FakePolisher: Making DeepFakes More Detection-Evasive by Shallow Reconstruction

Yihao Huang[1], Felix Juefei-Xu[2], Run Wang[3,*], Qing Guo[3], Lei Ma[4], Xiaofei Xie[3], Jianwen Li[1], Weikai Miao[1], Yang Liu[3,5], Geguang Pu[1,*]

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

ABSTRACT

At this moment, GAN-based image generation methods are still imperfect, whose upsampling design has limitations in leaving some certain artifact patterns in the synthesized image. Such artifact patterns can be easily exploited (by recent methods) for difference detection of real and GAN-synthesized images. However, the existing detection methods put much emphasis on the artifact patterns, which can become futile if such artifact patterns were reduced.

Towards reducing the artifacts in the synthesized images, in this paper, we devise a simple yet powerful approach termed FakePolisher that performs shallow reconstruction of fake images through a learned linear dictionary, intending to effectively and efficiently reduce the artifacts introduced during image synthesis. In particular, we first train a dictionary model to capture the patterns of real images. Based on this dictionary, we seek the representation of DeepFake images in a low dimensional subspace through linear projection or sparse coding. Then, we are able to perform shallow reconstruction of the 'fake-free' version of the DeepFake image, which largely reduces the artifact patterns DeepFake introduces. The comprehensive evaluation on 3 state-of-the-art DeepFake detection methods and fake images generated by 16 popular GAN-based fake image generation techniques, demonstrates the effectiveness of our technique. Overall, through reducing artifact patterns, our technique significantly reduces the accuracy of the 3 state-of-the-art fake image detection methods, *i.e.*, 47% on average and up to 93% in the worst case.

Our results confirm the limitation of current fake detection methods and calls the attention of DeepFake researchers and practitioners for more general-purpose fake detection techniques.

CCS CONCEPTS

• **Security and privacy → Human and societal aspects of security and privacy**; • **Computing methodologies → Computer vision**.

Yihao Huang's email: huangyihao22@gmail.com
* Corresponding authors. E-mail: runwang1991@gmail.com, ggpu@sei.ecnu.edu.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MM '20, October 12–16, 2020, Seattle, WA, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413732

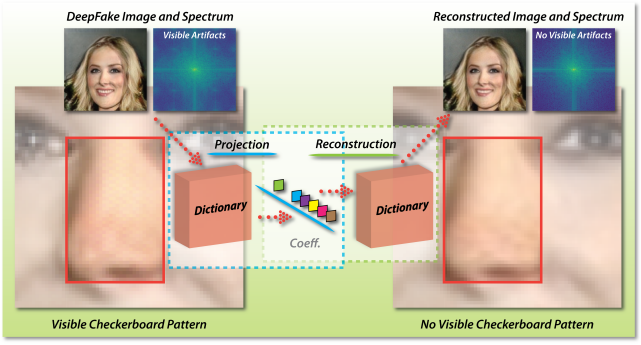

Figure 1: Before and after *FakePolisher* is applied: the left image is a fake image generated from the DeepFake method [11]. In the enlarged view, we can easily find obvious checkerboard patterns. Corresponding to these artifacts are the bright blobs at 1/4 and 3/4 of the width/height in the spectrum of the fake image. The artifacts are introduced by the upsampling methods of GAN-based image generation methods. We propose a shallow reconstruction method based on dictionary learning to remove the artifacts. The right image is the reconstructed image, which does not has obvious artifact in its enlarged view and spectrum.

KEYWORDS

Computer vision; DeepFake; Shallow Reconstruction

ACM Reference Format:
Yihao Huang, Felix Juefei-Xu, Run Wang, Qing Guo, Lei Ma, Xiaofei Xie, Jianwen Li, Weikai Miao, Yang Liu, Geguang Pu. 2020. *FakePolisher*: Making DeepFakes More Detection-Evasive by Shallow Reconstruction. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3394171.3413732

1 INTRODUCTION

The recent advances of fake information generation draw lots of attention and concern, with frequent and widespread media coverage and argument. Up to the present, **DeepFake** (*e.g.*, fake images, audios and videos) has become a real threat to our society due to its realism and impact scopes. Even worse, lots of tools such as FaceApp [1], ZAO [2] are available for fake image generation, further exacerbating the situation. In general, the backend techniques of DeepFake are mostly based on generative adversarial networks (GANs), which are used for synthesizing facial images and voices. Many of the current state-of-the-art DeepFake techniques reach a level that cannot be easily captured by human perceptions. For example, it can be really hard for humans to distinguish the real videos from faked ones only by our eyes and ears [43, 47]. We are entering an era where we cannot simply trust our eyes and ears. According to [20], humans detection peak accuracy reaches only

75% [46], where the real images are from a well-known dataset Flicker-Faces-HQ (FFHQ) and the fake images are generated by StyleGAN [30].

Although easily fooling the human, the state-of-the-art synthesized images can still be detected in many cases by current fake detection methods. The state-of-the-art synthesized methods often introduce artifact patterns into the image during generation, opening a chance for fake detectors [14, 59]. Due to the current technical limitation, even worse, the image manipulation footprint will be inevitably left in a synthesized image, either by partial image manipulation [11, 18, 34] or full image synthesis [29–31]. In particular, the partial image manipulation methods often use convolutional and pooling layers to transform a real image into feature maps. After feature map modification, they have to use the upsampling method in the decoder to amplify the feature maps into a high-resolution fake image. Similarly, full image synthesis takes a random vector and amplifies it with the decoder. Such inevitable manipulation footprints leave traces for automated fake detection.

Thus far, most state-of-the-art fake image detection methods are proposed based on convolutional neural network (CNN), which roughly fall into three categories by their input feature types, *i.e.*, image-based methods [4, 37, 38, 54], fingerprint-based methods [57], and spectrum-based methods [14, 59].

- *Image-based methods* adopt large and complex networks to perform fake detection, by directly working on the images as inputs.
- *Fingerprint-based methods* leverage both fingerprints of GAN and images as inputs for fake detection, based on the assumption that GANs carry certain model fingerprints, leaving stable fingerprints in their generated images. They even possibly allow us to identify which kind of DeepFake method is used for generation.
- *Spectrum-based methods* find out that all GAN architectures in the generation process leave some footprints in the frequency domain. Therefore, they propose to leverage the spectrum information for fake detection.

These three types of methods [14, 54, 57] all demonstrate their usefulness, in achieving the state-of-the-art performance for GAN-synthesized fake image detection.

We can see that the manipulation footprints during the synthesizing process open the chance for fake image detection. Existing techniques can possibly leverage such information, from different perspectives to different extent. Therefore, a new methodology that reduces the footprint introduced during the synthesized process could increase the chance of bypassing the fake detectors.

Although smoothing could be a possible way for fake footprint reduction, we find that it is generally infeasible to effectively reduce the footprint. For example, in Figure 2, the first image is a toy case with checkerboard patterns, generated by the following procedures. We first produce a checkerboard image with size 8*8. The checkerboard has two colors: white and orange. Then, we resize the image to 64*64 by using interpolation, which simulates the operation of upsampling. The histogram of it calculates the distribution of the values in the gray-scale version of the toy example. The third image is the blurred toy example. We apply a 5*5 kernel of Gaussian blur to the toy example to obtain this image. Although the checkerboard patterns are weakened, they still exist in the image. In the histogram of the gray-scale blurred toy example, we can also find that the values of peaks are regular and symmetric as that in the histogram of the toy example. The features can be easily detected.

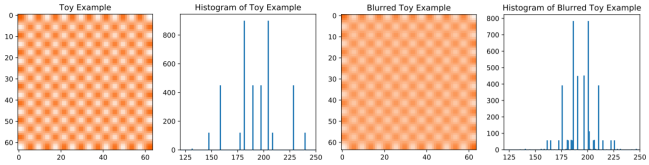

Figure 2: From left to right, the original toy example and its histogram are shown, followed by the blurred toy example and its histogram. The original toy example contains obvious checkerboard patterns. The checkerboard patterns also exist in the blurred toy example. We can find that the blur method can not effectively remove the checkerboard patterns.

In this paper, we propose the *FakePolisher*, a shallow reconstruction method with dictionary learning [5, 55] to reduce such fake footprints. In particular, we try to find the 'closest' representation, free of fake patterns, of its fake image counterparts. We first train a dictionary model to systematically capture the patterns of real images, based on which we seek the representation of DeepFake images in a low dimensional subspace through linear projection or sparse coding. Then, we perform shallow reconstruction of the 'fake-free' version of the DeepFake image, intending to largely reduce the manipulation footprints the DeepFake introduces. Our in-depth evaluation on 3 state-of-the-art DeepFake detection methods and fake images generated by 16 GAN-based methods demonstrates that our reconstructed images successfully fool all the three types of fake image detection methods. Although previous work does not explicitly mention they leverage manipulation footprint for fake image detection, our method successfully reduces the accuracy of the 3 state-of-the-art detection techniques significantly with an average accuracy decrease of 47%. This indicates that existing fake detection methods highly relies on the manipulation footprint introduced in the synthesizing phase. Our study presents a new challenge for future fake image detection methods in the domain of multimedia forensics, which need to look for more advanced fake patterns beyond only the footprints introduced by the generation phase.

The main contributions of this paper are summarized as follows.

- To reduce the footprint in GAN-synthesized fake images, we propose a post-processing shallow reconstruction method by using dictionary learning, which does not rely on any information of the GAN used for generation. In other words, it can be used as a black-box attack method to fool the fake image detectors.
- We conduct a comprehensive evaluation of our proposed approach in fooling three representative state-of-the-art of fake image detection methods over fake images generated by 16 GAN-based methods. By reducing the manipulation footprints, our method reduces the fake detection accuracy of these methods significantly. Our reconstructed images also exhibit high similarity to its original fake image counterpart.
- So far, it is still unknown whether existing fake detection methods leverage the manipulation footprints and in what ways. Our results answer this question, indicating that existing methods can highly leverage the manipulation of footprint information from different perspectives. Our results call for attention that more general fake detection mechanisms should be designed.

2 RELATED WORK

Since its advent, GAN [16] has been successfully applied to many application domains, especially in the generation process for images, natural languages, and audios, *etc.*

2.1 GAN-based Image Generation

Over the past several years, a lot of GAN-based image generation methods have been proposed, largely following two categories: full image synthesis and partial image manipulation.

Full image synthesis. Progressive growing GAN (ProGAN) [29] is able to synthesize high-resolution images via the incremental enhancement of the discriminator and the generator networks during the training process. StyleGAN [30] is an extension to the ProGAN architecture, hovers with the ability to control over the disentangled style properties of the generated images. StyleGAN2 [31] fixed the imperfection of StyleGAN to improve image quality. SNGAN [36] proposes a novel weight normalization technique called spectral normalization to stabilize the training of the discriminator. It is capable of generating images of better or equal quality relative to the previous training stabilization techniques. MMDGAN [32] combines the key ideas in both generative moment matching network (GMMN) and GAN.

Partial image manipulation. AttGAN [18] applies an attribute classification constraint to the generated image to guarantee the correct change of desired attributes. StarGAN [11] simply uses a single model to perform image-to-image translations for multiple facial properties. STGAN [34] simultaneously improves attribute manipulation accuracy as well as perception quality on the basis of AttGAN.

2.2 DeepFake Detection Methods

Tolosana *et al.* and Verdoliva *et al.* [48, 50] recently make comprehensive surveys on the DeepFake detection methods [4, 14, 37, 38, 42, 52–54, 57, 59]. Overall, they surveyed thirty-seven papers in total, all of which are CNN-based and can be classified into three categories depending on their feature inputs: image-based methods, fingerprint-based methods, and spectrum-based methods. Image-based methods directly use images as inputs with various networks to solve the problem. Fingerprint-based methods detect DeepFake with the features of GAN fingerprints. Spectrum-based methods consider that DeepFake artifacts are manifested as replications of spectra in the frequency domain. Thus they propose classifiers based on the spectrum of fake images.

3 METHOD

In this section, we give a more detailed discussion on the limitation of GAN-based methods and introduce our post-processing method.

3.1 Artifact of GAN-Based Image Generation

For both partial image manipulation [11, 18, 34] and full image synthesis [29–31], the generator of GAN-based image generation method has a decoder amplifies the random vectors or feature maps to images. Upsampling is a significant and indispensable design in the decoder. However, it is the upsampling design that makes the GAN-based image generation methods limited. In general, there are three types of upsampling methods: unpooling, transpose convolution and interpolation. It has been studied that transpose convolution results in checkerboard texture [39]. In the amplification procedure, unpooling operation assigns zero values to the new pixels. This regular magnification produces special textures that do not exist in real images. For interpolation operation, at an intuitive level, the new pixel values are calculated based on existing pixels, which is regularity. Interpolation brings periodicity into the second derivative signal of images [15]. Other papers [14, 19, 59] also introduced the imperfection of upsampling methods.

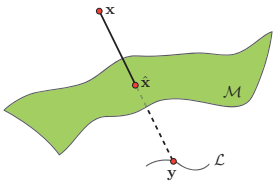

Figure 3: Geometric interpretation of how the shallow reconstruction works in order to bring the DeepFake image $\mathbf{x}$ onto the clean image manifold $\mathcal{M}$ thus finds its 'closest' counterpart $\hat{\mathbf{x}}$ on the manifold through the embedded representation $\mathbf{y}$ in the embedding space $\mathcal{L}$.

3.2 Shallow Reconstruction to the Rescue

In this paper, we propose *FakePolisher*, a post-processing method that performs a shallow modification of fake images. Our method is composed of three steps. First, we train a dictionary model with a real image dataset. The learned dictionary forms a subspace that is intrinsically low dimensional, which compactly captures the essential structures and representations of the real images. Second, we seek the representation of a DeepFake image using the aforementioned subspace either by linear projection or sparse coding depending on the over-completeness of the learned dictionary. Third, once such a representation is obtained, we reconstruct the 'fake-free' version of the DeepFake image by using the said dictionary.[1]

Intuitively, we are forcing the DeepFake image to find its 'closest' representation, on the subspace that subsequently leads to the reconstructed version of itself free of any fake patterns. By doing so, a shallow reconstruction can already effectively remove the fake patterns while preserving image fidelity to the greatest extent. In this context, deep reconstruction methods, on the contrary, can become futile because they potentially leave traces of upsampling artifacts as discussed above.

Geometrically, as shown in Figure 3, the learned dictionary forms an embedding space $\mathcal{L}$ that could be of lower or higher dimensionality. When a DeepFake image $\mathbf{x}$ comes along, we seek the 'closest' counterpart $\hat{\mathbf{x}}$ on the clean image manifold $\mathcal{M}$ by first obtaining the embedded representation $\mathbf{y}$, and reconstruct back to the clean image manifold $\mathcal{M}$.

3.3 Global vs. Local Dictionary Learning

When learning the dictionary on the real images that are free of fake patterns, one can choose to learn a local patch-based dictionary, leading to a patch-based image reconstruction; or, a global dictionary that spans the entire image, *i.e.*, dictionary atom is of the same size as the image to be reconstructed. Since we can view the global case as a local case with a large patch size, we will mainly discuss patch-based local reconstruction in detail since it already

[1] Throughout the paper, we use 'clean' to describe an image that is free of fake patterns. 'Clean' and 'fake pattern-free' are used interchangeably.

covers both cases. The choice between various patch sizes (local vs. global) is largely dictated by the actual application and the type of images that are being processed. For example, if the images are aligned faces, it is advisable to use a global dictionary since it is more efficient, without the need of patch-by-patch reconstruction. On the other hand, if the images are of ImageNet type, it is more reasonable to use a patch-based dictionary. One note is that in this case, it is still possible to use a global dictionary, it is just we are to foresee a drop in the reconstruction fidelity.

Next, we formulate the patch-based dictionary learning procedure. The training data (patch) matrix $\mathbf{Y} \in \mathbb{R}^{d \times n}$ is assumed with dimension d. All matrices have their elements arranged column-wise.

Dictionary learning methods have gained much popularity in tackling low-level computer vision problems. One widely adopted such an algorithm is the K-SVD [5]. K-SVD aims to be a natural extension of K-means clustering method with the analogy that the cluster centroids are the elements of the learned dictionary and the cluster memberships are defined by the sparse approximations (ℓ_1 or ℓ_0) of the signals in that dictionary. Formally, it provides a solution to the problem:

$$\text{(KSVD)} \quad \underset{\mathbf{D},\mathbf{X}}{\text{minimize}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \quad \text{subject to} \quad \forall i, \|\mathbf{x}_i\|_0 < K \quad (1)$$

where $\mathbf{Y}$, $\mathbf{D}$ and $\mathbf{X}$ are the data, the learned dictionary, and the sparse approximation matrix, respectively. Here $\|.\|_0$ is the pseudo-norm measuring sparsity. The sparse approximations of the data elements are allowed to have some maximum sparsity $\|\mathbf{x}\|_0 \leq K$.

In addition to the K-SVD method, we also explore a ubiquitously popular dictionary learning method: principal component analysis (PCA) [55]. In the original formulation of PCA, it tries to minimize the following objective function in an ℓ_2 sense. Of course, variants of PCA such as sparse PCA or ℓ_1-PCA, *etc.*, can also fit in with ease. Formally, PCA finds a solution to the problem:

$$\text{(PCA)} \quad \underset{\mathbf{D},\mathbf{X}}{\text{minimize}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \quad \text{subject to} \quad \mathbf{D}^\top\mathbf{D} = \mathbf{I} \quad (2)$$

where $\mathbf{Y}$, $\mathbf{D}$ and $\mathbf{X}$ are the data, the learned dictionary (principal components), and the dense coefficient matrix, respectively. The regularizer in the PCA optimization ensures that the learned dictionary atoms are orthogonal, which provides maximal reconstruction capability. The learned PCA dictionary is usually overdetermined (or undercomplete), which provides a good complement to the overcomplete K-SVD dictionary.

3.4 Shallow Reconstruction from Dense vs. Sparse Representation

Once the said dictionaries are learned from real images that are fake-pattern-free, we can project a DeepFake image patch onto the learned subspace and obtain a new representation on the learned manifold. The dimensionality of such a representation can be lower or higher than the original image-domain representation depending on the overcompleteness of the learned dictionary.

More specifically, for example, when we want to reconstruct a single image patch $\mathbf{y} \in \mathbb{R}^d$ using learned K-SVD dictionary, since it is overcomplete, we resort to pursuit algorithms such as the orthogonal matching pursuit (OMP) [49] to obtain the sparse coefficient vector $\mathbf{x}$ according to the following optimization:

$$\text{(OMP)} \quad \underset{\mathbf{x}}{\text{minimize}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 \quad \text{subject to} \quad \forall i, \|\mathbf{x}\|_0 < \tau \quad (3)$$

Note that there is a trade-off in choosing the sparsity τ while using OMP for obtaining the sparse representation. To determine the optimal reconstruction sparsity τ for the down-stream task, we conduct a pilot experiment that aims at selecting a τ value that is both relatively small (more efficient for the greedy OMP algorithm) and provides high-quality reconstruction. Also, the sparsity τ during the OMP step is independent and different from the sparsity K during K-SVD dictionary learning.

The shallow reconstruction is straight-forward with the learned K-SVD dictionary $\mathbf{D}$ and the obtained sparse representation $\mathbf{x}$. The reconstructed image patch $\hat{\mathbf{y}} = \mathbf{D}\mathbf{x}$. The aforementioned dictionary learning and reconstruction were previously used in various domain-domain mapping problems such as [3, 22–28].

As a comparison, shallow reconstruction from learned PCA dictionary requires first obtaining a dense representation for the image patch $\mathbf{y}$. As discussed above, the learned PCA dictionary $\mathbf{D}$ is usually overdetermined, and therefore, the resulting representation vector $\mathbf{x}$ on the manifold will be of lower dimensionality and dense. The representation vector $\mathbf{x}$ can be obtained through a least-square error solution in closed form which is extremely efficient:

$$\mathbf{x} = (\mathbf{D}^\top\mathbf{D})^{-1}\mathbf{D}^\top\mathbf{y} \quad (4)$$

To further make the shallow reconstruction using PCA more versatile, one can control what dimensions contribute more during the reconstruction by involving a selector vector $\mathbf{s} \in \mathbb{R}^d$ that embeds *e.g.*, prior knowledge such as confidence or importance of each dimension. In this case, the representation vector $\mathbf{x}$ can be obtained by incorporating a diagonal selector matrix $\mathbf{S}$, where $\mathbf{S} = \text{diag}(\mathbf{s})$. The solution becomes:

$$\hat{\mathbf{x}} = [(\mathbf{S}\mathbf{D})^\top(\mathbf{S}\mathbf{D})]^{-1}(\mathbf{S}\mathbf{D})^\top\mathbf{S}\mathbf{y} = [\mathbf{D}^\top\mathbf{S}^\top\mathbf{S}\mathbf{D}]^{-1}\mathbf{D}^\top\mathbf{S}^\top\mathbf{S}\mathbf{y} \quad (5)$$

$$= [\mathbf{D}^\top(\mathbf{S}^\top\mathbf{S})\mathbf{D}]^{-1}\mathbf{D}^\top(\mathbf{S}^\top\mathbf{S})\mathbf{y} \quad (6)$$

It can be observed that when $\mathbf{S}^\top\mathbf{S}$ is close to the identity matrix $\mathbf{I}$, the $\hat{\mathbf{x}}$ is close to the original $\mathbf{x}$. The same principal can be applied to the aforementioned K-SVD sparse reconstruction. Also, the selector vector $\mathbf{s}$ can be both real-numbered (dimension re-weighting) or binary (dimension selection).

The shallow reconstruction is also straight-forward with the learned PCA dictionary $\mathbf{D}$ and the obtained dense representation $\mathbf{x}$, with the reconstructed image patch $\hat{\mathbf{y}} = \mathbf{D}\mathbf{x}$.

Both K-SVD and PCA reconstructions are shallow in the sense that they can bring back the manifold representations to the image domain with a single-step projection. More importantly, with the reconstruction being shallow and single-step, it does not induce unnecessary fake patterns, as commonly found in those DeepFake images produced or manipulated by deep generative models.

3.4.1 Discussion: Comparison with Denoising Autoencoder and DefenseGAN. Denoising Autoencoder (DAE) [51] is an early attempt for image deep reconstruction, especially for the image denoising task. Compared to the shallow reconstruction we have discussed in this work, DAE is usually comprised of several layers of fully connected or convolutional layers in both the encoder and the decoder, interlaced with non-linearity.

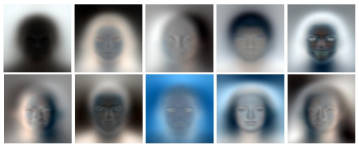

Figure 4: The PCA dictionary generated by us has 10,000 components. The size of each component is 224*224*3. Here we show the images of the first ten principal components.

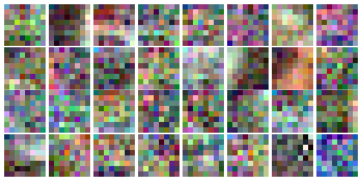

Figure 5: The K-SVD dictionary generated by us has 5,000 components. The size of each component is 8*8*3. The components of K-SVD dictionary are not sequenced. Thus we randomly show images of 32 components.

Apart from being much deeper and non-linear, the training process of the DAE takes the noisy version of the data as input, and the reconstructed version is then compared with the clean version, whose discrepancy amounts to the loss that needs to be minimized by tuning the weights in the encoder and the decoder. The model usually works well when the input image is corrupted with the same noise that the model has seen during the training process.

As a comparison, our shallow reconstruction method has the following main advantages: (1) the model is shallow and linear, which is much easier to train with little to none tuning required, and it is much more data efficient; (2) the training only requires a clean version of the images (as compared to the clean and noisy pairs as in the DAE), and such reconstruction can deal with arbitrary non-clean versions of the data, be it some types of noises, or DeepFake patterns that need to be removed.

Another line of recent work in the context of removing adversarial noise through deep image reconstruction is the DefenseGAN method [44]. The idea is to train a GAN generator based only on clean images and any adversarially noisy image can be noise-removed by DefenseGAN deep reconstruction. In some sense, it seems that it can be re-purposed for reconstructing DeepFake images that are free of fake patterns. However, a major issue remains because the deep reconstruction in the DefenseGAN also results in fake patterns, which is something our proposed shallow reconstruction is trying hard to prevent.

4 EXPERIMENTS

To demonstrate the effectiveness of our shallow reconstruction, we propose two different validation methods. One is to test the reconstructed images on various fake image detection methods, which indicates whether there exists some relation between these detection methods and the artifacts (*i.e.*, manipulation footprint). The other is using metrics to measure the similarity between fake images and reconstructed images, quantitatively measure our reconstructed change magnitude. These two validation methods are able to confirm the usefulness of our method. We also give some concrete examples to show the fake images and our reconstructed ones.

4.1 Experimental Setup

Subject Detection Methods and Dataset: We choose three state-of-the-art fake detection methods to verify the validity of our method, *i.e.*, **GANFingerprint** (fingerprint-based method) [57], **CNNDetector** (image-based method) [54], and **DCTA** (spectrum-based method) [14]. We use CelebA [35], LSUN [56], and FFHQ [30] as the real image dataset. CelebA and FFHQ are the human face dataset while LSUN includes the images of different rooms such as classroom, bedroom, *etc.*, which are widely used in previous work. Then, we leverage a total of 16 GAN-based methods for fake image generation on these datasets. In particular, ProGAN [29], SNGAN [36], CramerGAN [6] and MMDGAN [32] are the GAN-based image generation methods used by **GANFingerprint** and **DCTA**. For each GAN-based image generation method, the size of the testing dataset is 10,000. In **CNNDetector**, they choose 13 GAN-based image generation methods as the testing dataset. The methods include ProGAN [29], StyleGAN [30] , BigGAN [7], CycleGAN [61], StarGAN [11], GauGAN [41], CRN [45], IMLE [33], SITD [9], SAN [12], DeepFakes [43], StyleGAN2 [31], and Whichfaceisreal [20]. The size of the testing dataset of these GAN-based image generation methods range from hundreds to thousands. The objects in the datasets of CycleGAN, ProGAN, StyleGAN and StyleGAN2 have two or more categories. For example, in StyleGAN, it has three different categories: bedroom, car, cat. The datasets of other GANs have only one category.

Evaluation Settings: For PCA reconstruction, we use 50,000 real human images of CelebA to train the PCA dictionary model. The component number of PCA dictionary model is 10,000. For K-SVD reconstruction, we use 100,000 patches to train a K-SVD model of 5,000 components. Each patch is of size 8*8, clipped from real images of CelebA. The number of nonzero coefficients in the training procedure is 15. In K-SVD reconstruction, we drop 10% pixels of the fake image before reconstruction. This is an important procedure in K-SVD reconstruction for that it can destroy the fake textures of the fake images. What's more, it needs a lot of time to produce one K-SVD image. Therefore, we choose 200 of the 5,000 components of the K-SVD dictionary to reconstruct fake images. The reconstructed images of using 200 or 5,000 components are similar while the reconstruction time is significantly reduced. In the reconstructing procedure, the number of nonzero coefficients is 20. The graphical representation of the PCA dictionary and K-SVD dictionary are shown in Figure 4 and Figure 5.

Metrics: The main metric is the detection accuracy of the methods. We compare the detection accuracy of fake images and reconstructed images for each method. In addition, we also use cosine similarity (COSS), peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) for measuring the similarity between fake image and its corresponding reconstructed image. COSS is a common similarity metric that measures the cosine of the angle. We transform the RGB images to vectors before calculating COSS. PSNR is the most commonly used measurement for the reconstruction quality of lossy compression. SSIM is one of the most popular and useful metrics for measuring the similarity between two images. COSS, PSNR and SSIM metrics are better if a higher value is provided. The value ranges of COSS and SSIM are both in [0,1].

All the experiments were run on a Ubuntu 16.04 system with an Intel(R) Xeon(R) CPU E5-2699 with 196 GB of RAM, equipped with four Tesla V100 GPU of 32G RAM.

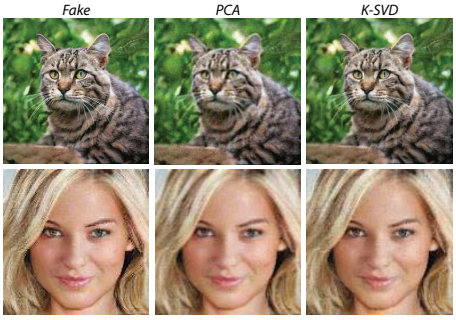

Figure 6: In the first row, the images in turn are the fake image produced by StyleGAN [30], PCA reconstructed image and K-SVD reconstructed image. In the second row, the images in turn are the fake image produced by SNGAN [36], PCA reconstructed image and K-SVD reconstructed image.

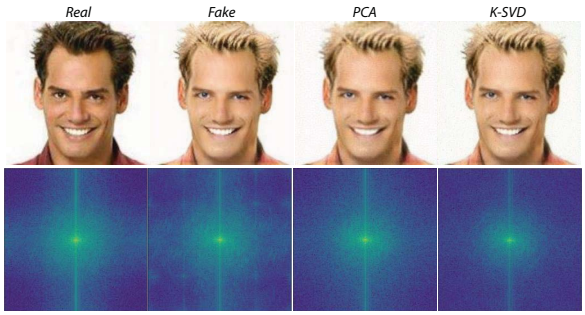

Figure 7: In the first row, the images in turn are a real image from CelebA, a fake image produced by StarGAN [11], PCA reconstructed image and K-SVD reconstructed image. In the second row, the images are the spectrum corresponding to the images above. As [59] mentioned, the GAN-synthesised images have obvious artifacts in the frequency spectrum. In the spectrum of the fake image, there are bright blobs at 1/4 and 3/4 of the width/height. In the spectrum of reconstructed images, the artifact does not exist. This means that our method effectively reduces fake texture from the fake image.

4.2 Examples of Reconstructed Image

Figure 6 gives the reconstructed image examples of our method, on a cat and a human, respectively. In the second row, PCA and K-SVD reconstruct the fake image successfully. In the first row, we can see that the fidelity of PCA reconstructed image is not as good as K-SVD reconstructed image. The reason is that the PCA dictionary we used is trained by real images of humans instead of cats. Thus, this suggests using K-SVD to reconstruct fake images if PCA dictionary with the same category is not available.

To analyze whether the reconstructed images contain artifacts (*i.e.*, manipulation footprints), we analyze the spectrums of the real image, fake image, PCA reconstructed image, K-SVD reconstructed image (see Figure 7). We can observe that only the spectrum of the fake image has bright blobs at 1/4 and 3/4 of the width/height. The blobs correspond to the manipulation footprints in the fake image. We also verified that the PCA and K-SVD reconstruction methods can reduce artifacts on other types of images such as cat, bedroom, *etc.*

4.3 GANFingerprint

In the original experiment [57] of **GANFingerprint**, their method can successfully detect whether an input image is real or fake with high accuracy. It can even judge which GAN-based image generation method is used to produce the fake image. In our experiment, we randomly choose 10,000 real images from CelebA. Then, for each GAN-based image generation method (*i.e.*, ProGAN, SNGAN, CramerGAN, MMDGAN), we produce 10,000 fake images, resulting in a total of 40,000. For each of PCA and K-SVD reconstruction method, we produce 40,000 images from these 40,000 fake images.

Table 1 summarizes the detailed detection accuracy of **GANFingerprint**. We use ProGAN as an example to explain the data in Table 1 (*i.e.*, columns 2-6, rows 2-4). For **ProGAN (Pro)**, it has five sub-items (columns): CelebA, ProGAN (Pro), SNGAN (SN), CramerGAN (Cramer), MMDGAN (MMD). They represent the possibility that the input ProGAN fake images be considered as one of them. **Fake**, **PCA-reconstructed** and **K-SVD-reconstructed** represent the type of input images. The row of **Fake** shows the results with fake images as inputs. As we can see, **GANFingerprint** can accurately classify the 10,000 images generated by ProGAN into subitem ProGAN with a detection accuracy of 99.91%. In the table, we highlight the difference in detection accuracy between reconstructed images and fake images (*e.g.*, by color and number). For example, in the row **PCA-reconstructed**, when we put the 10,000 PCA-reconstructed images into **GANFingerprint**, it misclassifies most of the images into CelebA (*i.e.*, real images). The ratio of images classified into CelebA raises from 0.03% to 88.90%. We use blue color and (+88.87) to highlight the difference. Similarly, the ratio of images classified into Pro decreases from 99.91% to 6.99%. We use red color and (-92.92) to show the difference. We can see that most of the fake images generated by ProGAN are misclassifies to be real images after our shallow reconstruction.

Similar conclusions could also be reached for the other three GAN-based image generation methods. PCA reconstruction reduces all of their classification accuracy effectively. K-SVD reconstruction also reduces the accuracy. The attack by PCA reconstruction shows slightly higher results compared with K-SVD reconstruction.

Table 2 shows the similarity between the fake images and reconstructed images. For both PCA and K-SVD, we use COSS, PSNR and SSIM as the metrics for similarity measurement. In the column of **ProGAN**, we can see that the values of COSS and SSIM are near 1.0, and the value of PSNR is more than 30, indicating high similarity. Likewise, for the other three GAN-based image generation methods, the reconstructed images are also very similar to the fake image counterparts. Compared with PCA reconstruction, images by K-SVD show higher similarity to original fake images.

4.4 DCTA

As mentioned above, **DCTA** has a same testing dataset as **GANFingerprint**. In the original experiment of **DCTA**, it transformed the images into spectrum images before classification. We follow the exact same evaluation setting, except that we use the reconstructed images to replace the fake images. Overall, the number of testing images in our experiment is 48,000. Each category of CelebA, ProGAN, SNGAN, CramerGAN, MMDGAN, has 9,600 images. The PCA-reconstructed and K-SVD-reconstructed images used are the same as that in **GANFingerprint**. The number of reconstructed images is also 48,000. Table 3 summarizes detection accuracy decrease. When using PCA-reconstruction, the accuracy decreases

Table 1: Detection accuracy before & after reconstruction of GAN-synthesized images in GANFingerprint

Accuracy(%)	ProGAN (Pro)					SNGAN (SN)				
	CelebA	Pro	SN	Cramer	MMD	CelebA	Pro	SN	Cramer	MMD
Fake	0.03	99.91	0.01	0.03	0.02	0.07	0.01	99.75	0.05	0.12
PCA-reconstructed	88.90 (+88.87)	6.99 (-92.92)	0.21 (+0.20)	0.07 (+0.04)	3.83 (+3.81)	46.10 (+46.03)	0.12 (+0.11)	50.86 (-48.89)	0.16 (+0.11)	2.76 (+2.64)
K-SVD-reconstructed	21.50 (+21.47)	78.10 (-21.81)	0.10 (+0.09)	0.20 (+0.17)	0.10 (+0.08)	48.15 (+48.08)	4.60 (+4.59)	46.35 (-53.40)	0.60 (+0.55)	0.30 (+0.18)
Accuracy(%)	CramerGAN (Cramer)					MMDGAN (MMD)				
	CelebA	Pro	SN	Cramer	MMD	CelebA	Pro	SN	Cramer	MMD
Fake	0.00	0.02	0.02	99.76	0.20	0.11	0.01	0.04	0.27	99.57
PCA-reconstructed	54.85 (+54.85)	0.35 (+0.33)	0.93 (+0.91)	35.07 (-64.69)	8.80 (+8.60)	45.94 (+45.83)	0.13 (+0.12)	0.20 (+0.16)	0.03 (-0.24)	53.70 (-45.87)
K-SVD-reconstructed	28.70 (+28.70)	14.90 (+14.88)	0.10 (+0.08)	55.60 (-44.16)	0.70 (+0.50)	47.40 (+47.29)	14.20 (+14.19)	0.30 (+0.26)	0.70 (+0.43)	37.40 (-62.17)

Table 2: Similarity between fake image & reconstructed image of GANs in GANFingerprint & DCTA

		ProGAN	SNGAN	CramerGAN	MMDGAN
PCA	COSS	0.999	0.999	0.998	0.999
	PSNR	32.33	32.67	31.85	32.28
	SSIM	0.960	0.960	0.957	0.959
K-SVD	COSS	0.999	0.999	0.999	0.999
	PSNR	33.224	33.526	32.897	33.304
	SSIM	0.972	0.972	0.971	0.972

Table 3: Detection accuracy before & after reconstruction of GAN-synthesized images in DCTA

	Accuracy(%)
Fake	88.99
PCA-reconstructed	16.42 (-72.57)
K-SVD-reconstructed	20.44 (-68.55)

from 88.99% to 16.42%. Similarly, 20.44% accuracy decreases when using K-SVD. The experimental results demonstrate the effectiveness of reconstructed images in misleading the fake detectors.

4.5 CNNDetector

In **CNNDetector**, it is evaluated on a large number of 13 GAN-based image generation methods. In their original experiments, the objects in the images are very different, containing animals, human faces, road, *etc.* For each GAN-based image generation method, the size of the testing dataset ranges from hundreds to thousands. They use detection accuracy and average precision (AP) as the metrics, on the same number of fake images and real images as the testing dataset. We follow the same evaluation setting, except replacing fake images with reconstructed images by our methods.

As we can see in Table 4, the two models used by **CNNDetector**: blur_jpg_prob0.1 (prob0.1) and blur_jpg_prob0.5 (prob0.5) achieve high accuracy. For convenience, we only introduce the data of using blur_jpg_prob0.1. In the first row, **Real & Fake** means using real images and fake images as the testing dataset (*i.e.*, the same testing dataset as used in **CNNDetector**).

To show the detection accuracy of real images and fake images, we conduct extra experiments on real images and fake images respectively. As shown in the second and third row, the testing datasets are Real images only and Fake images only. In the second row, we can observe that the model performs well. However, in the third row, the performance of the model of **CNNDetector** is different in various GAN-based image generation methods. It has various performance drops on SAN and DeepFakes although it achieves high accuracy on GANs, *e.g.*, ProGAN, StarGAN, CRN.

For PCA reconstruction of our method, we produce a corresponding reconstructed image for each fake image in the testing dataset. For K-SVD reconstruction, it needs a lot of time to produce one K-SVD image. Thus for each category of each GAN-based image generation method, we choose 100 fake images and produce 100 corresponding K-SVD reconstructed images. No matter the detection accuracy of fake images, the reconstructed images generated by us can reduce its performance.

As we can see in the fourth and fifth row, the testing datasets are PCA-reconstructed images and K-SVD-reconstructed images respectively. Compared to the detection accuracy of that on fake images, most of them are both decreased. Only the accuracy of SAN increases slightly. In the experiment of blur_jpg_prob0.5, all the detection accuracy decrease. The similarity of PCA/K-SVD reconstructed images and fake images is shown in Table 5. The images of CycleGAN, StyleGAN, StyleGAN2 and ProGAN have quite a few different categories. For these four multi-category GANs, they use different folders to store different categories of images. In the other nine GANs, some of them involve only one category (DeepFakes, IMLE, StarGAN, Whichfaceisreal, CRN). The others combine images of different categories into one folder. We call these GANs (BigGAN, GauGAN, SAN, SITD) uncertain-category and use '-' to represent the category.

4.6 Comparison Between Partial and Full Reconstruction

Sometimes, fake images are produced by only modifying parts of the real images. For example, DeepFake methods may change the hair color of a person or the color of a chair in the bedroom. For these situations, we propose partial reconstruction.

In the GANs of **GANFingerprint**, **DCTA** and **CNNDetection**, only **StarGAN** have partially modified fake images. Therefore, we use StarGAN to show the comparison between partial reconstruction and full reconstruction. The numbers of real images and fake images of StarGAN are 1,999. We produce 1,999 reconstructed images for partial and full reconstruction of PCA. We also produce 100 reconstructed images for K-SVD. Since it needs a lot of time to produce K-SVD images, the number of K-SVD reconstructed images is not as large as that of PCA reconstructed images.

For each of **GANFingerprint** and **DCTA**, we train a binary classification model with real images from CelebA and partially modified fake images from StarGAN.

Table 4: Detection accuracy before & after reconstruction of GAN-synthesized images in CNNDetecion

	Accuracy(%)/AP	ProGAN	StyleGAN	BigGAN	CycleGAN	StarGAN	GauGAN	CRN	IMLE	SITD	SAN	DeepFakes	StyleGAN2	Whichfaceisreal
prob0.1	Real & Fake	99.9/99.9	87.1/99.6	70.2/84.5	85.2/93.5	91.7/98.2	78.9/89.5	86.3/98.2	86.2/98.4	90.3/97.2	50.5/70.5	53.5/89.0	84.4/99.1	83.6/93.2
	Real	100/-	99.9/-	93.5/-	91.5/-	96.7/-	93.0/-	72.7/-	72.7/-	93.9/-	99.1/-	99.9/-	99.9/-	92.9/-
	Fake	99.9/-	74.2/-	46.8/-	78.8/-	86.7/-	64.8/-	99.8/-	99.8/-	86.7/-	1.83/-	6.86/-	68.8/-	74.3/-
	PCA	42.3 (-57.6)/-	3.90 (-70.3)/-	12.3 (-34.5)/-	35.8 (-43.0)/-	36.0 (-50.7)/-	14.2 (-50.6)/-	6.50 (-93.3)/-	19.4 (-80.4)/-	3.89 (-82.8)/-	3.20 (+1.37)/-	1.33 (-5.53)/-	11.9 (-56.9)/-	1.40 (-72.9)/-
	K-SVD	94.9 (-5.0)/-	33.7 (-40.5)/-	30.0 (-16.8)/-	68.7 (-10.1)/-	48.0 (-38.7)/-	51.0 (-13.8)/-	79.0 (-20.8)/-	88.0 (-11.8)/-	45.0 (-41.7)/-	8.0 (+6.17)/-	0.00 (-6.86)/-	33.5 (-35.3)/-	50.0 (-24.3)/-
prob0.5	Real & Fake	100/100	73.4/98.5	59.0/88.2	80.8/96.8	81.0/95.4	79.3/98.1	87.6/98.9	94.1/99.5	78.3/92.7	50.0/63.9	51.1/66.3	68.4/98.0	63.9/88.8
	Real	100/-	99.9/-	99.1/-	98.6/-	99.3/-	99.4/-	99.2/-	99.2/-	92.8/-	100/-	99.4/-	99.9/-	99.2/-
	Fake	100/-	46.9/-	18.9/-	62.9/-	62.7/-	59.2/-	76.0/-	88.9/-	63.9/-	0.00/-	2.5/-	36.9/-	28.6/-
	PCA	71.6 (-28.4)/-	3.00 (-43.9)/-	6.45 (-12.5)/-	30.9 (-32.0)/-	42.1 (-20.6)/-	22.8 (-36.4)/-	4.36 (-71.6)/-	16.7 (-72.2)/-	1.12 (-62.8)/-	0.00 (0)/-	1.89 (-0.61)/-	6.84 (-30.1)/-	0.70 (-27.9)/-
	K-SVD	96.7 (-3.30)/-	20.7 (-26.2)/-	9.00 (-9.90)/-	44.2 (-18.7)/-	37.0 (-25.7)/-	44.0 (-15.2)/-	22.0 (-54.0)/-	60.0 (-28.9)/-	36.0 (-27.9)/-	0.00 (0)/-	2.00 (-0.50)/-	13.0 (-23.9)/-	18.0 (-10.6)/-

Table 5: Similarity between fake image & reconstructed image of GANs in CNNDetection

		BigGan	DeepFakes	GauGAN	IMLE	SAN	SITD	StarGAN	Whichfaceisreal	CycleGAN						StyleGAN			StyleGAN2			
		-	person	-	road	-	-	person	person	horse	zebra	winter	orange	apple	summer	bedroom	car	cat	horse	church	car	cat
PCA	COSS	0.996	0.999	0.996	0.999	0.989	0.987	0.999	0.997	0.997	0.995	0.996	0.998	0.997	0.995	0.998	0.994	0.998	0.997	0.997	0.995	0.999
	PSNR	29.14	43.94	29.62	32.72	25.29	29.29	37.08	29.21	29.28	27.39	28.50	31.51	30.19	28.10	30.53	25.98	32.47	29.91	28.07	27.22	34.70
	SSIM	0.897	0.993	0.902	0.945	0.821	0.886	0.975	0.899	0.896	0.870	0.885	0.917	0.910	0.877	0.916	0.844	0.933	0.899	0.881	0.864	0.954
K-SVD	COSS	0.998	0.999	0.999	0.999	0.999	0.991	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.998	0.999	0.999
	PSNR	32.17	39.46	39.14	39.58	36.51	31.38	38.63	37.50	33.55	32.39	32.58	33.70	33.48	31.99	33.56	33.50	34.93	34.24	31.18	34.34	37.35
	SSIM	0.961	0.988	0.965	0.986	0.986	0.962	0.987	0.980	0.969	0.967	0.966	0.961	0.966	0.961	0.968	0.971	0.973	0.969	0.956	0.973	0.980

		ProGAN																				CRN
		airplane	motorbike	tvmonitor	horse	sofa	car	pottedplant	diningtable	sheep	bottle	person	train	dog	cow	bicycle	cat	bird	boat	chair	bus	road
PCA	COSS	0.998	0.995	0.997	0.996	0.998	0.995	0.994	0.996	0.996	0.997	0.996	0.996	0.997	0.997	0.994	0.997	0.997	0.996	0.997	0.995	0.998
	PSNR	29.74	26.14	28.17	27.94	29.61	27.49	26.15	27.12	28.13	29.21	28.95	27.49	29.59	28.24	26.02	30.39	29.16	27.94	28.74	26.68	31.13
	SSIM	0.913	0.867	0.898	0.883	0.991	0.884	0.858	0.881	0.878	0.908	0.901	0.875	0.905	0.882	0.860	0.917	0.899	0.880	0.904	0.867	0.932
K-SVD	COSS	0.999	0.998	0.998	0.999	0.999	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.999	0.998	0.998	0.999	0.998	0.998	0.999	0.998	0.999
	PSNR	33.73	30.76	32.37	32.49	32.94	32.01	30.43	31.73	32.01	32.49	33.13	31.65	33.34	32.00	30.70	33.90	32.82	31.88	32.78	31.37	37.49
	SSIM	0.974	0.965	0.972	0.968	0.970	0.970	0.959	0.968	0.963	0.968	0.973	0.963	0.970	0.965	0.963	0.973	0.968	0.964	0.971	0.965	0.985

Table 6: Comparison of detection accuracy between partial reconstruction and full reconstruction

Accuracy(%)	GANFingerprint	DCTA	CNNDetection(0.1)	CNNDetection(0.5)
Fake	99.6	76.1	86.7	62.7
fully PCA	92.6 (-7.0)	65.2 (-10.9)	36.0 (-50.7)	42.1 (-20.6)
partially PCA	92.5 (-7.1)	64.7 (-11.4)	26.0 (-60.7)	36.6 (-26.1)
fully K-SVD	87.0 (-12.6)	40.0 (-36.1)	48.0 (-38.7)	37.0 (-25.7)
partially K-SVD	87.0 (-12.6)	38.9 (-37.2)	52.0 (-34.7)	35.0 (-27.7)

Table 7: Similarity comparison between partial reconstruction and full reconstruction

	COSS	PSNR	SSIM
fully PCA-reconstructed	0.99932	35.597	0.97072
partially PCA-reconstructed	0.99936	35.221	0.97154
fully K-SVD-reconstructed	0.99978	38.626	0.98713
partially K-SVD-reconstructed	0.99978	38.623	0.98715

Table 6 summarizes the detection accuracy of fully reconstructed images and partially reconstructed images on three different types of fake detectors. Compared with full reconstruction, the detection accuracy of all the three fake detectors decrease similarly in partial reconstruction for both PCA and K-SVD. Table 7, shows the similarity metrics between reconstructed images and the original fake images. We can observe that the similarity of fake images and partial reconstruction is higher than that of fake images and full reconstruction.

To sum up, compared with fully reconstructed images, the partially reconstructed images are more similar to their original fake counterparts. Meanwhile, in terms of degrading the performance of fake detectors, their abilities are close, indicating the advantage of partial reconstruction for partially modified fake images.

5 CONCLUSIONS

In the paper, we propose the *FakePolisher*, a post-processing shallow reconstruction method based on dictionary learning without knowing any information of the GAN. The reconstructed images can easily fool the existing state-of-the-art detection methods. We also demonstrate that the existing detection methods are limited, which highly rely on the imperfection of upsampling methods. More powerful defense mechanisms for DeepFakes should be proposed. In future work, we plan to propose new methods that can remove the artifacts in fake images. Moreover, other shallow methods such as the ones based on advanced correlation filters [8, 10, 13, 17, 21, 40, 58, 60] are also potentially viable solutions to this problem, which we intend to explore further.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable feedback. This research was supported in part by Singapore National Cybersecurity R&D Program No. NRF2018NCR-NCR005-0001, National Satellite of Excellence in Trustworthy Software System No. NRF2018NCR-NSOE003-0001, NRF Investigatorship No. NRFI06-2020-0022. It was also supported by JSPS KAKENHI Grant No. 20H04168, 19K24348, 19H04086, and JST-Mirai Program Grant No. JPMJMI18BB, Japan. Weikai Miao is supported by the NSFCs of China (No. 61872144 and No. 61872146). Geguang Pu is supported by NSFC Project No. 61632005 and NSFC Project. No. 61532019. We gratefully acknowledge the support of NVIDIA AI Tech Center (NVAITC) to our research.

REFERENCES

[1] 2019. FaceAPP. https://faceapp.com/app.

[2] 2019. ZAO. https://apps.apple.com/cn/app/zao/id1465199127.

[3] Ramzi Abiantun, Felix Juefei-Xu, Utsav Prabhu, and Marios Savvides. 2019. SSR2: Sparse Signal Recovery for Single-Image Super-Resolution on Faces with Extreme Low Resolutions. *Pattern Recognition* (2019).

[4] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. 2018. Mesonet: a compact facial video forgery detection network. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 1–7.

[5] Michal Aharon, Michael Elad, and Alfred Bruckstein. 2006. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing* 54, 11 (2006), 4311–4322.

[6] Marc G Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. 2017. The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743* (2017).

[7] Andrew Brock, Jeff Donahue, and Karen Simonyan. 2018. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096* (2018).

[8] P. Buchana, I. Cazan, M. Diaz-Granados, F. Juefei-Xu, and M.Savvides. 2016. Simultaneous Forgery Identification and Localization in Paintings Using Advanced Correlation Filters. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 1–5.

[9] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. 2018. Learning to see in the dark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3291–3300.

[10] Z. Chen, Q. Guo, L. Wan, and W. Feng. 2018. Background-Suppressed Correlation Filters for Visual Tracking. In *ICME*. 1–6.

[11] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. 2018. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8789–8797.

[12] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. 2019. Second-order attention network for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 11065–11074.

[13] W. Feng, R. Han, Q. Guo, J. Zhu, and S. Wang. 2019. Dynamic Saliency-Aware Regularization for Correlation Filter-Based Object Tracking. *IEEE TIP* 28, 7 (2019), 3232–3245.

[14] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. 2020. Leveraging Frequency Analysis for Deep Fake Image Recognition. *arXiv preprint arXiv:2003.08685* (2020).

[15] Andrew C Gallagher. 2005. Detection of Linear and Cubic Interpolation in JPEG Compressed Images.. In *CRV*, Vol. 5. Citeseer, 65–72.

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[17] Qing Guo, Ruize Han, Wei Feng, Zhihao Chen, and Liang Wan. 2020. Selective Spatial Regularization by Reinforcement Learned Decision Making for Object Tracking. *IEEE TIP* 29 (2020), 2999–3013.

[18] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. 2019. Attgan: Facial attribute editing by only changing what you want. *IEEE Transactions on Image Processing* (2019).

[19] Yihao Huang, Felix Juefei-Xu, Run Wang, Xiaofei Xie, Lei Ma, Jianwen Li, Weikai Miao, Yang Liu, and Geguang Pu. 2020. FakeLocator: Robust Localization of GAN-Based Face Manipulations via Semantic Segmentation Networks with Bells and Whistles. *arXiv preprint arXiv:2001.09598* (2020).

[20] Carl Bergstrom Jevin West. 2019. which face is real. http://www.whichfaceisreal.com/.

[21] F. Juefei-Xu, K. Luu, and M. Savvides. 2015. Spartans: Single-sample Periocular-based Alignment-robust Recognition Technique Applied to Non-frontal Scenarios. *IEEE Transactions on Image Processing (TIP)* 24, 12 (Dec 2015), 4780–4795.

[22] F. Juefei-Xu, Dipan K. Pal, and M. Savvides. 2014. Hallucinating the Full Face from the Periocular Region via Dimensionally Weighted K-SVD. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 1–8.

[23] F. Juefei-Xu, D. K. Pal, and M. Savvides. 2015. NIR-VIS Heterogeneous Face Recognition via Cross-Spectral Joint Dictionary Learning and Reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 141–150.

[24] F. Juefei-Xu and M. Savvides. 2015. Encoding and Decoding Local Binary Patterns for Harsh Face Illumination Normalization. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 3220–3224.

[25] F. Juefei-Xu and M. Savvides. 2015. Single Face Image Super-Resolution via Solo Dictionary Learning. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2239–2243.

[26] F. Juefei-Xu and M. Savvides. 2016. Fastfood Dictionary Learning for Periocular-Based Full Face Hallucination. In *Proceedings of the IEEE Seventh International Conference on Biometrics: Theory, Applications, and Systems (BTAS)*. IEEE, 1–8.

[27] Felix Juefei-Xu and Marios Savvides. 2016. Learning to Invert Local Binary Patterns. In *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, Article 29, 14 pages.

[28] F. Juefei-Xu and M. Savvides. 2016. Multi-class Fukunaga Koontz Discriminant Analysis for Enhanced Face Recognition. *Pattern Recognition* 52 (Apr 2016), 186–205.

[29] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196* (2017).

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

[31] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2019. Analyzing and Improving the Image Quality of StyleGAN. *arXiv preprint arXiv:1912.04958* (2019).

[32] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. 2017. Mmd gan: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*. 2203–2213.

[33] Ke Li and Jitendra Malik. 2018. Implicit maximum likelihood estimation. *arXiv preprint arXiv:1809.09087* (2018).

[34] Ming Liu, Yukang Ding, Min Xia, Xiao Liu, Errui Ding, Wangmeng Zuo, and Shilei Wen. 2019. STGAN: A Unified Selective Transfer Network for Arbitrary Image Attribute Editing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3673–3682.

[35] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2018. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August* 15 (2018), 2018.

[36] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957* (2018).

[37] Huy H Nguyen, Fuming Fang, Junichi Yamagishi, and Isao Echizen. 2019. Multi-task learning for detecting and segmenting manipulated facial images and videos. *arXiv preprint arXiv:1906.06876* (2019).

[38] Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. 2019. Capsule-forensics: Using capsule networks to detect forged images and videos. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2307–2311.

[39] Augustus Odena, Vincent Dumoulin, and Chris Olah. 2016. Deconvolution and Checkerboard Artifacts. *Distill* (2016). https://doi.org/10.23915/distill.00003

[40] D. K. Pal, F. Juefei-Xu, and M. Savvides. 2016. Discriminative Invariant Kernel Features: A Bells-and-Whistles-Free Approach to Unsupervised Face Recognition and Pose Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 5590–5599.

[41] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. 2019. Gau-GAN: semantic image synthesis with spatially adaptive normalization. In *ACM SIGGRAPH 2019 Real-Time Live!* 1–1.

[42] Hua Qi, Qing Guo, Felix Juefei-Xu, Xiaofei Xie, Lei Ma, Wei Feng, Yang Liu, and Jianjun Zhao. 2020. DeepRhythm: Exposing DeepFakes with Attentional Visual Heartbeat Rhythms. *ACM International Conference on Multimedia (ACM MM)* (2020).

[43] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, et al. 2019. Faceforensics++: Learning to detect manipulated facial images. *arXiv preprint arXiv:1901.08971* (2019).

[44] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. 2018. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605* (2018).

[45] Baoguang Shi, Xiang Bai, and Cong Yao. 2016. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence* 39, 11 (2016), 2298–2304.

[46] T Simonite. 2019. Artificial intelligence is coming for our faces. https://www.cluzters.ai/Feed-view/14/16387/artificial-intelligence-is-coming-for-our-faces?c=1415.

[47] Massimiliano Todisco, Xin Wang, Ville Vestman, Md Sahidullah, Héctor Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Tomi Kinnunen, and Kong Aik Lee. 2019. Asvspoof 2019: Future horizons in spoofed and fake audio detection. *arXiv preprint arXiv:1904.05441* (2019).

[48] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. 2020. DeepFakes and Beyond: A Survey of Face Manipulation and Fake Detection. *arXiv preprint arXiv:2001.00179* (2020).

[49] Joel A Tropp and Anna C Gilbert. 2007. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory* 53, 12 (2007), 4655–4666.

[50] Luisa Verdoliva. 2020. Media Forensics and DeepFakes: an overview. *arXiv preprint arXiv:2001.06564* (2020).

[51] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*. 1096–1103.

[52] Run Wang, Felix Juefei-Xu, Yihao Huang, Qing Guo, Xiaofei Xie, Lei Ma, and Yang Liu. 2020. DeepSonar: Towards Effective and Robust Detection of AI-Synthesized Fake Voices. *ACM International Conference on Multimedia (ACM MM)* (2020).

[53] Run Wang, Felix Juefei-Xu, Lei Ma, Xiaofei Xie, Yihao Huang, Jian Wang, and Yang Liu. 2020. FakeSpotter: A Simple yet Robust Baseline for Spotting AI-Synthesized Fake Faces. *International Joint Conference on Artificial Intelligence (IJCAI)* (2020).

[54] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. 2019. CNN-generated images are surprisingly easy to spot... for now. *arXiv preprint arXiv:1912.11035* (2019).

[55] Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems* 2, 1-3 (1987), 37–52.

[56] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. 2015. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365* (2015).

[57] Ning Yu, Larry S Davis, and Mario Fritz. 2019. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *Proceedings of the IEEE International Conference on Computer Vision*. 7556–7566.

[58] Pengyu Zhang, Qing Guo, and Wei Feng. 2019. Fast and object-adaptive spatial regularization for correlation filters based tracking. *Neurocomputing* 337 (2019), 129 – 143.

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

[60] C. Zhou, Q. Guo, L. Wan, and W. Feng. 2017. Selective object and context tracking. 1947–1951.

[61] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*. 2223–2232.