FakeSpotter: A Simple yet Robust Baseline for Spotting AI-Synthesized Fake Faces

Run Wang[1*] , **Felix Juefei-Xu**[2] , **Lei Ma**[3] , **Xiaofei Xie**[1] , **Yihao Huang**[4] , **Jian Wang**[1] , **Yang Liu**[1,5]

[1]Nanyang Technological University, Singapore
[2]Alibaba Group, USA
[3]Kyushu University, Japan
[4]East China Normal University, China
[5]Institute of Computing Innovation, Zhejiang University, China

Abstract

In recent years, generative adversarial networks (GANs) and its variants have achieved unprecedented success in image synthesis. They are widely adopted in synthesizing facial images which brings potential security concerns to humans as the fakes spread and fuel the misinformation. However, robust detectors of these AI-synthesized fake faces are still in their infancy and are not ready to fully tackle this emerging challenge. In this work, we propose a novel approach, named *FakeSpotter*, based on monitoring neuron behaviors to spot AI-synthesized fake faces. The studies on neuron coverage and interactions have successfully shown that they can be served as testing criteria for deep learning systems, especially under the settings of being exposed to adversarial attacks. Here, we conjecture that monitoring neuron behavior can also serve as an asset in detecting fake faces since layer-by-layer neuron activation patterns may capture more subtle features that are important for the fake detector. Experimental results on detecting four types of fake faces synthesized with the state-of-the-art GANs and evading four perturbation attacks show the effectiveness and robustness of our approach.

1 Introduction

With the remarkable development of AI, particularly GANs, seeing is no longer believing nowadays. GANs (*e.g.*, Style-GAN [Karras *et al.*, 2019a], STGAN [Liu *et al.*, 2019], and StarGAN [Choi *et al.*, 2018]) exhibit powerful capabilities in synthesizing human imperceptible fake images and editing images in a natural way. Humans can be easily fooled by these synthesized fake images[1]. Figure 1 presents four typical fake faces synthesized with various GANs, which are really hard for humans to distinguish at the first glance.

The AI-synthesized fake faces not only bring fun to users but also raise security and privacy concerns and even panics to everyone including celebrities, politicians, *etc*. Some apps (*e.g.*, FaceApp, Reflect, and ZAO) employ face-synthesis

---

*Corresponding author, E-mail: runwang1991@gmail.com
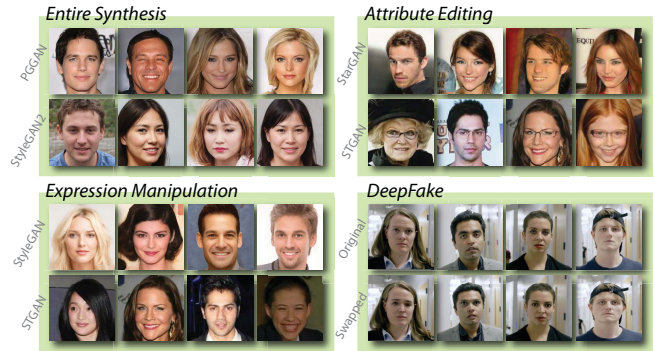[1]https://thispersondoesnotexist.com

Figure 1: Four types of fake faces synthesized with various GANs. For the entire synthesis, the facial images are non-existent faces in the world. For attribute editing, StarGAN changes the color of hair into brown and STAGN wears eyeglasses. For expression manipulation, both StyleGAN and STGAN manipulate the face with a smile expression. For DeepFake, the data is from the DeepFake dataset in FaceForensics++ [Rössler *et al.*, 2019] and they involve face swap.

techniques to provide attractive and interesting services such as face swap, facial expression manipulation with several taps on mobile devices. Unfortunately, abusing AI in synthesizing fake images raises security and privacy concerns such as creating fake pornography [Cole, 2018], where a victim's face can be naturally swapped into a naked body and indistinguishable to humans' eyes with several photos [Zakharov *et al.*, 2019]. Politicians will also be confused by fake faces, for example, fake official statements may be announced with nearly realistic facial expressions and body movements by adopting AI-synthesized fake face techniques. Due to the potential and severe threats of fake faces, it is urgent to call for effective techniques to spot fake faces in the wild. In this paper, the AI-synthesized fake face or fake face means that the face is synthesized with GANs unless particularly addressed.

Entire face synthesis, **facial attribute editing**, **facial expression manipulation**, and **DeepFake** are the four typical fake face synthesis modes with various GANs [Stehouwer *et al.*, 2020]. Entire face synthesis means that a facial image can be wholly synthesized with GANs and the synthesized faces do not exist in the world. Facial attribute editing manipulates single or several attributes in a face like hair, eyeglass, gender, *etc*. Facial expression manipulation alters one's facial expression or transforms facial expressions among persons. Deep-Fake is also known as the identity swap. It normally swaps

synthesized face between different persons and is widely applied in producing fake videos [Agarwal *et al.*, 2019]. More recently, there is some work that starts to study this topic. However, none of the previous approaches fully tackle the aforementioned four types of fake faces and thoroughly evaluate their robustness against perturbation attack with various transformations in order to show their potentials in dealing with fakes in the wild.

In this paper, we propose a novel approach, named *FakeSpotter*, which detects fake faces by monitoring neuron behaviors of deep face recognition (FR) systems with a simple binary-classifier. Specifically, FakeSpotter leverages the power of deep FR systems in learning the representations of faces and the capabilities of neurons in monitoring the layer-by-layer behaviors which can capture more subtle differences for distinguishing between real and fake faces.

To evaluate the effectiveness of FakeSpotter in detecting fake faces and its robustness against perturbation attacks, we collect numerous high-quality fake faces produced with the state-of-the-art (SOTA) GANs. For example, our entire synthesized fake faces are generated with 1) the freshly released StyleGAN2 [Karras *et al.*, 2019b], 2) the newest STGAN [Liu *et al.*, 2019] that performs facial attributes editing, 3) *DeepFake* that is composed of public datasets (*e.g.*, Face-Forensics++ and *Celeb-DF* [Li *et al.*, 2019]), and 4) the Facebook announced real-world DeepFake detection competition dataset (*i.e.*, DFDC). Experiments are evaluated on our collected four types of high-quality fake faces and the results demonstrate the effectiveness of FakeSpotter in spotting fake faces and its robustness in tackling four perturbation attacks (*e.g.*, **adding noise**, **blur**, **compression**, and **resizing**). FakeSpotter also outperforms prior work AutoGAN [Zhang *et al.*, 2019b] and gives an average detection accuracy of more than 90% on the four types of fake faces. The average performance measured by the AUC score is merely down less than 3.77% in tackling the four perturbation attacks under various intensities.

Our main contributions are summarized as follows.

- **New observation of neurons in spotting AI-synthesized fake faces.** We observe that layer-by-layer neuron behaviors can be served as an asset for distinguishing fake faces. Additionally, they are also robust against various perturbation attacks at various magnitudes.

- **Presenting a new insight for spotting AI-synthesized fake faces by monitoring neuron behaviors.** We propose the first neuron coverage based fake detection approach that monitors the layer-by-layer neuron behaviors in deep FR systems. Our approach provides a novel insight for spotting AI aided fakes with neuron coverage techniques.

- **Performing the first comprehensive evaluation on four typical AI-synthesized fake faces and robustness against four common perturbation attacks.** Experiments are conducted on our collected high-quality fake faces synthesized with the SOTA GANs and real dataset like DFDC. Experimental results have demonstrated the effectiveness and robustness of our approach.

2 Related Work

2.1 Image Synthesis

GANs have made impressive progress in image synthesis [Zhu *et al.*, 2017; Yi *et al.*, 2017] which is the most widely studied area of the applications of GANs since it is first proposed in 2014 [Goodfellow *et al.*, 2014]. The generator in GANs learns to produce synthesized samples that are almost identical to real samples, while the discriminator learns to differentiate between them. Recently, various GANs are proposed for facial image synthesis and manipulation.

In entire face synthesis, PGGAN [Karras *et al.*, 2018] and StyleGAN, created by NVIDIA, produce faces in high resolution with unprecedented quality and synthesize non-existent faces in the world. STGAN and StarGAN focus on face editing which manipulates the attributes and expressions of humans' faces, *e.g.*, changing the color of hair, wearing eyeglasses, and laughing with a smile or showing feared expression, *etc*. *FaceApp* and *FaceSwap* employ GANs to generate *DeepFake* which involves identity swap.

Currently, GANs can be well applied in synthesizing entire fake faces, editing facial attributes, manipulating facial expressions, and swapping identities among persons (*a.k.a.* DeepFake). Fake faces synthesized with the SOTA GANs are almost indistinguishable to humans in many cases. We are living in a world where we cannot believe our eyes anymore.

2.2 Fake Face Detection

Some researchers employ traditional forensics-based techniques to spot fake faces/images. These work inspect the disparities in pixel-level between real and fake images. However, they are either susceptible to perturbation attacks like compression that is common in producing videos with still images [Böhme and Kirchner, 2013], or do not scale well with the increasing amount of training data, as commonly found in shallow learning-based fake detection methods such as [Buchana *et al.*, 2016]. Another line in detecting fake images is leveraging the power of deep neural networks (DNNs) in learning the differences between real and fake which are also vulnerable to perturbation attacks like adding human-imperceptible noise [Goodfellow *et al.*, 2015].

In forensics-based fake detection, Nataraj *et al*. [Nataraj *et al.*, 2019] employ a DNN model to learn the representation in order to compute co-occurrence matrices on the RGB channels. McCloskey *et al*. [McCloskey and Albright, 2018] observe that the frequency of saturated pixels in GAN-synthesized fake images is limited as the generator's internal values are normalized and the formation of a color image is vastly different from real images which are sensitive to spectral analysis. Different from forensics-based fake detection, Stehouwer *et al*. [Stehouwer *et al.*, 2020] introduce an attention-based layer in convolutional neural networks (CNNs) to improve fake identification performance. Wang *et al*. [Wang *et al.*, 2020] use ResNet-50 to train a binary-classifier for CNN-synthesized images detection. AutoGAN [Zhang *et al.*, 2019b] trains a classifier to identify the artifacts inducted in the up-sampling component of the GAN.

Other work explores various *ad-hoc* features to investigate artifacts in images for differentiating real and synthesized fa-

cial images. For example, mismatched facial landmark points [Yang *et al.*, 2019], fixed size of facial area [Li and Lyu, 2019], and unique fingerprints of GANs [Zhang *et al.*, 2019b; Yu *et al.*, 2019], *etc.* These approaches will be invalid in dealing with improved or advanced GANs. Existing works are sensitive to perturbation attacks, but robustness is quite important for a fake detector deployed in the wild.

3 Our Method

In this section, we first give our basic insight and present an overview of FakeSpotter in spotting fake faces by monitoring neuron behaviors. Then, a neuron coverage criteria mean neuron coverage (*MNC*) is proposed for capturing the layer-by-layer neuron activation behaviors. Finally, FakeSpotter differentiates four different types of fake faces with a simple binary-classifier.

3.1 Insight

Neuron coverage techniques are widely adopted for investigating the internal behaviors of DNNs and play an important role in assuring the quality and security of DNNs. It explores activated neurons whose output values are larger than a threshold. The activated neurons serve as another representation of inputs that preserves the learned layer-by-layer representations in DNNs. Studies have shown that activated neurons exhibit strong capabilities in capturing more subtle features of inputs that are important for studying the intrinsic of inputs. DeepXplore [Pei *et al.*, 2017] first introduces neuron coverage as metrics for DNN testing to assure their qualities. Some work exploits the critical activated neurons in layers to detect adversarial examples for securing DNNs [Ma *et al.*, 2019b; Ma *et al.*, 2019a; Ma *et al.*, 2018b; Zhang *et al.*, 2019a].

Our work is motivated by the power of layer-wise activated neurons in capturing the subtle features of inputs which could be used for amplifying the differences between real and synthesized facial images. Based on this insight, we propose FakeSpotter by monitoring the neuron behaviors in deep FR systems (*e.g.*, VGG-Face) for fake face detection. Deep FR systems have made incredible progress in face recognition but are still vulnerable to identifying fake faces [Korshunov and Marcel, 2018]. In Figure 2, we present an overview of FakeSpotter using layer-wise neuron behavior as features with a simple binary-classifier to identify real and fake faces.

3.2 Monitoring Neuron Behaviors

In DNNs, a neuron is a basic unit and the final layer neuron outputs are employed for prediction. Given an input of trained DNN, the activation function ϕ (*e.g.*, Sigmoid, ReLU) computes the output value of neurons with connected neurons x_i in the previous layers, weights matrix W_i^k, and bias b_j. Activated neurons in each individual layers are determined by whether the output value is higher than a threshold ξ.

In this work, we propose a new neuron coverage criterion, named mean neuron coverage (*MNC*), for determining the threshold ξ. Existing approaches [Ma *et al.*, 2018a;
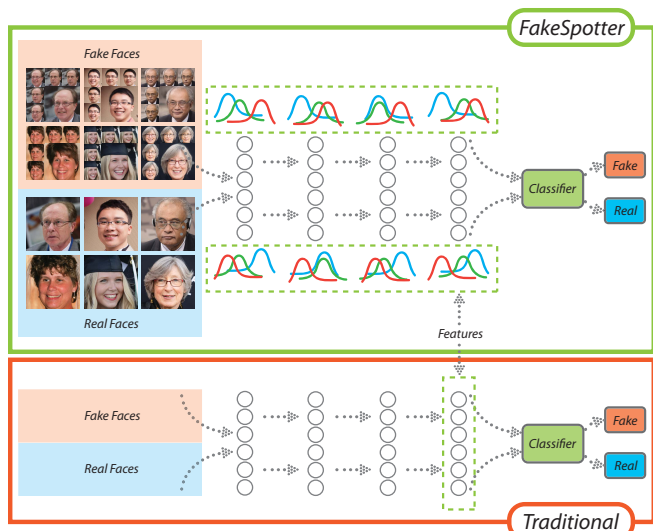

Figure 2: An overview of the proposed fake face detection method, FakeSpotter. Compared to the traditional learning-based method (shown at the bottom), the FakeSpotter uses layer-wise neuron behavior as features, as opposed to final-layer neuron output. Our approach uses a shallow neural network as the classifier while traditional methods rely on deep neural networks in classification.

Xie *et al.*, 2019] in calculating threshold ξ are mostly designed for testing DNNs and are not applicable for fake detection. Pei *et al.* [Pei *et al.*, 2017] define a global threshold for activating neurons in all layers, which is too rough.

In DNNs, each layer plays their own unique roles in learning the representations of inputs [Mahendran and Vedaldi, 2015]. Here, we introduce another strategy by specifying a threshold ξ_l for each layer l. The threshold ξ_l is the average value of neuron outputs in each layer for given training inputs. The layer l is the convolutional and fully-connected layers which are valuable layers preserving more representation information in the model. Specifically, we calculate the threshold ξ_l for each layer with the following formula:

$$\xi_l = \frac{\sum_{n \in N, t \in \mathcal{T}} \delta(n, t)}{|N| \cdot |\mathcal{T}|} \tag{1}$$

where N represents a set of neurons in the lth layer and $|N|$ is the total number of neurons in the N, $\mathcal{T} = \{t_1, t_2, ..., t_k\}$ is a set of training inputs and $|\mathcal{T}|$ indicates the number of training inputs in $\mathcal{T}$, $\delta(n, t)$ calculates the neurons output value where n is the neuron in N and t denotes the input in $\mathcal{T}$. Finally, our neuron coverage criterion *MNC* determines whether a neuron in the lth layer is activated or not by checking whether its output value is higher than the threshold ξ_l. We define the neuron coverage criterion *MNC* for each layer l as follows:

$$MNC(l, t) = |\{n | \forall n \in l, \delta(n, t) > \xi_l\}| \tag{2}$$

where t represents the input, n is the neuron in layer l, δ is a function for computing the neuron output value, and ξ_l is the threshold of the l-th layer calculated by formula (1).

3.3 Detecting Fake Faces

As described above, we capture the layer-wise activated neurons with *MNC*. We train a simple binary-classifier with shal-

Algorithm 1: Algorithm for detecting fake faces with neuron coverage in deep FR systems.

Input : Training dataset of fake and real faces $\mathcal{T}$, Test dataset of fake and real faces $\mathcal{D}$, Pre-trained deep FR model $\widetilde{M}$

Output: Label tag

1 L is the convolutional and fully-connected layers in $\widetilde{M}$.
2 ▷ Determine the threshold of neuron activation for each layer.
3 **for** $t \in \mathcal{T}$ **do**
4   N is a set of neurons in the lth layer of $\widetilde{M}$.
5   S saves neuron output value for a given input t.
6   **for** $l \in L,\ n \in N$ **do**
7         $S_l = \sum \delta(n, t)$
8         $\xi_l = \frac{1}{|L|} \cdot S$
9 ▷ Train a binary-classifier for detecting fake/real faces.
10 V counts activated neurons in L.
11 **for** $t \in \mathcal{T}$ **do**
12   **for** $l \in L,\ n \in N$ **do**
13   **if** $\delta(n, t) > \xi_l$ **then**
14            $V_l \leftarrow n$
15 Train a binary-classifier $\widetilde{C}$ with inputs V.
16 ▷ Predict whether a face from test dataset $\mathcal{D}$ is real or fake.
17 **for** $d \in \mathcal{D}$ **do**
18      $tag \leftarrow \operatorname{argmax} \widetilde{C}(d)$
19 **return** tag

Fake Faces	GAN Type	Manipulation	Real Source	Collection
Entire Synthesis	PGGAN	full	CelebA	self-synthesis
	StyleGAN2	full	FFHQ	officially released
Attribute Editing	StarGAN	brown-hair	CelebA	self-synthesis
	STGAN	eyeglasses	CelebA	self-synthesis
Expression Manipulation	StyleGAN	ctrl. smile intensity	FFHQ	self-synthesis
	STGAN	smile	CelebA	self-syntheis
DeepFake	F. F. ++	face swap	unknown	FaceForensics++
	DFDC	face/voice swap	unknown	Kaggle dataset
	Celeb-DF	face swap	YouTube	Celeb-DF(V2)

Table 1: Statistics of collected fake faces dataset. Column *Manipulation* indicates the manipulated region in face. Column *Real Source* denotes the source of real face for producing fake faces. Last column *Collection* means the way of producing fake faces, synthesized by ourselves or collected from public dataset. *F.F. ++* denotes FaceForensics++ dataset.

faces produced with the SOTA techniques and investigate its robustness against four common perturbation attacks. We present the experimental results of detection performance with a comparison of recently published work AutoGAN [Zhang *et al.*, 2019b] in Section 4.2 and robustness analysis in Section 4.3. In Section 4.4, we provide the comparison results in detecting a public DeepFake dataset *Celeb-DF*.

4.1 Experimental Setup

Data Collection. In our experiments, real face samples are collected from CelebA [Liu *et al.*, 2015] and Flicker-Faces-HQ (FFHQ) since they exhibit good diversity. We also utilize original real images provided by the public dataset FaceForensics++, DFDC[2], and *Celeb-DF*. To ensure the diversity and high-quality of our fake face dataset, we use the newest GANs for synthesizing fake faces (*e.g.*, StyleGAN2) using the public dataset (*e.g.*, *Celeb-DF*), and real dataset such as DFDC dataset. The DFDC dataset is the officially released version rather than the preview edition. Table 1 presents the statistics of our collected fake face dataset.

Implementation Details. We design a shallow neural network with merely five fully-connected layers as our binary-classifier for spotting fakes. The optimizer is SGD with momentum 0.9 and the starting learning rate is 0.0001, with a decay of $1e$-6. The loss function is binary cross-entropy.

In monitoring neuron behaviors with *MNC*, we utilize VGG-Face[3] with ResNet50 as backend architecture for capturing activated neurons as it can well balance detection performance and computing overhead. Our approach is generic to FR systems, which could be easily extended to other deep FR systems. In evaluating the robustness in tackling perturbation attacks, we select four common transformations, namely *compression*, *resizing*, *adding noise*, and *blur*.

Training and Test Dataset. Using the training dataset $\mathcal{T}$, we train the model with 5,000 real and 5,000 fake fakes for each individual GAN. In the test dataset $\mathcal{D}$, we use 1,000 real and 1,000 fake faces for evaluation. The training and test dataset are based on different identities. The training

low neural networks. The input of our classifier is the *general* neuron behavior rather than the *ad-hoc* raw pixels like traditional image classification models. Raw pixels could be easily perturbed by attackers and trigger erroneous behaviors.

Algorithm 1 describes the procedure of fake face detection. First, the thresholds for determining neuron activation in each layer are identified by our proposed neuron coverage criterion *MNC* with fake and real faces as the training dataset, denoted as $\mathcal{T}$. Then, a feature vector for each input face is formed as the number of activated neurons in each layer. Let $F = \{f_1, f_2, ..., f_i, ..., f_m\}$ and $R = \{r_1, r_2, ..., r_j, ..., r_m\}$ represent the feature vector of fake and real input faces respectively, where f_i and r_j are the number of activated neurons in the ith and jth layer, m is the total number of layers in deep FR system. Finally, we train a supervised binary-classifier, denoted as $\widetilde{C}$, by receiving the formed feature vectors of fake and real faces as inputs to predict the input is real or fake.

In prediction, an input face should be processed by a deep FR system to extract the neuron coverage behaviors with our proposed criterion *MNC*, namely the number of activated neurons in each layer. The activated neurons are formed as a feature to represent the input face. Then, the trained binary-classifier predicts whether the input is a real or fake face.

4 Experiments

In this section, we conduct experiments to evaluate the effectiveness of FakeSpotter in spotting four types of fake

---

[2]DeepFakes Detection Challenge (DFDC) Dataset by Facebook. https://www.kaggle.com/c/DeepFake-detection-challenge
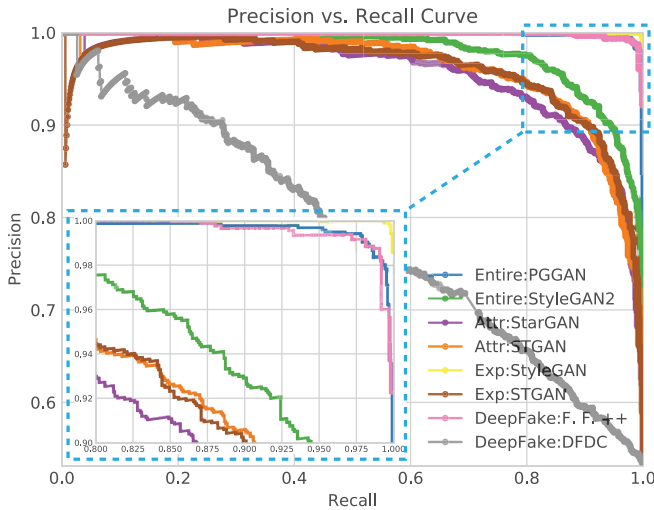[3]https://github.com/rcmalli/keras-vggface

Figure 3: Precision-recall curves of the four types of fake faces. The curve computes precision-recall for different probability thresholds.

Figure 4: Four perturbation attacks under different intensities. Legends refer to Figure3.

dataset $\mathcal{T}$ and test dataset $\mathcal{D}$ are employed for evaluating the effectiveness and robustness of FakeSpotter. The *Celeb-DF* dataset provides another independent training and test dataset for comparing the performance with existing thirteen methods in detecting fake videos on *Celeb-DF*.

Evaluation Metrics. In spotting real and fake faces, we adopt eight popular metrics to get a comprehensive performance evaluation of FakeSpotter. Specifically, we report precision, recall, F1-score, accuracy, AP (average precision), AUC (area under curve) of ROC (receiver operating characteristics), FPR (false positive rate), and FNR (false negative rate), respectively. We also use the AUC as a metric to evaluate the performance of FakeSpotter in tackling various perturbation attacks.

All our experiments are conducted on a server running Ubuntu 16.04 system on a total 24 cores 2.20GHz Xeon CPU with 260GB RAM and two NVIDIA Tesla P40 GPUs with 24GB memory for each.

4.2 Detection Performance

In evaluating the performance of FakeSpotter in detecting fake faces and its generalization to different GANs. We select four totally different types of fake faces synthesized with various GANs and compare with prior work AutoGAN. To get a comprehensive performance evaluation, we use eight different metrics to report the detection rate and false alarm rate.

Table 2 shows the performance of FakeSpotter and prior work AutoGAN in detecting fake faces measured by eight different metrics. AutoGAN is a recent open source work that leverages the artifacts existed in GAN-synthesized images and detects the fake image with a deep neural network-based classifier. Furthermore, to illustrate the performance of FakeSpotter in balancing the precision and recall, we present the precision and recall curves in Figure 3 as well.

Experimental results demonstrate that FakeSpotter outperforms AutoGAN and achieves competitive performance with a high detection rate and low false alarm rate in spotting the four typical fake faces synthesized by GANs. We also find that FakeSpotter achieves a better balance between precision
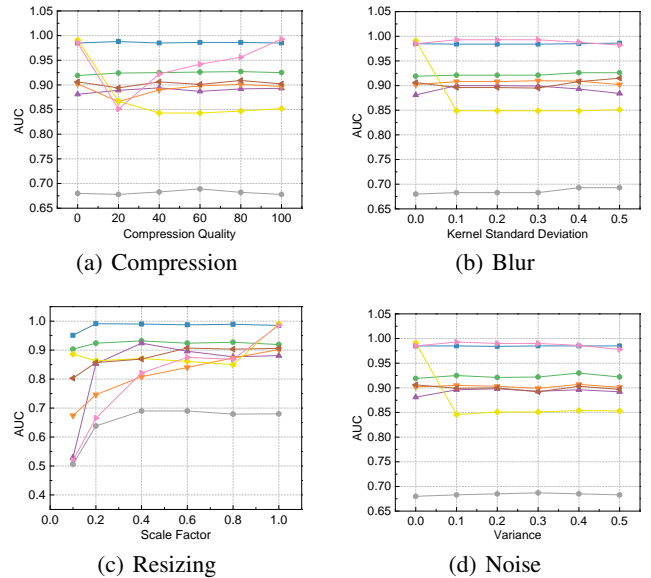
and recall on four types of fake faces from Figure 3. Further, we observe some interesting findings from Table 2.

First, fake faces synthesized with advanced GANs are difficult to be spotted by FakeSpotter. For example, in entire synthesis, FakeSpotter detects PGGAN with an accuracy of 98.6%, but gives an accuracy of 91.8% on StyleGAN2 (the best performed GAN in entire synthesis and just released by NVIDIA). In addition, entire face synthesis is easily spotted than partial manipulation of fake faces that may contain less fake footprints. These two findings indicate that well-designed GANs and minor manipulations could produce more realistic and harder-to-spot fake faces.

In Table 2, the performance of FakeSpotter in detecting DFDC is not ideal as other types of fake faces since fake faces in DFDC could be either a face swap or voice swap (or both) claimed by Facebook. In our experiments, some false alarms could be caused by the voice swap which is out the scope of FakeSpotter. A potential idea of detecting fakes with random face and voice swap combination is inferring the characteristic physical features of faces from voice, and vice versa.

4.3 Robustness Analysis

Robustness analysis aims at evaluating the capabilities of FakeSpotter against perturbation attacks since image transformations are common in the wild, especially in creating fake videos. The transformations should be less sensitive to human eyes. Here, we mainly discuss the performance of FakeSpotter in tackling four different perturbation attacks under various intensities. We utilize the AUC as metrics for the performance evaluation. Figure 4 plots the experimental results of FakeSpotter against the four perturbation attacks.

In Figure 4, the compression quality measures the intensity of compression, the maximum and minimum value are 100 and 0, respectively. Blur means that we employ Gaussian blur to faces. The value of Gaussian kernel standard deviation is

Fake Faces	GAN	precision		recall		F1		accuracy		AP		AUC		FPR		FNR	
		F. S.	A. G.	F. S.	A. G.	F. S.	A. G.	F. S.	A. G.	F. S.	A. G.	F. S.	A. G.	F. S.	A. G.	F. S.	A. G.
Entire Synthesis	PGGAN	0.986	0.926	0.987	0.974	0.986	0.949	0.986	0.948	0.979	0.915	0.985	0.948	0.013	0.026	0.016	0.078
	StyleGAN2	0.912	0.757	0.924	0.663	0.918	0.707	0.919	0.725	0.881	0.670	0.919	0.725	0.076	0.337	0.087	0.213
Attribute Editing	StarGAN	0.901	0.690	0.865	0.567	0.883	0.622	0.88	0.656	0.851	0.608	0.881	0.656	0.135	0.433	0.104	0.255
	STGAN	0.885	0.555	0.918	0.890	0.901	0.683	0.902	0.588	0.852	0.549	0.902	0.588	0.082	0.11	0.114	0.715
Expression Manipulation	StyleGAN	1.0	0.736	0.983	0.920	0.991	0.818	0.991	0.795	0.992	0.717	0.991	0.795	0.017	0.08	0.0	0.33
	STGAN	0.898	0.0	0.913	0.0	0.905	0.0	0.906	0.5	0.863	0.5	0.906	0.5	0.087	1.0	0.102	0.0
DeepFake	FaceForensics++	0.978	0.508	0.992	0.629	0.985	0.562	0.985	0.511	0.973	0.505	0.985	0.511	0.008	0.371	0.021	0.608
	DFDC	0.691	0.536	0.719	1.0	0.705	0.698	0.682	0.536	0.645	0.536	0.680	0.5	0.281	0.0	0.359	1.0
Average Performance (first three types)		0.930	0.611	0.932	0.669	0.931	0.630	0.931	0.702	0.903	0.660	0.931	0.702	0.068	0.331	0.071	0.265
Average Performance (all four types)		0.906	0.589	0.913	0.705	0.909	0.630	0.906	0.657	0.880	0.625	0.906	0.653	0.087	0.295	0.10	0.40

Table 2: Performance of FakeSpotter (*F. S.*) and AutoGAN (*A. G.*) in spotting the four types of fake faces. PGGAN and StyleGAN2 produce entire synthesized facial images. In attribute editing, StarGAN manipulates the color of the hair with brown, STGAN manipulates face by wearing eyeglasses. In Expression manipulation, StyleGAN and STGAN manipulate the expression of faces with the smile while StyleGAN can control the intensity of the smile. Average performance is an average results over the fake faces. Here, we provide two kinds of average performance, average performance on still images (including the first three types of fake faces) and all the four types of fake faces.

adjusted to control the intensity of blur while maintaining the Gaussian kernel size to (3, 3) unchanged. In resizing, scale factor is used for controlling the size of an image in horizontal and vertical axis. We add Gaussian additive noise to produce images with noise and the variance is used for controlling the intensity of the noise.

Experimental results demonstrated the robustness of the FakeSpotter in tackling the four common perturbation attacks. We find that the AUC score of FakeSpotter maintains a minor fluctuation range when the intensity of perturbation attacks increased. Due to the severe artifacts in F.F.++ and high intensity of facial expression manipulation in StyleGAN, their variation is a little obvious. The average AUC score of all the four types of fake faces decreased less than 3.77% on the four perturbation attacks under five different intensities.

4.4 Performance on *Celeb-DF(v2)*

Celeb-DF [Li *et al.*, 2019] is another large-scale DeepFake video dataset with many different subjects (*e.g.*, ages, ethic groups, gender) and contains more than 5,639 high-quality fake videos. In their project website, they provide some comparison results of existing video detection methods on several DeepFake videos including *Celeb-DF*. There are two versions of *Celeb-DF* dataset, *Celeb-DF(v1)* and *Celeb-DF(v2)* dataset, a superset of *Celeb-DF(v1)*.

We use *Celeb-DF(v2)* dataset for demonstrating the effectiveness of FakeSpotter further and get a more comprehensive comparison with existing work on fake video detection. We also utilize AUC score as metrics for evaluating our approach FakeSpotter as AUC score is served as the metrics in *Celeb-DF* project for comparing with various methods. Figure 5 shows the performance of FakeSpotter in spotting fake videos on *Celeb-DF(v2)*. Experimental results show that FakeSpotter reaches an AUC score 66.8% on the test dataset provided in *Celeb-DF(v2)* and outperforms all the existing work listed.

According to the experimental results in Figure 5, fake video detection is still a challenge, especially when some high-quality fake videos utilize various unknown techniques.
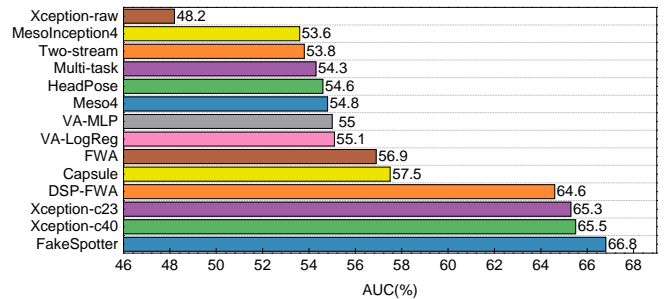

Figure 5: AUC score of various methods on *Cele-DF(V2)* dataset.

4.5 Discussion

Our approach achieves impressive results in detecting various types of fake faces and is robust against several common perturbation attacks. However, there are also some limitations. The performance of FakeSpotter in spotting DFDC is not as ideal as other types of fake faces. One of the main reasons is that fake faces in DFDC involve two different domain fake, face swap and voice swap. However, our approach only focuses on facial images without any consideration of the voice. This suggests that producing fake multimedia by incorporating various seen and unseen techniques may be a trend in the future. It poses a big challenge to the community and calls for effective approaches to detect these perpetrating fakes. In an adversarial environment, attackers could add adversarial noise to evade our detection, and there is a trade-off between generating imperceptible facial images and the success of evasion.

5 Conclusion and Future Research Directions

We proposed the FakeSpotter, the first neuron coverage based approach for fake face detection, and performed an extensive evaluation of the FakeSpotter on fake detection challenges with four typical SOTA fake faces. FakeSpotter demonstrates its effectiveness in achieving high detection rates and low false alarm rates. In addition, our approach also exhibits robustness against four common perturbation attacks. The neuron coverage based approach presents a new insight for detecting fakes, which we believe could also be extended to other fields like fake speech detection.

Everyone could potentially fall victim to the rapid development of AI techniques that produce fake artifacts (*e.g.*, fake speech, fake videos). The arms race between producing and fighting fakes is on an endless road. Powerful defense mechanisms should be developed for protecting us against AI risks. However, a public database with benchmark containing diverse high-quality fake faces produced by the SOTA GANs is still lacking in the community which could be our future work. In addition, an interplay between our proposed method and novel fake localization methods [Huang *et al.*, 2020] is also worth pursuing. Beyond DeepFake detection, we conjecture that the FakeSpotter can work well in tandem with non-additive noise adversarial attacks *e.g.*, [Wang *et al.*, 2019; Guo *et al.*, 2020] where the attacked images do not reveal the noise pattern and are much harder to accurately detect.

Acknowledgments

References

[Agarwal *et al.*, 2019] Shruti Agarwal, Hany Farid, Yuming Gu, Mingming He, Koki Nagano, and Hao Li. Protecting world leaders against deep fakes. In *CVPR Workshops*, pages 38–45, 2019.

[Böhme and Kirchner, 2013] Rainer Böhme and Matthias Kirchner. Counter-forensics: Attacking image forensics. In *Digital Image Forensics*, pages 327–366. Springer, 2013.

[Buchana *et al.*, 2016] P. Buchana, I. Cazan, M. Diaz-Granados, F. Juefei-Xu, and M.Savvides. Simultaneous Forgery Identification and Localization in Paintings Using Advanced Correlation Filters. In *ICIP*, 2016.

[Choi *et al.*, 2018] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, pages 8789–8797, 2018.

[Cole, 2018] Samantha Cole. We Are Truly F—ed: Everyone Is Making AI-Generated Fake Porn Now. https://www.vice.com/en_us/article/bjye8a/reddit-fake-porn-app-daisy-ridley/, 2018. (Jan 25 2018).

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014.

[Goodfellow *et al.*, 2015] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[Guo *et al.*, 2020] Qing Guo, Felix Juefei-Xu, Xiaofei Xie, Lei Ma, Jian Wang, Wei Feng, and Yang Liu. ABBA: Saliency-Regularized Motion-Based Adversarial Blur Attack. *arXiv preprint arXiv:2002.03500*, 2020.

[Huang *et al.*, 2020] Yihao Huang, Felix Juefei-Xu, Run Wang, Qing Guo, Xiaofei Xie, Lei Ma, Jianwen Li, Weikai Miao, Yang Liu, and Geguang Pu. FakeLocator: Robust Localization of GAN-Based Face Manipulations. *arXiv preprint arXiv:2001.09598*, 2020.

[Karras *et al.*, 2018] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. *ICLR*, 2018.

[Karras *et al.*, 2019a] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019.

[Karras *et al.*, 2019b] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. *arXiv preprint arXiv:1912.04958*, 2019.

[Korshunov and Marcel, 2018] Pavel Korshunov and Sébastien Marcel. Deepfakes: a new threat to face recognition? assessment and detection. *arXiv preprint arXiv:1812.08685*, 2018.

[Li and Lyu, 2019] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. *CVPRW*, 2, 2019.

[Li *et al.*, 2019] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-DF: A new dataset for deepfake forensics. *arXiv preprint arXiv:1909.12962*, 2019.

[Liu *et al.*, 2015] Ziwei Liu, Ping Luo, Xiaogang W., and X. Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[Liu *et al.*, 2019] Ming Liu, Yukang Ding, Min Xia, Xiao Liu, Errui Ding, Wangmeng Zuo, and Shilei Wen. STGAN: A unified selective transfer network for arbitrary image attribute editing. In *CVPR*, pages 3673–3682, 2019.

[Ma *et al.*, 2018a] Lei Ma, Felix Juefei-Xu, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Chunyang Chen, Ting Su, Li Li, Yang Liu, et al. DeepGauge: Multi-granularity testing criteria for deep learning systems. In *ASE*, pages 120–131, 2018.

[Ma *et al.*, 2018b] Lei Ma, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Felix Juefei-Xu, Chao Xie, Li Li, Yang Liu, Jianjun Zhao, et al. Deepmutation: Mutation testing of deep learning systems. In *ISSRE*, 2018.

[Ma *et al.*, 2019a] Lei Ma, Felix Juefei-Xu, Minhui Xue, Bo Li, Li Li, Yang Liu, and Jianjun Zhao. Deepct: Tomographic combinatorial testing for deep learning systems. In *SANER*, 2019.

[Ma *et al.*, 2019b] Shiqing Ma, Yingqi Liu, Guanhong Tao, Wen-Chuan Lee, and Xiangyu Zhang. NIC: Detecting adversarial samples with neural network invariant checking. In *NDSS*, 2019.

[Mahendran and Vedaldi, 2015] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, June 2015.

[McCloskey and Albright, 2018] Scott McCloskey and Michael Albright. Detecting GAN-generated imagery using color cues. *arXiv preprint arXiv:1812.08247*, 2018.

[Nataraj *et al.*, 2019] Lakshmanan Nataraj, Tajuddin Manhar Mohammed, BS Manjunath, Shivkumar Chandrasekaran, Arjuna Flenner, Jawadul H Bappy, and Amit K Roy-Chowdhury. Detecting GAN generated fake images using co-occurrence matrices. *arXiv preprint arXiv:1903.06836*, 2019.

[Pei *et al.*, 2017] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. DeepXplore: Automated whitebox testing of deep learning systems. In *SOSP*, 2017.

[Rössler *et al.*, 2019] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics++: Learning to detect manipulated facial images. In *ICCV*, 2019.

[Stehouwer *et al.*, 2020] Joel Stehouwer, Hao Dang, Feng Liu, Xiaoming Liu, and Anil Jain. On the detection of digital face manipulation. *CVPR*, 2020.

[Wang *et al.*, 2019] Run Wang, Felix Juefei-Xu, Xiaofei Xie, Lei Ma, Yihao Huang, and Yang Liu. Amora: Black-box Adversarial Morphing Attack. *arXiv preprint arXiv:1912.03829*, 2019.

[Wang *et al.*, 2020] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. CNN-generated images are surprisingly easy to spot... for now. *CVPR*, 2020.

[Xie *et al.*, 2019] Xiaofei Xie, Lei Ma, Felix Juefei-Xu, Minhui Xue, Hongxu Chen, Yang Liu, Jianjun Zhao, Bo Li, Jianxiong Yin, and Simon See. DeepHunter: a coverage-guided fuzz testing framework for deep neural networks. In *ISSTA*, 2019.

[Yang *et al.*, 2019] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. In *ICASSP*, 2019.

[Yi *et al.*, 2017] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *ICCV*, pages 2849–2857, 2017.

[Yu *et al.*, 2019] Ning Yu, Larry S Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *ICCV*, pages 7556–7566, 2019.

[Zakharov *et al.*, 2019] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. *arXiv preprint arXiv:1905.08233*, 2019.

[Zhang *et al.*, 2019a] Jie M Zhang, Mark Harman, Lei Ma, and Yang Liu. Machine learning testing: Survey, landscapes and horizons. *arXiv preprint arXiv:1906.10742*, 2019.

[Zhang *et al.*, 2019b] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and simulating artifacts in gan fake images. *arXiv preprint arXiv:1907.06515*, 2019.

[Zhu *et al.*, 2017] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.