DeepBlur: A Simple and Effective Method for Natural Image Obfuscation

Article · March 2021

CITATIONS
10

READS
726

2 authors, including:

Tao Li
Purdue University
39 PUBLICATIONS **624** CITATIONS

SEE PROFILE

DeepBlur: A Simple and Effective Method for Natural Image Obfuscation

Tao Li
Department of Computer Science
Purdue University
taoli@purdue.edu

Min Soo Choi
School of Industrial Engineering
Purdue University
choi502@purdue.edu

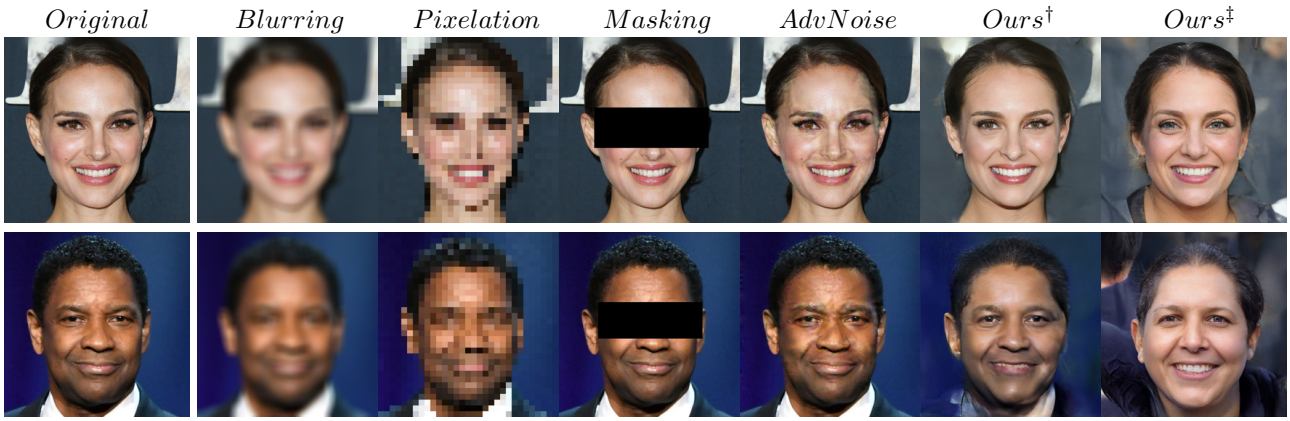

Figure 1. We propose a simple yet effective method for image obfuscation by blurring in latent space (i.e., DeepBlur). Comparing to existing methods (e.g., Gaussian blur, pixelation, masking, and adversarial noise), our approach preserves high perceptual quality while preventing unauthorized face recognition from both automatic systems and human adversaries.

Abstract

There is a growing privacy concern due to the popularity of social media and surveillance systems, along with advances in face recognition software. However, established image obfuscation techniques are either vulnerable to re-identification attacks by human or deep learning models, insufficient in preserving image fidelity, or too computationally intensive to be practical. To tackle these issues, we present DeepBlur, a simple yet effective method for image obfuscation by blurring in the latent space of an unconditionally pre-trained generative model that is able to synthesize photo-realistic facial images. We compare it with existing methods by efficiency and image quality, and evaluate against both state-of-the-art deep learning models and industrial products (e.g., Face++, Microsoft face service). Experiments show that our method produces high quality outputs and is the strongest defense for most test cases.

1. Introduction

Being in a digital era, we enjoy the benefits of smartphones and cameras which facilitate learning and social connections. In the meantime, however, billions of images being uploaded to public cloud servers every day, introducing serious privacy concerns, as an adversary may collect such data, identify "persons of interest" using either crowdsourcing or machine learning algorithms, and secretly monitor our daily lives. A recent New York Times article reveals that a private company collected over three billion online images and trained a large model capable of recognizing millions of people without consent [1].

With legal and privacy concerns, image obfuscation methods such as pixelation and blurring are often used to protect sensitive information, e.g., human faces and confidential texts. However, recent advances in deep learning make these approaches less effective, as it has been shown that blurred or pixelated facial images can be re-identified by deep neural networks at high accuracy [2]. Moreover, image distorted by these methods usually are less visually pleasing (see examples in fig. 1).

More recently, new approaches such as adversarial perturbation [3, 4, 5] and GAN-based image editing [6, 7, 8] have been proposed to tackle these issues. However, the former is subjective to the attack's deep learning model and generally fails when the model is unknown; they also cannot (and are not supposed to) protect against human adversaries (e.g., crowdsourcing). The latter, although may have bet-

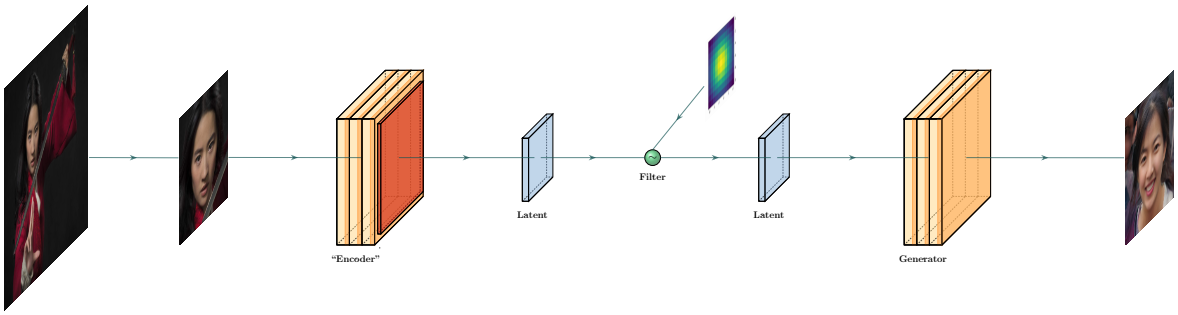

Figure 2. DeepBlur overview. Given an arbitrary input, we first crop, rotate, and align the image, and then feed the aligned face to an optimization pipeline (see fig. 3) and obtain a latent representation that can synthesize almost identical faces with a generative neural network. Depending on the application scenario (e.g., the required level of obfuscation), we apply a low-pass filter with desired kernel size to the latent representation and provide the smoothed counterpart to the generative model again, which generates a "deep blurred" result.

ter image quality and is capable of protecting against both human perception and automatic systems, tends to be computationally expensive and may produce visible artifacts on synthesized faces, especially when the GANs are conditionally trained [9].

This leads to our motivation: we need an image obfuscate method that is effective to protect against human and machine adversaries while preserving image quality, yet simple enough, both conceptually and computationally, to be deployed in practice at a large scale. Then we propose DeepBlur, a simple yet effective method for natural image obfuscation. Figure 2 outlines the approach.

We argue that DeepBlur has following advantages over existing methods:

- Compared to traditional methods (e.g., Gaussian blurring, pixelation, masking), DeepBlur is a stronger defense against deep learning-based recognition systems and is able to generate more visually pleasing results;
- Compared to adversarial perturbation-based methods, DeepBlur makes no assumption on the specific neural network or recognition system used by the attacker, and can defense against unauthorized recognition from both human crowdsourcing and automatic systems;
- Compared to GAN-based image editing methods (e.g., attribute editing, face inpainting and replacement), DeepBlur generally produces less artifacts (due to smoothing effect in latent space) and is more computationally friendly.

We will demonstrate these both qualitatively and quantitatively in the following sections.

The rest of the paper is organized as follows: in section 2 we review recent advances in image privacy research and face manipulation techniques; section 3 formalizes the attack model and explains the DeepBlur method, including our approach for latent representation search, deep blurring, and image generation; section 4 details experiment settings and evaluation metrics, and show both qualitative and quantitative results; section 5 further discuss computational concerns and show interesting deep blur visual effects. We concludes the paper in section 6.

2. Related Work

Privacy-Enhancing Techniques for Images. Classical methods such as pixelation and blurring (as shown in fig. 1) have been widely used to obfuscate facial images; but, as mentioned earlier, they fail to defeat against modern facial recognition systems powered by deep learning and often produce images that are not visually pleasing. Methods include distorting images to make them unrecognizable [7, 10], and producing adversarial patches in the form of bright patterns printed on sweatshirts or signs, which prevent facial recognition algorithms from even registering their wearer as a person [11, 12].

However, these are targeted against facial recognition systems designed without regard to privacy protection, and could be subject to targeted re-identification attacks such as [13, 14, 15]. In 2005, Newton *et al*. [16] introduced k-Same, the first privacy-preserving algorithm in the context of image databases, and Hao *et al*. [17] demonstrated that it is more effective than canonical methods. There is a trade-off between privacy and usability [6] and Gross *et al*. [13] introduced k-Same-Select to balance disclosure risk and classification accuracy. Zhang *et al*. [18] further designed an "obfuscate function" that adds random noises to samples to hide sensitive information in the dataset while preserving model accuracy. In 2018, Fan [19] proposed an obfuscation method that satisfies ε-differential privacy at pixel level, yet its image quality is low and only protects privacy of the pixels instead of the person. More recently, Li and Clifton [20] proposed to manipulate image latent space in a way that satisfies ε-differential privacy for the person and produces photo-realistic images.

Facial Image Editing. Face analysis is an important topic in computer vision with a wide range of real-world applications, such as expression and attribute recognition [21, 22], face super-resolution [23, 24], and virtual cosmetic enhancement [25, 26, 27]. The task of facial image editing aims at manipulating facial attributes of a given image at the semantic level. Current approaches include carefully designing loss functions [28, 29], introducing additional attribute labels or features [30, 31, 32], and using special architectures to train new models [33, 34] . However, the synthesized results by these conditionally trained models is incomparable to native unconditionally trained GANs, such as PGGAN [35] and StyleGAN [36]. Unlike AnonymousNet [6] and UP-GAN [8] which use conditional GANs for image obfuscation, DeepBlur leverages an unconditionally trained generative adversarial network and varies its latent space to control image synthesis, which produces image outputs of higher quality (see figs. 1 to 5).

Latent Space Properties of GANs. Despite the great success of GANs in image synthesis and editing, a full understanding of how semantics are encoded in their latent spaces is still missing. A major issue is how we can project a measure (e.g., Euclidean distance) that we observe in semantic space to its counterpart in latent space. Literature usually treat the latent space as Riemannian manifold [37, 38, 39]. Radford *et al*. [40] and Upchurch *et al*. [41] observed vector arithmetic properties in latent space and analyzed the disentanglement of multiple semantics. Studies in this domain are mostly empirical: Jahanian *et al*. [42] "steered" the latent space for camera motion and image scaling; Yang *et al*. [43] observed semantic hierarchy in scene synthesis models; Bau *et al*. [44] found correspondences between intermediate layers of GANs and visual objects such as buildings and trees; Shen *et al*. [9] interpreted facial semantics by varying latent codes in the latent space.

3. Preserving Image Privacy with DeepBlur

In this section, we detail the DeepBlur method for image privacy preservation. We first describe our assumptions for both users and attackers, and define three threat models, which explain what we mean by preserving image privacy.

3.1. Assumptions and Threat Models

In our scenario, a user wants to upload images to a remote server, where connections to the server or the server itself is compromised. The goal of the user is to share high quality images with obfuscated identities, with the hope that the user's identity will not be revealed even if an adversary has access to the images.

On the other side, we assume that the attacker's goal is to build a powerful face recognition system that can accurately identify a group of people. We also assume that the attacker has unlimited computing resources and can use external datasets to facilitate model training (the external datasets exclude images from the users unless otherwise specified). We say an obfuscation method is stronger than the other if it has a lower identification accuracy by the attacker's recognition system. Accordingly, we define three threat models.

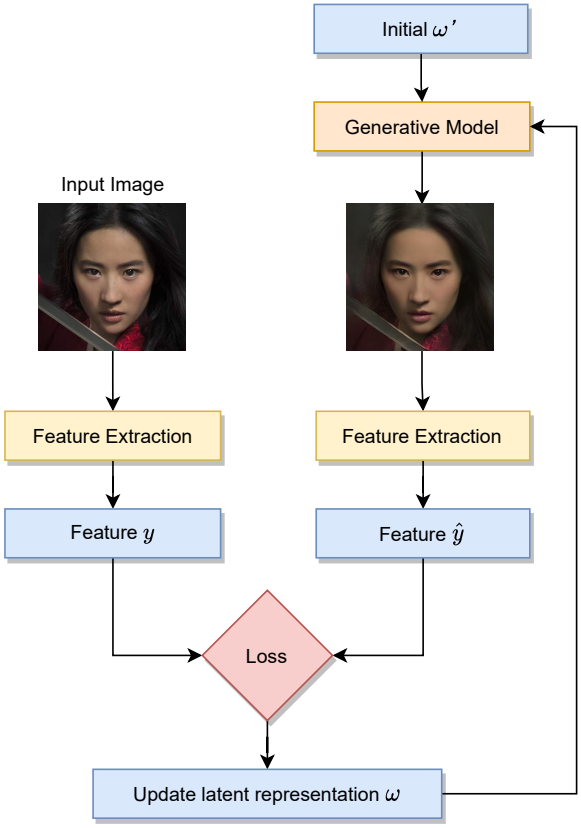

Figure 3. A general framework of latent representation search.

Threat Model T_1. This scenario simulates the case that the attacker has prior information of the users and tries to identify them from protected (i.e., obfuscated) images. For example, a paparazzo obtains some sensitive personal photos but the photos are obfuscated and he would like to know what celebrities are in the photos. In other words, the paparazzo can train his model using all publicly available photos of celebrities but the obfuscated photos were unseen.

Threat Model T_2. In this scenario, the attacker acquires a set of obfuscated images with identity labels and tries to identify users from a group of original images. For example, Eve has two classmates, Alice and Bob, who have accounts in an anonymous dating website with selfies protected by image obfuscation techniques. Eve downloads an

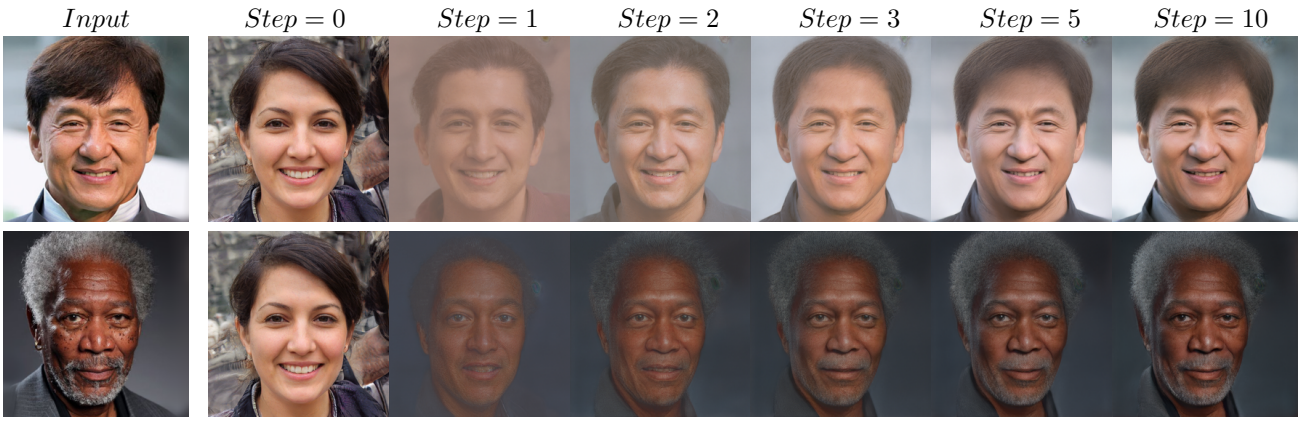

Figure 4. Examples of latent representation search. As illustrated in fig. 3, we formalize the task of finding the latent representation of images as an optimization problem. Starting from an "average face" of GAN (see fig. 9), we use L-BFGS algorithm [45] to find the latent representation that is able to minimize the loss between input image and the generated counterpart. The algorithm converges very quickly and usually a latent representation obtained after 10 steps can synthesize an image that looks reasonably close to the original.

obfuscated selfie and would like to train a model to predict whose the selfie is.

Threat Model T_3. For this model, the attacker acquires obfuscated images with labels and would like to associate the labels with another group of obfuscated images. For example, an attacker successfully breaks into the server of `victim.com` which stores credentials and obfuscated photos of thousands of users. Then he targets another website, `vulnerable.com`, whose preview mode shows obfsucated images of all users. To perform a credential stuffing attack (i.e., use credentials on `victim.com` to take over accounts on `vulnerable.com`), the attacker needs to train a model on the obfuscated images of `victim.com` and use it to label accounts on `vulnerable.com`.

We will evaluate our method under these three settings in section 4 and compare it with other obfuscation methods.

3.2. The DeepBlur Method

From a high level perspective, DeepBlur applies a low-pass filter to the latent space of an input image and then uses the blurred latent representation to synthesize the output. Figure 2 shows an overview of DeepBlur, including three essential steps: latent representation search (i.e., "encoder"), deep blurring, and image generation,

Latent representation search. Given an arbitrary input image, we first perform cropping and alignment, and then feed it into a feature extractor (e.g., VGG16 [46]) to obtain feature vector y. In the meantime, we put its counterpart, an image synthesized by a generative model (e.g., StyleGAN [36]) with the same procedure and obtain feature vector $\hat{y}$. After computing the loss between y and $\hat{y}$, we accordingly update the latent representation ω, feed it back to the generator, and repeat this process until the synthesized image is close enough to the original. Figure 3 illustrates the approach.

The reason that we don't use an autoencoder directly is that the latent representation obtained by such method is hard to produce images close to the original and with quality comparable to ours (see fig. 4); also, with an efficient optimization algorithm, the proposed search method can converge quite quickly. We will discuss this more in section 5.

Deep blurring. The two-dimensional Gaussian filter [47] is defined as follows:

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{1}$$

where x is the distance from the origin in x-axis, y is the distance from the origin in y-axis, and σ is the standard deviation of the Gaussian distribution. From the previous step, we obtained ω, the latent representation of the image. Then we apply a filter on it, and get the blurred representation,

$$\omega' = g(\omega). \tag{2}$$

Usually, ω is two-dimensional but the dimension may vary depending on the specific generative model and layer in use.

Image generation from latent representation. After the above deep blurring step, we obtained a latent representation of the image that was smoothed in its latent space. The process of image generation using GAN is to feed the latent values into a specific layer in the generator. In our case, we feed the blurred latent representation ω to the unconditionally pre-trained generative model in the searching step (see fig. 3). By changing the kernel size (i.e., σ), we can adjust the output image to a desired level of obfuscation. Figure 5 demonstrates deep blurred images with various kernel sizes.

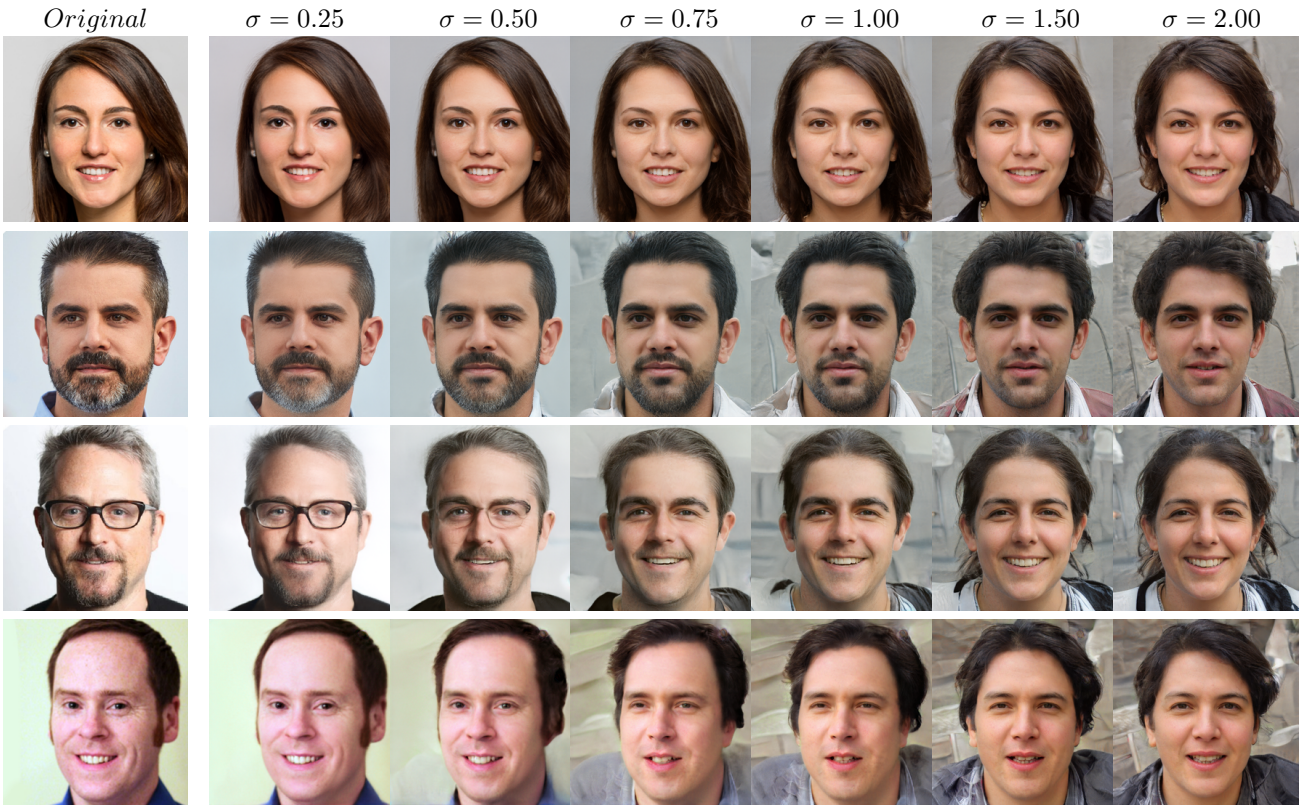

Figure 5. Examples of deep blurred images. Given an arbitrary facial images, the DeepBlur method is able to obfuscate the identity while preserving high visual fidelity, and the identity distance monotonically increases as σ getting larger. Note that the original images are from CVPR'21 Media Forensics Workshop committee and were not visible to the pre-trained generative model (see section 3.2) during training.

4. Experiment

Compared to existing methods, DeepBlur shows convincing performance in terms of both effectiveness against adversarial facial recognition systems and the quality of synthesized images. In this section, we first introduce the datasets and experimental settings in section 4.1, and assess image quality in section 4.2. We then evaluate obfuscation methods in section 4.3 under different attack settings.

4.1. Datasets

In our study, we mainly use two datasets: FlickrFaces-HQ (FFHQ) [36] and CelebFaces Attributes (CelebA) [48].

The former, FFHQ, consists of 70,000 high-resolution (i.e., 1024 $\times$ 1024) images, covering a wide spectrum of faces with various ages, ethnicities, and image backgrounds. It was collected by researchers in NVIDIA from Flickr. The style-based generator that we use in the latent search step and image generation step (see fig. 2) was trained in this dataset.

The latter, CelebA, is a large-scale human face dataset which contains more than 200,000 images from over 10,000 celebrities (i.e., different identities). We use the dataset to evaluate and compare our approach with others. For proof of concept, instead of using the entire dataset, we select a subset of 100 identities and 10 images for each, and split the 10 images as 7, 1, and 2, for training, validation, and testing, respectively. Note that necessary preprocessing procedures (e.g., face alignment) are performed for all images before running the experiments.

4.2. Image Quality Assessment

Structural Similarity Index Measure (SSIM) and Fréchet Inception distance (FID) are commonly used to measure image quality in terms of similarity and perceptual distance.

Definition 4.1. Given a reference image and a test image, the PSNR (in dB) between the two images is defined as

$$\text{PSNR} = 10 \cdot \log_{10}(\frac{\text{MAX}_\text{I}^2}{\text{MSE}}), \tag{3}$$

where MAX_I is the maximum possible pixel value of the image (typically 255) [49].

Definition 4.2.

$$\text{FID} = ||\mu_r - \mu_g||^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}), \tag{4}$$

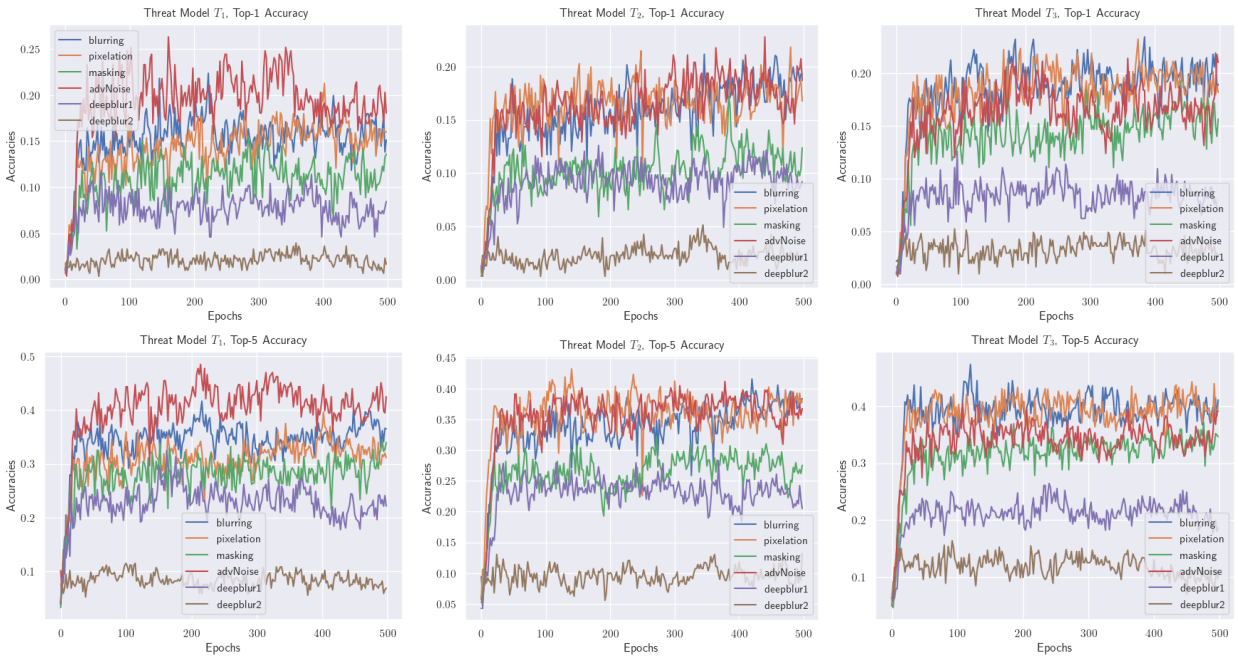

Figure 6. Comparison of obfuscation methods. Above, we show the test result from a VGG16 network [46] trained after 500 epochs under different settings (top row and bottom row report top-1 accuracies and top-5 accuracies respectively, and each column corresponds to a threat model specified in section 3.1). We compare Gaussian blurring on pixels, pixelation, masking, adversarial noise by Fawkes [5], and our method with two settings ($\sigma = 0.5$ and $\sigma = 1.0$). Note that our settings are the same as shown in figs. 1 and 5. More test results from other face recognition systems (e.g., commercial APIs) can be found in table 2.

where $X_r \sim \mathcal{N}(\mu_r, \Sigma_r)$ and $X_g \sim \mathcal{N}(\mu_g, \Sigma_g)$ are activations of Inception-v3 pool3 layer for real and generated samples, respectively.

A larger FID means the generated result is further away from the original in the identity space (as measured by the Fréchet distance between the two distributions), and a larger value of SSIM or MS-SSIM implies two images are structurally more similar. Table 1 compares obfuscation methods in terms of SSIM, MS-SSIM, and FID. Results imply that DeepBlur well preserves structural similarity and quality of images while the generated identities are far from the original, which aligns with our observation in fig. 1 and fig. 5. Note that SSIM may fail to capture nuances of human perception [52] and a smaller or larger value of the metrics does not necessarily imply higher or lower image quality. Thus, we only use the measurements for reference.

	SSIM	MS-SSIM	FID
Blurring	0.858	0.814	60.770
Pixelation	0.759	0.737	**264.946**
Masking	0.873	0.889	72.417
AdvNoise	0.482	0.393	42.764
Ours†	0.467	0.380	207.473
Ours‡	**0.430**	**0.349**	231.115

Table 1. Measures of identity distance. We use SSIM and MS-SSIM [50] to measure the similarity between original image and the obfuscated counterpart, and use Fréchet inception distance (FID) [51] for distance between the original and obfuscated identities. For our method, we set the σ values as 2 and 5 respectively.

4.3. Evaluation of Obfuscation Methods

We evaluate the obfuscation methods by attacking them under different threat model settings (i.e., T_1, T_2, and T_3 as specified in section 3.1) and with both canonical deep learning models and commercial face recognition systems.

The task of face recognition is essentially a classification problem. We first attack the obfuscation methods using VGG16 [46] which consists of 13 convolutional layers and 3 linear layers, and was the runner-up of ImageNet Large Scale Visual Recognition Competition (ILSVC) in 2014. We also use ResNet18 [56], the winner of the ILSVC 2015 challenge, to simulate the attack scenarios, which has 17 convolutional layers with skip connections and 1 linear layer and uses the pre-activation residual unit.

With years of development in deep learning and face recognition techniques, canonical models such VGG and ResNet may not reflect the real privacy threat today. Thus, we also attack the obfuscation methods with commercial grade face recognition systems, including Microsoft Azure Face API [57] and Face++ [58]. Microsoft Azure Face API

	Threat Model T_1				Threat Model T_2				Threat Model T_3			
	VGG19	ResNet18	Face++	Azure	VGG19	ResNet18	Face++	Azure	VGG19	ResNet18	Face++	Azure
Original	0.208	0.421	0.973	0.935	0.208	0.421	0.973	0.935	0.208	0.421	0.973	0.935
Pixelation	0.160	0.305	0.458	0.255	0.168	0.373	**0.116**	0.391	0.182	0.155	0.292	0.653
Blurring	0.151	0.157	0.922	0.592	0.193	0.056	0.942	0.684	0.190	0.397	0.912	0.871
Masking	0.136	0.072	0.614	0.289	0.124	0.223	0.646	0.201	0.157	0.114	0.537	**0.051**
AdvNoise	0.180	0.366	0.951	0.765	0.187	0.314	0.908	0.878	0.211	0.256	0.910	0.760
DeepBlur†	0.084	0.209	0.683	0.330	0.092	0.213	0.908	0.500	0.072	0.225	0.541	0.460
DeepBlur‡	**0.016**	**0.043**	**0.060**	**0.000**	**0.020**	**0.020**	0.180	**0.000**	**0.026**	**0.032**	**0.070**	0.126

Table 2. Comparison of obfuscation methods. We evaluate obfuscation methods by top-1 accuracies of four face recognition systems (i.e., VGG16, ResNet18, Face++, and Microsoft Azure Face API) under different threat model settings. DeepBlur† and DeepBlur‡ correspond to $\sigma = 0.5$ and $\sigma = 1.0$ respectively, and experiments show that the later one is the strongest at most times. Note that the experiments use the same settings as in figs. 1 and 5, and "original" means the original image without any obfuscation.

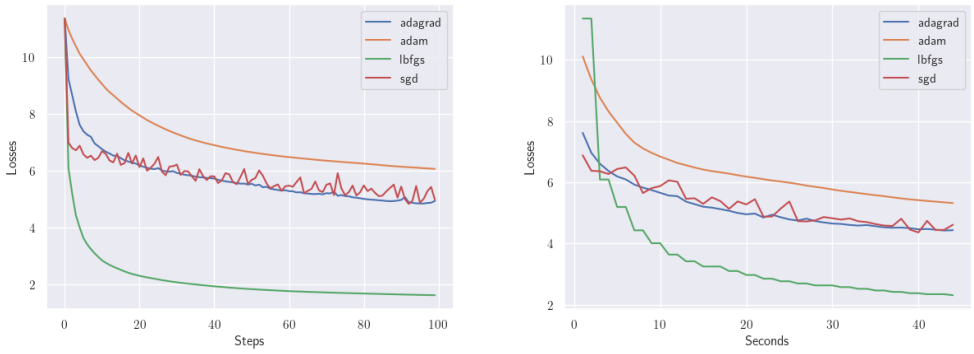

Figure 7. We evaluate four canonical optimization methods for latent representation search, including Adam [53], AdaGrad [54], SGD with momentum [55], and L-BFGS [45], and find that L-BFGS outperforms the rest in terms of efficiency. Note that a very accurate image latent representation isn't necessary for our task. Thus, we weight speed over precision and argue that L-BFGS is a good fit here.

is a part of Microsoft's cognitive services which provide various machine cognition algorithms from face detection to cluster similar faces. For their face identification API, a user can provide a set of mappings of identities and images for training and query the identity of an image out of the training set. The service returns the best matching along with confidence level among the given user pool. Face++ provides similar API service for face identification where a user can enter annotated image data, train a model, and query the identity of the unseen.

Figure 6 shows the experimental results under three threat model settings (i.e., T_1, T_2, and T_3 as specified in section 3.1) and reports both top-1 and top-5 accuracies. In the accuracy versus epoches plot, the attacker (i.e., VGG16) achieves the lowest accuracy with DeepBlur‡ (i.e., $\sigma = 1.0$), meaning it is the strongest defense among all compared methods. As mentioned in section 4.1, the test dataset has 100 identities. Thus, the expected accuracy for a random guess is 0.01, which is close to what this face recognition model can get with our approach. Note that $\sigma = 1.0$ isn't an unreasonably large value as it still preserves high image quality and certain facial semantics from the original (see fig. 5 and ours‡ in fig. 1). Table 2 lists the results for all four face recognition methods, showcasing that DeepBlur is the strongest method for most of the tests.

5. Analysis

In this section, we extend our analysis of DeepBlur, and further discuss its computational efficiency in section 5.1 and show empirical results in section 5.2 that may explain its superior visual quality and tricks for fast convergence in latent representation search.

5.1. Computational Efficiency of DeepBlur

As shown in fig. 2, our approach has three main components: latent representation search, deep blurring, and image generation. It is trivial that the first is the most computationally expensive step, as the second step is nothing more than linear filtering and the third step only takes one forward pass in the pre-trained generative model

We formalize the searching step as an optimization problem in section 3.2. and use derivative information to update the latent representation. Due to computational efficiency concerns, we mainly investigate two types of optimization algorithms: first-order algorithms, e.g., stochastic

Figure 8. Deep-blurring effects. By smoothing in latent space, DeepBlur removes artifacts and occlusions in the images at semantic level. For example, **top** are original images and **bottom** are the deep-blurred counterparts, where watermark, hair, nose sleeve, and hands are removed from the frontal faces, respectively.

Figure 9. Averaging effects. If we apply a very large kernel (e.g., $\sigma = 100$), the model will take the average of almost all latent values and generate an "average face" of GAN. In above examples, **top** are the original images and **bottom** are the averaged ones. Although the inputs are different, the averaged images are almost identical and only have subtle differences in background.

gradient descent (SGD) [59], adaptive gradient algorithm (AdaGrad) [54], Adam [53]; and second-order algorithms such as BFGS [45]. To accelerate convergence, SGD adds momentum of previous weight when updating the current; AdaGrad leverages adaptive learning rates; Adam combines momentum with the adaptive learning rate method; and BFGS approximates second-order derivatives and uses them for weight updates.

In general, second-order methods require more computing resources per step as higher order information is required. However, they usually take less steps to converge, especially for functions close to convex, and robust against saddle points, which is the case when we start the latent representation search from an "average face." Figure 3 shows that the search algorithm converges very quickly with an appropriate initialization, and a latent representation obtained after only 10 steps can be used by the generator to synthesize an image that looks close to the original. Figure 7 compares the discussed methods in terms of numbers of iterations and elapsed time versus losses, and demonstrates that L-BFGS (limited memory BFGS) has the best performance among the four, which aligns with our intuition and thus it is used in our framework for all the experiments.

5.2. Deep Blurring Effects

In the experiments, we also observe some interesting visual effects by deep blurring, which provide deeper insight of the method. For example, fig. 8 shows that deep blurring may remove artifacts and occlusions on frontal faces when applying a low-pass filter to the latent representation that control the semantics. Figure 9 shows the "average face", which is achieved by filtering with a very large kernel size (i.e., taking the average of all latent values). We found that using the latent representation of the "average face" as initial value for latent representation search (see fig. 3), instead of random initialization, makes the searching step converge faster, which can be explained by the property of second-order methods we discussed in section 5.1. These visual effects, along with results shown in figs. 1 to 5, align with empirical findings in literature, and provide new evidence of linearity and continuity in the latent space of GANs.

6. Conclusion

To conclude the paper, we present DeepBlur, a simple yet effective method for natural image obfuscation. By blurring the latent space of a generative model, DeepBlur is able to alter the identity in the image while preserving high visual quality. We evaluate the method both qualitatively and quantitatively, and show that it is effective against both human perception and state-of-the-art facial recognition systems. Our experiments demonstrate that DeepBlur has advantages in either image quality, computational efficiency, effectiveness against unauthorized identification attacks, or all of the above when comparing to established methods. In the future, we plan to extend our method to broader applications.

References

[1] Kashmir Hill. The secretive company that might end privacy as we know it. *The New York Times*, 18:2020, 2020. 1

[2] Richard McPherson, Reza Shokri, and Vitaly Shmatikov. Defeating image obfuscation with deep learning. *arXiv preprint arXiv:1609.00408*, 2016. 1

[3] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018. 1

[4] Zuxuan Wu, Ser-Nam Lim, Larry S Davis, and Tom Goldstein. Making an invisibility cloak: Real world adversarial attacks on object detectors. In *European Conference on Computer Vision*, pages 1–17. Springer, 2020. 1

[5] 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 (USENIX Security 20), pages 1589–1604, 2020. 1, 6

[6] Tao Li and Lei Lin. AnonymousNet: Natural face de-identification with measurable privacy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1, 2, 3

[7] Qianru Sun, Ayush Tewari, Weipeng Xu, Mario Fritz, Christian Theobalt, and Bernt Schiele. A hybrid model for identity obfuscation by face replacement. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 553–569, 2018. 1, 2

[8] Hanxiang Hao, David Güera, Amy R Reibman, and Edward J Delp. A utility-preserving gan for face obscuration. *arXiv preprint arXiv:1906.11979*, 2019. 1, 3

[9] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2, 3

[10] Yifan Wu, Fan Yang, and Haibin Ling. Privacy-protective-gan for face de-identification. *arXiv preprint arXiv:1806.08906*, 2018. 2

[11] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2

[12] Zuxuan Wu, Ser-Nam Lim, Larry Davis, and Tom Goldstein. Making an invisibility cloak: Real world adversarial attacks on object detectors. *arXiv preprint arXiv:1910.14667*, 2019. 2

[13] Ralph Gross, Edoardo Airoldi, Bradley Malin, and Latanya Sweeney. Integrating utility into face de-identification. In *International Workshop on Privacy Enhancing Technologies*, pages 227–242. Springer, 2005. 2

[14] Ralph Gross, Latanya Sweeney, Fernando De la Torre, and Simon Baker. Model-based face de-identification. In *2006 Conference on computer vision and pattern recognition workshop (CVPRW'06)*, pages 161–161. IEEE, 2006. 2

[15] Ralph Gross, Latanya Sweeney, Jeffrey Cohn, Fernando De la Torre, and Simon Baker. Face de-identification. In *Protecting privacy in video surveillance*, pages 129–146. Springer, 2009. 2

[16] Elaine M Newton, Latanya Sweeney, and Bradley Malin. Preserving privacy by de-identifying face images. *IEEE transactions on Knowledge and Data Engineering*, 17(2):232–243, 2005. 2

[17] Hanxiang Hao, David Güera, János Horváth, Amy R Reibman, and Edward J Delp. Robustness analysis of face obscuration. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 176–183. IEEE, 2020. 2

[18] Tianwei Zhang, Zecheng He, and Ruby B Lee. Privacy-preserving machine learning through data obfuscation. *arXiv preprint arXiv:1807.01860*, 2018. 2

[19] Liyue Fan. Image pixelization with differential privacy. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 148–162. Springer, 2018. 2

[20] Tao Li and Chris Clifton. Differentially private imaging via latent space manipulation. *arXiv preprint arXiv:2103.05472*, 2021. 2

[21] Caifeng Shan, Shaogang Gong, and Peter W McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and vision Computing*, 27(6):803–816, 2009. 3

[22] Xudong Liu and Guodong Guo. Attributes in multiple facial images. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 318–324. IEEE, 2018. 3

[23] Ratheesh Kalarot, Tao Li, and Fatih Porikli. Component attention guided face super-resolution network: Cagface. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 370–380, 2020. 3

[24] Junjun Jiang, Chenyang Wang, Xianming Liu, and Jiayi Ma. Deep learning-based face super-resolution: A survey. *arXiv preprint arXiv:2101.03749*, 2021. 3

[25] Xudong Liu, Tao Li, Hao Peng, Iris Chuoying Ouyang, Taehwan Kim, and Ruizhe Wang. Understanding beauty via deep facial features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 3

[26] Xudong Liu, Ruizhe Wang, Chih-Fan Chen, Minglei Yin, Hao Peng, Shukhan Ng, and Xin Li. Face beautification: Beyond makeup transfer. *arXiv preprint arXiv:1912.03630*, 2019. 3

[27] Tao Li. Beauty learning and counterfactual inference. In *CVPR Workshops*, pages 111–113, 2019. 3

[28] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org, 2017. 3

[29] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1415–1424, 2017. 3

[30] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, and Marc'Aurelio Ranzato. Fader networks: Manipulating images by sliding attributes. In *Advances in Neural Information Processing Systems*, pages 5967–5976, 2017. 3

[31] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Towards open-set identity preserving face synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6713–6722, 2018. 3

[32] Yujun Shen, Bolei Zhou, Ping Luo, and Xiaoou Tang. Facefeat-gan: A two-stage approach for identity-preserving face synthesis. *arXiv preprint arXiv:1812.01288*, 2018. 3

[33] Chris Donahue, Zachary C. Lipton, Akshay Balsubramani, and Julian J. McAuley. Semantically decomposing the latent spaces of generative adversarial networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 3

[34] Yujun Shen, Ping Luo, Junjie Yan, Xiaogang Wang, and Xiaoou Tang. Faceid-gan: Learning a symmetry three-player

gan for identity-preserving face synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 821–830, 2018. 3

[35] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 3

[36] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 3, 4, 5

[37] Nutan Chen, Alexej Klushyn, Richard Kurle, Xueyan Jiang, Justin Bayer, and Patrick van der Smagt. Metrics for deep generative models. In Amos J. Storkey and Fernando Pérez-Cruz, editors, *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, volume 84 of *Proceedings of Machine Learning Research*, pages 1540–1550. PMLR, 2018. 3

[38] Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: on the curvature of deep generative models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 3

[39] Line Kuhnel, Tom Fletcher, Sarang Joshi, and Stefan Sommer. Latent space non-linear statistics. *arXiv preprint arXiv:1805.07632*, 2018. 3

[40] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. 3

[41] Paul Upchurch, Jacob Gardner, Geoff Pleiss, Robert Pless, Noah Snavely, Kavita Bala, and Kilian Weinberger. Deep feature interpolation for image content changes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7064–7073, 2017. 3

[42] Ali Jahanian, Lucy Chai, and Phillip Isola. On the"steerability" of generative adversarial networks. *arXiv preprint arXiv:1907.07171*, 2019. 3

[43] Ceyuan Yang, Yujun Shen, and Bolei Zhou. Semantic hierarchy emerges in deep generative representations for scene synthesis. *arXiv preprint arXiv:1911.09267*, 2019. 3

[44] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B Tenenbaum, William T Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. *arXiv preprint arXiv:1811.10597*, 2018. 3

[45] Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013. 4, 7, 8

[46] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 4, 6

[47] Richard A Haddad, Ali N Akansu, et al. A class of fast gaussian binomial filters for speech and image processing. *IEEE Transactions on Signal Processing*, 39(3):723–727, 1991. 4

[48] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 5

[49] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369. IEEE, 2010. 5

[50] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003. 6

[51] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017. 6

[52] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 6

[53] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7, 8

[54] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011. 7, 8

[55] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999. 7

[56] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 6

[57] Alessandro Del Sole. Introducing microsoft cognitive services. In *Microsoft Computer Vision APIs Distilled*, pages 1–4. Springer, 2018. 6

[58] Face^{++} Face Searching API, 2021. 6

[59] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145–151, 1999. 8