Responsible Machine learning - Final technical assignment

Abdellah Elmrini Jérémie Dentan Nathanaël Cuvelle–Magar

6 février 2023

Introduction

Aujourd'hui à l'état de l'art dans de nombreux domaines, les algorithmes de *deep learning* (DL) ont révolutionné notre façon d'aborder de nombreuses tâches, en particulier en *computer vision* [7]. Néanmoins, ces excellentes performances sont contrebalancées par certaines fragilités, au premier rang desquelles la vulnérabilité aux attaques adversariales. Mises en évidence dans de nombreux articles [9, 5], elles exploitent généralement le fait que les raisons poussant un classifieur à prédire un certain résultat ne sont pas celles perçues par les analystes humains [1]. Une modification imperceptible d'une image peut ainsi conduire à une classification complètement erronée, comme illustré par Goodfellow et al. [5] (FGSM, pour *Fast Gradient Sign Attack*). Cette vulnérabilité est ainsi un problème majeur pour quiconque souhaite utiliser un *deep neural network* (DNN) dans un système sécurisé. Malgré tout, des attaques comme FGSM nécessitent de connaître l'architecture et les poids du réseau que l'on souhaite attaquer, et imposent donc une configuration *white box*. L'existence d'attaques *black box*, ne nécessitant pas de connaître le modèle cible et ses poids, représente donc un enjeu de sécurité important, qui a conduit au développement d'un champ de recherche dédié à la transférabilité des attaques [11]. Du son [16, 17] au texte [18] en passant par la *computer vision* [13], l'objectif est de concevoir des exemples adversariaux à partir d'un modèle de génération pour tromper un modèle cible. Demontis et al. [3] mettent en évidence plusieurs critères influençant les performances de ces attaques, dont la simplicité du modèle générateur. S'appuyant sur le fait que les DNNs entraînés sur de grands jeux de données apprennent des représentations internes génériques transférables à de nouvelles tâches et à de nouveaux jeux de données, Naseer et al. [12] proposent d'utiliser une métrique perceptuelle fondée sur les représentations internes des réseaux VGG pour concevoir des attaques transférables. Les auteurs utilisent ainsi implicitement la *Natural Abstraction Hypothesis*, selon laquelle "une grande variété d'architectures cognitives vont apprendre à utiliser approximativement les mêmes objets/concepts abstraits de haut niveau pour raisonner sur le monde" [2].

Notre but dans ce projet est d'implémenter l'attaque proposée par Naseer et al. [12] et de la confronter aux résultats généraux sur la transférabilité des attaques proposées par Demontis et al. [3]. En s'appuyant sur les résultats de ce dernier papier, nous chercherons également, dans un second temps, à établir si un processus de génération plus robuste pour les exemples adversariaux peut permettre d'améliorer leur transférabilité au-delà de la complexité du modèle de génération choisi.

Ce rapport est accompagné d'un répertoire Github qui en implémente les principales expérimentations, disponible à l'addresse suivante : `https://github.com/DentanJeremie/adversarialTransferts`.

1 Transférabilité via des métriques perceptuelles

Dans cette partie, on cherche à évaluer la transférabilité de l'attaque NRDM, pour *Neural Representation Distortion Method*, proposée par Naseer et al. [12] et à vérifier l'applicabilité des résultats de Demontis et al. [3] dans ce cadre.

1. Dans ce projet, nous nous concentrerons sur les *evasion attacks* et n'aborderons pas la question des *poisoning attacks*.

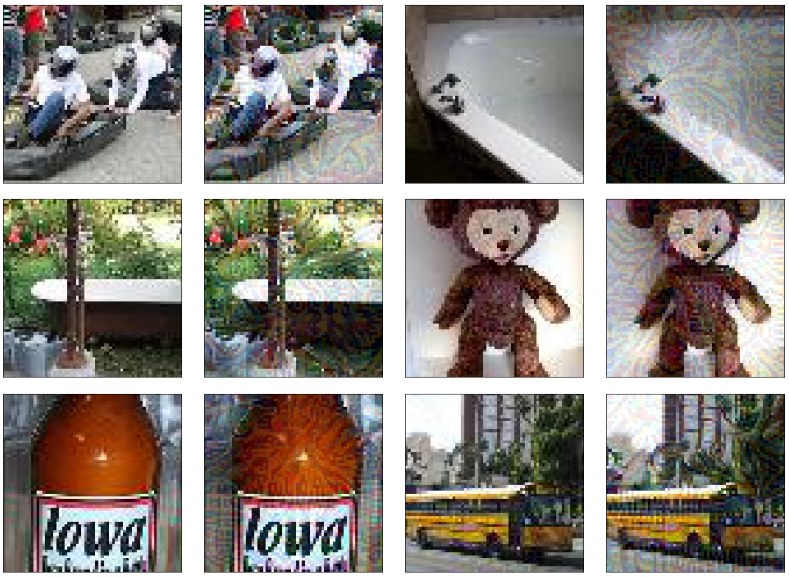

FIGURE 1 – Exemples d'images adversariales perceptuelles.

1.1 Rappels sur la transférabilité des attaques

Comme évoqué en introduction, Naseer et al. [12] utilisent une métrique perceptuelle pour créer des attaques dont la transférabilité repose sur la générécité des représentations apprises par les *neural networks* (NNs) lorsqu'ils sont entraînés sur de grands jeux de données, comme ImageNet [14]. Plus précisément, les auteurs proposent de concevoir les exemples adversariaux de la façon suivante :

$$\underset{x'}{\operatorname{argmax}} \left\{ \mathcal{F}(x')|_k - \mathcal{F}(x)|_k \quad / \quad ||x - x'||_\infty \leq \epsilon \right\},$$

où $\mathcal{F}$ est un NN générateur, k est l'indice d'une couche, $\mathcal{F}(x)|_k$ désigne l'évaluation de $\mathcal{F}$ tronqué à la couche k en x, et ϵ est le budget de perturbation autorisé. L'idée est ainsi de chercher l'image d'entrée dont la représentation interne de niveau k diffère le plus de celle de l'image d'origine, à budget de perturbation fixé. Une description détaillée de l'algorithme utilisé est donnée dans l'article original. Nous en proposons une implémentation disponible sur GitHub [2], utilisant VGG16 [15] comme modèle générateur (pré-entraîné sur ImageNet). La figure 1 illustre les résultats obtenus sur le jeu de données Tiny ImageNet [10], avec 5 étapes de descente de gradient, $\epsilon = 16/256$ et k correspondant à la couche "conv33".

De leur côté, Demontis et al. [3] étudient les facteurs favorisant la transférabilité des attaques. Ils en identifient trois principaux :

- la taille des gradients d'entrée, qui est liée à la complexité du modèle cible, puisque les classifieurs cibles complexes tendent à présenter des gradients plus importants ;
- l'alignement des gradients des modèles générateurs et cibles, qui est, pour les *evasion attacks*, lié à la complexité du générateur, puisque des générateurs de faibles complexités donnent des gradients plus stables et mieux alignés, en moyenne, avec ceux de la cible ;
- la variabilité du paysage de la fonction de perte.

1.2 Evaluation des attaques perceptuelles

Les résultats de Demontis et al. [3] ayant été obtenus pour une tâche de classification binaire et pour des algorithmes d'apprentissage variés, i.e. pas uniquement des NNs (SVM, logistic regression, etc.) [3], nous souhaitons voir s'ils restent valables dans le contexte de l'attaque proposée par Naseer

2. https://github.com/DentanJeremie/adversarialTransferts

3. Les auteurs ne distinguent pas réellement entre les architectures. Notre objectif est donc de voir si leurs résultats sont toujours valables si l'on compare des algorithmes de type NN entre eux.

et al. [12]. Plus particulièrement, nous souhaitons voir si les facteurs proposés peuvent être reliés au choix de la couche d'index k dans NRDM, cette dernière contrôlant en un sens la complexité du modèle de génération. Nous chercherons également à voir si l'efficacité de cette attaque dépend du nombre de paramètre du NN cible, utilisé comme un proxy de sa complexité. Enfin, nous évaluerons l'impact du nombre d'étapes de descente de gradients réalisées sur l'efficacité des exemples adversariaux produits. Les résultats obtenus sont donnés dans les tableaux 1 (target VGG16), 2 (target DenseNet201 [8]) et 3 (target ResNet50 [6]). Dans toutes ces expériences, ϵ est fixé à 16/256.

TABLE 1 – VGG *accuracy* sur les données corrompues (performances initiales 0.4042)

Attaque	3 étapes	5 étapes	7 étapes	10 étapes
vgg_conv22	0.1376	0.1060	0.0964	0.0937
vgg_conv33	0.0704	0.0475	0.0420	0.0412
vgg_conv43	0.0444	0.0341	0.0296	0.0288

TABLE 2 – DenseNet *accuracy* sur les données corrompues (performances initiales 0.6489)

Attaque	3 étapes	5 étapes	7 étapes	10 étapes
vgg_conv22	0.2558	0.2518	0.2458	0.2398
vgg_conv33	0.2347	0.2107	0.2047	0.1964
vgg_conv43	0.2297	0.2086	0.2090	0.2037

TABLE 3 – ResNet *accuracy* sur les données corrompues (performances initiales 0.7354)

Attaque	3 étapes	5 étapes	7 étapes	10 étapes
vgg_conv22	0.2768	0.2565	0.2447	0.2378
vgg_conv33	0.2548	0.2218	0.2107	0.2105
vgg_conv43	0.2363	0.2184	0.2170	0.2106

On fait les observations suivantes :

- L'augmentation du nombre d'itération de la descente de gradient améliore presque toujours les performances de l'attaque. Néanmoins, on observe que son impact finit par se tasser.
- L'impact de la complexité du modèle générateur varie suivant le nombre d'étapes de *gradient descent* (GD) réalisées et il faut donc distinguer les cas. Lorsque ce nombre est faible $(3, 5)$, il apparaît qu'une attaque fondée sur une couche plus élevée fonctionne mieux quel que soit le modèle cible. On peut l'interpréter en considérant que, si l'on tronque le réseau trop bas, les perturbations obtenues dépendront trop du réseau en question. Contrairement à ce qui se passe quand on tronque à une couche supérieure, ce ne seront pas des "*high level features*". Ce constat semble aller à l'encontre du principe de simplicité pour le modèle générateur. Néanmoins, on observe que pour un nombre d'étapes de GD plus important $(7, 10)$, cette dynamique n'est plus celle obtenue pour les modèles DenseNet201 et ResNet50. En effet, on observe alors un optimum pour une attaque basée sur la couche de profondeur intermédiaire "vgg_conv33", indiquant peut être un compromis entre "*high level features*" et faible complexité du générateur. Le cas où VGG16 est le modèle cible correspond quant à lui à un cas particulier, puisqu'augmenter l'index k conduit à rapprocher le modèle générateur du modèle cible. Il semble donc logique de ne pas observer le même comportement dans cette configuration.
- L'impact de la complexité du modèle cible semble être confirmé. En effet, les modèles VGG16, DenseNet201 et ResNet50 présentent, respectivement, 138357544, 20013928 et 25557032 paramètres, et on vérifie que les performances de l'attaque NRDM sont d'autant plus importantes que le modèle a de paramètres. Pour quantifier ces performances, on considère le ratio entre les performances originales du modèle et celles obtenues pour la meilleure configuration d'attaque. Pour DenseNet201 et ResNet50, cela revient à considérer 10 d'étapes de GD et la couche "vgg_conv33" pour la génération. Pour VGG16, on considère la même configuration afin de permettre une comparaison équitable. En effet, comme évoqué plus haut, les meilleures performances obtenues pour des couches plus élevées sont à relier au fait que VGG est le modèle

générateur. On obtient les résultats suivants : 3.30 pour DenseNet201, 3.49 pour ResNet50 et 9.81 pour VGG16.

Par conséquent, nos résultats semblent aller dans le sens de ceux obtenus par Demontis et al. [3]. Il pourrait notamment être intéressant de vérifier le rapport entre nombre de paramètres du modèle cible et la performance maximale de l'attaque NRDM qui semble, pour les modèles considérés ici, linéaire. Nous laissons cette vérification à de prochains travaux.

2 Conception robuste des exemples adversariaux

La partie précédente a permis de confirmer, dans une certaine mesure, les résultats de Demontis et al. [3] quant à l'importance de la simplicité du modèle de génération pour les exemples adversariaux. On note néanmoins que la méthode utilisée pour les construire repose sur une descente gradient classique, correspondant à une métrique L^2. Or, des travaux réalisés dans le domaine de la visualisation des *features* montrent qu'un tel processus d'optimisation peut se révéler sensible aux hautes fréquences, tandis qu'une paramétrisation Fourier/décorrélée permet de réduire leur impact [1]. Par suite, nous proposons dans cette partie d'implémenter l'attaque NRDM avec une telle paramétrisation en s'appuyant sur la librairie PyTorch Lucent [4]. L'idée est de réduire la complexité des exemples adversariaux, d'une façon complémentaire à celle du modèle de génération.

On commence par illustrer la différence entre les perturbations obtenues suivant la paramétrisation adoptée. On considère les images d'entrée présentées en figure 2.

FIGURE 2 – Images d'entrée utilisées pour illustrer l'impact du choix de paramétrisation.

On considère deux configurations :

— Pour la première image, représentant le Big Ben, dont l'état d'origine est présenté sur la figure 2 (gauche) et les perturbations sur la figure 3, on utilise le modèle Inceptionv1 et on maximise la *Mean Squared Error* entre les *features* obtenus en sortie de la couche "mixed3b" pour l'image d'origine et l'image paramétrée. Cette dernière est initialisée à la valeur de l'image d'origine, sans bruit additionnel. 10 étapes de GD sont calculées (l'image perturbée obtenue n'est pas clippée, l'objectif étant ici purement illustratif). Les résultats pour les paramétrisations classiques (gauche) et Fourier/décorrélées (droite) sont illustrés en figure 3.

— Pour la seconde image, représentant un chien, dont l'état d'origine est présenté sur la figure 2 (droite) et les perturbations sur la figure 4, on utilise le modèle VGG16 et on maximise la *Mean Squared Error* entre les *features* obtenus en sortie de la couche "features_30" (dernière couche avant le classifieur) pour l'image d'origine et l'image paramétrée. Cette dernière est initialisée à la valeur de l'image d'origine, avec un bruit additionnel gaussien d'amplitude $\epsilon = 16/256$. 1 (gauche), 10 (milieu) et 100 (droite) étapes de GD sont calculées (l'image perturbée obtenue

4. https://github.com/greentfrapp/lucent

n'est pas clippée, l'objectif étant ici purement illustratif). Les résultats pour les paramétrisations classiques (haut) et Fourier/décorrélées (bas) sont illustrés en figure 4.

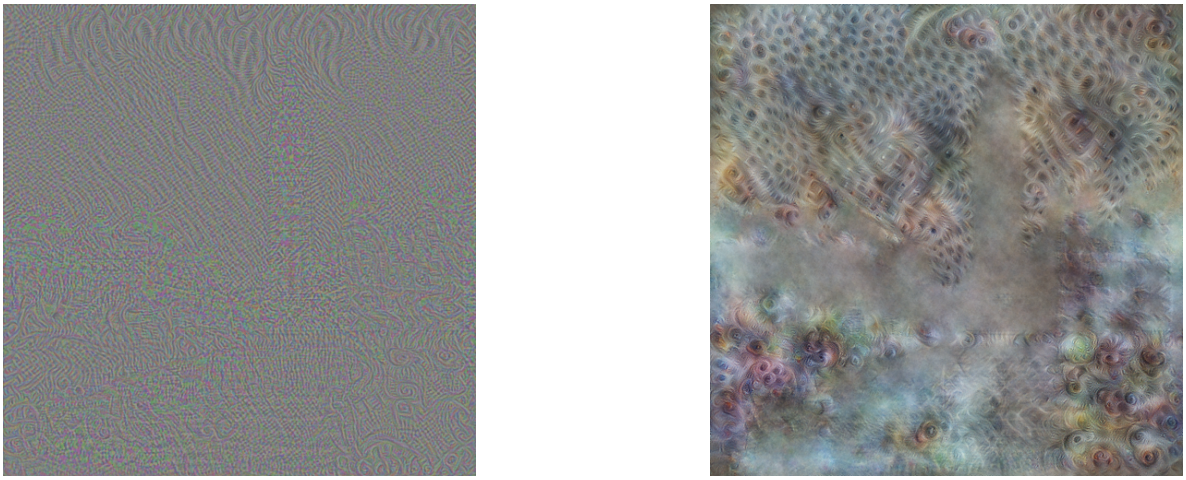

FIGURE 3 – Perturbation (i.e. image perturbée moins image d'origine) pour la figure 2 (gauche).

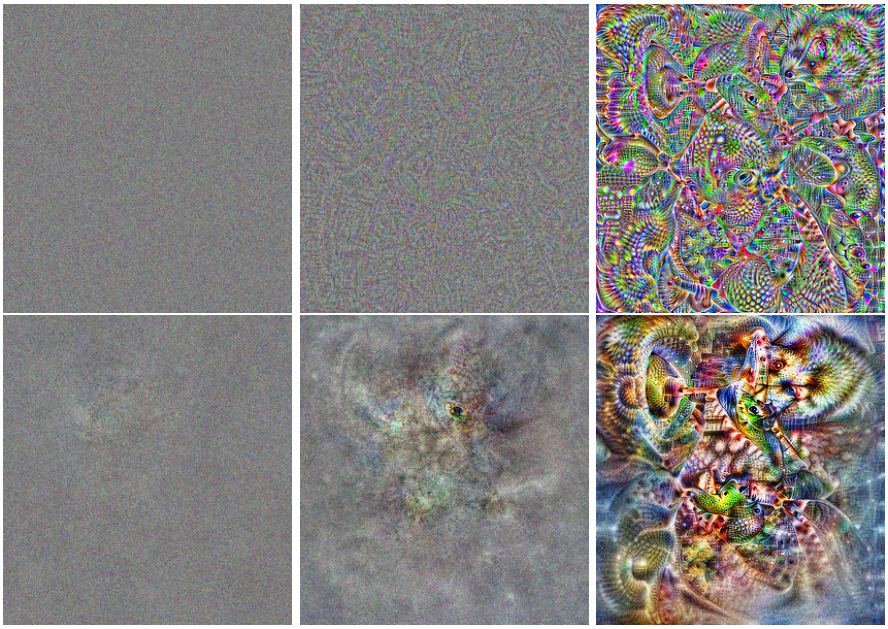

FIGURE 4 – Perturbation (i.e. image perturbée moins image d'origine) pour la figure 2 (droite).

En accord avec les résultats présentés dans [1], on observe qu'une optimisation dans l'espace Fourier/décorrélé semble, d'une part, mieux retranscrire la structure de l'image et, d'autre part, réduire l'impact des hautes fréquences.

De la même façon qu'en partie 1, on compare les performances obtenues avec ces deux versions de NRDM. Les résultats sont les suivants, pour DenseNet201 (resp. ResNet50) en modèle cible : 0.5782 (resp. 0.6230) pour une paramétrisation classique et 0.5846 (resp. 0.6549) pour une paramétrisation Fourier/décorrélée [5]. On n'observe donc pas d'amélioration significative des performances de l'attaque lorsque la seconde est utilisée, ce qui semble suggérer qu'une optimisation limitant l'impact des hautes fréquences n'est pas directement reliée à la réduction de complexité préconisée par Demontis et al. [3].

5. Ces valeurs sont calculées pour un échantillon de 300 exemples adversariaux avec VGG16 comme modèle génératif, en sélectionnant la couche 5 du réseau et pour 100 étapes de GD. ϵ est toujours fixé à 16/256.

À titre de complément, les tableaux 4 et 5 présentent les résultats obtenus dans diverses configurations de l'attaque avec paramétrisation Fourier/décorrélée. On note que, dans ce cadre, les attaques généralisent d'autant mieux qu'elles ont été générées via des couches peu profondes.

TABLE 4 – DenseNet *accuracy* sur les données corrompues (performances initiales 0.66)

Attaque	100 étapes	250 étapes
vgg, couche 5	0.5846	0.4632
vgg, couche 7	0.6130	0.5367

TABLE 5 – ResNet *accuracy* sur les données corrompues (performances initiales 0.73)

Attaque	100 étapes	250 étapes
vgg, couche 5	0.6549	0.5623
vgg, couche 7	0.7093	0.5879

Conclusion

En résumé, il apparaît que les résultats de Demontis et al. [3] restent, dans une certaine mesure, valables dans le cadre de l'attaque NRDM et, plus généralement, dans celui où le générateur et la cible sont des NNs. L'analyse de la transférabilité des exemples adversariaux obtenus nous a notamment permis de mettre en évidence un compromis entre, d'une part, l'importance de considérer des *features* d'assez haut niveau, non spécifiques au modèle, et, d'autre part, la nécessité de limiter le nombre de paramètres (i.e. la complexité) du modèle générateur (i.e. l'indice k).
L'étude de ce critère de complexité du générateur nous a également permis de considérer une voie alternative pour sa réduction, pour un nombre de paramètres fixé, en modifiant la paramétrisation de l'image perturbée pour passer dans l'espace de Fourier/décorrélé. Bien que l'impact des hautes fréquences ait ainsi pu être contenu par rapport à une descente de gradient L^2 classique, notre étude ne nous a pas permis de conclure qu'une telle approche puisse mener à une réelle amélioration dans la transférabilité de l'attaque.

Dans la continuité de ce projet et de cet approfondissement du critère de simplicité du générateur, il serait intéressant de creuser plus avant la question de la complexité du modèle cible. Pour cela, il pourrait notamment être pertinent d'étudier ses liens avec la notion de superposition dans les réseaux de neurones et ses conséquences sur la vulnérabilité aux attaques adversariales [4]. En effet, dans les deux cas, il apparaît que la vulnérabilité aux attaques est liée à une forme de complexité du modèle, que ce soit sous la forme d'un nombre important de paramètres dans Demontis et al. [3], ou d'une forte superposition des *features* dans Elhage et al. [4]. Une expérience pour réaliser cette comparaison pourrait, par exemple, consister à construire un jeu de données d'images simples (ex. formes géométriques simples de couleurs variables) permettant d'identifier les *features* appris par le réseau (forme, couleur), de façon à pouvoir ensuite établir si l'optimisation du critère perceptuel de NRDM revient effectivement à utiliser la superposition pour tromper le modèle cible. Nous laissons ces perspectives à de prochains travaux.

Références

[1] Feature visualization. https://distill.pub/2017/feature-visualization/. Accessed : 2023-01-28.

[2] Natural abstraction. https://www.alignmentforum.org/tag/natural-abstraction#:~:text=The%20Natural%20Abstraction%20hypothesis%20says,dimensional%20than%20the%20system%20itself. Accessed : 2023-01-28.

[3] Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio, Alina Oprea, Cristina Nita-Rotaru, and Fabio Roli. Why do adversarial attacks transfer ? explaining transferability of evasion and poisoning attacks. In *USENIX Security Symposium*, 2018.

[4] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, T. J. Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Baker Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *ArXiv*, abs/2209.10652, 2022.

[5] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014.

[6] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015.

[7] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers : Surpassing human-level performance on imagenet classification. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

[8] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2016.

[9] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *ArXiv*, abs/1611.01236, 2016.

[10] Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge. 2015.

[11] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Xiaodong Song. Delving into transferable adversarial examples and black-box attacks. *ArXiv*, abs/1611.02770, 2016.

[12] Muzammal Naseer, Salman Hameed Khan, Shafin Rahman, and Fatih Murat Porikli. Task-generalizable adversarial attack based on perceptual metric. *arXiv : Computer Vision and Pattern Recognition*, 2018.

[13] Muzammal Naseer, Salman Hameed Khan, M. H. Khan, Fahad Shahbaz Khan, and Fatih Murat Porikli. Cross-domain transferability of adversarial perturbations. In *Neural Information Processing Systems*, 2019.

[14] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115 :211–252, 2014.

[15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[16] 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*, 2019.

[17] Vinod Subramanian, Arjun Pankajakshan, Emmanouil Benetos, Ning Xu, SKoT McDonald, and Mark Sandler. A study on the transferability of adversarial attacks in sound event classification. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 301–305, 2020.

[18] Liping Yuan, Xiaoqing Zheng, Yi Zhou, Cho-Jui Hsieh, and Kai-Wei Chang. On the transferability of adversarial attacks against neural text classifier. In *Conference on Empirical Methods in Natural Language Processing*, 2020.