A STUDY ON THE TRANSFERABILITY OF ADVERSARIAL ATTACKS IN SOUND EVENT CLASSIFICATION

Vinod Subramanian[1,2], *Arjun Pankajakshan*[1], *Emmanouil Benetos*[1],
Ning Xu[2], *SKoT McDonald*[2], *Mark Sandler*[1]

[1] Centre for Digital Music, Queen Mary University of London, UK [2] ROLI Ltd., UK

{v.subramanian, a.pankajakshan, emmanouil.benetos, mark.sandler}@qmul.ac.uk {ning,skot.mcdonald}@roli.com

ABSTRACT

An adversarial attack is an algorithm that perturbs the input of a machine learning model in an intelligent way in order to change the output of the model. An important property of adversarial attacks is transferability. According to this property, it is possible to generate adversarial perturbations on one model and apply it the input to fool the output of a different model. Our work focuses on studying the transferability of adversarial attacks in sound event classification. We are able to demonstrate differences in transferability properties from those observed in computer vision. We show that dataset normalization techniques such as z-score normalization does not affect the transferability of adversarial attacks and we show that techniques such as knowledge distillation do not increase the transferability of attacks.

Index Terms— Adversarial attacks, transferability, audio tagging, sound event classification

1. INTRODUCTION

Adversarial attacks are algorithms that add a small perturbation to the input of a machine learning model in order to change the output. The perturbed inputs are called *adversarial examples*. An interesting property of adversarial examples is that they can be transferred between machine learning models [1]. In computer vision, research shows that adversarial attacks can transfer between different model architectures, datasets and machine learning types [2, 3].

The *transferability* of adversarial attacks implies that a machine learning model is not safe even if the model architecture and model weights are kept hidden. Transferability also implies similarities in the underlying feature representation of different machine learning models, even if they have different model architectures and model types.

Adversarial attacks can be broadly classified into *perfect-knowledge attacks* and *limited-knowledge attacks* [3]. In a perfect-knowledge scenario, the adversarial attack has complete knowledge about the classifier and training data–and can use this knowledge to directly fool the classifier by computing gradients or using some other optimization algorithm if the classifier is not differentiable. In a limited-knowledge scenario, that adversarial attack has access to some information, such as the training data or model architecture, but may not have access to the model weights; therefore, the attack has to try and imitate the target classifier using another machine learning model. The model we want to attack is called the *target model* and the model we generate adversarial attacks on to transfer to the target model is called the *substitute model*.

In computer vision, generating adversarial examples in the limited-knowledge setting is proven to be possible [3, 2]. In these experiments, it is assumed that there is no knowledge of the training data or model architecture and that only a limited amount of queries are possible on the target model. Despite all these restrictions, they are able to generate adversarial attacks consistently and reliably. In sound event classification, Esmailpour et al. [4] carried out studies on transferability of adversarial attacks between Support Vector Machines (SVM) and deep learning models. Their research does not delve into transferability between deep learning models. In the perfect-knowledge scenario, there is a growing body of work in automatic speech recognition that focuses on generating adversarial attacks unique for speech recognition [5, 6, 7, 8]. In sound event classification and music, there is research that demonstrates the existence of perfect-knowledge attacks as well [9, 10]. Given the lack of research in sound event classification on adversarial attacks, it is important to gain an understanding of the properties of adversarial attacks using the domain knowledge of the field. Yang et al. [11] explain how the temporal aspects of automatic speech recognition and differences in the setup of speech recognition systems change some of the properties of adversarial attacks. Since very little work exists on transferability in sound event classification, we perform transferability experiments across deep learning models where the attack has knowledge of the dataset but not knowledge of the model architecture. The main contributions of our work include:

1. Observing the effects of input transformations on the transferability of adversarial attacks.
2. Investigating whether *knowledge distillation* [12] can be used to learn the weaknesses of the teacher model.
3. Investigating whether z-score normalization across the dataset during training affects transferability.
4. Identifying patterns in the audio files that are exhibiting the transferability property.

2. METHOD

2.1. Datasets

The dataset for all the experiments in this work is the FSDKaggle2018 dataset [13] that was released for task 2 of the DCASE 2018 challenge on "General-purpose audio tagging of Freesound content with AudioSet labels". The dataset has 41 labels and it is a single label classification task. The labels range from musical instruments such as snare drum and clarinet to urban sounds such as bus and gunshots. There are roughly 9K audio files in the training data and 1.6K

This work has received funding from the EU's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 765068. EB is supported by RAEng Research Fellowship RF/128.

Model	Training	Test
VGG13	0.9714	0.8093
CRNN	0.9768	0.8437
GCNN	0.9803	0.8437
dense_mel	0.9876	0.89875
dense_wav	0.9698	0.86125

Table 1. Model accuracy on training and test data.

audio files in the test data. The test data is used in our experiments to generate adversarial attacks.

2.2. Deep learning models

The deep learning models used in this work are from groups that participated in task 2 of the DCASE 2018 challenge [13]. From Iqbal et al. [14], we use the VGG13, CRNN and GCNN architectures. The VGG13 model is a convolutional model with a fully connected layer at the end. The CRNN model is a convolutional recurrent neural network that replaces the fully connected layer at the end of the VGG13 model with a bi-directional recurrent layer. The GCNN model is a gated convolutional neural network that replaces the normal convolutions in the VGG13 with gated convolutions. All models have a log mel-spectrogram input with a window size of 1024, hop size of 512, and 64 mel bands between 0-16kHz. The log mel-spectrogram of each input is normalized by the absolute maximum value of that particular input's log mel-spectrogram and is z-score normalized across the training data.

From Jeong and Lim [15], we use the densenet models. The densenet architecture uses dense layers where the input is concatenated with the output to improve the gradient flow. There are two versions of the densenet model, the log mel-spectrogram version (dense_mel) and the raw audio version (dense_wav). The log mel-spectrogram version has a window size of 1024, a hop size of 128 and 64 mel bands between 0-16kHz. Batch normalization is applied to the raw audio before it is passed to the log mel-spectrogram layer. All the models use audio sampled at 32kHz. All models use mixup augmentation [16] while training. Model performance on the FSD-Kaggle2018 training and test data is shown in Table 1.

2.3. Adversarial attack

We use the Carlini and Wagner attack [17] which uses the $L2$ as a regularizer to reduce the perturbation added to the input. The goal of an adversarial attack can be to generate a *targeted attack* or an *untargeted attack*. A targeted attack is where the algorithm is trying to fool the classifier into a specific target label. An untargeted attack is where the algorithm is trying to minimize the output probability of the current label until the output prediction changes. The untargeted attack does not care about what label the classifier is fooled to. We formalize the Carlini and Wagner attack as follows. Assume the input to classifier C is x, δ is the input perturbation, f is some function and c is a constant:

$$\begin{aligned} &\text{Minimize } L2(x, x+\delta) + c \cdot f(x+\delta) \\ &\qquad\qquad \text{such that } x+\delta \in [0,1]^n \end{aligned}$$

The purpose of the function f is to simplify the the optimization problem [17, 1]. In our work f is defined as:

$$\begin{aligned} C(x+\delta) = t &\text{ is true if } f(x+\delta) \leq 0 \\ f(x') = (max(Z(x')_i) &- Z(x')_t)^+ \; i \neq t \end{aligned}$$

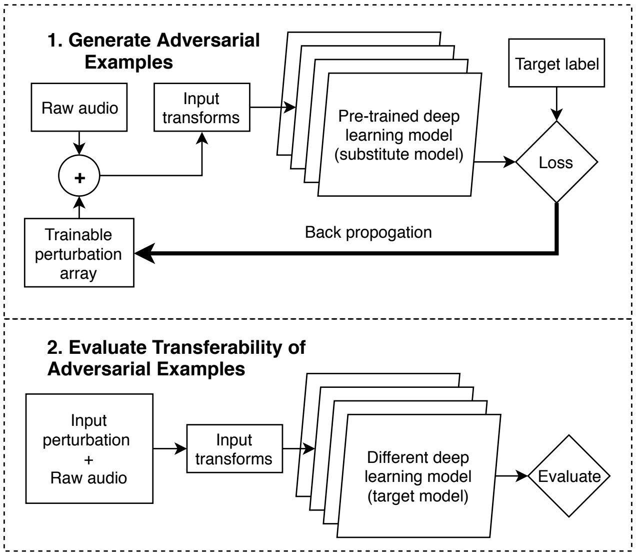
1. Generate Adversarial Examples
Raw audio
Input transforms
Pre-trained deep learning model (substitute model)
Target label
Loss
+
Trainable perturbation array
Back propogation
2. Evaluate Transferability of Adversarial Examples
Input perturbation + Raw audio
Input transforms
Different deep learning model (target model)
Evaluate

Fig. 1. Block diagram for all the experimental setups.

Z denotes the logits layer of the classifier, t is the target class and $x' = x + \delta$. We only use the targeted version of the Carlini and Wagner attack. The target label for each audio file is selected randomly ensuring that the target label is different from the ground truth.

There is a trade-off between generating adversarial attacks quickly and generating the best adversarial attack for an audio file by hyperparameter tuning. We are more interested in generating adversarial attacks quickly. In order to speed up the generation of adversarial attacks we stop optimization once the signal-to-noise ratio (SNR) becomes lower than 20dB, once the output probability of the target model is higher than 0.95 or once the number of iterations exceeds 5000.

3. EXPERIMENTS AND RESULTS

The general experimental setup is shown in Fig. 1. There is a model on which we generate adversarial examples; we call this model the *substitute model*. Then we observe how the generated adversarial examples transfer to the other models. The other models are called the *target models*. In each experiment, we alter the substitute model and try to observe the changes in how adversarial examples are transferred to the target models. The target models are from the list of CRNN, GCNN, dense_mel and dense_wav.

The audio files of interest for these experiments are the intersection of audio files on which adversarial attacks were successfully generated on the substitute model and the audio files that were correctly classified by the target model. The **correct%** refers to the percentage audio files in the test data that are audio files of interest. The **transfer%** refers to the percentage of adversarial attacks on the audio files of interest that fool the target model. The **target%** refers to the percentage of adversarial attacks on the audio files of interest that fool the target model such that the output label of the target model is the same as the substitute model for the adversarial attack.

3.1. Experiment 1

In the first experiment, the VGG13 model is the substitute model and the training configuration is the same as the original paper [14].

Target model	**Transfer%**	**Target%**	**Correct%**
CRNN	41.4	23.64	84.65
GCNN	43.9	30.6	84.72
dense_mel	8.9	0.2	90.26
dense_wav	4.5	0.2	83.90

Table 2. Experiment 1: Baseline system with VGG13 substitute model

Target model	**Transfer%**	**Target%**	**Correct%**
CRNN	43.1	9.2	85.02
GCNN	37.5	8.5	84.88
dense_mel	12.2	0.5	90.32
dense_wav	6.1	0.2	83.75

Table 3. Experiment 2: Knowledge distillation using dense_mel output to train VGG13

We generate adversarial attacks on the VGG13 model and evaluate the attacks on the CRNN, GCNN, dense_mel and dense_wav models. This experiment serves as baseline towards observing the percentage of adversarial attack transfer. We are interested in observing whether different architectures affect transferability. We were able to generate 1479 adversarial attacks out of the 1600 test audio files on the substitute model. The results of the transferability of attacks can be seen in Table 2.

Transferability is high between the VGG13 model and the CRNN and GCNN models at ∼40%, but it fails for both configurations of the dense models. Between the dense models, the transferability is better to the dense_mel model, which also has a log mel-spectrogram input. This suggests that the adversarial attacks are less likely to transfer if the input representation is different. It is interesting that the adversarial examples transfer successfully between the VGG13 and CRNN model because the adversarial attack has no knowledge of recurrent layers or their gradients. The target% shows similar trends to the transfer%. The numbers are predictably lower for target% due to the extra condition as described at the start of section 3.

3.2. Experiment 2

Experiment 2 modifies the VGG13 substitute model to try and improve the transferability of adversarial attacks on the target models. By using *knowledge distillation* [12] we aim to impart more knowledge of the target models to the substitute model. In knowledge distillation, one model is trained on the soft labels of the other model. In this experiment, we train two VGG13 models–one using the soft labels of the dense_mel and one using the soft labels of the CRNN. We do not use mixup augmentation [16] while training.

The VGG13 model trained on the dense_mel labels successfully generated adversarial attacks on 1416 out of the 1600 test audio files and the VGG13 model trained on the CRNN labels successfully generated adversarial attacks on 1414 out of the 1600 test audio files. Results are shown in Tables 3 and 4.

Target model	**Transfer%**	**Target%**	**Correct%**
CRNN	41.4	9.2	84.72
GCNN	36.7	9.5	84.86

Table 4. Experiment 2: Knowledge distillation using CRNN output to train VGG13

Target model	**Transfer%**	**Target%**	**Correct%**
CRNN	38.9	21.5	84.72
GCNN	41.9	30.4	84.79

Table 5. Experiment 3 with VGG13 as substitute model trained on no z-score normalization.

There is a slight improvement in the transfer% to the dense_mel when the VGG13 model is trained with the soft labels of the dense_mel but a similar improvement did not happen for transferability to the CRNN when training the VGG13 with the CRNN labels. On contrary, using knowledge distillation with the CRNN labels decreased the target% on the CRNN and GCNN models by a large amount. We take a closer look at these results in the discussion section to try and explain the observations with more clarity.

3.3. Experiment 3

Normalizing data across the training set is a common machine learning technique. The VGG13, CRNN and GCNN models use z-score normalization while training. We change the VGG13 substitute model by training it without z-score normalization. The attack was successful on 1480 of the 1600 test audio files. The results are shown in Table 5. Both the transfer% and the target% are more or less unchanged from when the substitute model was trained with z-score normalization. This suggests that the features learned by the deep learning models are similar irrespective of if they are trained with z-score normalization.

4. DISCUSSION

The experiments and results give us a broad overview of the transferability of adversarial attacks. In this section, we look for patterns in the type of audio files that are transferred.

4.1. SNR and output probability

Signal-to-noise ratio (SNR) is computed using the original audio file from the test dataset as the signal and the perturbation added by the adversarial attack as the noise. If the SNR is lower, it means more perturbation was added to generate the adversarial attack. Intuitively, the more noisy a signal is, the more likely the adversarial attack will succeed. Similarly, if the targeted model is highly confident (probable) in its prediction on the original audio file, it might reduce the chance of an adversarial attack changing its prediction. Table 7 plots the data of the output probabilities on the original audio files of interest and the SNR of the adversarial attacks on the audio files of interest.

From the results, there is a clear positive correlation between the output probability of the substitute and target model with the successful transferability of the adversarial attack. While there is a small difference in the average SNR between the successfully transferred and failed adversarial attacks, the standard deviation for both is around 6dB. The larger standard deviation suggests that the small difference in SNR does not mean much.

4.2. Label-wise performance

So far, we have looked at the transferability properties without focusing on specific labels. It is possible that some labels transfer at a much higher rate than other labels. Ideally, we expect that transferability should be independent of labels and that the transferability

Target	**First(transfer%)**	**Second(transfer%)**	**Third(transfer%)**
Substitute model: original vgg13 model			
CRNN	Fireworks(88)	Double_bass(70)	Squeak(70)
GCNN	Drawer(85)	Electric_piano(80)	Gong(76)
dense_mel	Electric_piano(40.62)	Flute(33.96)	Clarinet(25.92)
dense_wav	Cello(20.00)	Hi-hat(17.64)	Gong(13.33)
Substitute model: vgg13 trained with densenet mel labels			
dense_mel	Flute(56.60)	Electric_piano(43.75)	Hi-hat(32.26)
CRNN	Fireworks(94.74)	Trumpet(90.62)	Flute(72.22)
GCNN	Flute(80.39)	Electric_piano(73.33)	Fireworks(72.22)
Substitute model: vgg13 trained with crnn labels			
CRNN	Fireworks(100)	Trumpet(90.62)	Flute(84.31)
GCNN	Fireworks(92.30)	Cello(75.55)	Flute(72.55)

Table 6. Top-3 labels' transferability under different substitute models.

	VGG13 prob.		**Target prob.**		**SNR(dB)**	
target	s	f	s	f	s	f
CRNN	0.50	0.65	0.59	0.74	28.27	29.72
GCNN	0.50	0.66	0.52	0.70	28.62	29.30
dense_mel	0.48	0.58	0.79	0.92	28.84	29.06
dense_wav	0.49	0.58	0.57	0.90	26.55	28.94

Table 7. Comparing the average SNR and output probability of the substitute and target model on adversarial attacks that transferred successfully (s) and failed to transfer (f).

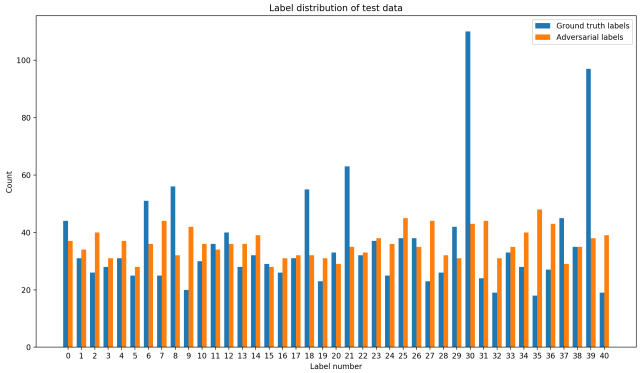

Fig. 2. Label distribution of the ground truth labels and the substitute model's adversarial labels.

per label should be close to the overall rate of transferability. In order to understand this better, we analyze the relationship of the ground truth labels from the test dataset and the adversarial target labels of the substitute model with transferability. Figure 2 shows the label distribution of the ground truth labels and the adversarial target labels of the substitute model.

The test dataset of FSDKaggle2018 does not have a uniform distribution between classes. Label numbers 30 and 39 have over 80 audio files when the average number per label is around 30. The target labels have a more uniform distribution because they are selected randomly. Table 6 shows the top 3 labels' transferability between the substitute model and the target model.

We can see that certain labels perform much better than the average performance such as Fireworks, Electric_piano, Flute etc. It is interesting that there seems to be a specific set of labels that is more transferable. The CRNN is consistently fooled by Fireworks, the GCNN by the Electric_piano etc.

Table 6 sheds more clarity on what knowledge distillation is doing. The broader results from Experiment 2 implies that knowledge distillation does not succeed in imparting meaningful knowledge to the substitute model. However, we can now see that knowledge distillation improves the transferability on certain labels. For example the VGG13 was already fooling Electric_piano and Flute the most against the dense_mel model, but training with the dense_mel labels improved the ability to fool these labels from 40.62% to 43.75% and 33.96% to 56.60% respectively. In fact, this ability to fool those particular labels more spills over to fooling the CRNN and GCNN. With this VGG13 model, Flute is suddenly much more transferable for the CRNN and GCNN models. Similarly, training with the CRNN labels improves the Fireworks and Flute performance on the CRNN and make Fireworks the most transferable for the GCNN model. This suggests that there are label specific features that are being exploited to generate adversarial attacks.

5. CONCLUSION

The results suggest that the transferability of adversarial attacks is affected by the input transformation. When the input transformation is identical, it is similar to the transferability experiments in computer vision and there is transferability between models. When the hyperparameters of the input transform are different, as in the case of the dense_mel, or the input representation is completely different, as in the case of the dense_wav, the transferability is almost nonexistent. However, z-score normalization of the input does not seem to affect the transferability of adversarial attacks. Using knowledge distillation to learn the features of another deep learning model does not affect transferability as well, however a closer analysis of the improvement in transferability at the label level suggests that there is some part of the features that is being learned.

This paper scratches the surface of adversarial attacks in sound event classification. There is obvious room for improvement to increase both the transfer% and target%. It is important to improve the transferability of adversarial attacks so that we can understand the security threats that a sound event classification are going to face when deployed in the real world. From an interpretability perspective, future work will focus on disentangling the complicated relationship between model architecture, input representations and training data.

6. REFERENCES

[1] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations (ICLR)*, Canada, 2014.

[2] Nicolas Papernot, Patrick D. McDaniel, and Ian J. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv:1605.07277*, 2016.

[3] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli, "Evasion attacks against machine learning at test time," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Berlin, Heidelberg, 2013, pp. 387–402, Springer.

[4] M. Esmaeilpour, P. Cardinal, and A. Lameiras Koerich, "A robust approach for securing audio classification against adversarial attacks," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2147–2159, 2020.

[5] Nicholas Carlini and David Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *IEEE Security and Privacy Workshops (SPW)*, USA, 2018, pp. 1–7.

[6] Yao Qin, Nicholas Carlini, Garrison Cottrell, Ian Goodfellow, and Colin Raffel, "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition," in *International Conference on Machine Learning (ICML)*, USA, 2019, pp. 5231–5240.

[7] Lea Schonherr, Katharina Kohls, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa, "Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding," *Network and Distributed System Security Symposium (NDSS)*, 2019.

[8] Xuejing Yuan, Yuxuan Chen, Yue Zhao, Yunhui Long, Xiaokang Liu, Kai Chen, Shengzhi Zhang, Heqing Huang, XiaoFeng Wang, and Carl A Gunter, "Commandersong: A systematic approach for practical adversarial voice recognition," in *USENIX Security Symposium*, USA, 2018, pp. 49–64.

[9] Vinod Subramanian, Emmanouil Benetos, and Mark Sandler, "Robustness of adversarial attacks in sound event classification," in *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, USA, 2019, pp. 239–243.

[10] Corey Kereliuk, Bob L Sturm, and Jan Larsen, "Deep learning and music adversaries," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 2059–2071, 2015.

[11] Zhuolin Yang, Bo Li, Pin-Yu Chen, and Dawn Song, "Characterizing audio adversarial examples using temporal dependency," in *International Conference on Learning Representations (ICLR)*, USA, 2019, OpenReview.net.

[12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," *arXiv:1503.02531*, 2015.

[13] Eduardo Fonseca, Manoj Plakal, Frederic Font, Daniel P. W. Ellis, Xavier Favory, Jordi Pons, and Xavier Serra, "General-purpose tagging of freesound audio with audioset labels: Task description, dataset, and baseline," in *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, UK, 2018, pp. 69–73.

[14] Turab Iqbal, Qiuqiang Kong, Mark D Plumbley, and Wenwu Wang, "General-purpose audio tagging from noisy labels using convolutional neural networks," in *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, UK, 2018, pp. 212–216.

[15] Il-Young Jeong and Hyungui Lim, "Audio tagging system for dcase 2018: focusing on label noise, data augmentation and its efficient learning," Tech. Rep., Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE), UK, 2018.

[16] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz, "mixup: Beyond empirical risk minimization," in *arXiv:1710.09412*, 2017.

[17] Nicholas Carlini and David Wagner, "Towards evaluating the robustness of neural networks," in *IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 39–57.