Varun Chandrasekaran, Chuhan Gao, Brian Tang, Kassem Fawaz*, Somesh Jha, and Suman Banerjee

Face-Off: Adversarial Face Obfuscation

Abstract: Advances in deep learning have made face recognition technologies pervasive. While useful to social media platforms and users, this technology carries significant privacy threats. Coupled with the abundant information they have about users, service providers can associate users with social interactions, visited places, activities, and preferences–some of which the user may not want to share. Additionally, facial recognition models used by various agencies are trained by data scraped from social media platforms. Existing approaches to mitigate associated privacy risks result in an imbalanced trade-off between privacy and utility. In this paper, we address this trade-off by proposing Face-Off, a privacy-preserving framework that introduces strategic perturbations to images of the user's face to prevent it from being correctly recognized. To realize Face-Off, we overcome a set of challenges related to the black-box nature of commercial face recognition services, and the scarcity of literature for adversarial attacks on metric networks. We implement and evaluate Face-Off to find that it deceives three commercial face recognition services from Microsoft, Amazon, and Face++. Our user study with 423 participants further shows that the perturbations come at an acceptable cost for the users.

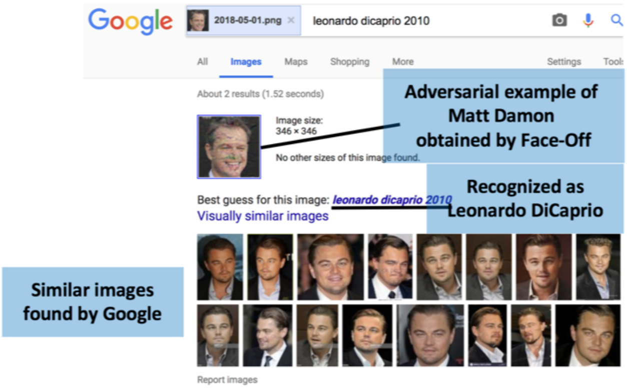
Keywords: face recognition, privacy

DOI 10.2478/popets-2021-0032

Received 2020-08-31; revised 2020-12-15; accepted 2020-12-16.

1 Introduction

Enabled by advances in deep learning, face recognition permeates several contexts, such as social media, online photo storage, and law enforcement [1]. Platforms such as Facebook, Google, and Amazon provide users with various services built atop face recognition, including automatic tagging and grouping of faces. Users share their photos with these platforms, which detect and recognize the faces present in these photos. Automated face recognition, however, poses significant privacy threats, induces bias, and violates legal frameworks [1, 2]. These platforms operate proprietary (black-box) recognition models that allow for associating users with social interactions, visited places, activities, and preferences–some of which the user may not want to share.



About 2 results (1.52 seconds)

Image size:
346 × 346

No other sizes of this image found.

Best guess for this image: *leonardo dicaprio 2010*

Visually similar images

Report images

Adversarial example of Matt Damon obtained by Face-Off

Recognized as Leonardo DiCaprio

Similar images found by Google

Fig. 1. Adversarial example of `Matt Damon` generated by Face-Off recognized as `Leonardo DiCaprio` by Google image search.

Existing approaches to mitigate these privacy risks result in an imbalanced trade-off between privacy and utility. Such approaches rely on (a) blurring/obscuring/morphing faces [3], (b) having the users utilize physical objects, such as eyeglass frames, clothes, or surrounding scenery with special patterns [4, 5], and (c) evading the face detector (the necessary condition for face recognition) [6]. These solutions, however, exhibit two main drawbacks to users. First, the user can no longer meet their original goal in using the social media platform, especially when various applications built atop of face detection (such as face-enhancement fea-

Varun Chandrasekaran: University of Wisconsin–Madison, E-mail: chandrasekaran@cs.wisc.edu
Chuhan Gao: Microsoft, work done while at University of Wisconsin–Madison, E-mail: chuhan@cs.wisc.edu
Brian Tang: University of Wisconsin–Madison, E-mail: bjtang2@wisc.edu
***Corresponding Author: Kassem Fawaz:** University of Wisconsin–Madison, E-mail: kfawaz@wisc.edu
Somesh Jha: University of Wisconsin–Madison, E-mail: jha@cs.wisc.edu
Suman Banerjee: University of Wisconsin–Madison, E-mail: suman@cs.wisc.edu

*First two authors contributed equally.

tures) are broken. Second, specially manufactured objects for physical obfuscation are not omnipresent and might not be desirable by the user.

Relying on insights from prior work [4–6], we propose a new paradigm to improve the trade-off between privacy and utility for such users. In adversarial machine learning, carefully crafted human-imperceptible perturbations cause misclassifications [7–9]. In this paper, *we extend this approach from classification models to metric learning,* as used in face recognition systems. In particular, we propose Face-Off, a system that preserves the user's privacy against *real-world* face recognition systems. By carefully designing the adversarial perturbation, Face-Off targets only face recognition (and not face detection), preserving the user's original intention along with context associated with the image. Face-Off detects a user's face from an image to-be-uploaded, applies the necessary adversarial perturbation, and returns the image with a perturbed face. However, the design of Face-Off faces the following challenges:

- Unlike classification networks, metric learning networks (used for face verification/recognition) represent inputs as feature embeddings [10–12]. Real-world face recognition maps the feature embedding of an input image to the closest cluster of faces. Existing approaches target classification networks and must be retrofit for metric learning.
- As the models used by these organizations are proprietary, Face-Off needs to perform *black-box attacks*. This issue is already challenging in the classification domain [7, 9, 13]. Further, Face-Off cannot use the service provider's face recognition API as a black-box oracle to generate the adversarially perturbed face [4] for two reasons. First, generating an adversarial example requires querying the API extensively, which is not free and is often rate-limited[1]. Second, querying the black-box model defeats our purpose for privacy protection as it sometimes begins with releasing the original face [4].

We address the first challenge by designing two new loss functions targeting metric learning. These loss functions aim to pull the input face away from a cluster of faces belonging to the user (in the embedding space), which results in incorrect face recognition. Both loss functions can be integrated with the state-of-the-art adversarial attacks against classification networks [9, 15].

To meet the second challenge, we leverage *transferability*, where an adversarial example generated for one model is effective against another model targeting the same problem. We rely on surrogate face recognition models (which we have full access to) to generate adversarial examples. Then, Face-Off amplifies the obtained perturbation by a small multiplicative factor to enhance transferability. This property reduces the probability of metric learning networks correctly recognizing the perturbed faces. Further, we explore amplification, beyond classifiers [16, 17], and show it enhances transferability in metric learning and reduces attack run-time.

We evaluate Face-Off across three major commercial face recognition services: Microsoft Azure Face API [18], AWS Rekognition [19], and Face++ [20]. Face-Off generates perturbed images that transfer to these three services, preventing them from correctly recognizing the input face. Our adversarial examples also transfer to Google image search (refer Figure 1) successfully with the target labels. Based on a longitudinal study (*across two years*), we observe that commercial APIs have not implemented defense mechanisms to safeguard against adversarial inputs. We show that using adversarial training [15] as a defense mechanism deteriorates natural accuracy, dropping the accuracy by 11.91 percentage points for a subset of the VGGFace2 dataset. Finally, we perform two user studies on Amazon Mechanical Turk with 423 participants to evaluate user perception of the perturbed faces. We find that users' privacy consciousness determines the degree of acceptable perturbation; privacy-conscious users are willing to tolerate greater perturbation levels for improved privacy.

In summary, our contributions are:

1. We propose two new loss functions to generate adversarial examples for metric networks (§ 2.2). We also highlight how amplification improves transferability for metric networks (§ A.1).
2. We design, implement, and evaluate Face-Off, which applies adversarial perturbations to prevent real-world face recognition platforms from correctly tagging a user's face (§ 6). We confirm Face-Off's effectiveness across three major commercial face recognition services: Microsoft Azure Face API, AWS Rekognition, and Face++.
3. We perform two user studies (with 423 participants) to assess the user-perceived utility of the images that Face-Off generates (§ 7).

[1] Approaches based on gradient-free optimization [14] are prohibitively expensive.

2 Background

This section describes the machine learning (ML) notation required in this paper. We assume a data distribution D over $\mathbf{X} \times \mathbf{Y}$, where $\mathbf{X}$ is the sample space and $\mathbf{Y} = \{y_1, \cdots, y_L\}$ is the finite space of labels. For example, $\mathbf{X}$ may be the space of all images, and $\mathbf{Y}$ may be the labels of the images.

Empirical Risk Minimization: In the *empirical risk minimization (ERM)* framework, we wish to solve the following optimization problem:

$$w^* = \min_{w \in H} \mathbb{E}_{(\mathbf{x},y)\sim D} \mathcal{L}(w, \mathbf{x}, y),$$

where H is the hypothesis space and $\mathcal{L}$ is the loss function (such as cross-entropy loss [21]). We denote vectors in bold (*e.g.,* $\mathbf{x}$). Since the distribution is usually unknown, a learner solves the following problem over a dataset $S_{train} = \{(\mathbf{x}_1, y_1), \cdots .(\mathbf{x}_n, y_n)\}$ sampled from D:

$$w^* = \min_{w \in H} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(w, \mathbf{x}_i, y_i)$$

Once the learner has solved the optimization problem, it obtains a solution $w^* \in H$ which yields a classifier $F : \mathbf{X} \to \mathbf{Y}$ (the classifier is usually parameterized by w^* *i.e.,*, F_{w^*}, but we will omit this dependence for brevity).

2.1 Metric Embeddings

A *deep metric embedding* f_θ is function from $\mathbf{X}$ to $\mathbb{R}^m$, where $\theta \in \Theta$ is a parameter chosen from a parameter space and $\mathbb{R}^m$ is the space of m-dimensional real vectors. Throughout the section, we sometimes refer to $f_\theta(\mathbf{x})$ as the *embedding of* $\mathbf{x}$. Let $\phi : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^+$ be a distance metric[2] on $\mathbb{R}^m$. Given a metric embedding function f_θ, we define $d_f(\mathbf{x}, \mathbf{x}_1)$ to denote $\phi(f_\theta(\mathbf{x}), f_\theta(\mathbf{x}_1))$. $[n]$ denotes the set $\{1, \cdots, n\}$.

Loss Functions: Deep embeddings use different loss functions than typical classification networks. We define two of these loss functions: contrastive and triplet.

For a constant $\gamma \in \mathbb{R}^+$, the *contrastive* loss is defined over the pair $(\mathbf{x}, y)$ and $(\mathbf{x}_1, y_1)$ of labeled samples as:

$$\mathcal{L}(\theta, (\mathbf{x}, y), (\mathbf{x}_1, y_1)) = \mathbb{I}_{y=y_1} \cdot d_f^2(\mathbf{x}, \mathbf{x}_1) + \mathbb{I}_{y \neq y_1} \cdot [\gamma - d_f^2(\mathbf{x}, \mathbf{x}_1)],$$

where $\mathbb{I}_E$ is the indicator function for event E (and is equal to 1 if event E is true and 0 otherwise).

The *triplet* loss is defined over three labeled samples– $(\mathbf{x}, y)$, $(\mathbf{x}_1, y)$ and $(\mathbf{x}_2, y_2)$, given a constant $\gamma \in \mathbb{R}^+$, as:

$$\mathcal{L}(\theta, (\mathbf{x}, y), (\mathbf{x}_1, y), (\mathbf{x}_2, y_2)) = [d_f^2(\mathbf{x}, \mathbf{x}_1) - d_f^2(\mathbf{x}, \mathbf{x}_2) + \gamma]_+,$$

where $[x]_+ = \max(x, 0)$ and $y \neq y_2$.

Inference: Let $A = \{(\mathbf{a}_1, c_1), \cdots, (\mathbf{a}_k, c_k)\}$ be a reference dataset (*e.g.,* a set of face and label pairs). Note that A is the dataset used during inference time and different from the dataset S_{train} used for training. Let $A_y \subset A$ be the subset of the reference dataset with label y (*i.e.,* $A_y = \{(\mathbf{a}_j, y) \mid (\mathbf{a}_j, y) \in A\}$). Additionally, we denote the centroid of set A_y as $\beta_{y,f} \in \mathbb{R}^m$. Formally, the centroid of label y is defined as follows:

$$\beta_{y,f} = \frac{1}{|A_y|} \sum_{(\mathbf{a}_i, y) \in A_y} f_\theta(\mathbf{a}_i)$$

If we have a sample $\mathbf{x}$ and a reference dataset A, let

$$j^* = \arg\min_{j \in [k]} \phi(\beta_{c_j,f}, f_\theta(\mathbf{x})).$$

We predict the label of $\mathbf{x}$ as c_{j^*}.

Recognition vs. Matching: For face recognition, the training set S_{train} corresponds to a large labeled dataset of individuals' faces. During inference, the face recognition service has access to a reference dataset A; these could correspond to tagged images on Facebook, for example. When a user uploads a new image, the service searches for the centroid that is closest to the image in the embedding space and returns the label corresponding to the centroid. In the face matching setting, the service provider receives two faces and returns the distance between them in the embedding space.

2.2 Attacks on Metric Embeddings

Attack Overview: We define two types of attacks on metric embedding networks: untargeted and targeted. In the formulations given above, we assume that $\mathbf{X}$ is a metric space with μ defined as a metric on $\mathbf{X}$ (*e.g.,* $\mathbf{X}$ could be $\mathbb{R}^2$ with μ representing ℓ_∞, ℓ_1, or ℓ_p norms for $p \geq 2$). δ is the perturbation we add to inputs from $\mathbf{X}$. We summarize the attacks below:

1. *Untargeted attack* on $\mathbf{x}$ can be described as follows:

$$\begin{gathered} \min_{\delta \in \mathbf{X}} \ \mu(\delta) \\ s.t.\ j^* \neq \arg\min_{j \in [k]} \phi(\beta_{c_j,f}, f_\theta(\mathbf{x} + \delta)) \end{gathered}$$

This attack aims to find a small perturbation that pushes the perturbed example's embedding to a closer centroid than the original one.

[2] For all definitions that follow, ϕ represents the 2-norm.

2. Targeted attack (with label $t \neq j^*$) can be described as follows:

$$\begin{gathered}\min_{\delta \in \mathbf{X}} \ \mu(\delta) \\ s.t. \ \arg\min_{j \in [k]} \phi(\beta_{c_j,f}, f_\theta(\mathbf{x}+\delta)) = t\end{gathered}$$

This attack aims to find a small perturbation that pushes the perturbed example's embedding to the centroid corresponding to a target label.

Approach Overview: Let $\mathbf{x}$ be a sample that we wish to perturb. Intuitively, an attack increases the distance between the perturbed sample's embedding (with y as the true label) and that of all those other samples with label y. Empirically, we found this objective to be stronger than just pushing an embedding of a perturbed sample away from the centroid $\beta_{y,f}$. Next, for a deep embedding f_θ and set A_y, define

$$d'_f(\mathbf{z}, A_y) = \frac{1}{|A_y|} \sum_{(\mathbf{a}_i, y) \in A_y} d_f(\mathbf{x}, \mathbf{a}_i)$$

Observe that $d'_f(\mathbf{x}, A_y)$ is a differentiable function of $\mathbf{x}$, and thus prior work (*e.g.,* FGSM [7]) can be used to generate the adversarial perturbation.

Concrete Formulation: For the untargeted case, we pose the attacker's optimization problem as:

$$\begin{gathered}\max_{\delta \in \mathbf{X}} \ d'_f(\mathbf{x}+\delta, A_y) \\ s.t. \ ||\delta||_p \leq \epsilon\end{gathered}$$

For targeted attacks, the adversary wishes to label the face as target t; we refer to the term $d'_f(\mathbf{x}+\delta, A_t)$ as the *target loss*. We define the following function $G(\mathbf{x}', t)$ (also known as *hinge loss*), where $\mathbf{x}' = \mathbf{x}+\delta$ is the perturbed sample as follows:

$$G(\mathbf{x}+\delta, t) = [d'_f(\mathbf{x}+\delta, A_t) - \max_{y \neq t} d'_f(\mathbf{x}+\delta, A_y) + \kappa]_+$$

where the margin κ denotes the desired separation from the source label's samples. Once $G(\mathbf{x}', t)$ is defined, we can adapt existing algorithms, such as Carlini & Wagner (CW) [9], to construct the perturbation. Thus, for targeted attacks, the adversary wishes solve the following optimization problem:

$$\begin{gathered}\min \ ||\delta||_p \\ s.t. \ ||\delta||_p \leq \epsilon; \\ G(\mathbf{x}+\delta, t) \leq 0\end{gathered}$$

Amplification: Amplifying a perturbation δ by $\alpha > 1$ as scaling δ with α. If the attack algorithm generates a perturbed sample $\mathbf{x}+\delta$, amplification returns $\mathbf{x}+\alpha \cdot \delta$.

3 Face-Off: Overview

We provide an overview of Face-Off, which aims to preserve visual privacy against social media platforms.

3.1 System and Threat Models

Online users upload photos of themselves (or others) to social media platforms (such as Facebook or Instagram). These platforms first utilize a *face detector* [22] to identify the faces in the photo, and then apply *face recognition* [23] to tag the faces. The face recognition module can employ (a) verification (*i.e.,*, determine whether an uploaded face matches a candidate person, tag, or label), or (b) top-1 matching (*i.e.,*, find the top candidate for a match given a set of candidates). In particular, these platforms have these two properties:

1. They use *proprietary, black-box models* for face recognition. The models are trained on *private datasets* using architectures or parameters which are not public.
2. They can process user-uploaded images of *varying sizes, resolutions, and formats.* The platform (with high probability) recognizes faces in all of them.

Upon tagging the people in the photo, the platform can perform additional inferences beyond the user's expectations [24]. For example, the platform can infer the behavior of specific people, the places they visit, the activities they engage in, and their social circles [2]. Additionally, these *labeled* photos can be scraped by various services and later used by various governmental agencies [1]. The prolonged analysis of user-uploaded images allows the platform (and other agencies that use these photos) to profile users, which enables targeted advertising [25], and lays the foundations for surveillance at the behest of a nation-state [26]. Thus, it is essential to safeguard the privacy of user-uploaded media, specifically images, from social network providers.

Actors: In this setting, we assume that the social network provider is an adversarial entity. The provider will analyze face tags to infer information about the user.

3.2 High-level Operation

Face-Off aims to minimally modify images that the user wishes to upload such that the cloud provider cannot correctly recognize their face. Based on insights from adversarial ML (and specifically evasion attacks [27] as

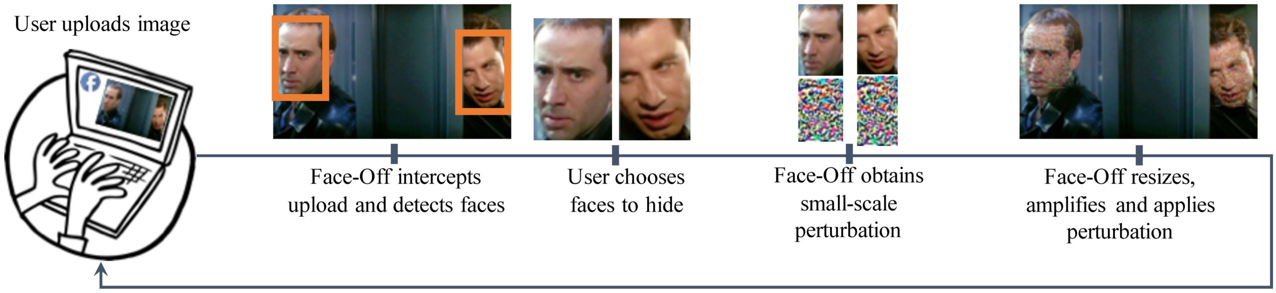

User uploads image
Face-Off intercepts upload and detects faces
User chooses faces to hide
Face-Off obtains small-scale perturbation
Face-Off resizes, amplifies and applies perturbation

Fig. 2. High-level overview of Face-Off's processing pipeline.

highlighted in § 2.2), Face-Off applies small perturbation to the user inputs such that they are misclassified by facial recognition. Face-Off sits between the user and the social network provider and generates pixel-level perturbations (or *masks*) to induce user-specified misclassifications. Face-Off operates as follows (Figure 2):

1. Face-Off detects and extracts faces from user-uploaded photos. Advances in deep learning have made this process very accurate. In our implementation, we utilize an MTCNN [28], which has a detection accuracy of 95%.
2. Our mask generation process requires inputs of a fixed size; Face-Off resizes the detected faces. This resizing process is also error-free.
3. Face-Off generates the mask for the resized face. We highlight the efficacy of this approach in § 6.
4. Finally, Face-Off adds the generated mask to the resized face and returns a resized perturbed image. Sometimes, the generated masks are amplified by a scalar constant α.

Observe that, apart from step 3, *none of the other steps in the mask generation process induces erroneous artifacts in our pipeline.* Note that inaccuracies in detection do not lead to errors, but leads to the inability to generate a mask. In our threat model, users do not upload images that cause failures in the face detector [6]. The procedure for mask generation (step 3) always runs to completion, and we study the impact of various hyper-parameter choices on the success of the approach in § 4.

Challenges: Achieving the functionality is challenging; we highlight several challenges below:

C1. Extensive work on evasion attacks (or generating adversarial examples) *focuses on classification.* However, Face-Off requires attacks for metric embeddings (refer § 2), which are *different.* Thus, new attack formulations, which include customized loss functions are required.

C2. Models used by the platform are black-box in nature, and there is a *lack of knowledge* of their internals. Most prior work on generating adversarial examples involves white-box access. To circumvent the issues associated with black-box access, we utilize *surrogate models* which we train. As these substitute models are similar to the proprietary models, we expect the generated adversarial examples to transfer [29].

C3. Since social media platforms are *capable of pre-processing* the images (through compression or resizing), it is essential that the perturbation generated for the scaled image transfers as well.

In the rest of the paper, we highlight how Face-Off overcomes these challenges. We reiterate that designing a system like Face-Off is a challenging proposition because it provides privacy at inference time. Our work makes no assumptions about the nature of the models used by social network providers nor the data or mechanism they use for training. Adversarial knowledge and control of the model, the mechanism in which it is trained, and training data allow for different attack strategies (based on data poisoning [30], for example). However, such threat models are unrealistic in practice.

We refer the curious reader to Appendix A.1 which highlights the theoretical intuition behind Face-Off.

3.3 Face-Off Design

Attacks: We design Face-Off by borrowing elements from two popular approaches: projected gradient descent (PGD) by Madry *et al.* [15], and the Carlini & Wagner (CW) approach [9]. As noted in § 2.2, we utilize two custom loss functions: (a) target loss and (b) hinge loss on surrogate model to which we have white-box access. Table 1 details the hyper-parameters of our attack implementations. In § 4, we describe in detail how (a) the choice of surrogate model, (b) choice of various attack hyper-parameters, and (c) choice of additional parameters such as amplification factor α and

Parameters	PGD	$\mathbf{CW}_\infty$	$\mathbf{CW}_{small}$	$\mathbf{CW}_{large}$
Pert. bound (ε)	0.1	-	-	-
Norm (p)	2	∞	2	2
Iterations (N)	20/200	100	100	800
Search Steps	-	10	8	15
Learning Rate (η)	0.1	0.1	0.1	0.1
Initial Const.	0.3	0.3	0.3	0.3
Hinge Loss	✓	✓	✓	✓
Target Loss	✓	✓	✓	✓

Table 1. Attack hyper-parameters

margin κ impact attack success. All our code is available at https://github.com/wi-pi/face-off.

Surrogate Models: We utilize two state-of-the-art face verification architectures: (a) the triplet loss architecture (*i.e.,* FaceNet [10]), and (b) center loss architecture (henceforth referred to as CenterNet [31]). For both architectures, we utilize the code and pre-trained weights from the original repositories, and convert all implementations to `keras` [32] to ensure compatibility with our perturbation generation framework (which was built using `tensorflow` [33] and the `cleverhans` library [34]). The original implementations (collectively referred to as *small* models) accept inputs of the shape $96 \times 96 \times 3$ and $112 \times 96 \times 3$ respectively. We trained another variant of both these models (collectively referred to as *large* models), using the procedures outlined in the original papers[3], to accept inputs of shape $160 \times 160 \times 3$. Salient features, including test accuracy on the Labeled Faces in the Wild (LFW) dataset [35], of these models are in Table 2. In all models, the 2-norm between embeddings is used as the distance function $\phi(.,.)$.

4 Parameter Choices

Recall that our objective is to (a) generate adversarial examples (or masked samples) on a local surrogate model to which we have white-box access, and (b) transfer these examples to the black-box victim models used by social network providers. We conduct a detailed analysis to understand black-box transferability for both targeted (Figure 3) and untargeted attacks (Figure 4).

Setup: We utilize images of celebrities, including some diversity in age, gender, and race. They are `Barack Obama, Bill Gates, Jennifer Lawrence, Leonardo DiCaprio, Mark Zuckerberg, Matt Damon, Melania Trump, Meryl Streep, Morgan Freeman, and Taylor Swift`. Thus, our experiments include images from 10 labels (totaling 90 source-target pairs) for *portrait* images alone. We consider 2 models (CSVC, FLVT from Table 2) to generate the adversarial examples using 2 different adversarial loss functions (target and hinge) for 3 attacks (ℓ_2 and ℓ_∞ variants of CW and an ℓ_2 variant of PGD, as defined in § 2.2). We evaluate 6 choices of margin κ (*i.e.,* $\kappa = 0, 5, 10, 11, 12, 13$), along with 40 choices of amplification factor α (*i.e.,* $\alpha \in [1, 8]$ at intervals of 0.2), running for 2 settings (few and many) in terms of the number of execution iterations N. The remaining hyper-parameters are as specified in Table 1. Due to space constraints, we only report the results using the FLVT model as the surrogate and the FLCT model as the victim. Results from other model combinations show similar trends.

Metrics: We measure the top-1 matching accuracy *i.e.,* given a set of candidate labels (all 10 in our case); a correct match is one where the distance to the correct label is the smallest (in the embedding space). For targeted attacks, we define the *success metric* (a value that lies in $[0, 1]$) as the ratio between the number of times an adversarial example matches the intended target (*i.e.,* attack success) and the number of tests (*i.e.,* number of attacks). The larger the *success metric*, the more effective is the attack. For untargeted attacks, we define the *success metric* as one minus the ratio of the number of times the label of the adversarial example is the true (source) label and the number of adversarial samples created. Again, the higher the success metric, the more successful is the attack. We detail the conclusions obtained from our ablation study below. To measure the dependence of one factor (say α or κ) on the success metric, we keep all other parameters fixed (unless explicitly stated otherwise) as specified in Table 1.

Description of Plots: Figures 3 and 4 highlight the impact of amplification (α) and margin (κ) on the success metric. Each point in the plot is the average of the success metric across all 90 source-target pairs used.

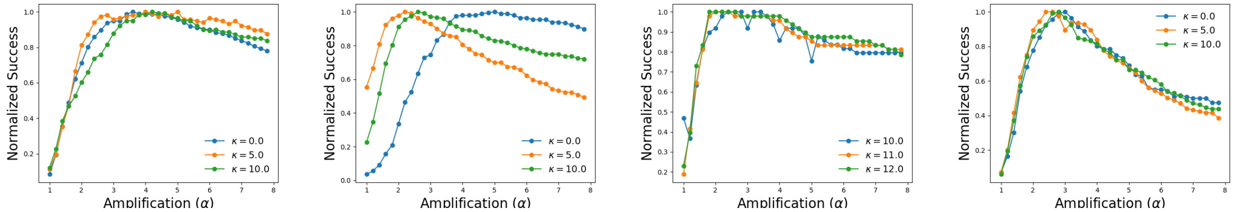
4.1 Choice of Attack

Our analysis clearly shows that the exact choice of attack (CW or PGD) does not significantly impact transferability. Figures 3b and 3d (as well as Figures 4b and 4d) show that, for a given pair of surrogate and

[3] FaceNet (large) was obtained from the official `github` repository, while CenterNet (large) was trained from scratch.

Abbreviation	Architecture	Dataset	Loss	Input Shape	Embedding Size	Test Accuracy
FSVT	FaceNet	VGGFace2	Triplet	$96 \times 96 \times 3$	128	99.65 %
CSVC	CenterNet	VGGFace2	Center	$112 \times 96 \times 3$	512	99.28 %
FLVT	FaceNet	VGGFace2	Triplet	$160 \times 160 \times 3$	128	99.65 %
FLVC	FaceNet	VGGFace2	Center	$160 \times 160 \times 3$	512	98.35 %
FLCT	FaceNet	CASIA	Triplet	$160 \times 160 \times 3$	512	99.05 %

Table 2. Salient features of the white-box models used for offline mask generation.



(a) CW$_{\infty}$ **(b)** CW$_{small}$ **(c)** CW$_{large}$ **(d)** PGD

Fig. 3. White-box results for the *targeted* top-1 attack with the FLVT model as the surrogate and the FLCT model as the victim.

victim models, the success metric is relatively comparable across attacks (in both the targeted and untargeted setting). The minor variations can be attributed to variations in hyper-parameters as specified in Table 1. This suggests that the loss functions proposed in § 2.2 enable successful transferability of generated adversarial examples (more than the exact attack).

ℓ_2 vs. ℓ_∞ attack: Given a fixed set of other execution parameters, the exact choice of norm does not impact attack success (as witnessed in Figure 3a and 4a). We conducted an (IRB approved) user study involving 50 participants (each shown 20 pairs of ℓ_∞ and ℓ_2 variants of CW-based masked samples) to determine if one type of attack was more favorable (due to the imperceptibility of the perturbation). The users were nearly undecided between the two conditions: out of the 1000 assessments, 468 favored the ℓ_∞ attack and 532 favored ℓ_2 attack. We could not reject the null hypothesis that both conditions are equally favorable to the users ($p = 0.15$).

Takeaway: Exact choice of attack or norm does not (greatly) influence transferability or perceptibility.

4.2 Amplification & Margin

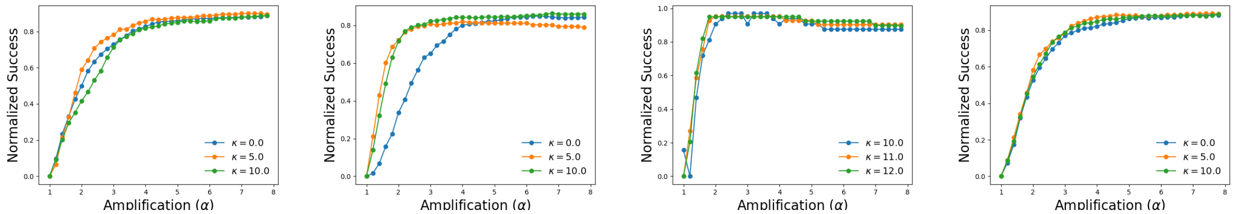
Across all attacks, we observe that success is directly correlated with increasing κ and α. This result holds regardless of the target model (upon which transferability is being measured), and is independent of the exact loss function used. It is, however, crucial to understand the difference between κ and α. The choice of κ has a direct impact on the run-time of the approach. However, amplification is post-processing applied to the generated images (and is off the critical path). Thus, one can suitably compensate for low κ by increasing α, and improve the run-time of the approach. From our results in Figures 3 and 4, we observe that α more directly influences the transferability in comparison to κ. In particular, we observe that values of $\kappa \geq 5$ and $\alpha \geq 2$ are ideal for transferability.

Takeaway: α influences transferability more than κ; smaller values of κ are preferred to reduce the run-time.

4.3 Number of Iterations

We now focus on the number of iterations N and its impact on success. We only consider the CW attack for the 2-norm. We plot the impact of κ and α on success across two trails: the first with a fewer number of iterations ($N = 100$) in Figures 3b and 4b, and the second with more iterations ($N = 800$) in Figures 3c and 4c. We observe that increasing N results in increased transferability (and this holds with increasing κ and α). We note, however, that increasing N is a time-consuming process as it lies on the critical path.

Takeaway: Although increasing N increases the run-time, it improves transferability.



(a) CW$_\infty$ **(b)** CW$_{small}$ **(c)** CW$_{large}$ **(d)** PGD

Fig. 4. White-box results for the *untargeted* top-1 attack with the FLVT model as the surrogate and the FLCT model as the victim.

API	Confidence	Threshold (τ)	Cost
Azure Face	$[0, 1]$	0.5	$0.001
Rekognition	$[0, 100]$	50	$0.001
Face++	$[0, 100]$	Dynamic	Free

Table 3. Online API black-box models

5 Evaluation Setup

In § 3.3, we detailed how to construct the surrogate models and other attack details. Here, we describe the commercial victims (§ 5.1) and our evaluation setup (§ 5.2).

5.1 Victim Models

We evaluate Face-Off using 3 popular commercial recognition APIs: (a) Azure Face API [18], (b) Face++ [20], and (c) Amazon Rekognition [19]. These APIs accept two images as input (henceforth referred to as one query). They return a confidence value indicating how similar these images are, and a matching threshold τ (the two images correspond to the same face when the confidence value is above the threshold). The salient features of these APIs are available in Table 3. To ensure a consistent comparison, all confidence scores are normalized to values in $[0, 1]$, and the threshold $\tau = 0.5$.

5.2 Experimental Setup

Our experiments were carried out on a server with 2 Titan XPs, 1 Quadro P6000, 40 CPU cores, 125 GB of memory, and ran Ubuntu 16.04 LTS.

1. Requirements: We choose to generate adversarial examples for celebrity faces (as in § 4) due to their vast availability on the public internet. For each celebrity image $\mathbf{x}$ (whose label is referred to as the *source* label s), we need to obtain (a) a corresponding *target* label t, (b) a set of source images to calculate $d'_f(\mathbf{x}, A_s)$, and (c) a set of target images to calculate $d'_f(\mathbf{x}, A_t)$ (§ 2.2).

2. Processing Pipeline: The processing pipeline is detailed in § 3.2. The cropped faces are downsized accordingly using bilinear interpolation of `OpenCV`[4], and used to obtain adversarial perturbations. We refer to these perturbed images as *cropped* images. Face-Off extracts the perturbation mask, resizes it, and applies it to the original subject after amplification. We refer to these perturbed images (after upsizing) as *uncropped* images.

3. Measuring Transferability: A successful attack is one where the sample is misclassified based on the top-1 accuracy. This is the case since we compare the adversarial image with the source image (and the original label) from which it was generated. In § 6.3, we discuss top-n matching accuracy for the targeted attacks.

4. Understanding the Plots: For all results presented in § 6, we plot the transferability (measured by the confidence value returned by the corresponding APIs) against the norm of the perturbation (*i.e.*, $||\alpha \cdot \delta||_{p=2,\infty}$) for varying values of κ and ε (as in the case of PGD). We intend to highlight how the perceptibility of the perturbation (observed with increasing norm) impacts transferability. We also assess if both cropped uncropped faces transfer to the victim model. For brevity, we only plot the results using uncropped images. Results related to cropped images can be found in Appendix A.5.

6 Evaluation

Our evaluation is designed to answer the following questions (and we provide our findings as responses):

Q1. Do the generated adversarial examples transfer to commercial black-box APIs specified

[4] https://docs.opencv.org/2.4

in § 5.1? Using the 2-norm variants of PGD and CW, we are able to successfully transfer the generated adversarial examples to all 3 commercial APIs (refer § 6.1).

Q2. Can commercial APIs deploy defense mechanisms (such as adversarial training) to safeguard themselves against masked samples? Based on a longitudinal study (*across two years*), we observe that commercial APIs have not implemented defense mechanisms to safeguard against adversarial inputs. We also observe that adversarial training induces a substantial decrease in recognition accuracy (§ 6.2). Further, we show that using adversarial training [15] as a defense mechanism deteriorates natural accuracy. When evaluation is performed using a subset of VGGFace2, accuracy drops by 11.91 percentage points (Table 4).

Q3. If transferability is measured by the top-n accuracy instead of the top-1 accuracy, how effective are the generated masked samples? We observe that, for targeted masked samples, the top-3 accuracy is higher than the top-1 accuracy. However, increasing α decreases this value as well (§ 6.3).

6.1 Transferability to Commercial APIs

Due to space constraints, we report results only for the ℓ_2 variants of CW and PGD and for the uncropped setting; the cropped setting displays similar trends as evident from Appendix A.5. We use the CSVC model (Table 2) as the surrogate to generate the adversarial samples. We fix the number of iterations to 500 iterations for the CW attack and 50 iterations for the PGD attack; other hyper-parameters are in Table 1. The experimental parameters are as follows: we varied κ (for CW) and ε (for PGD) in $[0, 5.8]$ at increments of 0.2, and varied α from $[1, 5]$ at increments of 0.1. The results are obtained using images of `Matt Damon` as the source and `Leonardo Di Caprio` as the target, and all experiments were carried out in August 2018. Across all plots, lower confidence values correspond to better privacy gains.

CW: Consistent with § 4.2, transferability increases as amplification increases, and we observe transferability starting at $\|\alpha \cdot \delta\|_2 \approx 6$ for Azure (Figure 5a), $\|\alpha \cdot \delta\|_2 \approx 4.5$ for Rekognition (Figure 5b), and $\|\alpha \cdot \delta\|_2 \approx 12$ for Face++ (Figure 5c). The slope decreases beyond a specific point across all three models, suggesting that increasing the amplification factor α will only produce marginal privacy gains. We also observe limited correlation between the value of κ and matching confidence. This observation suggests that the choice of α is more relevant for transferability than the choice of κ. Observe, however, that for a given value of α, the transferability varies for different APIs.

PGD: Here, the choice of ε has a greater impact on the rate of transferability (*i.e.,* how quickly, in terms of amplification, the confidence value reaches the $\tau = 0.5$). The larger values of ε correspond to faster transferability (steeper slope), as highlighted across all APIs in Figure 7. As before, transferability across different models requires different levels of $\|\alpha \cdot \delta\|_2$ (consequently α). However, unlike images generated by the CW attack, the perturbation is larger for those generated by PGD.

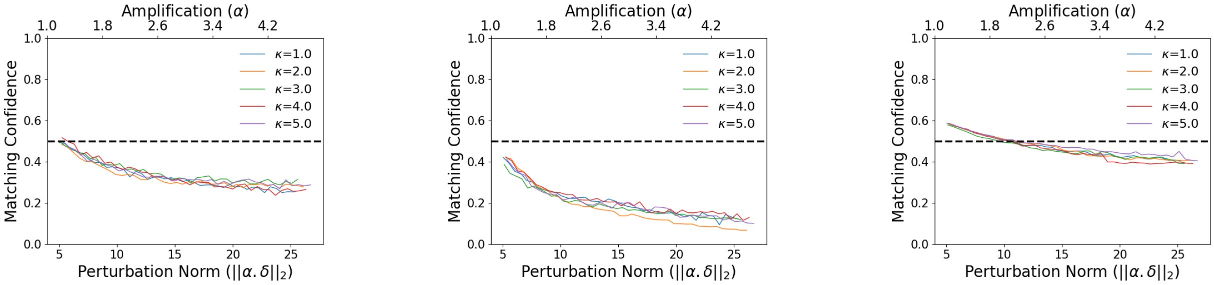
6.2 Measuring API Robustness

Longitudinal Study: We conducted a longitudinal study to verify if commercial APIs have improved their robustness against adversarial samples. For this experiment, we utilize the images generated in August 2018 (from § 6.1) and verify if they transfer to the APIs in August 2020, *two years after initial testing.* We present the results in Figures 6 and Figures 8 (for uncropped images modified using the exact same configuration of CW and PGD as in § 6.1). We observe that transferability persists across all 3 APIs (to variable degrees) despite the passage of time and potential retraining conducted by the API providers. The trends observed in § 6.1 still hold. This fidning suggests that APIs *have not deployed mechanisms* to provide adversarial robustness.

Dataset	Base (ℓ_2)	AT (ℓ_2)	Base (cos)	AT (cos)
LFW [35]	52.57%	38%	53.54%	38.47%
VGGFace2 [36]	60.13%	48.22%	60.60%	48.25%
Celeb [37]	82.29%	80.41%	83.39%	81.03%

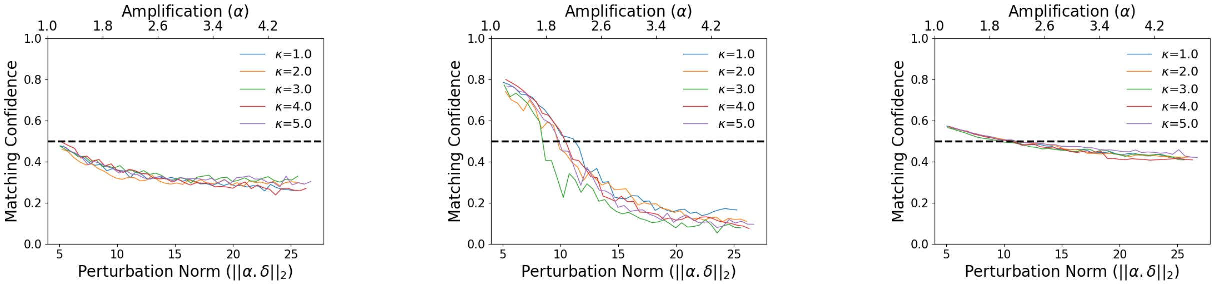
Table 4. Top-1 accuracies after adversarial training. Note that adversarial training decreases top-1 accuracy. **Base** refers to the baseline natural accuracy (before adversarial training), and **AT** refers to the natural accuracy after adversarial training. ℓ_2 denotes the 2-norm, and cos denotes the cosine similarity measure.

Adversarial Training: A curious reader may wonder if adversarial training [15] can safeguard APIs against adversarial inputs, such as those generated through Face-Off. It is well understood that, in classification settings, adversarial robustness is achieved at the expense of natural accuracy [38]. To empirically validate this observation in the context of metric embeddings, we train a variant of the FLVC model both naturally and adversar-



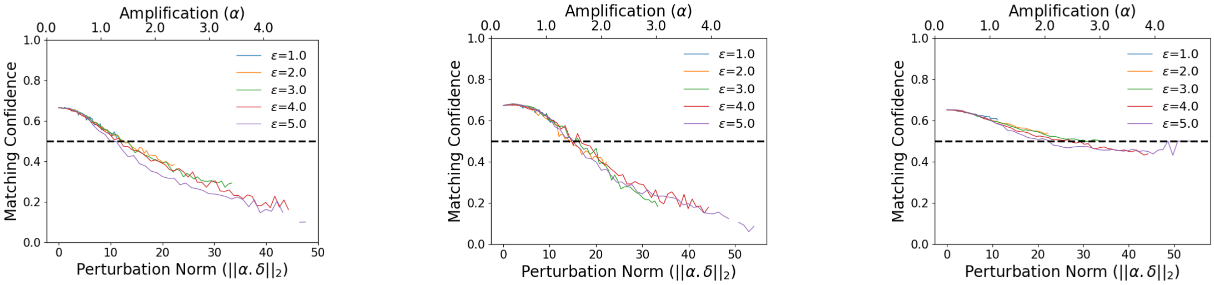
(a) Azure **(b)** AWS Rekognition **(c)** Face++

Fig. 5. In *2018*: Transferability of uncropped images generated using CW attack



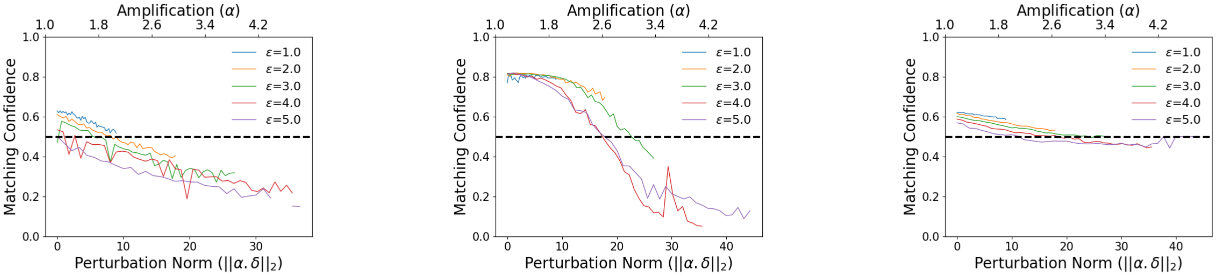
(a) Azure **(b)** AWS Rekognition **(c)** Face++

Fig. 6. In *2020*: Transferability of uncropped images generated using CW attack



(a) Azure **(b)** AWS Rekognition **(c)** Face++

Fig. 7. In *2018*: Transferability of uncropped images generated using PGD attack



(a) Azure **(b)** AWS Rekognition **(c)** Face++

Fig. 8. In *2020*: Transferability of uncropped images generated using PGD attack

ially for 3000 epochs on a subset of 50 labels (*i.e.,* 23435 images) sampled from the VGGFace2 dataset [36]. We trained the model with a mixture of 50% adversarial examples and 50% natural examples. We generated adversarial examples using the ℓ_2 variant of CW (and the hinge loss) with $\kappa = 6$ and $\alpha = 3$ on half of the labeled images in our training set. The robustness results are presented in Table 4. The **Base** columns refer to the *natural accuracy* (*i.e.,* accuracy on clean/non-adversarial samples) when the model is trained naturally. The **AT** columns refer to the natural accuracy of the model when the model is trained adversarially. For natural accuracy, using three datasets for inference, we can observe that *in all cases*, the top-1 accuracy (measured using both 2-norm and cosine similarity) decreases with adversarial training. Thus, our findings are consistent with prior work in the classification regime [15]. When we test the *adversarial accuracy* (*i.e.,* accuracy on adversarial samples) of the model using adversarial samples generated from VGGFace2, we observe an increase from 41.49% to 48.59% (when top-1 accuracy is calculated using the 2-norm) and from 41.70% to 48.40% (when top-1 accuracy is calculated using the cosine similarity measure)[5]. We report additional details and results in Appendix A.2.

6.3 Top-n Recognition

Finally, we study the efficacy of our targeted attack if the adversary uses top-n recognition instead of top-1 recognition. We generated masked inputs using (a) the FLVT model as the surrogate, (b) the hinge loss, and (c) the CW$_{large}$ attack for 6 of the 10 labels described earlier. We consider all pair-wise combinations for the following identities: `Matt`, `Leonardo`, `Barack`, `Morgan`, `Melania` and `Taylor`. Table 5 contains the results of top-3 recognition success for Face++, Azure, and AWS Rekognition (in sequence). Note that a success event is one where the correct label is in the top 3 labels predicted for the input. Each entry in the table is the average of the 30 pairings we consider. We observe that across all the APIs: (a) increasing κ increases attack success (*i.e.,*, decreases accuracy), and (b) increasing α also increases attack success (*i.e.,*, decreases accuracy). Note that the trends we observe are consistent independent of the exact attack we use (as denoted by results for the ℓ_2 variant of PGD using the FLVT model in Table 6), or the exact surrogate model–though they impact the magnitude of attack success.

7 User Study

Thus far, we have studied the efficacy of our approach on transferability. In this section, we check if the perturbed (and amplified) images are *user-friendly*, *i.e.,* if the users are willing to upload such images to social media platforms. In the first study (§ 7.1), users see images amplified by different values of α and are asked if they would upload said images. In the second study (§ 7.2), users upload images of their choice to Face-Off's online service. After the service returns a perturbed image, we ask the user to ascertain the utility of this image. Both studies are approved by our IRB and are conducted on Amazon's MTurk platform. The main difference between the two studies is the control condition. The first study controls the images to assess the user's perception of the perturbation at the potential cost of the results' ecological validity. The second study allows users to upload images of their choice to get more realistic assessments.

7.1 Perturbation Tolerance

We conducted the first user study to assess *user-perceived utility* of the images that Face-Off generates. We performed this assessment for samples generated using ℓ_2 variants of both CW and PGD attacks (as described in § 6.1). Through this study, we aim to understand the amount of perturbation users are willing to tolerate. We consider two types of images: (a) *portrait*, where the face is the focus of the image (Figure 9), and (b) *background*, where the background is the focus of the image, and the face is in the image (Figure 10). We summarize our findings below:

- Privacy-conscious individuals are willing to tolerate larger perturbation levels for improved privacy.
- For the portrait image, 40% of respondents had no problem uploading a perturbed image to the social media platform (the exact tolerable perturbation level, however, differed among respondents).
- For the background image, the vast majority of users exhibited tolerance to higher image perturbation.

[5] These results are not presented in Table 4.

	α					α					α				
κ	1.8	3.4	5.0	6.6	8.2	1.8	3.4	5.0	6.6	8.2	1.8	3.4	5.0	6.6	8.2
0	96%	78%	54%	40%	22%	78%	34%	16%	12%	6%	95%	75%	55%	40%	35%
5	100%	74%	58%	34%	24%	80%	40%	20%	6%	4%	95%	75%	50%	35%	35%
10	96%	74%	46%	32%	24%	76%	30%	8%	6%	4%	95%	65%	45%	35%	35%

Table 5. Top-3 recognition accuracy (using the 2-norm) for Face++, Azure, AWS Rekognition respectively. Masked samples were generated using the FLVT model as the surrogate and the CW$_{large}$ attack (refer Table 1).

	α					α					α				
κ	1.8	3.4	5.0	6.6	8.2	1.8	3.4	5.0	6.6	8.2	1.8	3.4	5.0	6.6	8.2
0	95%	70%	15%	0%	0%	75%	20%	5%	0%	0%	90%	75%	60%	40%	10%
5	95%	65%	20%	0%	0%	75%	15%	5%	0%	0%	85%	75%	70%	20%	0%
10	80%	55%	0%	0%	0%	70%	10%	0%	0%	0%	85%	75%	60%	40%	10%

Table 6. Top-3 recognition accuracy (using the 2-norm) for Face++, Azure, AWS Rekognition respectively. Masked samples were generated using the FLVT model as the surrogate and the PGD attack (refer Table 1).

7.1.1 Study Design

Participant Recruitment: We recruited a total of 167 and 163 Amazon MTurk *master* workers for the portrait and background study, respectively. With this number of users, each image in the portrait study received at least five ratings, and each image in the background study received 163 ratings. Each worker was compensated \$1 for their effort, with an average completion time of 6 minutes. We present the demographics of the participants in Table 8 in Appendix A.3.

Study Protocol: We asked the user to rate a different image every time. For each participant in the portrait group, we display (a) 20 random images (each on a different page) where $\alpha \in [1.4, 2.4]$. For participants in the background group, we show the same 20 images where $\alpha \in [1.4, 2.4]$. We choose this range as it enables (some degree of) transferability (as witnessed in Figures 5 and 6).

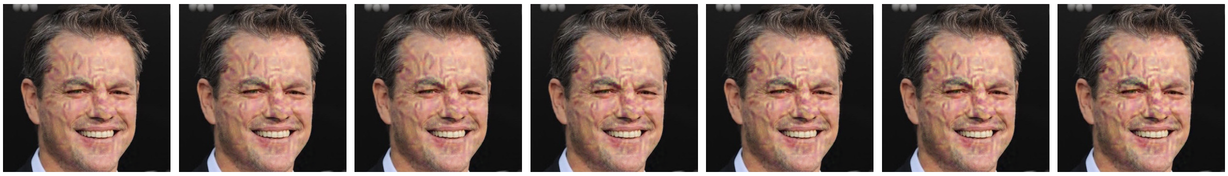
After rating the images, we asked the respondents a set of four questions on a 5-point Likert scale to gauge their privacy concern levels. We utilize the “Concern for Privacy" [39] scale which is modeled after the well-known “Information Privacy Concern scale" of Smith *et al.* [40]. We use this set of questions to divide the respondents into two groups: (a) *Privacy Conscious (PC)*, and (b) *Not Privacy Conscious (NPC)*. Respondents belonging to the first group are those who have answered all four questions with either `Strongly Agree` (SA) or `Agree` (A). The second group of respondents is those who responded with `Neutral` (N), `Disagree` (D), or `Strongly Disagree` (SD) any of the questions. Finally, we require the respondents to answer a set of general demographic questions. The respondents were made aware that these questions are optional and require no personally identifiable information.

Image Evaluation: For each displayed image, we asked the respondents to answer the following questions: (a) “*I have no problem in uploading this photo to social media*", and (b) “*I would upload the image to social media to prevent automatic tagging of my face*" on a 5-point Likert scale.

7.1.2 Results

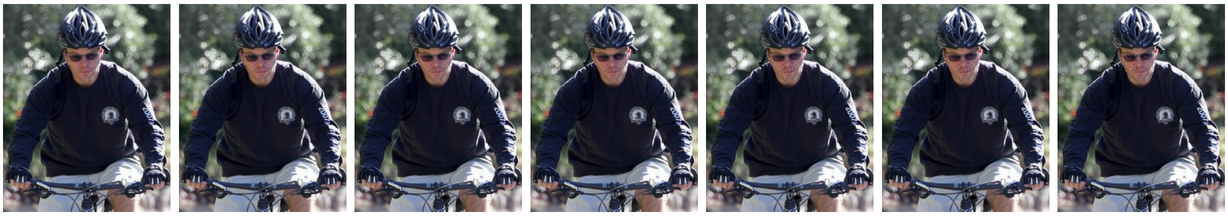
1. Portrait Images: Each user was shown a set of images with a corresponding α unknown to the user. We grouped responses into 5 buckets corresponding to the ranges $[1.4, 1.6), [1.6, 1.8), [1.8, 2), [2, 2.2), [2.2, 2.4]$. These five buckets represent increasing levels of perturbation.

We first discuss results for the NPC category (Figure 11a); as α increases, the number of users who do not wish to upload the image (the SD category *i.e.,* the last column) also increases. The inverse is also true; as the perturbation is lower, the number of participants who wish to upload the image is higher (*i.e.,* the first column). Similar observations can be made for the PC category (Figure 11b). Combining both groups, we found that 40% of the respondents are impartial to uploading at least one of the perturbed images. It is worth noting that portrait images are a unique case, where the subject is the highlight of these images, with a minimal background context (Figure 11). Thus, the perturbation is more explicitly visible on the user’s face, accounting for the received responses.



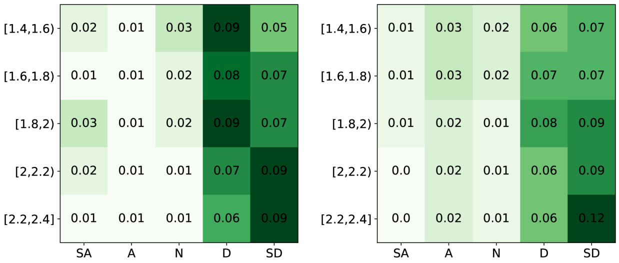
(a) $\alpha = 1.6$ (b) α =1.7 (c) α =1.8 (d) α =1.9 (e) α =2.0 (f) α =2.1 (g) α =2.2

Fig. 9. Photos used in the user study for portrait case with the perturbation increasingly amplified.



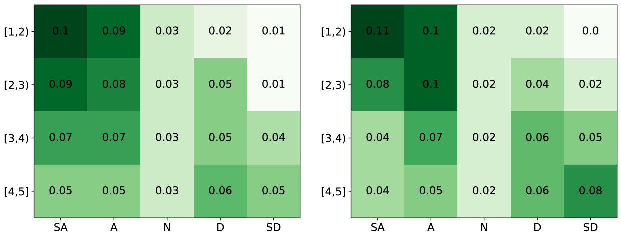
(a) Original (b) $\alpha = 1$ (c) $\alpha = 1.2$ (d) $\alpha = 1.4$ (e) $\alpha = 1.6$ (f) $\alpha = 1.8$ (g) $\alpha = 2$

Fig. 10. Photos used in the user study for the background case with the perturbation increasingly amplified. The perturbation is focused on the face region.



(a) Not Privacy Conscious (b) Privacy Conscious

Fig. 11. The distribution of responses for the **portrait** scenario for the PC and NPC groups. Each cell value contains the portion of responses for a specific α range and user satisfaction value.



(a) Not Privacy Conscious (b) Privacy Conscious

Fig. 12. The distribution of responses for the **background** scenario for the PC and NPC groups. Each cell value contains the portion of responses for a specific α range and user satisfaction value.

Nevertheless, we observe that the PC respondents are slightly more inclined to accept perturbed images than NPC users. This observation is evident from comparing Figure 11a and Figure 11b, where the latter figure shows a higher density of responses in the SA-A region. Further, we find the user perception to be dependant on privacy consciousness in all five amplification ranges ($p < 0.05$ according to the χ^2 test after applying the Holm-Bonferroni [41] method to correct for multiple comparisons).

2. Background Images: Figure 12 shows the user responses for background images to be far more favorable than the portrait case. As evident from Figure 10, the face constitutes a small region of such images, with other relevant features. Thus, resizing, amplifying, and adding the adversarial perturbation does not make as noticeable a difference as in the portrait image case. Except for the last range of α values, most of the respondents agreed to upload the perturbed image to the social media platform. In the first three ranges of α, we did not observe privacy consciousness to be a factor in the respondents' answers ($p > 0.05$ according to the χ^2 test after applying the Holm-Bonferroni method to correct for multiple comparisons). In one case, one of the respondents indicated that they do not observe any difference in the images and wondered whether we were testing respondents by showing the same image for every question. The only exception was the last α range, where large image perturbation is high enough to be unacceptable to our PC respondents.

Finally, the choice of the image exhibits a clear distinction in the user's perception of the perturbation

($p = 0$ according to the χ^2 test when comparing background and portrait responses over the same ranges of α). This distinction holds for all respondents. Users are typically more interested in preserving their privacy in situations related to a certain activity, behavior, or social context [42]. A portrait image contains little context about user activity or behavior. On the other hand, background images contain more context related to user activity, behavior, location, and social circles. For these images, users have a high incentive to avoid being automatically tagged and tracked by social media platforms.

In summary, we observed that Face-Off helps balance the privacy-utility trade-off of users. Most of the respondents have no problem uploading the background image, regardless of their privacy stance. Even for portrait images, a part of PC respondents accepted uploading perturbed images.

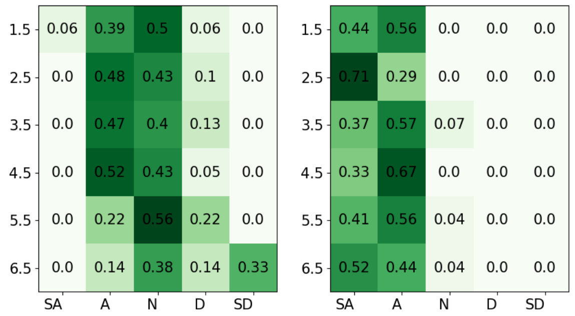
7.2 End-to-End Usability

While the first study suggests that privacy-conscious users are willing to upload perturbed images, the images themselves were not relevant to the users. We address this shortcoming through another user study with a more realistic setting, where users upload images to Face-Off's online portal. This study design improves the ecological validity from the first study as we show the users perturbed versions of images relevant to them. We describe the specifics of the study below.

7.2.1 Study Design

Participant Recruitment: We recruited a total of 100 Amazon MTurk workers. Each worker was compensated \$2 for their effort, with an average completion time of 10 minutes. After filtering responses that fail our attention checkers, we report the results based on 93 participants. We present the demographics of the participants in Table 8 in Appendix A.3.

Study Protocol: We used a between-subject study by asking each user to first upload an image of their choice to our portal (Appendix A.6). The only requirements were to ensure that each image contained a person/people of significance, and the faces of these people were easily identifiable. The uploaded face is compared with a similar target identity (in the embedding space), which is used for the attack. The service randomly assigns the user a value of α (κ is fixed to be 10), and proceeds to return a perturbed variant of the user-uploaded image. We avoided explicit mentions of privacy in the survey's introduction to reduce priming effects. Then, the user answers the same questions as in § 7.1–to understand if they are tolerant of the perturbation and willing to upload the masked input, and to ascertain their privacy preference. Similar to the previous study, participants were grouped into two categories: *Privacy Conscious (PC)* or *Not Privacy Conscious (NPC).*



(a) Not Privacy Conscious **(b)** Privacy Conscious

Fig. 13. The distribution of responses for the scenario where the users upload images to be be perturbed. Each cell value contains the portion of responses for a specific α value and user satisfaction value.

7.2.2 Results

The distribution of individual responses from this study can be found in Figure 13a for PC participants, and Figure 13b for NPC participants. For the PC group, we found that the value of α has little effect on the user's decision to upload the image. For all the values of α, nearly all the users in the PC group agree to upload the perturbed image to a social media platform. For the NPC, we observe a shift in the users' decision; these users are likely to disagree to upload the perturbed photos for larger values of α. Still, for this group, most participants fall within the neutral and agree categories, indicating the acceptability of the perturbations.

We observe that the user responses in the second study followed a similar trend to those in the background scenario (Figure 12 from § 7.1). While we did not have access to the uploaded photos for privacy reasons[6], we conjecture that the users uploaded images feature

[6] Photos were immediately deleted after being processed by our online service.

other objects as well as faces, similar to the background case of § 7.1.

8 Discussion

Here, we state some observations and limitations of Face-Off. We stress that the findings we describe below are not conclusive and simply mirror our experiences with experimentation with various datasets, compression strategies, facial recognition models, and online APIs.

8.1 Observations

First, we observed that gender and race appear to play an important role in determining the target label should one use the targeted attack. The embeddings of people belonging to the same gender or race are closer in the embedding space (across all surrogate models we use). Also, despite extensively perturbing particular identities (using large values of α), such as those belonging to minorities, these perturbations do not transfer to the cloud APIs. This observation may suggest bias in the training data used by both victim and surrogate models, and has been thoroughly investigated in prior work [43].

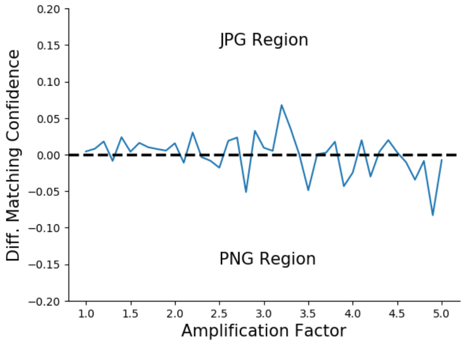


Fig. 14. Masked samples survive compression.

Second, we observed that our masked samples survive lossy compression at the expense of modest losses in privacy (reduction of matching confidence). In Figure 14, we plot the difference in confidence values (on the Azure Face API) between a test image and the generated masked sample when the masked sample is passed as a PNG (lossless compression) and a JPG (lossy compression). The upper half of the plot are regions where the JPG has lower confidence than the PNG image, and vice-versa. We observe that the magnitude of the difference in confidence values is minimal, suggesting that the choice of compression standard does not impact the results.

8.2 Limitations

We highlight some limitations of Face-Off, which we hope to address in future work.

1. Like other black-box attack schemes [7, 13, 17, 29, 44], our approach does not provide guarantees on transferability. Even with high values of α, the masked samples may still not transfer. However, our approach always enhances privacy if privacy were to be measured by the decreasing confidence with which these metric networks are able to match faces. In addition, amplification offers users the flexibility to balance the trade-off between privacy and utility; PC users can increase amplification and obtain greater privacy at the expense of more visible perturbation.
2. While the time required for generating the masked sample can potentially bottleneck real-time image upload (refer Appendix A.4), we envision alternate deployment strategies, such as offline masked sample generation, to circumvent this bottleneck.
3. A scenario that Face-Off cannot circumvent is when other people on social media platforms tag faces. This provides the social media platform an additional signal (and some feedback) to fix incorrect predictions.
4. The most significant limitation of our work, similar to all other adversarial example generation strategies, is the ever-improving robustness of black-box models [45]. However, as we show in § 6.2, increasing robustness is at the expense of natural accuracy.
5. In practice, determining the right surrogate model to use to maximize transferability is a challenging problem. Right now, we exhaustively try all candidates. The same can be said for choosing the *optimal* target label.

9 Related Work

We discuss relevant work below. These can broadly be classified as work related to generating adversarial examples in a black-box setting, and work designed to preserve privacy in online platforms.

9.1 Black-Box Attacks

Prior work demonstrate that some adversarial examples generated for one model may also be misclassified by another model [7, 13, 29]. For example, Papernot *et al.* [13], propose a strategy of training a local model (as a surrogate) using synthetically generated inputs. The victim DNN labels these inputs. The adversarial examples generated by the surrogate are shown to be likely misclassified by the target DNN as well.

Another line of work does not utilize the black-box models for the example generation process *i.e.,* the black-box model is never queried; work from Moosavi-Dezfooli *et al.* showed the existence of a universal perturbation for each model which can transfer across different images [46]. Tramer *et al.* conducted a study of the transferability of different adversarial instance generation strategies applied to different models [44]. The authors also proposed to use an ensemble of multiple models to generate adversarial examples to obtain increased transferability [47]. In a similar vein, Rajabi *et al.* [48] propose an approach to generate a universal perturbation (generated in a black-box manner) to be applied to all images. Finally, Sabour *et al.* [49] propose an approach to generate embedding perturbations, but in the white-box setting.

9.2 Privacy

Prior to our work, initial explorations have been made to utilize adversarial examples for protecting visual privacy [50]. Raval *et al.* developed a perturbation mechanism that jointly optimizes privacy and utility objectives [51]. Targeting face recognition systems, Sharif *et al.* developed a physical attack approach [4, 5]. The proposed algorithm first performs an adversarial attack on digital face images and limits the perturbation to an eyeglass frame-shaped area. Then the adversarial perturbation is printed into a pair of physical eyeglasses and can be worn by a person to dodge face detection, or to impersonate others in these face recognition systems. Being able to bring an adversarial attack into the physical world, this approach preserves visual privacy against face recognition. Additional prior work [52, 53] operate in the classification setting, where the loss objective (and attack formulation) are different from ours. Work by McPherson *et al.* [54] is a solution to an orthogonal problem, where the perturbations added are structured and human perceptible.

The work of Bose *et al.* [6] attempts to induce failure events given black-box access to *facial detectors.* Given white-box access to a face detector, the proposed scheme trains a generator against it for a given image. The generated adversarial perturbation aims to dodge the face detector so that the faces are not detected. Concurrent work from Shan *et al.* [55] explores the same problem. Using data poisoning attacks, they obfuscate faces at high success rates. In doing so, their approach could have an impact on large face recognition models trained using public images. However, their threat model differs in that they rely on online services using user data *to train* their deep learning models. However, social media platforms may opt to use a pre-trained model for face recognition tasks to avoid retraining on potentially malicious images. Face-Off operates at test time, so it transfers *regardless of the platform's training policy.*

10 Conclusion

In this paper, we present Face-Off, a system designed to preserve user privacy from facial recognition services. We design two new loss functions to attack metric learning systems, and extensively evaluate our approach using various models, architectures, parameters, and hyper-parameters across three commercial face recognition APIs. Our results affirm the utility of Face-Off. Through our evaluation, we observe several artifacts that suggest training dataset, and algorithmic bias against specific sub-populations.

11 Acknowledgements

The authors would like to thank the reviewers for their constructive reviews. Suman Banerjee, Varun Chandrasekaran, and Chuhan Gao are/were supported in part through the following US NSF grants: CNS-1838733, CNS-1719336, CNS-1647152, CNS-1629833, and CNS-2003129. Kassem Fawaz and Somesh Jha are supported by DARPA under agreement number 885000.

References

[1] Techcrunch, "Clearview said its facial recognition app was only for law enforcement as it courted private companies," https://techcrunch.com/2020/02/27/clearview-facial-recognition-private-companies/.

[2] S. Ahern, D. Eckles, N. S. Good, S. King, M. Naaman, and R. Nair, "Over-exposed?: Privacy patterns and considerations in online and mobile photo sharing," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2007.

[3] R. Gross, L. Sweeney, J. Cohn, F. De la Torre, and S. Baker, "Face de-identification," in *Protecting privacy in video surveillance*. Springer, 2009, pp. 129–146.

[4] M. Sharif, S. Bhagavatula, L. Bauer, and M. 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*. ACM, 2016, pp. 1528–1540.

[5] ——, "Adversarial generative nets: Neural network attacks on state-of-the-art face recognition," *arXiv preprint arXiv:1801.00349*, 2017.

[6] A. J. Bose and P. Aarabi, "Adversarial attacks on face detectors using neural net based constrained optimization," *arXiv preprint arXiv:1805.12302*, 2018.

[7] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

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

[10] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[11] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.

[12] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2017.

[13] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against deep learning systems using adversarial examples," *arXiv preprint*, 2016.

[14] P. Zhao, S. Liu, P.-Y. Chen, N. Hoang, K. Xu, B. Kailkhura, and X. Lin, "On the design of black-box adversarial examples by leveraging gradient-free optimization and operator splitting method," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 121–130.

[15] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[16] X. Cao and N. Z. Gong, "Mitigating evasion attacks to deep neural networks via region-based classification," in *Proceedings of the 33rd Annual Computer Security Applications Conference*. ACM, 2017, pp. 278–287.

[17] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," *arXiv preprint arXiv:1611.02770*, 2016.

[18] Microsoft, "Azure face api," https://azure.microsoft.com/en-us/services/cognitive-services/face/.

[19] Amazon, "Aws rekognition," https://aws.amazon.com/rekognition/.

[20] Face++, "Face++," https://www.faceplusplus.com.

[21] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Advances in neural information processing systems*, 2018, pp. 8778–8788.

[22] R. Feraund, O. J. Bernier, J.-E. Viallet, and M. Collobert, "A fast and accurate face detector based on neural networks," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 1, pp. 42–53, 2001.

[23] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition*. IEEE Computer Society, 1991, pp. 586–587.

[24] Techcrunch, "Facebook faces fresh criticism over ad targeting sensitive attributes," https://techcrunch.com/2018/05/16/facebook-faces-fresh-criticism-over-ad-targeting-of-sensitive-interests/.

[25] Wired, "Facebook's targeted ads are more complex than it lets on," https://www.wired.com/story/facebooks-targeted-ads-are-more-complex-than-it-lets-on/.

[26] B. C. for Justice, "The government is expanding its social media surveillance capabilities," https://www.brennancenter.org/our-work/analysis-opinion/government-expanding-its-social-media-surveillance-capabilities.

[27] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2013, pp. 387–402.

[28] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[29] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv preprint arXiv:1605.07277*, 2016.

[30] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," *arXiv preprint arXiv:1206.6389*, 2012.

[31] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European Conference on Computer Vision*. Springer, 2016, pp. 499–515.

[32] Google, "Keras," https://keras.io.

[33] ——, "Tensorflow," https://www.tensorflow.org.

[34] ——, "Cleverhans," https://github.com/tensorflow/cleverhans.

[35] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” 2008.

[36] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, “Vggface2: A dataset for recognising faces across pose and age,” in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 2018, pp. 67–74.

[37] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, “Ms-celeb-1m: A dataset and benchmark for large-scale face recognition,” in *European conference on computer vision*. Springer, 2016, pp. 87–102.

[38] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, “Robustness may be at odds with accuracy,” *arXiv preprint arXiv:1805.12152*, 2018.

[39] G. R. Milne and M. J. Culnan, “Strategies for reducing online privacy risks: Why consumers read (or don’t read) online privacy notices,” *Journal of Interactive Marketing*, vol. 18, no. 3, pp. 15 – 29, 2004.

[40] H. J. Smith, S. J. Milberg, and S. J. Burke, “Information privacy: Measuring individuals’ concerns about organizational practices,” *MIS Quarterly*, vol. 20, no. 2, pp. 167–196, 1996. [Online]. Available: http://www.jstor.org/stable/249477

[41] S. Holm, “A simple sequentially rejective multiple test procedure,” *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, 1979. [Online]. Available: http://www.jstor.org/stable/4615733

[42] A. Besmer and H. Richter Lipford, “Moving beyond untagging: Photo privacy in a tagged world,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’10. New York, NY, USA: ACM, 2010, pp. 1563–1572. [Online]. Available: http://doi.acm.org/10.1145/1753326.1753560

[43] J. Buolamwini and T. Gebru, “Gender shades: Intersectional accuracy disparities in commercial gender classification,” in *Conference on fairness, accountability and transparency*, 2018, pp. 77–91.

[44] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, “The Space of Transferable Adversarial Examples,” *ArXiv e-prints*, Apr. 2017.

[45] J. M. Cohen, E. Rosenfeld, and J. Z. Kolter, “Certified adversarial robustness via randomized smoothing,” in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, 2019, pp. 1310–1320. [Online]. Available: http://proceedings.mlr.press/v97/cohen19c.html

[46] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, “Universal adversarial perturbations,” *arXiv preprint*, 2017.

[47] F. Tramèr, A. Kurakin, N. Papernot, D. Boneh, and P. McDaniel, “Ensemble adversarial training: Attacks and defenses,” *arXiv preprint arXiv:1705.07204*, 2017.

[48] A. Rajabi, R. B. Bobba, M. Rosulek, C. V. Wright, and W.-c. Feng, “On the (im)practicality of adversarial perturbation for image privacy,” in *21st Privacy Enhancing Technologies Symposium ({PETS} 21)*, 2021, pp. 95–106.

[49] S. Sabour, Y. Cao, F. Faghri, and D. J. Fleet, “Adversarial manipulation of deep representations,” *arXiv preprint arXiv:1511.05122*, 2015.

[50] J. Jia and N. Z. Gong, “Defending against machine learning based inference attacks via adversarial examples: Opportunities and challenges,” 2019.

[51] N. Raval, A. Machanavajjhala, and L. P. Cox, “Protecting visual secrets using adversarial nets,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2017, pp. 1329–1332.

[52] Q. Sun, L. Ma, S. Joon Oh, L. Van Gool, B. Schiele, and M. Fritz, “Natural and effective obfuscation by head inpainting,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5050–5059.

[53] Q. Sun, A. Tewari, W. Xu, M. Fritz, C. Theobalt, and B. Schiele, “A hybrid model for identity obfuscation by face replacement,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 553–569.

[54] R. McPherson, R. Shokri, and V. Shmatikov, “Defeating image obfuscation with deep learning,” *arXiv preprint arXiv:1609.00408*, 2016.

[55] S. Shan, E. Wenger, J. Zhang, H. Li, H. Zheng, and B. Y. Zhao, “Fawkes: Protecting privacy against unauthorized deep learning models,” in *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020, pp. 1589–1604.

A Appendix

A.1 Theoretical Intuition

We discuss our intuition for why Face-Off is effective in the untargeted attack case; an extension to the targeted case is trivial. Let $f : \mathbf{X} \to \mathbb{R}^m$ be the surrogate embedding (*e.g.,* generated by one of the models in Table 2) and $g : \mathbf{X} \to \mathbb{R}^m$ be the victim embedding (*e.g.,* generated by an online model). Recall that $\delta \in \mathbf{X}$ is the output of the untargeted attack algorithm (as defined in § 2.2) that perturbs $\mathbf{x} \in \mathbf{X}$; the attack algorithm uses the surrogate embedding f. We consider the setting where the input to the attack is the sample $\mathbf{x}$ with a label s; both f and g label $\mathbf{x}$ as s, where $s = c_{j^*}$ such that $j^* = \arg\min_{j \in [k]} \phi(\beta_{c_j,f}, f(\mathbf{x}))$ and $j^* = \arg\min_{j \in [k]} \phi(\beta_{c_j,g}, g(\mathbf{x}))$ (*i.e.,* inputs produce the same label using both metric learning networks)–as defined in § 2.1. Define the following variable:

$$r(\mathbf{x}, \alpha, f, s) = \phi(f(\mathbf{x} + \alpha \cdot \delta), \beta_{s,f}) - \phi(f(\mathbf{x}), \beta_{s,f}).$$

$r(\mathbf{x}, \alpha, f, s)$ denotes the change in the distance in the embedding space of $f(\mathbf{x})$ from the centroid $\beta_{s,f}$ when we add the adversarial perturbation δ amplified by $\alpha \geq 1$. Our intuition is that $r(\mathbf{x}, \alpha, f, s)$ grows with the amplification factor α.

We define $R(\alpha, f)$ as the expectation of $r(\mathbf{x}, \alpha, f, s)$:

$$R(\alpha, f) = \mathbb{E}_{\mathbf{x} \sim D_{\mathbf{X}}}[r(\mathbf{x}, \alpha, f, s)]$$

We empirically validate our intuition in Figure 15, where amplification increases the value of $R(\alpha, f)$.

Embedding Similarity: We state that two embeddings f and g are similar if the following holds:

$$\forall \mathbf{x} \in \mathbb{R}^n, \ \phi(g(\mathbf{x}), f(\mathbf{x})) \leq \omega(\mathbf{x}).$$

Then, using the triangle inequality on the metric $\phi(\cdot, \cdot)$, it is clear to see that:

$$r(\mathbf{x}, \alpha, g, s) \geq r(\mathbf{x}, \alpha, f, s) - 4\omega(\mathbf{x})$$

In other words, if $r(\mathbf{x}, \alpha, f, s) > 4\omega(\mathbf{x})$, then $r(\mathbf{x}, \alpha, g, s) > 0$. This implies that for embedding g, $g(\mathbf{x} + \alpha \cdot \delta)$ is farther from the centroid $\beta_{s,g}$ than $g(\mathbf{x})$, meaning the attack transfers to the victim model. Taking the expectation of both the sides in the equation given above we get

$$R(\alpha, g) \geq R(\alpha, f) - 4\mathbb{E}_{\mathbf{x} \sim D}[\omega(\mathbf{x})]$$

In particular, if $\omega(\mathbf{x})$ is bounded by Δ, we obtain:

$$R(\alpha, g) \geq R(\alpha, f) - \Delta$$

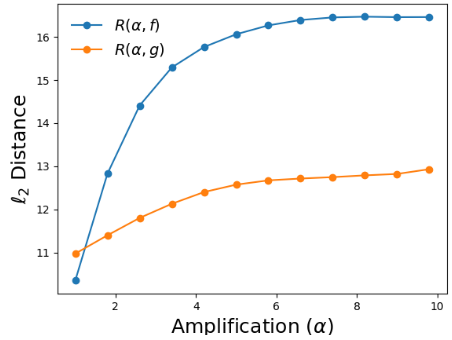


Fig. 15. The relationship between $R(\alpha, f)$ and $R(\alpha, g)$; amplifying the perturbation increases both terms.

We empirically validate the claims above using the FLVT model as f and the CSVC model as g (refer Table 2). Figure 15 reports $R(\alpha, f)$ and $R(\alpha, g)$. The values of $R(\alpha, f)$ and $R(\alpha, g)$ are averaged over 5 input samples. It is evident from the plot that amplification makes the perturbation *more* adversarial on both the surrogate and victim models.

A.2 Adversarial Training: Additional Results

Dataset	Base (ℓ_2)	AT (ℓ_2)	Base (cos)	AT (cos)
LFW	79.9%	74.4%	75.8%	73.2%
VGGFace2	83.6%	71.8%	82.4%	67.6%
Celeb	85.7%	81.7%	80.4%	76.3%

Table 7. Natural matching accuracies after adversarial training. Note that adversarial training decreases natural matching accuracy. As before, **Base** refers to the baseline natural accuracy (before adversarial training), and **AT** refers to the natural accuracy after adversarial training.

We choose to evaluate the results of adversarial training for both top-1 accuracy (Table 4) and matching accuracy (Table 7). The top-1 case refers to the closest embedding from a bucket of labels whereas matching deals with the binary classification *i.e.,* match vs. mismatch.

Attribute	**P**	**B**	**UU**
Demographics			
Num. Workers	167	163	93
Male	60.1%	68.7%	75.26%
Female	39.9%	31.3%	24.74%
Average Age (in years)	37	38	37
Privacy Preference			
Conscious (PC)	75.44%	78%	62.4%
Not Conscious (NPC)	24.56%	22%	37.6%
Education			
Some High School	1.19%	1.22%	1.07%
High School	11.97%	10.42%	1.07%
Some College	17.96%	17.79%	5.37%
Associate's	10.17%	9.81%	3.22%
Bachelor's	47.90%	47.85%	77.41%
Graduate	10.77%	12.83%	11.82%

Table 8. Demographics of participants of study reported in § 7

A.3 Demographics

We report the demographic information of the participants of our user studies (refer § 7) in Table 8. Columns **P** and **B** refer to Portrait and Background studies (refer § 7.1) and column **UU** refers to the User Uploaded images study (refer § 7.2).

A.4 Run-time

We report micro-benchmarks related to run-time performance of our algorithms in Table 9. Note that the ℓ_2 variant of CW uses 8 binary search steps, and the ℓ_∞ variant of CW terminates at 10 trials.

Attack	**Norm**	**Model**	**Avg. run-time (s)**	**Batch Size**
CW	2	CSVC	31.25	5
CW	2	FLVT	127.81	5
CW	∞	CSVC	126.00	1
CW	∞	FLVT	373.16	1
PGD	2	CSVC	6.40	5
PGD	2	FLVT	70.51	5

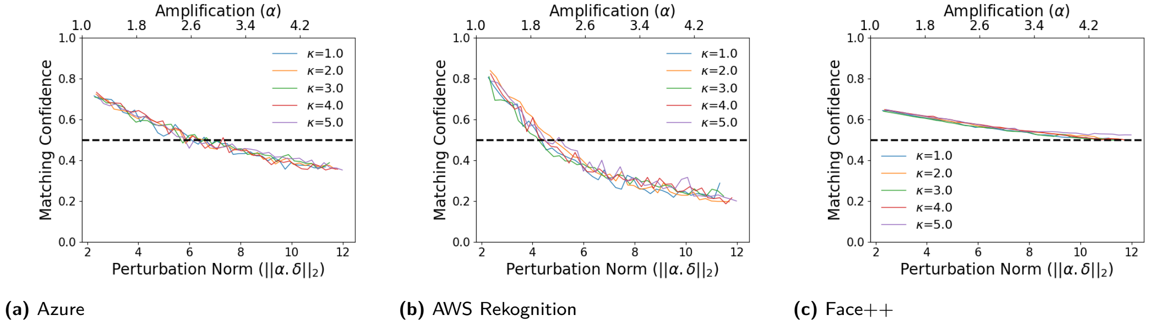
Table 9. Run-time for mask generation. Each attack uses $N = 100$ iterations. Run-times were evaluated on a server with 2 Titan XPs and 1 Quadro P6000. **Model** refers to the surrogate model used to generate the masks.

A.5 Cropped Images

In this experiment, we crop the adversarial inputs and compare it with a cropped reference (*i.e.*, an image with the true label of the cropped adversarial input). We observe that, similar to the uncropped images in § 6.1 and 6.2, the cropped images transfer to the black-box cloud APIs as well. This is the case for both CW (refer Figures 16 and 17) and PGD (refer Figures 18 and 19) attacks.

A.6 Deployed Service

Figure 20 contains screenshots from the service we deploy. A video highlighting its usability can be found here: https://youtu.be/LJtcpZmz7JY

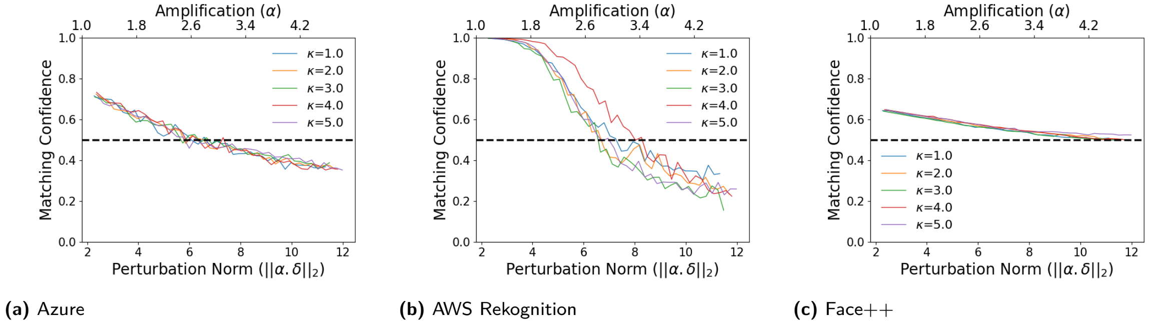


(a) Azure

(b) AWS Rekognition

(c) Face++

Fig. 16. 2018: Transferability of cropped images generated using CW attack

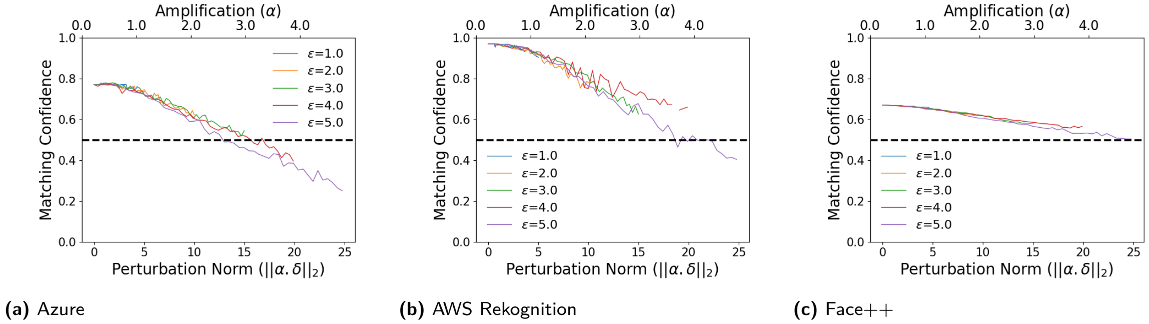


(a) Azure

(b) AWS Rekognition

(c) Face++

Fig. 17. 2020:Transferability of cropped images generated using CW attack

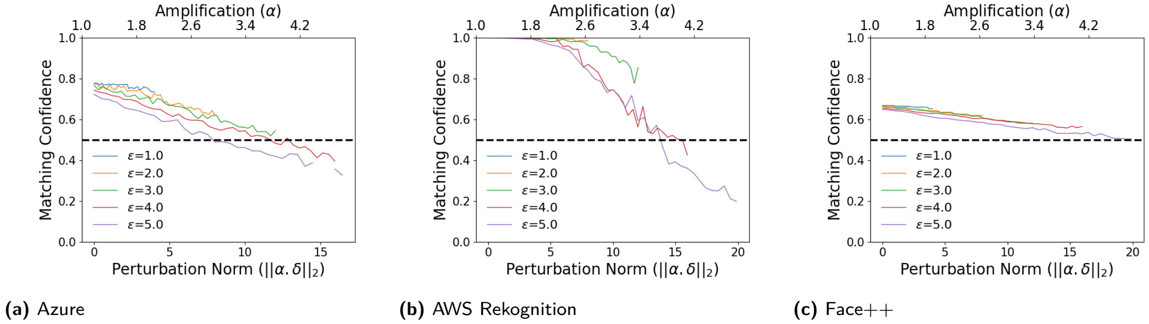


(a) Azure

(b) AWS Rekognition

(c) Face++

Fig. 18. 2018:Transferability of cropped images generated using PGD attack

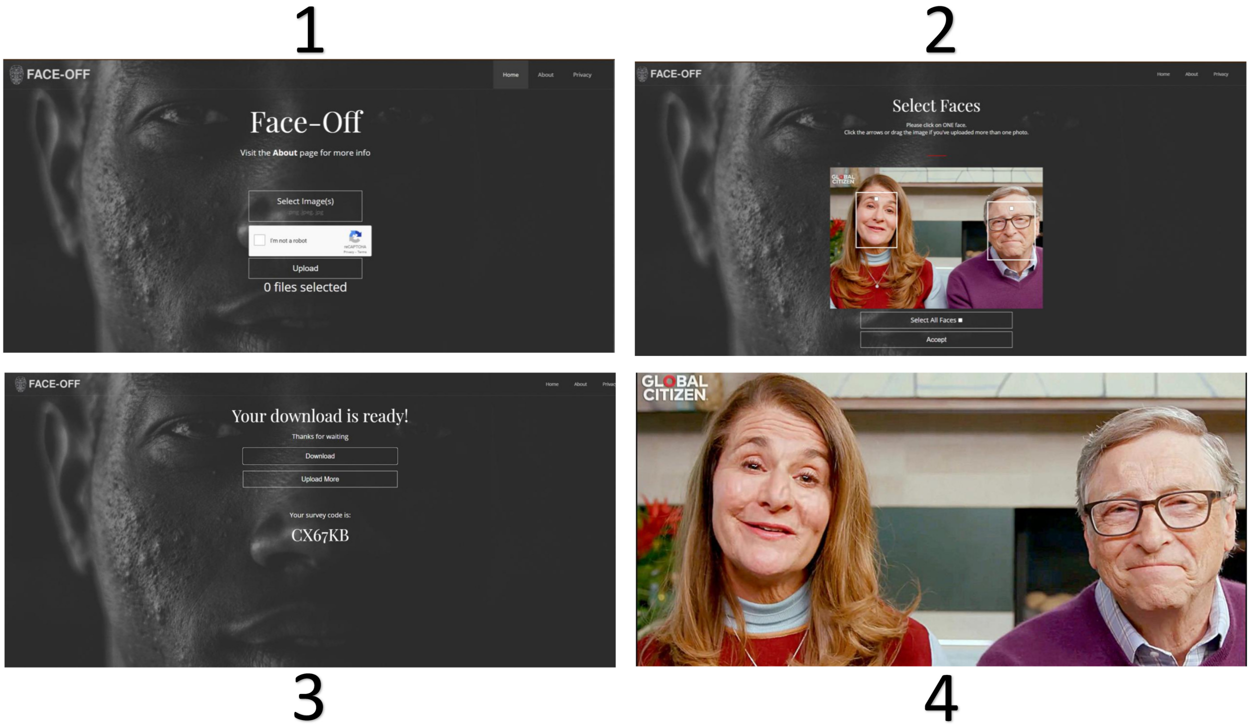


(a) Azure

(b) AWS Rekognition

(c) Face++

Fig. 19. 2020: Transferability of cropped images generated using PGD attack



Face-Off

Visit the **About** page for more info

Select Faces

Please click on ONE face.
Click the arrows or drag the image if you've uploaded more than one photo.

Your download is ready!

Thanks for waiting

Your survey code is:

CX67KB

Fig. 20. Website view of Face-Off's pipeline.