Face-Off: Improving Imperceptibility of Adversarial Examples

Brian Tang
University of Wisconsin - Madison

1 INTRODUCTION

As face recognition becomes more prevalent in contexts such as social media, photo storage, and law enforcement, it becomes increasingly important to consider the privacy of users' data. Automated face recognition systems can exploit uploaded photos to associate users with locations and activities. Recent work allows privacy-conscious users to obfuscate their faces from face recognition without any loss of usability. These approaches rely on adversarial perturbations[3] or data poisoning[11] to improve the privacy utility trade-off as opposed to completely blurring faces. Doing so will induce errors in the face recognition model and result in misclassifications. However, current systems such as Face-Off[3] amplify adversarial perturbations to ensure transferability which negatively impacts usability. Image scaling attacks, a new discovery in image processing addresses this issue by taking advantage of resizing algorithms to generate a new image from the downscaling process[13]. By applying this step in the data postprocessing step of Face-Off, we can greatly decrease the perceptibility of the adversarial perturbations while increasing the strength of the attack[1].

2 BACKGROUND

2.1 Adversarial Machine Learning

In the traditional deep learning classification setting, adversarial examples are images with minor imperceptible perturbations which result in an incorrect classification output from a model[4, 12]. Adversarial examples are defined with the following threat model: Given white-box access to a model's parameters, weights, and architecture, and where $f(X) = Y$, find an X' close to X such that $f(X') \neq Y$. For the context of Face-Off, we assume the user (adversary) has black-box access to the model in which they cannot query the model. Querying the face recognition model would defeat the purpose of Face-Off and leak user data. Thus, Face-Off must rely on the transferability property of adversarial examples, where adversarial examples generated for one deep learning also result in misclassifications in other models[7, 8]. Face-Off uses adversarial attack algorithms such as Projected Gradient Descent (PGD)[6] and Carlini-Wagner (CW)[2] to apply an imperceptible layer of strategic noise to the original image. Adversarial examples can also be amplified[1, 5] to increase the likelihood the attack will transfer to other models as well as decrease matching confidence.

2.2 Face Recognition and Face-Off

Typically, face recognition determines matches between faces by detecting a face within a photo and matching it with another face (or bucket of faces). A distance metric (l_2 norm or cosine similarity) is used in tandem with a threshold to calculate the *closeness* of faces. Face-Off applies a layer of calculated adversarial perturbations onto an uploaded face which allows a user to mask their data from proprietary datasets and malicious third-parties[3]. Through these mismatches, the user's face is no longer able to be automatically recognized, thus giving privacy conscious users an option for usable face obfuscation. One of the main drawbacks of Face-Off is the privacy utility trade-off inherent from amplifying adversarial perturbations.

2.3 Image Scaling Attacks

Image scaling attacks take advantage of image scaling algorithms by injecting a camouflaged image within a larger resolution image so that downscaling the image using an algorithm such as bilinear interpolation results in a completely different image. An attack image $S' \sim S$ is created by a source image S and target image T such that the output image $D \simeq T$. This attack can be used to induce failures within deep learning models via data poisoning attacks[10] and can likewise be used to hide adversarial perturbations. Image scaling attacks can be detected and defended against[9], and can produce perceptible artifacts in low resolution images, especially if the target image is very different from the source image. Hiding adversarial examples as the target image may bypass existing defenses and reduce the visibility of any perturbations.

3 CONTRIBUTIONS

The following contributions will be made to this research space:

(1) Popular face recognition APIs and social media sites will be queried for their model input dimensions.
(2) Image scaling will be applied in the Face-Off pipeline to improve usability (smaller perturbations) and transferability.
(3) A study will evaluate the perceptility score of the perturbed faces.

We hypothesize these contributions will greatly improve both transferability and imperceptibility of the adversarial perturbations generated by Face-Off.

[1]Concurrent research is being done to evaluate the effectiveness of camouflaging powerful adversarial perturbations using image scaling

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

4 TIMELINE

(1) **October**: Reproduce code from relevant research papers.
(2) **November**: Add image scaling attack algorithm to postprocessing step of Face-Off.
(3) **November 10**: Query face recognition APIs (Microsoft Azure, AWS Rekognition, Face++, Google, Facebook) for information about input dimensionality, image scaling algorithm, etc.
(4) **November 20**: Generate adversarial examples using other face recogntion architectures.
(5) **December 5**: Perform online evaluation to ensure faces are successfully obfuscated.
(6) **December 10**: Evaluate perceptibility score (LPIPS metric) of adversarial examples vs. image scaling + adversarial examples on several face recognition datasets (Labeled Faces in the Wild, Famous Celebrities).
(7) **December 10**: Update both project websites with functional demos and information.

5 MID-TERM PROGRESS REPORT

5.1 Challenges

After additional literature review and initial experimentation, some challenges have arisen with the direction of the project. Image scaling attacks rely on the interpolation algorithm being performed on the exact same image, since the perturbations are required to be embedded within the image at exact pixel positions, so that the scaling algorithm is exploited. Several implementation issues specific to the face recognition domain develop as a result.

(1) Face detection strategies (MTCNN, dlib, SSD, OpenCV) may not universally detect and segment images the same, so replacing a face with an image scaling attack may not necessarily result in the desired output since pixel positions could be slightly offset.
(2) Face recognition APIs involve a face detector, face recognition architecture, specific input dimensions, and the use of a specific scaling algorithm to perform face alignment. These need to be known in advance to effectively apply the image scaling attack technique.
(3) Generating image scaling attacks is computationally expensive which could decrease the usability aspect of Face-Off.

These issues came up as I was running preliminary experiments with generating images and evaluating them on face recognition APIs. As a result, I need to perform an extensive online evaluation by uploading faces created by altering certain parameters in the Face-Off pipeline such as input size, face detection algorithm, and image scaling interpolation type. Populating Table 1 will address these issues and allow Face-Off to use image scaling in its obfuscation pipeline to reduce the visible perturbation artifacts present in its output images. I will generate a set of 3x5x4x5 = 300[2] images to upload to each API in order to infer these parameters. In addition to input dimension, interpolation, and detector information, I intend to generate adversarial examples using face recognition architectures other than FaceNet (such as FBDeepFace, VGGFace, OpenFace) to further improve the likelihood of a perturbed face being misclassified.

[2](OpenCV, TensorFlow, Pillow)x(Nearest, Linear, Cubic, Lanczos, Area)x(MTCNN, dlib, SSD, OpenCV)x(160x160, 152x152, 96x96, 224x224, 55x47)

API	Scaling	Input	Detector
Azure	Linear	100x100 to 164x164	MTCNN
AWS	Linear	120x120 to 164x164	MTCNN or OpenCV

Table 1: Online API configurations

5.2 Progress

I completed points (1) and (2) in the timeline. I was able to modify the image scaling code and add it to the Face-Off obfuscation pipeline. Figure 1 shows a comparison between the perceptibility of image scaling and adversarial perturbation. The perturbations on the adversarial attack are much larger, but the image scaling perturbations have an obvious grid-like effect on the image. To address this, the image scaling attack can be applied to the delta mask as seen in Figure 2 to produce a smaller overall adversarial perturbation which is the motivation of this work. The resulting adversarial image after downscaling will remain effective.

Now that the pipeline is setup, the bulk of my evaluations will be done in the next two months, as my schedule is starting to free up after school and job applications.

Figure 1: Left: Unperturbed image; Middle: Image scaling attack (output is a 112x96 Matt Damon); Right: Adversarially perturbed image

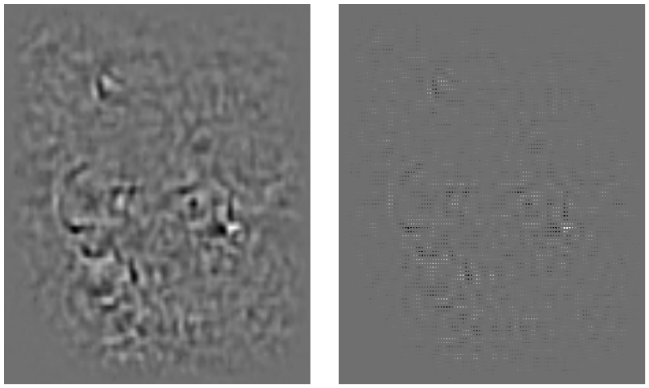

Figure 2: Left: Adversarial perturbation mask; Right: Perturbation mask + Image scaling attack

REFERENCES

[1] Xiaoyu Cao and Neil Zhenqiang Gong. Mitigating evasion attacks to deep neural networks via region-based classification. In *Proceedings of the 33rd Annual Computer Security Applications Conference. ACM* (2017).

[2] Nicholas Carlini and David Wagner. 2016. Towards Evaluating the Robustness of Neural Networks. *arXiv:1608.04644 [cs.CR]* (2016).

[3] Varun Chandrasekaran, Chuhan Gao, Brian Tang, Kassem Fawaz, and Somesh Jha. Face-Off: Adversarial Face Obfuscation. In *The 21st Privacy Enhancing Technologies Symposium* (2021).

[4] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and Harnessing Adversarial Examples. *arXiv:1412.6572 [stat.ML]* (2014).

[5] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2016. Delving into transferable adversarial examples and black-box attacks. *arXiv:1611.02770* (2016).

[6] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards Deep Learning Models Resistant to Adversarial Attacks. *arXiv:1706.06083 [stat.ML]* (2017).

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

[8] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2016. Practical black-box attacks against deep learning systems using adversarial examples. *arXiv:1602.02697* (2016).

[9] Erwin Quiring, David Klein, Daniel Arp, Martin Johns, and Konrad Rieck. Adversarial Preprocessing: Understanding and Preventing Image-Scaling Attacks in Machine Learning. In *Proceedings of the 29th USENIX Security Symposium* (2020).

[10] Erwin Quiring and Konrad Rieck. 2020. Backdooring and Poisoning Neural Networks with Image-Scaling Attacks. *arXiv:2003.08633* (2020).

[11] Shawn Shan, Emily Wenger, Jiayun Zhang, Huiying Li, Haitao Zheng, and Ben Y. Zhao. Fawkes: Protecting Privacy against Unauthorized Deep Learning Models. In *Proceedings of the 29th USENIX Security Symposium* (2020).

[12] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intruiging properties of neural networks. *arXiv:1312.6199 [cs.CV]* (2014).

[13] Qixue Xiao, Yufei Chen, Chao Shen, Yu Chen, and Kang Li. Seeing is Not Believing: Camouflage Attacks on Image Scaling Algorithms. In *Proceedings of the 28th USENIX Security Symposium* (2019).