**	1.00	0.998	**-0.002**
Class E	1.00	0.980	**-0.020**	1.00	0.980	**-0.020**
Overall	0.994	0.967	**-0.027**	0.994	0.967	**-0.027**

Table 5.1. Success rate of generated audio attacks on underwater soundscape classifier.

Training with the PMP caused a noticeable improvement in the integrated loudness of the generated signal. Overall, the MSE penalty function created perturbations that were 4 dB louder than the source signal. In contrast, training with the loudness PMP shows that on average the generated perturbations are 0.5 dB quieter than the source signals, Table 5.2. This is in direct contrast to the performance seen on the distortion metric defined in Equation 2.5. By this measure, we saw that the generated signals had an average distortion of -6 dB and 10 dB for the MSE penalty and PMP respectively. The distortion metric suggests that an attack produced by MSE is less likely to be detected by an observer than the attack produced with the PMP, and relative loudness measure suggests the opposite. This is not a surprising result. Training the model using MSE penalty function will produce perturbation signals that are similar to the source signal, reducing the value of the distortion metric. The examples generated by our method did not perform as well as either of the prior works that measured results using the distortion metric. Those papers reported distortion metrics as

low as -30 dB [8], [12]. This is most likely due to those methods employing signal clipping to a specified value. For example, Xie et al. [12] use a clipping value $\tau = 0.05$ which will decrease the value of their distortion metric where in the worst case it could be -6 dB. Furthermore, this metric is based on a singular value from the tested signals, whereas our measure of relative loudness is measurement that is comprehensive of the entire signal. For this reason, we prefer the relative loudness as a way to measure the quality of the generated examples.

Class	MSE Penalty (dB)	Loudness Penalty (dB)	**Delta (dB)**	Truncated MSE Penalty (dB)	Truncated Loudness Penalty (dB)	**Delta (dB)**
Class A	-5.133	-0.210	**4.923**	-4.775	-0.343	**4.432**
Class B	-2.186	1.738	**3.924**	-1.023	0.712	**1.735**
Class C	-5.356	-0.468	**4.888**	-5.040	-0.626	**4.414**
Class D	-4.693	-0.080	**4.613**	-4.100	-0.460	**3.64**
Class E	-3.587	1.106	**4.693**	-2.375	0.115	**2.490**
Overall	-4.024	0.503	**4.527**	-3.180	-0.093	**3.087**

Table 5.2. Relative loudness of generated perturbation on underwater soundscape classifier in dB.

The generated signals, shown in figure 5.1, show clear edge artifacts. Those being the large spikes in signal intensity at the beginning and end of the adversarial signals.

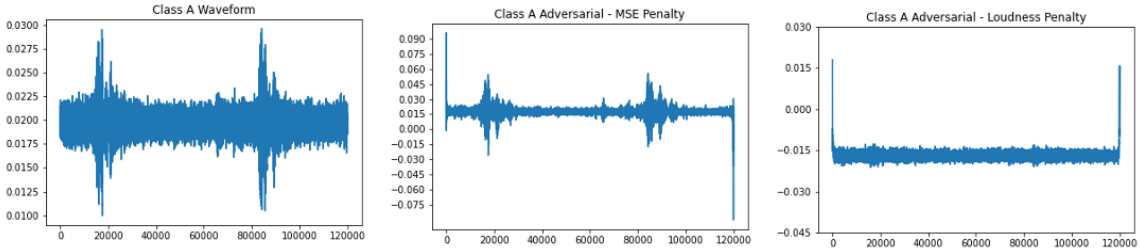

Figure 5.1. Example waveforms from underwater soundscape attacks.

This leads to the concern that the success of the attacks could be a result of these edge artifacts. To test this we reevaluate the signals by truncating the generated perturbations to remove the produced edge artifacts. The results of these tests are in the right side of tables 5.1 and 5.2. From these results we see a near zero change in the success rates of the attacks.

This indicates that the adversarial properties of the generated signals were not primarily the result of the edge artifacts. There is a decrease in the improvement in relative loudness between the two attacks, however, there is still a notable improvement on this metric.

5.3 Speech Command Attacks

Using our loudness-based PMP had an even larger impact on the quality of attacks on the SC09 classifier than the underwater soundscape classifier. Table 5.3 shows the success rate of the attacks on all 10 classes. We see high success rates with both penalty functions when generating attacks on this dataset. Unlike the underwater sound data there is not a single class that performs significantly worse than the other classes. The SC09 dataset is a balanced dataset. That is, there is an equal representation of all classes in the training and test data. This observations reinforces the conclusion that the under performance on class B observed in Section 5.2 is caused by the lack of training examples for that class.

Class	MSE Penalty	Loudness Penalty	**Delta**	Truncated MSE Penalty	Truncated Loudness Penalty	**Delta**
Zero	0.987	0.987	**0.0**	0.987	0.987	**0.0**
One	0.996	0.975	**-0.021**	0.996	0.975	**-0.021**
Two	1.00	1.00	**0.0**	1.00	1.00	**0.0**
Three	1.00	0.996	**-0.004**	1.00	0.996	**-0.004**
Four	0.996	0.992	**-0.004**	0.996	0.992	**-0.004**
Five	1.00	0.992	**-0.008**	1.00	0.992	**-0.008**
Six	0.987	0.975	**-0.012**	0.987	0.975	**-0.012**
Seven	0.979	0.983	**0.004**	0.979	0.983	**0.004**
Eight	0.970	0.992	**0.022**	0.970	0.992	**0.022**
Nine	1.00	0.992	**-0.008**	1.00	1.00	**0.0**
Overall	0.992	0.989	**-0.003**	0.992	0.989	**-0.003**

Table 5.3. Success rate of generated audio attacks on SC09 classifier.

Attacks developed with the loudness-based PMP show a significant improvement in the relative loudness of generated signals, Table 5.4. We see an average improvement of 8.7 dB in the relative loudness by training with our PMP. Furthermore, by listening to the generated

signals we are able to verify the perceptual improvement indicated by the calculated metrics for a smaller set of test examples. In those examples, we observe the signals generated using the MSE penalty had an artificial quality to them. Those generated with our PMP did not have this same quality. Instead, save for edge artifacts from the sound generation process, the attacks generated using our PMP sounded indistinguishable from the source signals.

Class	MSE Penalty (dB)	Loudness Penalty (dB)	**Delta (dB)**	Truncated MSE Penalty (dB)	Truncated Loudness Penalty (dB)	**Delta (dB)**
Zero	2.588	11.946	**9.358**	2.588	12.052	**9.464**
One	4.570	15.220	**10.65**	4.580	15.019	**10.439**
Two	1.865	12.487	**10.622**	1.879	12.496	**10.617**
Three	2.094	11.983	**9.889**	2.099	12.095	**9.996**
Four	2.770	13.380	**10.61**	2.777	13.567	**9.790**
Five	6.172	12.584	**6.412**	6.175	12.792	**6.617**
Six	4.501	13.482	**8.981**	4.498	13.436	**8.938**
Seven	5.096	12.821	**7.725**	5.010	12.944	**7.934**
Eight	3.437	12.219	**8.782**	3.449	12.255	**8.806**
Nine	5.427	10.317	**4.890**	5.427	10.415	**4.989**
Overall	4.086	12.815	**8.729**	4.089	12.858	**8.769**

Table 5.4. Relative loudness of generated perturbation on SC09 classifier in dB.

Unfortunately, a limitation of written presentation of audio signal processing is that a visual representation of the waveform does not always do a good job of capturing the auditory similarity between two signals, such as Figure 5.2. That is for the majority of the audio samples manually verified the waveform at the right of Figure 5.2 sounds more like the original signal than the signal in the center of Figure 5.2 despite the generation artifacts.

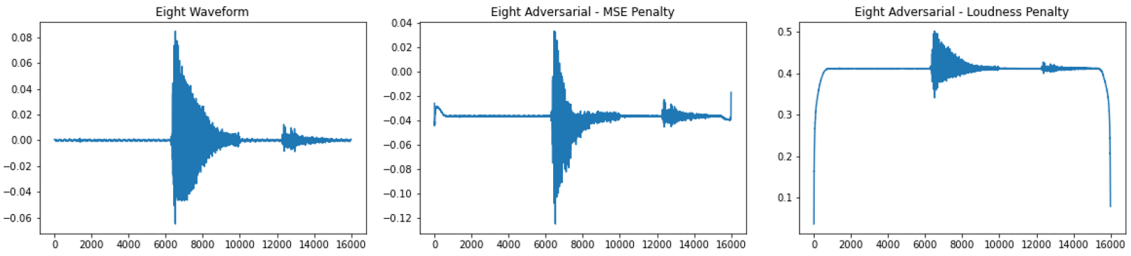

Figure 5.2. Waveforms generated for a class Eight example.

Examining the waveforms in Figure 5.2 we see evidence of two types of artifacts in the generated signals. Specifically, we see edge artifacts, like we saw with the signals discussed in section 5.2, and a DC bias was introduced in the generation process. The DC bias is a property of the waveform not being centered at zero. In the examples show by Figure 5.2, we see the MSE penalty signal being centered at -0.04 and the loudness penalty signal being centered at 0.4. Both of these artifacts can be addressed with some systematic post-processing steps. First, the DC offset can be estimated and removed from the signal by applying a linear shift to the signal. Then, the same truncation method used in 5.2 is applied to the signal. After both of these steps, the generated waveforms have a shape much more similar to the original signal, Figure 5.3.

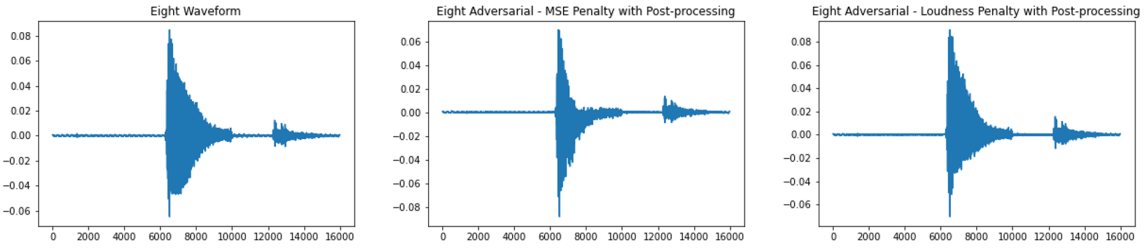

Figure 5.3. Waveforms generated for a class Eight example with post processing steps.

Including our loudness-based PMP in the training process of the generative network, we are able to achieve significant improvements in the measured perceptual quality of the adversarial examples without sacrificing attack success rate. More importantly, we are able to reduce the observed perceptual differences between the adversarial example and the source signal.

CHAPTER 6:
Conclusions and Future Work

6.1 Conclusions

This thesis work originally set out to demonstrate the feasibility of conducting adversarial attacks on underwater soundscape classification networks. We demonstrate the ability for open-source, image based adversarial attack libraries to create adversarial attacks on an underwater soundscape classification network. Five of the tested attacks had over 80% success rate with an epsilon value of 0.01. These spectrogram images also demonstrated a property of being imperceptibly different from the source spectrograms.

Additionally, we demonstrate the ability to create adversarial audio waveforms using a generative neural network. To capture the imperceptibility demonstrated by the image-based attacks we develop a novel training procedure that incorporates audio perceptual metrics as perceptually motivated penalty functions (PMPs). We provide a general formulation of this type of penalty function that can be used to construct additional penalties based on perceptual metrics. The proposed training scheme is demonstrated by developing a PMP based on the signal's integrated loudness. With this penalty function we were able to generate audio signals that are imperceptibly different from the source signals without placing explicit constraints on the magnitude of the generated signal. This is an improvement over the same training procedure that utilizes a more traditional MSE penalty function instead of a PMP.

The developed attack generation method is not limited to attacking classifiers in the underwater soundscape domain. The attack method is modular in its design to allow for the development of attacks on new datasets, target classifiers, and additional penalty functions. We show this by constructing attacks on a speech command classifier. Inclusion of a PMP in training this attack has an even greater impact on audio quality improvement than in the attacks on the underwater soundscape classifier.

In the case of both datasets, we are able to use a loudness-based PMP to construct adversarial audio waveforms that successfully fool the target classifier over 96% of the time. Furthermore, these attacks possess a measure quality increase over the same attack trained

using a MSE penalty function. Attacks on the underwater soundscape classifier benefited from a 4.5 dB average improvement and attacks on the SC09 classifier show an 8.7 dB average improvement in relative loudness. Training generative attack schemes using these penalty functions will allow for the creation of audio adversarial attacks that are more likely to go unnoticed by a human observer than those generated with a traditional MSE penalty function.

6.2 Future Work

This work demonstrates the feasibility of PMP in training a generative adversarial attack by using the integrated loudness of the signals. There are plenty of other reasonable perceptual metrics that could be used to construct similar penalty functions that could prove more effective than the signal loudness. For example, attacks in the speech space could benefit from penalty functions built using STOI, PESQ, or virtual speech quality objective listener (ViSQOL).

By using a generative architecture, the time required to generate singular examples is relatively low when compared to other audio attack methods [12]. Fast generation times enabled by the offline training of the generative model make this style of attack suitable to extension into an attack in a physical space. Pulling in methods demonstrated in [16], an attack scheme capable of attacking an audio classification system over the air, could be possible.

The type of target classifier could also be explored in more depth. For example, Bayesian neural networks introduce a probabilistic measure of uncertainty to the network's output. A high uncertainty measure may be able to indicate to the classification system that a specific example is an adversarial example instead of a benign input. Uncertainty measurement may make this type of classifier harder to attack with a black box, generative attack shown in this work.

Adversarial training is a useful approach for creating classification networks that are robust towards the types of attacks described in this work. Adversarial training functions by including adversarial examples as input examples to a neural network during the training process [7]. The attack method show in this work could be used for this purpose. Since,

the method is developed using a generative neural network, adversarial examples can be generated during the training process with out imposing a serious penalty to the training time of the new classifier.

Finally, the PMP training scheme proposed in this work could be extended to a targeted attack scheme. This work chose to simplify the problem by focusing on the untargeted case. The methods used by [12] could be easily altered to include the PMP style of penalty function into the training scheme. A work choosing to do this would be able to draw more meaningful conclusions about the inclusions of PMP on that fully established training methodology.

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[2] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, *Deep Speech: scaling up end-to-end speech recognition*, 2014, _eprint: 1412.5567.

[3] M. Gilday, "Frago 01/2019: A Design for Maintaining Maritime Superiority," 2019. Available: https://media.defense.gov/2020/Jul/23/2002463491/-1/-1/1/CNO%20FRAGO%2001_2019.PDF

[4] R. Hilger and B. Christman, "Strategic innovation from the deckplate," *Undersea Warfare Magazine*, pp. 7–24, 2019. Available: https://ufdcimages.uflib.ufl.edu/AA/00/04/80/32/00050/Summer-2019.pdf

[5] M. Thomas, B. Martin, K. Kowarski, B. Gaudet, and S. Matwin, "Marine mammal species classification using convolutional neural networks and a novel acoustic representation," in *Machine Learning and Knowledge Discovery in Databases*, vol. 11908, U. Brefeld, E. Fromont, A. Hotho, A. Knobbe, M. Maathuis, and C. Robardet, Eds. Cham: Springer International Publishing, 2020, pp. 290–305, series Title: Lecture Notes in Computer Science. Available: http://link.springer.com/10.1007/978-3-030-46133-1_18

[6] H. Yang, J. Li, S. Shen, and G. Xu, "A deep convolutional neural network inspired by auditory perception for underwater acoustic target recognition," *Sensors*, vol. 19, no. 5, p. 1104, Jan. 2019, number: 5 Publisher: Multidisciplinary Digital Publishing Institute. Available: https://www.mdpi.com/1424-8220/19/5/1104

[7] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv:1412.6572 [cs, stat]*, Mar. 2015, arXiv: 1412.6572. Available: http://arxiv.org/abs/1412.6572

[8] N. Carlini and D. Wagner, "Audio adversarial examples: targeted attacks on speech-to-text," in *2018 IEEE Security and Privacy Workshops (SPW)*, May 2018, pp. 1–7.

[9] N. Thiem, M. Orescanin, and J. B. Michael, "Reducing artifacts in GAN audio synthesis," *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1268–1275, 2020.

[10] J. Rauber, W. Brendel, and M. Bethge, "Foolbox: a Python toolbox to benchmark the robustness of machine learning models," *arXiv:1707.04131 [cs, stat]*, Mar. 2018, arXiv: 1707.04131. Available: http://arxiv.org/abs/1707.04131

[11] A. Pfau, "Multi-label classification of underwater soundscapes using deep convolutional neural Networks," Master's thesis, NPS, Monterey, CA, 2020.

[12] Y. Xie, Z. Li, C. Shi, J. Liu, Y. Chen, and B. Yuan, "Enabling fast and universal audio adversarial attack using generative model," *arXiv:2004.12261 [cs, eess]*, Apr. 2020, arXiv: 2004.12261. Available: http://arxiv.org/abs/2004.12261

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[14] A. Kurakin, I. Goodfellow, S. Bengio, and others, *Adversarial examples in the physical world*.

[15] T. Liiv and A. Strömberg, *Iterative gradient-based adversarial attacks on neural network image classifiers*, 2019.

[16] H. Yakura and J. Sakuma, "Robust audio adversarial example for a physical attack," *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 5334–5341, Aug. 2019, arXiv: 1810.11793. Available: http://arxiv.org/abs/1810.11793

[17] D. Stoller, S. Ewert, and S. Dixon, "Wave-U-Net: a multi-scale neural network for end-to-end audio source separation," *19th International Society for Music Information Retrieval Conference*, 2018.

[18] D. Santos-Domínguez, S. Torres-Guijarro, A. Cardenal-López, and A. Pena-Gimenez, "ShipsEar: an underwater vessel noise database," *Applied Acoustics*, vol. 113, pp. 64–69, 2016.

[19] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*. IEEE, 2017, pp. 39–57.

[20] H. Hosseini, B. Xiao, M. Jaiswal, and R. Poovendran, "On the limitation of convolutional neural networks in recognizing negative images," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2017, pp. 352–358.

[21] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: reliable attacks against black-box machine learning models," in *International Conference on Learning Representations*, 2018.

[22] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial networks*, 2014, _eprint: 1406.2661.

[23] J. Roberts and M. Kochenderfer, "Mathematical Optimization: penalty functions," 2014. Available: https://web.stanford.edu/group/sisl/k12/optimization/MO-unit5-pdfs/5.6penaltyfunctions.pdf

[24] "ITU-R BS.1770-4 algorithms to measure audio programme loudness and true-peak audio level," International Telecommunications Union, Recommendation, Oct. 2015.

[25] C. J. Steinmetz and J. D. Reiss, "pyloudnorm: a simple yet flexible loudness meter in Python."

[26] P. Warden, "Speech Commands: a dataset for limited-vocabulary speech recognition," *ArXiv e-prints*, Apr. 2018, _eprint: 1804.03209. Available: https://arxiv.org/abs/1804.03209

[27] "Simple audio recognition: recognizing keywords | Tensorflow Core." Available: https://www.tensorflow.org/tutorials/audio/simple_audio

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California