2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN) | 979-8-3503-4793-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/DSN58367.2023.00038

ReFace: Adversarial Transformation Networks for Real-time Attacks on Face Recognition Systems

Shehzeen Hussain*, Todd Huster†, Chris Mesterharm†, Paarth Neekhara* and Farinaz Koushanfar*
*University of California San Diego
†Peraton Labs
Email: ssh028@eng.ucsd.edu, thuster@peratonlabs.com

***Abstract*—In this work, we propose ReFace, a real-time, highly-transferable attack on face recognition models based on Adversarial Transformation Networks (ATNs). Past attacks on face recognition models require the adversary to solve an input-dependent optimization problem using gradient descent making the attack impractical in real-time. Such adversarial examples are also tightly coupled to the victim model and are not as successful in transferring to different models. We find that the white-box attack success rate of a pure U-Net ATN falls substantially short of gradient-based attacks like PGD on large face recognition datasets. We therefore propose a new architecture for ATNs that closes this gap while maintaining a 10000X speedup over PGD. Furthermore, we find that at a given perturbation magnitude, our ATN adversarial perturbations are more effective in transferring to new face recognition models than PGD. We demonstrate that our attacks transfer effectively to models with different architectures, loss functions, and training procedures. ReFace attacks can successfully deceive commercial face recognition services via transfer attack and reduce face identification accuracy from 82% to 16.4% for AWS SearchFaces API and Azure face verification accuracy from 91% to 50.1%.**

***Index Terms*—adversarial attacks, face recognition, real-time attack, security**

I. Introduction

Face recognition and verification systems are widely used for identity authentication in government surveillance, military applications, public security settings such as airports, hotels, banks as well as smartphones to unlock applications. Over recent years, Convolutional Neural Networks (CNNs) have achieved state-of-the-art results on several face recognition and verification benchmarks outperforming traditional computer vision algorithms that rely on hand engineered features. With the widespread adoption of face recognition models in surveillance and other security sensitive applications, careful vulnerability analysis is imperative to ensure their safe deployment.

Several works have shown that deep neural networks (DNNs) are vulnerable to adversarial examples, causing the model to make an incorrect prediction with higher confidence [5], [10], [15], [23], [30]. Particularly, past attacks [11], [37] on face recognition systems have garnered immense media attention [1], [2] by utilizing projected gradient descent (PGD) [26] based approaches to achieve high fooling success rates. However, designing such adversarial examples requires the adversary to solve an optimization problem for each input. This makes the attack impractical in real-time since the adversary would need to re-solve the data-dependent optimization problem from scratch for every new input. The aforementioned methods for generating adversarial examples may cause a timing bottleneck that could hinder real-time image uploads, making it impractical to deploy such attacks. This bottleneck becomes even more pronounced when dealing with videos, where adversarial examples need to be generated for multiple frames per second. For example, during surveillance real-time face recognition models operate on live video streams from security cameras or webcam interfaces. In order to expose any real-time security vulnerabilities in such systems, it is necessary to generate an adversarial video stream in real-time as well.

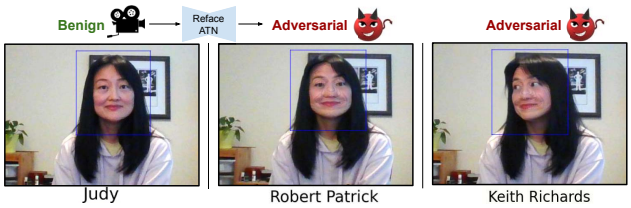

Fig. 1. Real-time ReFace attack on a face recognition model operating on a live video stream. ReFace uses an Adversarial Transformation Network (ATN) to inject adversarial perturbations into the video frames causing the face recognition model to mis-predict the identity of the subject in the video. Sample screenshots from the attack demo video posted on the project webpage.

To generate adversarial attacks against classification systems in real-time, some past works, such as Adversarial Transformation Networks (ATNs) [6], have attempted to learn a perturbation function with a neural network. ATNs are encoder-decoder neural networks that are trained to generate an adversarial image directly from an input image without having to perform multiple forward-backward passes on the victim classification model during inference, thereby making the attack possible in real time. However, ATNs have only been explored for classification tasks. The training objective studied thus far for an ATN is to push the classifier's output outside the decision boundary of the correct class. Unlike a classification model, where model outputs are class probabilities, the output space of a typical face recognition system is an embedding vector. A face recognition system is trained to cluster the embeddings of the same identity together in the embedding space while ensuring they are well separated from the embeddings of other

2158-3927/23/$31.00 ©2023 IEEE
DOI 10.1109/DSN58367.2023.00038

identities. Therefore when attacking such a setup, the attack objective requires the adversary to target the embedding space rather than the decision boundaries of the classifier.

To perform attacks on face recognition models, we first develop training objectives that target the embedding space of face recognition models and optimize metrics that degrade the identification and verification performance of such models. To minimize perceptibility of our perturbations, we incorporate Learned Perceptual Image Patch Similarity L_{pips} perceptual loss [45] in addition to the L_∞ constraint during training. Next, to perform real-time attacks, we design a new ATN based on the U-net [34] architecture, since U-nets have been notably effective in many prior image-to-image translation tasks [19], [20]. We find that while a U-net based ATN can generate real-time adversarial examples, the attack performance falls short as compared to per-image gradient based attacks such as PGD [26] at the same magnitude of adversarial perturbation. This is because gradient-based attacks generate highly tailored adversarial examples that are optimized on a single image. We address the performance gap between ATN and PGD attacks through neural architectural improvements to our ATN model which we describe in Section III-D.

Having bridged the gap with gradient based attacks on seen victim models, we evaluate the transferability of our adversarial samples to unseen models. Since ATNs are trained on a diverse set of images, we find that perturbations generated from an ATN are more transferable to unseen architectures as compared to per-input PGD attack, while being much faster to compute. To further improve our attack transferability, we adapt our ATN training framework to target an ensemble of face recognition models with various backbone architectures. Our best ATN attacks on unseen models successfully reduce the performance of face recognition models to the level of random guessing or worse. We present a demo video of our attack in real-time on our project webpage [1] with sample images presented in Figure 1. Finally, we demonstrate our attack effectiveness against cloud-hosted face recognition APIs in a complete black-box setting.

The technical contributions of our work are as follows:

- We propose a real-time attack framework to study the robustness of face recognition systems and demonstrate that our proposed ATN can synthesize adversarial examples several orders of magnitude faster than existing attacks on face recognition systems while achieving comparable attack success metrics as past works. To the best of our knowledge this is the first real-time attack on face recognition systems, in contrast to previous works which perform gradient based attacks or study real-time attack only in the classification domain.
- We bridge the performance gap between real-time ATN attacks and PGD attacks by developing a Residual U-net architecture that allows us to effectively increase the capacity of the ATN (Section III-D). Our ResU-Net ATN approaches PGD performance in white-box attacks and outperforms PGD on black-box transfer attacks.
- We develop and release a benchmarking library for face recognition models (Section IV-A)[2], implemented in the PyTorch framework. This allows us to evaluate our attacks on diverse set of architectures and loss functions. This library may be used to develop more robust face recognition models and to provide benchmarks of models' performance in an adversarial setting.
- We demonstrate the effectiveness of our real-time attacks on commercial face recognition services such as Amazon Face Rekognition and Microsoft Azure Face. Our attacks reduce face identification accuracy from 82% to 16.4% for AWS SearchFaces and face verification accuracy from 91% to 50.1% for Microsoft Azure.

II. Related Work

A. Adversarial Examples

An adversarial example is an input sample which has been perturbed in a way that is intended to cause misclassification by a victim machine learning model [7], [28], [43]. Prior work on attacks have demonstrated that adversarial examples can circumvent state-of-the-art image classification models while remaining indistinguishable from benign images for humans [10], [15], [17], [18], [26], [30], [31], [40]. However many of these works are gradient based attacks, which cannot be performed in real-time. To address this limitation, the authors of UAPs [27] demonstrated that there exist universal *input-agnostic* perturbations which when added to any image will cause the image to be misclassified by a victim network. The existence of such perturbations poses a threat to machine learning models in practical settings since the adversary may simply add the same pre-computed perturbation to a new image and cause misclassification in real-time. Also addressing the real-time challenge, the authors of [6] designed Adversarial Transformation Networks (ATNs) that follow an encoder-decoder architecture and output an adversarial perturbation for each input image, without having to compute gradients from the victim classification model during inference [6]. Unlike UAPs, ATNs generate input-specific perturbations. However these ATN attacks are specific to image classification tasks and cannot be directly used to attack face recognition models that use task-specific model and loss functions as opposed to the standard cross-entropy loss used by classifiers.

B. Facial Recognition Systems

Unlike typical classification algorithms, facial recognition systems do not have a fixed set of classes. Instead, a face recognition system must establish a person's identity and can operate in two different modes 1) *face verification* or 2) *face identification*. Both problems are generally solved with metric learning [25], [36], [41]. Recognition algorithms learn an embedding of face images in which the distance between embedding vectors indicates whether or not the vectors came

[1]Demo video: https://refaceattack.github.io/

[2]Model test bed to be released upon publication

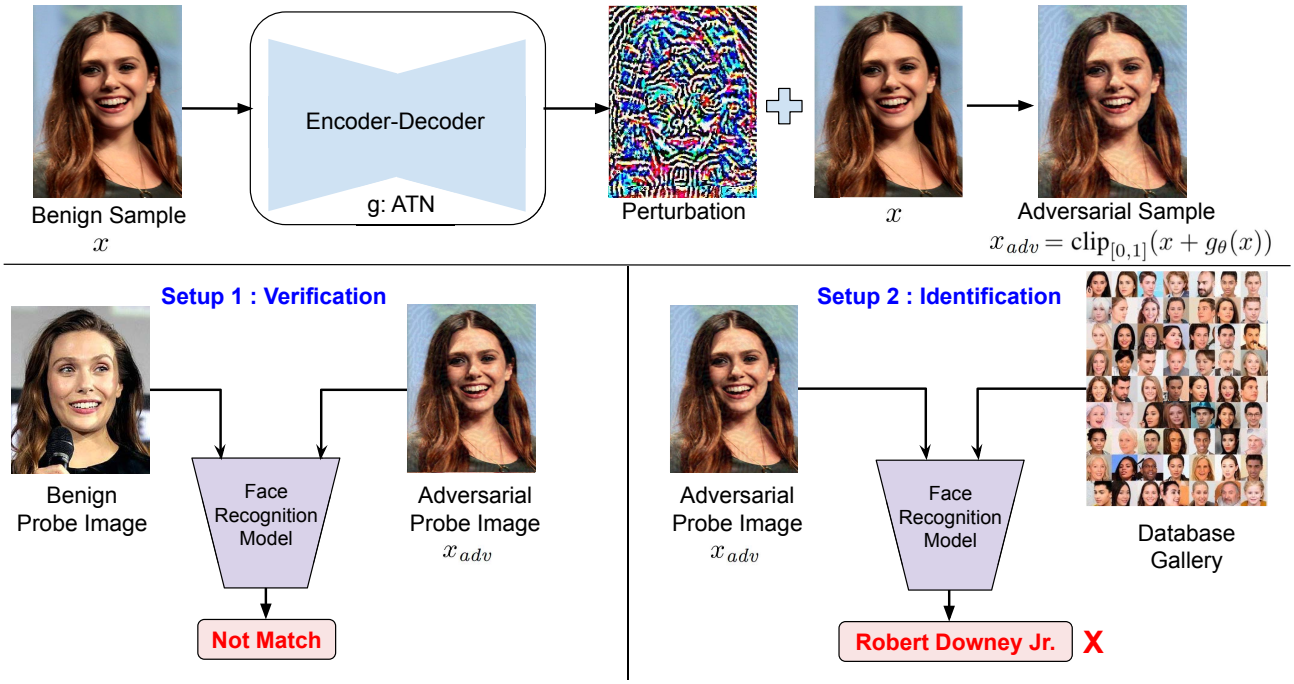

Fig. 2. Overview of ReFace adversarial perturbation generator (top) and attack application on face verification and identification systems (bottom). The ATN perturbation generator takes as input a benign image and generates a quasi-imperceptible perturbation which when added to the benign image causes the face-recognition model to misbehave for both verification and identification tasks.

from the same identity. Given this learned embedding function, face verification amounts to applying a threshold to the distance between two embedding vectors, and face identification amounts to ranking images by closeness to the query image. While most state-of-the-art facial recognition algorithms use this general approach, they use different strategies to learn the metric. DeepID [41] is a CNN based network which follows a training process where each unique training identity is treated as a separate class and the network is trained using softmax loss. The second-to-last layer then serves as the embedding, which is effective on unseen identities. SphereFace [36] builds on DeepID by replacing traditional softmax loss with angular softmax loss. Angular softmax ensures that Euclidean distance in the embedding space produces optimal decision boundaries between identities. Similarly, ArcFace [13] loss is also adopted by recent state-of-the-art face recognition models, and these models are used as the main testbench in recent literature [22], [37] to study the effectiveness of adversarial attacks. In our work, we study the effectiveness of adversarial attacks using ATNs on face recognition models trained with SphereFace, DeepID and ArcFace loss, in addition to black-box models trained with unseen loss functions.

C. Adversarial Attacks on Face Recognition

While several works have studied adversarial attacks on face recognition models, these are relatively fewer in literature as compared to image classification attacks. Some prior works include physical adversarial examples in the form of objects such as glasses [38], [39] and hats [22] that can fool models to make wrong prediction on the person wearing the object. The authors of [33] attempt to target face "classification" networks which operate differently from face recognition networks that perform face verification and identification. Prior works such as [11], [14], [35], [37] generate adversarial examples for face recognition systems by optimizing the perturbation for each image using white-box access to a face-recognition model. One such attack Face-Off [11] demonstrates that it is possible to generate adversarial faces by optimizing in the model embedding space using PGD [26] and CW [10] attack, however reports an attack run-time from 6 seconds to 373 seconds per image while using 2 GPUs. The authors of Face-Off evaluate the strongest attack algorithms to perform adversarial attack on Face Recognition models. They report that PGD algorithm used in their experiments is stronger than CW attack. This finding is corroborated in other papers [44]. Another gradient based attack Lowkey [12] generates image-specific adversarial samples for face recognition models and demonstrates their transferability to public cloud provider APIs, however reports an attack run-time of 32 seconds per image. To generate adversarial examples in black-box settings, the authors of [14] utilize an evolutionary optimization technique, but require at least 1,000 queries to the target face recognition system before a realistic adversarial face can be synthesized. Similarly, the more recently proposed black-box attack by [8] on face recognition systems requires at least 1700 queries to generate successful attacks. The time for

generating adversarial examples using the above techniques can potentially bottleneck real-time image upload making the attacks impractical for deployment. The timing bottleneck gets even more significant for videos in which we need to generate adversarial examples for several frames per second. In contrast, we propose a framework to adversarially modify query images in real-time, such that the performance of face recognition models deteriorate significantly in both white-box and transfer based black-box attack settings. Unlike past works on face recognition, our proposed approach enables real-time attacks on video streams which we demonstrate via successful attacks over web-cam interfaces (demo video linked in the second page).

III. Methodology

A. Victim Models

A typical face recognition pipeline first detects and crops faces. Next, they map each cropped image x to an embedding vector y using $F : x \mapsto y$. Typically, such models are trained on a dataset of facial images and identity labels, with the objective of clustering embeddings of images from the same identity together and ensuring separability between embeddings of images from different identities. State-of-the-art face recognition models are commonly trained with objectives that effectively optimize a cosine distance metric e.g. SphereFace [25], DeepID [41] or ArcFace [13] loss. During inference, a face recognition model can be used for one of the following goals:

1. Verification - A face recognition model can be used to verify whether two images belong to the same person or not. In this setting, the model compares the embeddings of two probe images and reports a match if the distance between the embeddings of the two models is below a certain threshold.

2. Identification - In this setting, the face recognition system tries to associate a person with an identity from a set of identities in *gallery images* stored in the system's database. When presented with a *probe image*, the system compares the embedding of the probe image with the gallery images to find the closest matching neighbour in the gallery and determine the identity of the probe image.

In our work, we attack CNN-based face recognition models in real time and assess the success rate against both of the above goals. To simplify experimentation, we do not include the detection and cropping step in our attacks pipeline. Instead we use the pre-cropped images provided by standard datasets.

B. Attack Goal

Threat Model: Given benign facial input images, our goal is to adversarially modify the inputs in real-time, such that the modified inputs cause the face recognition model to mispredict the embedding vectors, thereby degrading the verification and identification performance of the face recognition model. In order to adversarially modify each image, we design a perturbation generator that operates in real-time to add a quasi-imperceptible adversarial perturbation to the given input image. When attacking a *face verification* system, we adversarially perturb one of the two probe images. In this attack setting, our goal is to reduce the true recall rate of the verification system (performance on positive pairs). When attacking a *face identification* system, we assume the probe images have been adversarially perturbed while the dataset of gallery images is benign. In this attack setting, our goal is to lower the recognition rate of the face identification system.

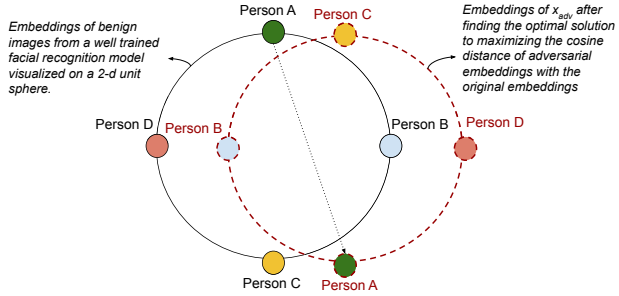

Fig. 3. *Visualizing the optimum solution to our attack objective:* Our attack objective pushes the originally predicted embedding vectors to the opposite end of the unit sphere thereby hampering the performance of the face-recognition model.

Problem Formulation: To achieve the above objectives, we train a perturbation generator g_θ, parameterized by θ, which takes as input an image x and generates an adversarial perturbation $g_\theta(x)$ that can be added to x to synthesize an adversarial example x_{adv}. The optimization objective of g_θ is to maximize the cosine distance the embeddings of the adversarial and original image, while constraining the amount of the perturbation added to the image. This is different from the objective for fooling classification systems, where the commonly used objective for untargeted attacks is to maximize the cross-entropy loss with the correct label. L_p norm is a widely used distance metric for measuring the distortion between the original and adversarial inputs. Prior works [15] recommend constraining the maximum distortion of any individual pixel using the L_∞ norm. To further reduce the perceptibility of the perturbation we incorporate L_{pips} [45] loss during training. L_{pips} distance measures the visual similarity between two images by comparing the embeddings from a pre-trained CNN model.

Mathematically, our attack objective is as follows:

$$\begin{gathered} \forall_{x \in X} \ \textit{maximize} \ [d(F(x_{adv}), F(x)) - \lambda L_{pips}(x_{adv}, x)] \quad (1) \\ \textit{where} \ x_{adv} = \text{clip}_{[0,1]}(x + g_\theta(x)) \\ \textit{s.t} \ ||g_\theta(x)||_\infty < \epsilon \end{gathered}$$

where $d(F(x_{adv}), F(x))$ is the cosine distance between embeddings of the adversarial and original image and λ is the loss coeffecient for L_{pips}. In Figure 3 we illustrate how an optimum solution to the above problem of maximizing the cosine distance completely degrades the performance of a face recognition model. A visualization of such embedding clusters for a hypothetical case of four individuals on a 2-D unit sphere is shown on the left in Figure 3. If we

were to find the optimum solution to our attack objective in an unbounded attack setting, the embeddings clusters for adversarial images will move to the opposite end of the unit sphere (to maximize the cosine distance). This clearly results in hampering both verification and identification performance of the model since the embeddings of benign and adversarial examples are completely rotated to the opposite ends in the unit sphere.

In our work, we model g_θ as a neural encoder-decoder architecture called an Adversarial Transformation Network (ATN) (Section III-C).

C. ATN: Adversarial Transformation Network

An ATN is a neural network trained to produce adversarial images, with the form $g_\theta : \mathcal{X} \rightarrow \mathcal{X}$. Since the network only needs one forward pass to compute the perturbation, it is less expensive than an iterative gradient-based optimization procedure. We obtain an adversarial image from a benign image using the neural network N_θ as follows:

$$g_\theta(x) = \epsilon \cdot \tanh(N_\theta(x)) \tag{2}$$

With this formulation we enforce the constraint $||x_{adv} - x||_\infty < \epsilon$ since the output of $\tanh$ is bounded between $[-1, 1]$.

Algorithm 1 Ensemble attack training procedure

Inputs: Victim Models $\mathbb{F} = F_1, \ldots F_n$, image dataset X
Output: Perturbation engine (g_θ) parameters θ
HyperParams: Learning rate α, L_∞ bound ϵ, L_{pips} loss coefficient λ
Initialize ATN: N_θ
Batch training images: $X_{batched} \leftarrow Batch(X)$
for *epoch* in 0 to N_{epochs} **do**
  for x in $X_{batched}$ **do**
    $x_{adv} \leftarrow \text{clip}_{[0,1]}(x + \epsilon \cdot \tanh(N_\theta(x)))$
    $loss \leftarrow 0$
    for F_i in F **do**
      $loss \leftarrow loss + (-d(F_i(x), F_i(x_{adv})))$
    end for
    $loss \leftarrow loss/len(F)$
    $loss \leftarrow loss + \lambda L_{pips}(x_{adv}, x)$
    $\theta \leftarrow \theta - \alpha \cdot \nabla_\theta(loss)$
  end for
end for
return θ

We train the ATN to generate adversarial examples using the procedure described in Algorithm 1. Our ATN can be trained to target one or more face recognition models in the model set $\mathbb{F}$. During each mini-batch iteration, we generate a batch of adversarial images from the ATN and compute the cosine distance between embeddings of benign and adversarial images. We accumulate the loss for all models in the set $\mathbb{F}$ and can optionally add the L_{pips} loss to minimize the perceptibility of the adversarial perturbation. Finally, we backpropogate through all models in the set $\mathbb{F}$ to compute the gradient of the loss with respect to the parameters θ of the ATN and update the ATN parameters using mini-batch gradient descent with a learning rate α. Targeting an ensemble of face recognition models during training can result in more transferable adversarial attacks. In our experiments, we verify this hypothesis and demonstrate that ATNs trained to target an ensemble of models result in better transferability to unseen models.

D. The search for an effective ATN architecture

The input and output domains of the ATN have the same spatial dimension, so a logical choice for the network architecture is a U-net [34]. U-nets are commonly used for several image-to-image translation problems. The architecture consists of several down-sampling layers followed by an equal number of up-sampling layers. The feature maps from the down-sampling layers have skip connections that are concatenated to the up-sampling layers with matching resolution. Previous work with ATNs used different architectures, but in our preliminary experiments, we found that U-nets were far more effective than alternate architectures at the same level of perturbation.

However, we still found that there was a large gap between a U-net based ATN and an iterative gradient-based white-box attack, *even on the training data*. This is illustrated in Figure 5 in our experiments comparing PGD-30 (i.e., 30 iterations of PGD) to the U-net ATN. From the universal approximation theorem [32], a neural network could in principle represent a close approximation of the PGD-30 function. As this neural network would have lower training loss than the U-net ATN, it appears that this architecture is *underfitting*. We therefore explored ways to add capacity to the ATN. We found that adding layers and making the layers wider both led to small gains in performance with diminishing returns.

One feature of the U-net is that every layer changes the spatial resolution. The deeper layers of the U-net necessarily operate at very low spatial resolutions. Intuitively, it may be useful to be able to express complex hierarchical functions at higher resolutions. We developed a new Residual U-net architecture, illustrated in Figure 4, that replaces individual convolution and transpose convolution layers in a U-net with groups of residual blocks. We use 2-layer pre-activation blocks with ReLU and batch normalization. One skip connection per downsample is carried over to the decoder, which allows arbitrary numbers of residual blocks at each step. We denote the number of blocks in each group as a vectors $\mathbf{E}$ and $\mathbf{D}$ for the encoder and decoder, respectively. While similar architectures have been proposed in the past [4], [24], they are not widely used and have not been used in adversarial perturbation literature.

We found that adding layers in this architecture was considerably more effective than in the pure U-net ATN. We performed an architecture search to find an effective balance between computational cost and attack effectiveness. The optimal architecture from this process had five downsampling steps with $\mathbf{E} = [1, 1, 2, 3, 5]$ and $\mathbf{D} = [1, 1, 1, 1, 1]$. We use a base width of 64 channels and double the width at each

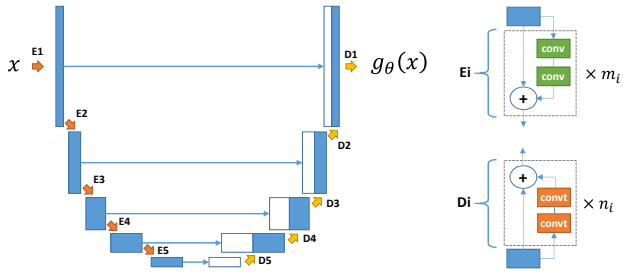

Fig. 4. Residual U-net architecture: We replace the strided convolutions and transposed convolutions in the U-Net architecture with residual blocks. Each residual block contains multiple convolutions (in the encoder) or transposed convolution (in the decoder) layers.

	Networks			*Verification*		*Identification*
Name	Architecture	# Models	Loss	V-AUC	V-Acc.	R1-Acc.
RN-SF-1	ResNet	1	SphereFace	0.99	95.2	84.4
RN-DID-1	ResNet	1	DeepID	0.98	93.3	78.0
IN-SF-1	InceptionNet	1	SphereFace	0.99	94.4	78.5
RN-AF-1	ResNet	1	ArcFace	0.98	92.8	89.0
RN-SF-6	ResNet	6	SphereFace	0.99	94.3	82.0
RN-DID-6	ResNet	6	DeepID	0.98	93.0	77.4
IN-SF-4	InceptionNet	4	SphereFace	0.99	94.4	78.9
RN-AF-4	ResNet	4	ArcFace	0.98	93.3	90.0

TABLE I
VICTIM MODEL SETS USED FOR CONDUCTING OUR ATTACK EVALUATIONS. EXPERIMENTS ARE CONDUCTED ON BOTH SINGLE AND ENSEMBLE MODEL SETS. THE VERIFICATION AND IDENTIFICATION METRICS ARE AVERAGES OVER THE WHOLE MODEL SET REPORTED ON THE *clean* UNPERTURBED VGGFACE2 TEST SET.

downsample step except for the last. Using this ResU-net architecture, the ATN approached the performance of PGD-30 (plotted in Figure 5a) with roughly 10,000× less runtime. We refer the readers to the code-base included in our supplementary material for the precise model implementation.

IV. EXPERIMENTS

We perform experiments to evaluate our proposed attack in both white-box and transfer-attack settings. We perform the attack at different levels of adversarial perturbations and study how factors such as victim model architecture, loss functions and random initialization affect the success rate of the attacks. We also perform a timing analysis and demonstrate that our attacks can be performed in real-time and achieve a high success rate in both white-box and transfer attack settings. We compare our attack performance against state-of-the-art adversarial attacks on face recognition models such as Face-Off [11] and Fawkes [37] that utilize the PGD attack algorithm. Finally, we perform our transfer attack on black-box public APIs (Amazon Rekognition and Microsoft Azure) and demonstrate that our attacks can significantly reduce both the verification and identification performance of such APIs.

A. Dataset and Models

We develop a benchmarking framework in PyTorch to evaluate both white-box and transfer attack performance of adversarial examples generated using our ATNs. Our experiments are designed to examine how factors such as network architecture, training loss functions, and random initialization affect the transferability of attacks. We used two main CNN architectures for the face recognition models: pre-activation ResNet [16] and Inception-v4 [42]. Within these architectures, we varied the number of blocks leading to networks ranging from 22 to 118 layers which were trained with three different loss functions: DeepID [41], SphereFace [25] and ArcFace [13]. The dimension of the output embedding vector for all test-bed models is 512. The face recognition models are trained on the training partition of the VGGFace2 dataset [9]. We start with the standard crops provided by the dataset and perform random resized cropping for data augmentation during training. VGGFace2 dataset contains 3.31 million images across 9131 identities which is larger and more diverse as compared to other face recognition datasets such as FaceScrub [29] or UMDFaces [3]. We choose VGGFace2 dataset for training the test-bed face recognition models because models trained on larger and more diverse datasets are more robust and generalize better to unseen images [9].

Table I presents the test-bed of face recognition models that we use for training our ATNs. The model sets comprise single and ensemble versions for each architecture, enabling us to evaluate the efficacy of ensemble attacks on unknown models. For ensemble models, the reported metrics are averaged over all individual models in the ensemble. For the training and testing of the ATNs, we utilize the VGGFace2 validation set that was not used in the training of the test-bed face recognition models. Specifically, we partitioned the validation set of VGGFace2 into two distinct subsets, each containing a comparable number of images and non-overlapping identities. The resulting subsets comprise a training subset of 84,953 images and a testing subset of 84,443 images, used for training and testing the ATN models.

B. Evaluation Metrics

We evaluate the performance of face recognition models on both verification and identification tasks with the metrics described below.

Face Verification Metrics: For each identity in the test set, we prepare all possible pairs of distinct images that have the same identity. To keep our problem balanced, we randomly sample an equal number of non-matching pairs. On the test set of VGGFace2, this creates a total of 917,692 verification tests where half have a pair of images with matching identities (positive labels) and half have different identities (negative labels). Given this binary classification problem, we report the following metrics:

1. Verification AUC (V-AUC): We use the cosine distance between the embedding of the two images along with the verification label to generate a Receiver Operating Characteristic curve (ROC). Our metric is the standard area under the ROC curve (AUC).

			Single Defender Models				Ensemble Defender Models			
			RN-SF-1	RN-DID-1	IN-SF-1	RN-AF-1	RN-SF-6	RN-DID-6	IN-SF-4	RN-AF-4
Verification AUC	*No Attack*	Clean	0.99	0.98	0.99	0.98	0.99	0.98	0.99	0.98
	Single Model Attack	RN-SF-1	0.03	0.59	0.87	0.77	0.59	0.71	0.89	0.71
		RN-DID-1	0.58	0.04	0.88	0.72	0.64	0.55	0.88	0.68
		IN-SF-1	0.75	0.77	0.03	0.82	0.77	0.79	0.54	0.80
		RN-AF-1	0.60	0.62	0.88	0.04	0.68	0.72	0.90	0.60
	Ensemble Model Attack	RN-SF-6	0.06	0.12	0.35	0.17	0.07	0.14	0.37	0.19
		RN-DID-6	0.13	0.07	0.45	0.18	0.10	0.08	0.44	0.18
		IN-SF-4	0.37	0.43	0.05	0.45	0.31	0.43	0.05	0.46
		RN-AF-4	0.15	0.14	0.34	0.06	0.12	0.15	0.38	0.08
Verification Accuracy	*No Attack*	Clean	95.20%	93.30%	94.40%	92.80%	94.30%	93.00%	94.40%	93.30%
	Single Model Attack	RN-SF-1	48.78%	59.29%	67.90%	63.43%	64.34%	65.47%	82.70%	68.20%
		RN-DID-1	62.54%	49.45%	82.28%	64.12%	64.40%	60.63%	81.58%	69.54%
		IN-SF-1	71.21%	68.91%	48.43%	72.12%	70.90%	69.98%	63.64%	71.41%
		RN-AF-1	61.34%	60.12%	68.10%	49.21%	67.34%	66.62%	83.45%	64.57%
	Ensemble Model Attack	RN-SF-6	48.37%	48.47%	53.44%	49.89%	48.17%	48.32%	53.53%	48.87%
		RN-DID-6	48.85%	48.67%	57.20%	50.23%	48.13%	48.57%	56.44%	49.35%
		IN-SF-4	50.79%	50.98%	47.93%	51.21%	49.46%	50.86%	47.99%	51.21%
		RN-AF-4	49.53%	50.12%	58.12%	48.73%	49.62%	48.92%	57.41%	48.45%
Rank-1 Accuracy	*No Attack*	Clean	84.40%	78.00%	78.50%	89.00%	82.00%	77.40%	78.90%	90.00%
	Single Model Attack	RN-SF-1	0.05%	12.60%	43.45%	18.32%	17.96%	19.36%	43.96%	17.56%
		RN-DID-1	16.36%	0.02%	43.10%	19.87%	17.41%	12.57%	41.01%	18.76%
		IN-SF-1	30.31%	26.14%	0.02%	32.88%	26.75%	25.59%	16.36%	27.65%
		RN-AF-1	17.21%	13.10%	46.37%	0.03%	18.45%	21.23%	45.46%	15.21%
	Ensemble Model Attack	RN-SF-6	0.10%	0.22%	5.31%	0.98%	0.09%	0.22%	5.05%	0.43%
		RN-DID-6	0.72%	0.05%	8.59%	1.03%	0.27%	0.05%	7.62%	0.45%
		IN-SF-4	2.20%	2.20%	0.01%	3.10%	1.03%	2.03%	0.01%	2.41%
		RN-AF-4	0.81%	0.34%	6.43%	0.05%	0.31%	0.27%	8.12%	0.04%

TABLE II

WHITE-BOX AND TRANSFER ATTACK RESULTS OF ATN ATTACK AT $\epsilon = 0.03$. A LOWER VALUE FOR ALL THREE METRICS INDICATES A MORE SUCCESSFUL ATTACK. THE DIAGONAL ENTRIES IN EACH OF THE THREE TABLES REPRESENTS A WHITE-BOX ATTACK WHILE ALL OTHER ENTRIES REPRESENT A TRANSFER (BLACK-BOX) ATTACK.

2. Verification Accuracy (V-Acc.): To determine the accuracy, we need a threshold for the cosine distance, across which the example is labelled positive or negative. For each model, we set this to *equal error rate* threshold of the model on the (clean) VGGFace2 validation set.

Face Identification Metric: We use the VGGFace2 test set to create a random gallery with 100 unique identities. For each of these 100 identities, we select a probe image with one of the identities appearing in the gallery and compute its distance to each image in the gallery. This creates 100 identification tests. We repeat this gallery test on 1000 random galleries to create a total of 100,000 identification tests. When evaluating attacks, we perturb the probe image and leave the gallery unmodified. We report the *Rank-1 Accuracy (R-1)*, which is the percentage of tests where the image in the gallery with the minimum distance to the probe image has the same identity as the probe image.

C. Baseline Attacks

We compare the effectiveness of ATN attacks against three alternate attacks:

1) Universal Adversarial Perturbations (UAP): UAP is a single input-agnostic perturbation vector that can be added to all images to fool the victim models. UAP can be formulated as a simplified ATN where the ATN formulation reduces to : $g_\theta(x) = \epsilon \cdot \tanh(\theta_{h \times w \times c})$. That is, instead of modeling ATN as a neural network, the ATN is modelled using a perturbation vector $\theta_{h \times w \times c}$ which is trained using the same procedure given by Algorithm 1.
2) Fast Gradient Sign Method (FGSM): FGSM [15] attack obtains an adversarial example for an image by obtaining the gradient of the optimization objective with respect to the image and then perturbing the image in the direction of the gradient with step size ϵ. That is,

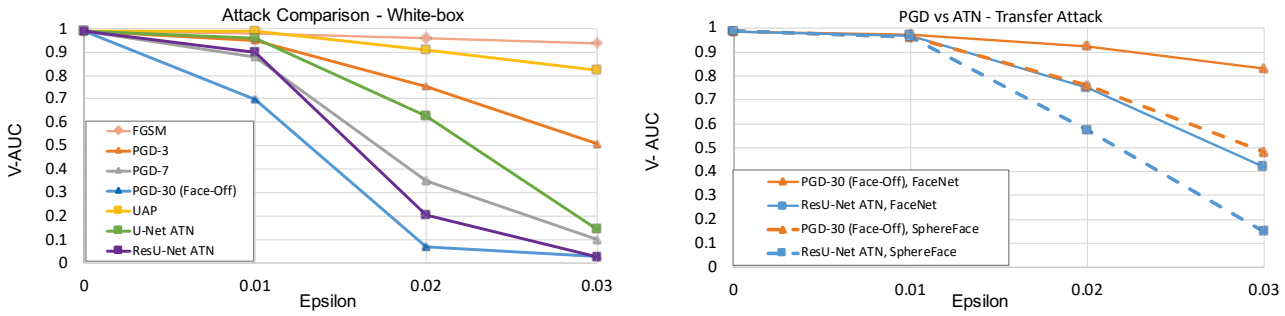

Fig. 5. Comparison of PGD and ATN based attacks. (a) compares white-box attacks on the single RN-SF-1 model. (b) compares transfer attacks optimized on the six RN-SF-6 models and evaluated on two different models.

$x_{adv} = clip_{[0,1]}(x + \epsilon \cdot sign(\nabla_x L(x)))$ where $L(x)$ is the optimization objective given by Equation 1.

3) Face-Off [11] / Projected Gradient Descent (PGD): PGD [26] attack is a multi-step iterative variant of the FGSM attack. Unlike ATNs and UAPs, PGD attack requires several forward and backward passes through a victim face-recognition model to find an adversarial perturbation that is highly optimized for a single image. We perform PGD attack as a baseline because it has been commonly adopted by past attacks such as Fawkes [37], Face-Off [11], [44] and achieves highest white-box attack success rates. In our experiments *PGD-n* refers to PGD with n iterations.

V. Results

We train six ATN models each targeting one of the model sets listed in Table I. The ATNs are trained using mini-batch gradient descent with a batch size 32 for 500K iterations using Adam optimizer [21] with a learning rate $2e$-4. Our primary evaluation is conducted using the ResU-Net ATN architecture described in Section III-D at max L_∞ distortion $\epsilon = 0.03$ in $[0, 1]$ pixel scale. We present the white-box and transfer attack results of our primary evaluation in Table II. Additionally, we present the results and comparisons for the pure U-Net architecture in Section V-B and comparison against alternate attacks at $\epsilon = [0.01, 0.02, 0.03]$ in Figure 5. Finally, we present examples of adversarial images generated at $\epsilon = [0.03]$ from each of the techniques in Figure 6.

A. Single Model Attack vs Ensemble Attack

Aversarial perturbations trained on an ensemble of victim models exhibit better transferability across model architectures than those trained on a single model. That is, the attack success metrics on unseen models for ATNs trained on ensemble models (RN-SF-6, RN-DID-6 and IN-SF-4) are significantly better than ATNs trained on single models (RN-SF-1, RN-DID-1 and IN-SF-1 respectively). The only difference amongst the models in an ensemble is their weight initialization. It is interesting to note that this difference in weight initialization offers enough variance in the model set to train significantly more generalizable perturbations, at the same level of distortion as compared to the single-model attacks.

B. ATN vs. PGD based attacks

State-of-the-art attacks such as Fawkes [37] and Face-Off [11] utilize PGD based attack against face recognition systems. As such, we implement the PGD based attack algorithm proposed in Face-Off on our models and architectures to compare the effectiveness of ATNs and PGD on both seen and unseen models. We optimize PGD-30 and ATN attacks on the same surrogate models and perform the attack on a random subset of 10,000 images from the test set. For a fair comparison, we drop the L_{pips} term from the loss and train purely to maximize the cosine distance with an L_∞ constraint. Figure 5a shows white-box attack success rate of PGD and ATN attacks on the RN-SF-1 model. As discussed in Section III-D, the Residual U-Net ATN architecture developed by us provides a large improvement over a basic U-net architecture and bridges the white-box performance gap between the ATN and PGD-30.

We also performed an ensemble attack on the six models from RN-SF-6. Running PGD-30 against six surrogate models simultaneously took more than three seconds per image, while the ATN's forward pass was the same complexity as other experiments - more than 10,000× faster than PGD-30. Table III reports the timing comparison of ATN and PGD attacks.

	Avg Wall-Clock Time (seconds)	
Process	GPU	CPU
RN-SF-1	2.93e-2	1.02e-1
ATN	2.83e-3	5.67e-2
UAP	1.89e-4	5.39e-3
PGD	3.73	365.2

TABLE III
AVERAGE WALL-CLOCK TIME IN SECONDS REQUIRED FOR GENERATING A SINGLE ADVERSARIAL IMAGE ON GPU (NVIDIA TITAN X) AND CPU PLATFORMS USING DIFFERENT ATTACKS. TIME FOR RN-SF-1 PROCESS INDICATES THE FORWARD PASS COMPUTATION TIME FOR A SINGLE RESNET FACE RECOGNITION MODEL.

In addition to being fast, ATNs learn attacks that generalize effectively to new models. We evaluated how well the perturbed images transferred to two different models. First, we evaluated against a ResNet+SphereFace model that is similar to the RN-SF-6 models, but has a different number of layers. Second, we evaluated against an open source model from the FaceNet repository[3]. This model uses a different architecture (Inception ResNet), loss (DeepID) and training procedure. We did not do any parameter tuning based on this model, so it serves as an independent validation of the transferability of our attacks.

Figure 5b compares the attacks at different L_∞ thresholds. As expected, transferring an attack from RN-SF-6 to the FaceNet model was more difficult than the ResNet+SphereFace model. However, in both cases the ATN attack is effective at $\epsilon = 0.02$ and transfers much better than PGD to the new models.

C. ATN vs. UAP

We find that attacks utilizing ATNs outperform the UAP attacks at the same level of perturbation. Since the goal of finding a single input-agnostic perturbation is more challenging than finding one perturbation per image, a higher amount of distortion is required for a successful attack using UAPs as compared to the ATN based attacks. This is indicated by less successful attack metrics (higher V-AUC) from UAPs in Figure 5a. However, it is important to note that UAPs pose a significant threat to face recognition models since they can be easily shared amongst attackers and are simpler to implement as compared to ATNs.

D. ATN Architecture Complexity vs Performance

To study the relationship between the architecture of the ATN model, computational cost and attack effectiveness, we train different ATN architectures to attack the RN-SF-1 face-recognition model. First, we consider different variants of the U-net architecture by progressively increasing the base channel-width from 16 to 64 and the number of downsampling/upsampling layers from 3 to 5. Next, we consider ResU-net architecture with 5 downsampling/upsamping layers with the default 2, 3, and 5 residual blocks in the last three downsampling layers (the default ATN model for all experiments described in Section V). We compare the number of parameters, inference time, and attack effectiveness of the different architectures in Table IV. We find that scaling up the architecture complexity and size helps improve attack performance with a marginal increase in the average wall-clock time which is real-time for all ATN attacks and several orders of magnitude faster than PGD-based attack (provided in Table III). Our proposed ResU-net architecture allows adding intermediate residual blocks to increase the number of parameters without requiring additional downsampling/upsampling layers or increasing the base-channel width.

[3]https://github.com/davidsandberg/facenet

	Model Size			*Time (seconds)*		*Attack Performance*
ATN Arch.	*#Layers*	*#channels*	*#Params*	GPU	CPU	Ver. AUC
U-net	3	32	$337k$	$1.01e$-3	$6.12e$-3	0.80
U-net	3	64	$1.45m$	$1.25e$-3	$1.35e$-2	0.75
U-net	5	16	$1.04m$	$1.37e$-3	$1.05e$-2	0.35
U-net	5	32	$4.17m$	$1.45e$-3	$1.76e$-2	0.21
U-net	5	64	$16.6m$	$1.73e$-3	$3.12e$-2	0.15
ResU-net	5	64	$48.6m$	$2.83e$-3	$5.67e$-2	0.03

TABLE IV
MODEL SIZE, INFERENCE TIME AND ATTACK EFFECTIVENESS COMPARISON FOR DIFFERENT ARCHITECTURES OF THE ATN MODEL. *#Layers* INDICATE THE NUMBER OF DOWNSAMPLING/UPSAMPLING LAYERS. *#Channels* INDICATE THE BASE CHANNEL WIDTH. INFERENCE TIME IS REPORTED AS THE AVERAGE WALL CLOCK TIME FOR A SINGLE IMAGE ON A SINGLE GPU (NVIDIA TITAN X) AND CPU. ATTACK EFFECTIVENESS IS REPORTED AT $\epsilon = 0.03$. LOWER VALUES OF VERIFICATION AUC INDICATE A MORE EFFECTIVE ATTACK.

VI. ATTACKING PUBLIC APIS

We demonstrate the effectiveness of our attacks against commercial face recognition systems. These systems are black-box, proprietary, and are abstracted away through a web-based API. We evaluate our perturbations against the Amazon (AWS) Rekognition and Microsoft Azure Face services.

Face Verification: In this setting, we target the *CompareFaces* API in AWS and the *verify_face_to_face* API in the Azure Face client. We prepare a total of 1000 image pairs (500 positive and 500 negative pairs sampled randomly from the VGGFace2 test set) and report the verification metrics in Table V.

Face Identification: We target the *SearchFaces* API in AWS Rekognition. The API accepts a gallery of N faces $x_1, x_2, x_3 ... x_N$ and a query image x_q, and returns similar faces to the query image from those in the gallery, ranked in order of similarity to the query image. We generate a gallery of 500 benign faces each with unique identities randomly sampled from the VGGFace2 test set and 500 adversarial samples by adversarially perturbing alternate images of the same identities as those in the gallery, resulting in a total of 500 trials. We report the Rank-1 accuracy of this experiment in Table V.

	Verification				*Identification*
	V-Acc. (%)		Recall (%)		Rank-1 Acc. (%)
Input type	AWS	Azure	AWS	Azure	AWS
Clean images	95.5	91.0	91.0	83.0	82.0
Ensemble ATN	64.7	50.1	30.2	2.1	16.4

TABLE V
ATN ATTACK RESULTS AT $\epsilon = 0.03$ AGAINST AWS AND AZURE FACE RECOGNITION APIS. THE ATN WAS TRAINED JOINTLY ON RN-SF-6 AND IN-SF-4. RECALL(%) INDICATES THE VERIFICATION ACCURACY ON ONLY THE POSITIVE PAIRS IN THE EVALUATION SET. FOR VERIFICATION, WE USE THE DEFAULT MATCH THRESHOLD 0.5 FOR BOTH AWS AND AZURE.

For attacking the above APIs, we train an ATN jointly on the ensemble of RN-SF-6 and IN-SF-4 models. Training the attack against an ensemble of different architectures helps achieve more effective attack generalization against black-box models. As reported by the results in Table V, for images perturbed

Fig. 6. Sample adversarial images generated by ReFace attack at $\epsilon = 0.03$ and their benign counterparts.

by our Ensemble ATN, we achieve a significant drop in both verification and identification metrics as compared to the API performance on Clean images. This is indicated by the lower verification accuracy, recall rate and identification accuracy for the Ensemble ATN attack.

VII. Conclusion

In this work we propose ReFace, a real-time, highly-transferable attack on face recognition models based on Adversarial Transformation Networks. Using our Residual U-Net ATN model, we bridge the performance gap between ATN and gradient-based PGD attacks while being several orders of magnitude faster than PGD attacks. Unlike prior work, our method enables real-time attacks on video streams which we demonstrate via successful attacks on face recognition over web-cam interfaces. We demonstrate that adversarial examples generated using ATNs can effectively bypass face recognition systems in both white-box and black-box transfer attack settings. Our work bridges the attack effectiveness gap between real-time ATN attacks and PGD attacks by developing a Residual U-net architecture that allows us to effectively increase the capacity of the ATN. Our ResU-Net ATN approaches PGD performance in white-box attacks and outperforms PGD on black-box transfer attacks. Adversarial examples generated from our framework can bypass commercial face recognition APIs in a complete black-box setting and reduce face identification accuracy from 82% to 16.4%. Our extensive experiments validate that ReFace attacks can effectively target the embedding space of face recognition models, and therefore serve as a strong benchmark to investigate the adversarial robustness of future models.

VIII. Acknowledgements

This research was, in part, funded under Defense Advanced Research Projects Agency (DARPA) contract HR00112090093. This research was, in part, funded by the U.S. Government. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. This work was, in part, supported by NSF TILOS under award number CCF-2112665, ARO MURI under award number W911NF-21-1-0322 and Intel PrivateAI Collaborative Research Institute. Approved for Public Release, Distribution Unlimited.

References

[1] Fawkes press release. In *https://sandlab.cs.uchicago.edu/fawkes/press*.
[2] The new york times. In *https://www.nytimes.com/2020/08/03/technology/fawkes-tool-protects-photos-from-facial-recognition.html*.
[3] Umdfaces dataset. In *http://umdfaces.io/*.
[4] Dina Abdelhafiz, Sheida Nabavi, Reda Ammar, Clifford Yang, and Jinbo Bi. Residual deep learning system for mass segmentation and classification in mammography. *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, 2019.
[5] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, 2018.
[6] Shumeet Baluja and Ian Fischer. Learning to attack: Adversarial transformation networks. In *Proceedings of AAAI*, 2018.

[7] 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 Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný, editors, *Machine Learning and Knowledge Discovery in Databases*, 2013.

[8] Junyoung Byun, Hyojun Go, and Changick Kim. Geometrically adaptive dictionary attack on face recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022.

[9] Q. Cao, Li Shen, Weidi Xie, O. Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, 2018.

[10] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. *2017 IEEE Symposium on Security and Privacy (SP)*, 2017.

[11] Varun Chandrasekaran, Chuhan Gao, Brian Tang, Kassem Fawaz, Somesh Jha, and Suman Banerjee. Face-off: Adversarial face obfuscation. *Proceedings on Privacy Enhancing Technologies*, 2021.

[12] Valeriia Cherepanova, Micah Goldblum, Harrison Foley, Shiyuan Duan, John P Dickerson, Gavin Taylor, and Tom Goldstein. Lowkey: Leveraging adversarial attacks to protect social media users from facial recognition. In *International Conference on Learning Representations*, 2021.

[13] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.

[14] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. Efficient decision-based black-box adversarial attacks on face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[15] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *stat*, 1050:20, 2015.

[16] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. *ArXiv*, abs/1603.05027, 2016.

[17] Shehzeen Hussain, Paarth Neekhara, Brian Dolhansky, Joanna Bitton, Cristian Canton Ferrer, Julian McAuley, and Farinaz Koushanfar. Exposing vulnerabilities of deepfake detection systems with robust attacks. *ACM Journal of Digital Threats: Research and Practice*, 2022.

[18] Shehzeen Hussain, Paarth Neekhara, Malhar Jere, Farinaz Koushanfar, and Julian McAuley. Adversarial deepfakes: Evaluating vulnerability of deepfake detectors to adversarial examples. In *WACV*, 2021.

[19] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

[20] Mikhail E Kandel, Yuchen R He, Young Jae Lee, Taylor Hsuan-Yu Chen, Kathryn Michele Sullivan, Onur Aydin, M Taher A Saif, Hyunjoon Kong, Nahil Sobh, and Gabriel Popescu. Phase imaging with computational specificity (pics) for measuring dry mass changes in subcellular compartments. *Nature communications*, 2020.

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[22] Stepan Komkov and Aleksandr Petiushko. Advhat: Real-world adversarial attack on arcface face id system. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021.

[23] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *5th International Conference on Learning Representations, ICLR*, 2017.

[24] Heyi Li, Dongdong Chen, Bill Nailon, Mike E. Davies, and Dave Laurenson. Improved breast mass segmentation in mammograms with conditional residual u-net. *ArXiv*, abs/1808.08885, 2018.

[25] Weiyang Liu, Y. Wen, Zhiding Yu, Ming Li, B. Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6738–6746, 2017.

[26] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[27] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[28] Paarth Neekhara, Shehzeen Hussain, Prakhar Pandey, Shlomo Dubnov, Julian McAuley, and Farinaz Koushanfar. Universal Adversarial Perturbations for Speech Recognition Systems. In *Proc. Interspeech*, 2019.

[29] Hong-Wei Ng and Stefan Winkler. A data-driven approach to cleaning large face datasets. In *2014 IEEE international conference on image processing (ICIP)*. IEEE, 2014.

[30] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2016.

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

[32] Allan Pinkus. Approximation theory of the mlp model in neural networks. *Acta Numerica*, 8:143 – 195, 1999.

[33] Arezoo Rajabi, Rakesh B Bobba, Mike Rosulek, Charles V Wright, and Wu-chi Feng. On the (im) practicality of adversarial perturbation for image privacy. *Proceedings on Privacy Enhancing Technologies*, pages 85–106, 2021.

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI*. Springer International Publishing, 2015.

[35] Andras Rozsa, Manuel Günther, and Terranee E. Boult. Lots about attacking deep features. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 168–176, 2017.

[36] Florian Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.

[37] Shawn Shan, Emily Wenger, Jiayun Zhang, Huiying Li, Haitao Zheng, and Ben Y Zhao. Fawkes: Protecting privacy against unauthorized deep learning models. In *29th USENIX Security Symposium*, 2020.

[38] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. Association for Computing Machinery, 2016.

[39] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. A general framework for adversarial examples with objectives. *ACM Transactions on Privacy and Security (TOPS)*, 2019.

[40] Yucheng Shi, Siyu Wang, and Yahong Han. Curls & whey: Boosting black-box adversarial attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[41] Y. Sun, Xiaogang Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[42] Christian Szegedy, S. Ioffe, V. Vanhoucke, and Alexander Amir Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017.

[43] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013.

[44] Ying Xu, Kiran Raja, Raghavendra Ramachandra, and Christoph Busch. Adversarial attacks on face recognition systems. In *Handbook of Digital Face Manipulation and Detection*. Springer, Cham, 2022.

[45] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.