**FakePolisher: Making DeepFakes More Detection-Evasive by Shallow Reconstruction**

Yihao Huang  
East China Normal University  
China

Felix Juefei-Xu  
Alibaba Group  
USA

Run Wang  
Nanyang Technological University  
Singapore

Qing Guo  
Nanyang Technological University  
Singapore

Lei Ma  
Kyushu University  
Japan

Xiaofei Xie  
Nanyang Technological University  
Singapore

Jianwen Li  
East China Normal University  
China

Weikai Miao  
East China Normal University  
China

Yang Liu  
Nanyang Technological University  
Singapore

Geguang Pu  
East China Normal University  
China

**ABSTRACT**

The recently rapid advances of generative adversarial networks (GANs) in synthesizing realistic and natural DeepFake information (e.g., images, video) cause severe concerns and threats to our society. 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.

To reduce the artifacts in the synthesized images, deep reconstruction techniques are usually futile because the process itself can leave traces of artifacts. In this paper, we devise a simple yet powerful approach termed FakePolisher that performs shallow reconstruction of fake images through 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 with this dictionary, 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 removing 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 for the attention of DeepFake researchers and practitioners for more general-purpose fake detection techniques.

**1 INTRODUCTION**

The recent advances of fake information generation draw lots of attention and concern, with a 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 [39, 43]. We are entering an era where we cannot simply trust our eyes and ears. According to [17], humans detection peak accuracy can reach only 75% [42]. The real image is from a well-known dataset Flicker-Faces-HQ (FFHQ) and fake image is generated by StyleGAN [27].
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 [12, 53]. 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 [10, 15, 31] or full image synthesis [26–28]. In particular, the partial image manipulation methods often use convolutional and pooling layers to transform a real image into some 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 open the chances for automated fake detection.

Thus far, most state-of-the-art fake image detection methods are proposed based on convolutional neural network (CNN), which can be classified into three categories by their input feature types, i.e., image-based methods [4, 34, 35, 49], fingerprint-based methods [52], and spectrum-based methods [12, 53].

- **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 to identify which kind of DeepFake method is used for generation.
- **Spectrum-based methods** find out that all GAN architectures in the generation process leaves some footprints in the frequency domain. Therefore, they propose to leverage the spectrum information for fake detection.

These three types of methods [12, 49, 52] 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 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 synthesized process could potentially increase the chance in 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 on the toy example to get 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.

In this paper, we propose the **FakePolisher**, a shallow reconstruction method with dictionary learning [5, 50] to reduce the 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 by our method 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 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 footprint information from different perspectives. Our results call for attention that more general fake detection mechanisms should be proposed.
2 RELATED WORK
Since its advent, GAN [14] has been successfully applied to many domains, especially in the generation process for images, natural languages, and audios, etc.

2.1 GAN-based Image Generation
Over the last 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 methods. Progressive growing GAN (ProGAN) [26] is able to synthesize high-resolution images via the incremental enhancement of the discriminator and the generator networks during the training process. StyleGAN [27] is an extension to the ProGAN architecture, hover with the ability to control over the disentangled style properties of the generated images. StyleGAN2 [28] fixed the imperfection of StyleGAN to improve image quality. SNGAN [33] 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 [29] combines the key ideas in both generative moment matching network (GMMN) and GAN.
Partial image manipulation methods. AttGAN [15] applies an attribute classification constraint to the generated image to guarantee the correct change of desired attributes. StarGAN [10] simply uses a single model to perform image-to-image translations for multiple facial properties. STGAN [31] simultaneously improves attribute manipulation accuracy as well as perception quality on the basis of AttGAN. In our experiment, only StarGAN is involved.

2.2 DeepFake Detection Methods
Tolosana et al. and Verdoliva et al. [44, 46] recently make comprehensive surveys on the DeepFake detection methods [4, 12, 34, 35, 48, 49, 52, 53].
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 use the features of GAN fingerprints. Spectrum-based methods consider that DeepFake artifacts are manifested as representations of spectra in the frequency domain. Thus they propose classifier models based on the spectrum of fake images.

3 METHOD
In this section, we give 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 [10, 15, 31] and full image synthesis [26–28], the generator of GAN-based image generation method has a decoder which used to amplify 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 proved by Google Brain that transpose convolution results in checkerboard texture [36]. 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 [13]. Three other papers [12, 16, 53] also have introduced the imperfection of upsampling methods.

3.2 Shallow Reconstruction to the Rescue
The proposed FakePolisher is a post-processing method that performs a shallow modification of fake images. Our method is composed of three steps. Firstly, 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. Secondly, 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. Thirdly, once such a representation is obtained, we can reconstruct its ‘closest’ counterpart \( \hat{x} \) on the manifold through the embedded representation \( y \) in the embedding space \( \mathcal{L} \).

![Figure 3: Geometric interpretation of how the shallow reconstruction works in order to bring the DeepFake image \( x \) onto the clean image manifold \( M \) thus finds its ‘closest’ counterpart \( \hat{x} \) on the manifold through the embedded representation \( y \) in the embedding space \( \mathcal{L} \).](image-url)
global case as a local case with a large patch size, we will mainly discuss patch-based local reconstruction in detail since it already 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 will mathematically formulate the patch-based dictionary learning procedure. The training data (patch) matrix \( Y \in \mathbb{R}^{d \times n} \) 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 \((\ell_1 \text{ or } \ell_0)\) of the signals in that dictionary. Formally, it provides a solution to the problem:

\[
\text{(K-SVD)} \quad \min_{D,X} \|Y - DX\|_F^2 \quad \text{subject to} \quad \forall_i, \|x_i\|_0 < K \tag{1}
\]

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

In addition to the K-SVD method described above, we also explore a ubiquitously popular dictionary learning method: principal component analysis (PCA) [50]. In the original formulation of PCA, it tries to minimize the following objective function in an \( \ell_2 \) sense. Of course, variants of PCA such as sparse PCA or \( \ell_1 \)-PCA, etc., can also fit in with ease. Formally, PCA finds a solution to the problem:

\[
\text{(PCA) } \min_{D,X} \|Y - DX\|_F^2 \quad \text{subject to } \quad D^\top D = I \tag{2}
\]

where \( Y, D \) and \( X \) 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 over-complete 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 \( y \in \mathbb{R}^d \) using learned K-SVD dictionary, since it is overcomplete, we resort to pursuit algorithms such as

\[
\text{(OMP) } \min_{x} \|y - Dx\|_2^2 \quad \text{subject to } \quad \forall_i, \|x_i\|_0 < \tau \tag{3}
\]

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

The shallow reconstruction is straightforward with the learned K-SVD dictionary \( D \) and the obtained sparse representation \( x \). The reconstructed image patch \( \hat{y} = Dx \). The aforementioned dictionary learning and reconstruction were previously used in various domain-domain mapping problems such as [3, 19–25].

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

\[
x = (D^\top D)^{-1} D^\top y \tag{4}
\]

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 \( s \in \mathbb{R}^d \) that embeds e.g., prior knowledge such as confidence or importance of each dimension. In this case, the representation vector \( x \) can be obtained by incorporating a diagonal selector matrix \( S \), where \( S = \text{diag}(s) \). The solution becomes:

\[
\hat{x} = [(SD)^\top (SD)]^{-1} (SD)^\top Sy \tag{5}
\]

\[
= [D^\top (S^\top SD)]^{-1} D^\top S^\top Sy \tag{6}
\]

\[
= [D^\top (S^\top SD)]^{-1} D^\top (S^\top S)y \tag{7}
\]

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

The shallow reconstruction is also straightforward with the learned PCA dictionary \( D \) and the obtained dense representation \( x \). The reconstructed image patch \( \hat{y} = Dx \).

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) [47] is an early attempt for image deep reconstruction, especially for the image denoising task. Compared to the shallow reconstruction we have discussed in
which is much easier to train with little to none tuning required, whose discrepancy amounts to the loss that needs to be minimized.

To demonstrate the effectiveness of our shallow reconstruction, we choose three state-of-the-art fake detection methods to verify the validity of our method, GANFingerprint (fingerprint-based method) [40], CNNDetector (image-based method) [49], and DCTA (spectrum-based method) [12]. We use CelebA [32], LSUN [51], and FFHQ [27] 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 [26], SNGAN [33], CramerGAN [6], and MMDGAN [29] 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 thirteen GAN-based image generation methods as the testing dataset. The methods include ProGAN [26], StyleGAN [27], BigGAN [7], CycleGAN [54], StarGAN [10], GauGAN [38], CRN [41], IMLE [30], SITD [9], SAN [11], DeepFakes [39], StyleGAN2 [28], Whichfaceisreal [17]. 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 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

---

**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.
similarity metric that measures the cosine of the angle between two non-zero vectors. 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.

### 4.2 Examples of Reconstructed Image

In Figure 6 and Figure 7, we give the reconstructed image examples of our method, on a human and a cat, respectively. In Figure 6, PCA and K-SVD reconstruct the fake image successfully. In Figure 7, 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 have artifacts (i.e., manipulation footprints), we show the spectrums of the real image, fake image, PCA reconstructed image, K-SVD reconstructed image in Figure 8. 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 [52] 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 shows the 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 misclassified to be real images after our shallow reconstruction.

Similar conclusion could also be reached for the other three GAN-based image generation methods. PCA-reconstructed images reduce all of their classification accuracy effectively. K-SVD reconstruction also reduces the detection accuracy. The attack by PCA reconstruction shows slightly higher results compared with K-SVD reconstruction.
Table 2: Similarity between fake image & reconstructed image of GANs in GANFingerprint & DCTA

| Accuracy(%) | ProGAN | SNGAN | CramerGAN | MMDGAN |
|-------------|--------|-------|-----------|--------|
| PCA         |        |       |           |        |
| COSS        | 0.999  | 0.999 | 0.998     | 0.999  |
| PSNR        | 32.33  | 32.67 | 31.85     | 32.28  |
| Cramer      | 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  |
| K-SVD-reconstructed | 16.42 | 72.37 | 20.44 | 68.55 |

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

| Accuracy(%) | CelebA | Pro | SN | Cramer | MMD |
|-------------|--------|----|----|--------|-----|
| Fake        | 88.99  |    |    |        |     |
| PCA-reconstructed | 16.42 | 72.37 | 20.44 | 68.55 |
| K-SVD-reconstructed | 20.44 | 68.55 |  |  |

As we can see in the Table 4, the two models used by CNNDetector: blur_jpg_prob0.1 and blur_jpg_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. We follow the same evaluation setting, except replacing fake images with reconstructed images by our methods.

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 the Table 4, the two models used by CNNDetector: blur_jpg_prob0.1 and blur_jpg_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 drop on SAN and DeepFakes although it achieves high accuracy on GANs, e.g., ProGAN, StarGAN, CRN, IMLE.

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 a little. In the experiment of blur_jpg_prob0.5, all the detection accuracy decrease. The similarity of PCA/K-SVD reconstructed images and fake images are 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.
4.6 Comparison Between Partial and Full Reconstruction

Sometimes, fake images are produced by only modifying part 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 999. We produce 1999 reconstructed images for partial and full reconstruction of K-SVD. We also produce 100 partially PCA-reconstructed and K-SVD. In Table 7, there are the values of similarity metrics for partial reconstruction and full reconstruction is higher than that of fake images and full reconstruction. We can observe that the similarity of fake images and partial reconstruction is better.

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

5 CONCLUSIONS

In the paper, in order to remove the artifacts introduced by the upsampling method used in GAN-based image generation methods, we novel approach to partial reconstruction.

In the other nine GANs, some of them involve only one category (DeepFakes, IMLE, StarGAN, Whichfaciaceseal, 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.
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 three typical detection methods that are state-of-the-art. Thus our study demonstrates that the existing detection methods are highly relevant to the imperfection of upsampling methods and more powerful defense mechanism for DeepFakes should be proposed. In future work, we plan to improve the GAN-based image generation method and 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, 18, 37] are also potentially viable solutions to this problem, and we intend to explore further.

REFERENCES
[1] 2019. FaceAPP. https://faceapp.com/app
[2] 2019. ZAO. https://apps.apple.com/cn/app/zao/id465199127.
[3] Ramzi Abiantun, Felix Juefei-Xu, Utav 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 Noizck, 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] Michael 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 Danielekha, Will Dabney, Shuki 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 based Alignment-robust Recognition Technique Applied to Non-frontal Scenarios. arXiv preprint arXiv:2001.09598 (2020).
[8] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. 2018. Learning to see in the wild. In Proceedings of the IEEE International Conference on Computer Vision (ICCV). IEEE, 1–5.
[9] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. MMD gan: Towards deeper understanding of moment matching network. In Advances in Neural Information Processing Systems. 2203–2213.
[10] Carl Bergstrom Jevin West. 2019. which face is real. http://www.whichfaceisreal.com/.
[11] Andrew C Gallagher. 2005. Detection of Linear and Cubic Interpolation in JPEG Compressed Images.. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 11065–11074.
[12] Augustus Odena, Vincent Dumoulin, and Chris Olah. 2016. Deconvolution and Fake Detection. arXiv preprint arXiv:1804.04958 (2019).
[13] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, et al. 2019. Artificial intelligence is coming for our faces. arXiv preprint arXiv:1803.05957 (2018).
[14] Augustus Odena, Vincent Dumoulin, and Chris Olah. 2016. Deconvolution and Checkerboard Artifacts. Distill (2016). https://doi.org/10.23915/distill.00003
[15] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, et al. 2019. Faceforensics++: Learning to detect manipulated facial images. arXiv preprint arXiv:1906.06876 (2019).
[16] Shilei Wen. 2019. STGAN: A Unified Selective Transfer Network for Arbitrary Faces attributes (celeba) dataset. arXiv preprint arXiv:1901.08971 (2019).
[17] Augustin Odena, Vincent Dumoulin, and Chris Olah. 2016. Deconvolution and Checkerboard Artifacts. Distill (2016). https://doi.org/10.23915/distill.00003
[18] 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:2003.08685 (2020).
[19] 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.
[46] Luisa Verdoliva. 2020. Media Forensics and DeepFakes: an overview. arXiv preprint arXiv:2001.06564 (2020).
[47] 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.
[48] 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).
[49] 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:1912.11033 (2019).
[50] Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. Chemometrics and intelligent laboratory systems 2, 1-3 (1987), 37–52.
[51] 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).
[52] 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.
[53] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. 2019. Detecting and simulating artifacts in gan fake images. arXiv preprint arXiv:1907.06515 (2019).
[54] 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.