Face De-Spoofing: Anti-Spoofing via Noise Modeling

Amin Jourabloo*, Yaojie Liu*, Xiaoming Liu

Department of Computer Science and Engineering,
Michigan State University
{jourablo,liuyaoj1,liuxm}@msu.edu

Abstract. Many prior face anti-spoofing works develop discriminative models for recognizing the subtle differences between live and spoof faces. Those approaches often regard the image as an indivisible unit, and process it holistically, without explicit modeling of the spoofing process. In this work, motivated by the noise modeling and denoising algorithms, we identify a new problem of face de-spoofing, for the purpose of anti-spoofing: inversely decomposing a spoof face into a spoof noise and a live face, and then utilizing the spoof noise for classification. A CNN architecture with proper constraints and supervisions is proposed to overcome the problem of having no ground truth for the decomposition. We evaluate the proposed method on multiple face anti-spoofing databases. The results show promising improvements due to our spoof noise modeling. Moreover, the estimated spoof noise provides a visualization which helps to understand the added spoof noise by each spoof medium.

Keywords: Face anti-spoofing, Generative model, CNN, Image decomposition

1 Introduction

With the increasing influence of smart devices in our daily lives, people are seeking for secure and convenient ways to access their personal information. Biometrics, such as face, fingerprint, and iris, are widely utilized for person authentication due to their intrinsic distinctiveness and convenience to use. Face, as one of the most popular modalities, has received increasing attention in the academia and industry in the recent years (e.g., iPhone X). However, the attention also brings a growing incentive for hackers to design biometric presentation attacks (PA), or spoofs, to be authenticated as the genuine user. Due to the almost no-cost access to the human face, the spoof face can be as simple as a printed photo paper (i.e., print attack) and a digital image/video (i.e., replay attack), or as complicated as a 3D Mask and facial cosmetic makeup. With proper handling, those spoofs can be visually very close to the genuine user's live face. As a result, these call for the need of developing robust face anti-spoofing algorithms.

As the most common spoofs, print attack and replay attack have been well studied previously, from different perspectives. The cue-based methods aim to detect liveness cues [1, 2] (e.g., eye blinking, head motion) to classify live videos. But these methods can be fooled by video replay attacks. The texture-based methods attempt to compare texture difference between live and spoof faces, using pre-defined features such as

* denotes equal contribution by the authors.

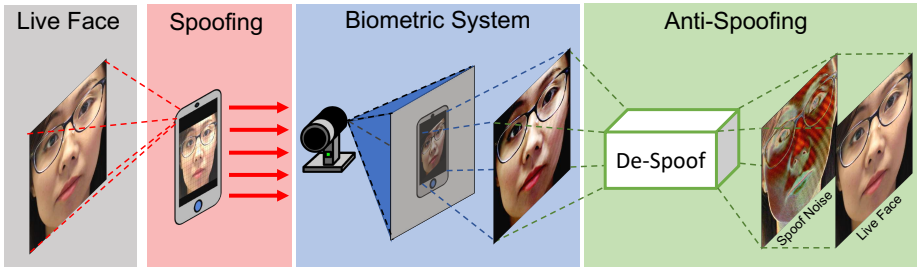

Fig. 1. The illustration of face spoofing and anti-spoofing processes. De-spoofing process aims to estimate a spoof noise from a spoof face and reconstruct the live face. The estimated spoof noise should be discriminative for face anti-spoofing.

LBP [3, 4], HOG [5, 6]. Similar to texture-based methods, CNN-based methods [7, 2, 8] design a unified process of feature extraction and classification. With a softmax loss based binary supervision, they have the risk of overfitting on the training data. Regardless of the perspectives, almost all the prior works treat face anti-spoofing as a *black box* binary classification problem. In contrast, we propose to open the black box by modeling the process of how a spoof image is generated from its original live image.

Our approach is motivated by the classic image de-X problems, such as image de-noising and de-blurring [9–12]. In image de-noising, the corrupted image is regarded as a degradation from the additive noise, e.g., salt-and-pepper noise and white Gaussian noise. In image de-blurring, the uncorrupted image is degraded by motion, which can be described as a process of convolution. Similarly, in face anti-spoofing, the spoof image can be viewed as a re-rendering of the live image but with some "special" noise from the spoof medium and the environment. Hence, the natural question is, *can we recover the underlying live image when given a spoof image, similar to image de-noising*?

Yes. This paper shows "how" to do this. We call the process of decomposing a spoof face to the spoof noise pattern and a live face as *Face De-spoofing*, shown in Fig. 1. Similar to the previous de-X works, the degraded image $\mathbf{x} \in \mathbb{R}^m$ can be formulated as a function of the original image $\hat{\mathbf{x}}$, the degradation matrix $\mathbf{A} \in \mathbb{R}^{m \times m}$ and an additive noise $\mathbf{n} \in \mathbb{R}^m$.

$$\mathbf{x} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{n} = \hat{\mathbf{x}} + (\mathbf{A} - \mathbb{I})\hat{\mathbf{x}} + \mathbf{n} = \hat{\mathbf{x}} + N(\hat{\mathbf{x}}), \tag{1}$$

where $N(\hat{\mathbf{x}}) = (\mathbf{A} - \mathbb{I})\hat{\mathbf{x}} + \mathbf{n}$ is the image-dependent noise function. Instead of solving $\mathbf{A}$ and $\mathbf{n}$, we decide to estimate $N(\hat{\mathbf{x}})$ directly since it is more solvable under the deep learning framework [13–17]. Essentially, by estimating $N(\hat{\mathbf{x}})$ and $\hat{\mathbf{x}}$, we aim to peel off the spoof noise and reconstruct the original live face. Likewise, if given a live face, face de-spoofing model should return itself plus *zero* noise. Note that our face de-spoofing is designed to handle paper attack, replay attack and possibly make-up attack, but our experiments are limited to the first two PAs. The benefits of face de-spoofing are twofold: 1) it reverses, or undoes, the spoofing generation process, which helps us to model and visualize the spoof noise pattern of different spoof mediums. 2) the spoof noise itself is discriminative between live and spoof images and hence is useful for face anti-spoofing.

While face de-spoofing shares the same challenges as other image de-X problems, it has a few distinct difficulties to conquer:

No Ground Truth: Image de-X works often use synthetic data where the original undegraded image could be used as ground truth for supervised learning. In contrast, we have no access to $\hat{\mathbf{x}}$, which is the corresponding live face of a spoof face image.

No Noise Model: There is no comprehensive study and understanding about the spoof noise. Hence it is not clear how we can constrain the solution space to *faithfully* estimate the spoof noise pattern.

Diverse Spoof Mediums: Each type of spoofs utilizes different spoof mediums for generating spoof images. Each spoof medium represents a specific type of noise pattern.

To address these challenges, we propose several constraints and supervisions based on our prior knowledge and the conclusions from a case study (in Section 3.1). Given that a live face has no spoof noise, we impose the constraint that $N(\hat{\mathbf{x}})$ of a live image is *zero*. Based on our study, we assume that the spoof noise of a spoof image is ubiquitous, i.e., it exists everywhere in the spatial domain of the image; and is repetitive, i.e., it is the spatial repetition of certain noise in the image. The repetitiveness can be encouraged by maximizing the high-frequency magnitude of the estimated noise in the Fourier domain.

With such constraints and auxiliary supervisions proposed in [18], a novel CNN architecture is presented in this paper. Given an image, one CNN is designed to synthesize the spoof noise pattern and reconstruct the corresponding live image. In order to examine the reconstructed live image, we train another CNN with auxiliary supervision and a GAN-like discriminator in an end-to-end fashion. These two networks are designed to ensure the quality of the reconstructed image regarding its discriminativeness between live and spoof, and the visual plausibility of the synthesized live image.

To summarize, the main contributions of this work include:

◇ We offer a new perspective for detecting the spoofing face from print attack and replay attack by inversely decomposing a spoof face image into the live face and the spoofing noise, without having the ground truth of either.

◇ A novel CNN architecture is proposed for face de-spoofing, where appropriate constraints and auxiliary supervisions are imposed.

◇ We demonstrate the value of face de-spoofing by its contribution to face anti-spoofing and the visualization of the spoof noise patterns.

2 Prior Work

We review the most relevant prior works to ours from two perspectives: texture-based face anti-spoofing and de-X problems.

Texture-based Face Anti-spoofing Texture analysis is widely adopted in face anti-spoofing as well as other computer vision tasks [19, 20], where defining an effective feature representation is the key endeavor. Early works apply the hand-crafted feature descriptors, such as LBP [3, 4, 21], HoG [5, 6], SIFT [22] and SURF [23], to project the faces to a low-dimension embedding. However, those hand-crafted features are not specifically designed to capture the subtle differences in the spoofing faces, and thus the embedding may not be discriminative. In addition, those features may not be robust to variations such as illumination, pose, and etc. To overcome some of these difficulties,

researchers tackle the problem in different domains, such as HSV and YCbCr color space [24, 25], temporal domain [26–29] and Fourier spectrum [30].

Heading into the deep learning era, researchers aim to build deep models for a higher accuracy. Most of the CNN works treat face anti-spoofing as a binary classification problem and apply the softmax loss function. Compared to hand-crafted features, such models [29] achieve remarkable improvements in the intra-testing (i.e., train and test within the same dataset). However, during the cross-testing (i.e., train and test in different datasets), these CNN models exhibit a poor generalization ability due to the overfitting to training data. Atoum et al. [31] and Liu et al. [**?**] observe the overfitting issue of the softmax loss, and both propose novel auxiliary-driven loss functions instead of softmax to supervise the CNN. These works bring us the insight that we need to involve the domain knowledge to solve face anti-spoofing.

To the best of our knowledge, all the previous methods are discriminative models. There are only a few papers [22, 2] trying to categorize the types and properties of the spoof noise pattern, such as color distortion and moiré pattern. In this work, we analyze the properties of spoof noise and design a GAN-fashion generative model [32] to estimate the spoof noise pattern and peel it off the spoof image. We believe by decomposing the spoof image, CNN can analyze the spoof noise more directly and effectively, and gain more knowledge in tackling face anti-spoofing.

De-X problems De-X problems, such as de-noising, de-blurring, de-mosaicing, super-resolution and inpainting [33, 13–15, 34, 35, 16, 17, 36–38], are classic low-level vision problems that remove the degradation effect or artifacts from the image. General de-noising works assume additive Gaussian noise and researchers propose non-local filters [33] or CNNs [13, 34] to exploit the inherent similarity within the images. For de-mosaicing and super-resolution, many models, such as ResNet in [14, 15] and joint models in [16, 17, 35], are learnt from the given pairs of low-quality input and high-quality ground truth. In image inpainting, users mark the area to inpaint in a mask map and apply the filling based on the existing patch texture and the overall view structure in the unmasked region [39, 36, 37].

One advantage of existing de-X problems is that most of the image degradation can be easily synthesized. This brings two benefits: 1) it provides the model training with the input degraded samples and *golden* ground-truth original images for supervision. 2) it is easy to synthesize a large amount of data for training and evaluation. On the contrary, degradation due to spoofing is versatile, complex, and subtle. It consists of 2-stage degradation: one from the spoof medium (e.g., paper and digital screen), and the other from the interaction of the spoof medium with the imaging environment. Each stage includes a large number of variations, such as medium type, illumination, non-rigid deformation and sensor types. Combination of these variations makes the overall degradation varies greatly. As a result, it is almost impossible to mimic realistic spoofing by synthesizing a degradation, which is a distinct challenge of face de-spoofing compared to the conventional de-X problems.

Without the ground truth of the degraded image, face de-spoofing becomes a very challenging problem. In this work, we propose an encoder-decoder architecture with novel loss functions and supervisions to solve the de-spoofing problem.

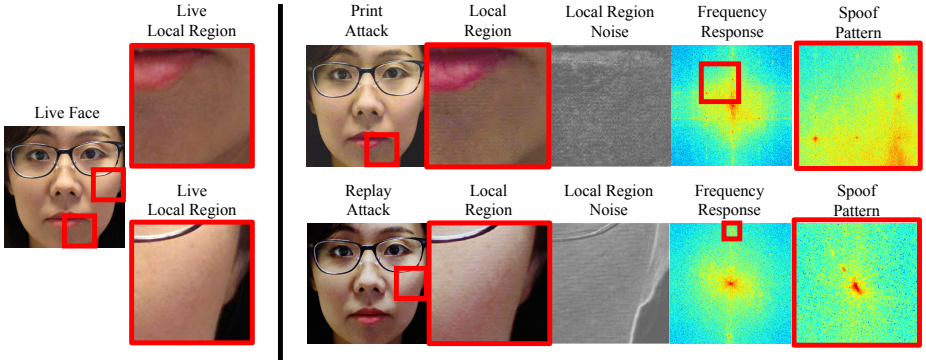

Fig. 2. The illustration of the spoof noise pattern. **Left:** live face and its local regions. **Right:** Two registered spoofing faces from print attack and replay attack. For each sample, we show the local region of the face, intensity difference to the live image, magnitude of 2D FFT, and the local peaks in the frequency domain that indicates the spoof noise pattern. Best viewed electronically.

3 Face De-spoofing

In this section, we start with a case study of spoof noise pattern, which demonstrates a few important characteristics of the noise. This study motivates us to design the novel CNN architecture that will be presented in Sec. 3.2.

3.1 A Case Study of Spoof Noise Pattern

The core task of face de-spoofing is to estimate the spoofing-relevant noise pattern in the given face image. Despite the strength of using a CNN model, we are still facing the challenge of learning *without* the ground truth of the noise pattern. To address this challenge, we would like to first carry out a case study on the noise pattern with the objectives of answering the following questions: 1) is Eqn. 1 a good modeling of the spoof noise? 2) what characteristics does the spoof noise hold?

Let us denote a genuine face as $\hat{\mathbf{I}}$. By using printed paper or video replay on digital devices, the attacker can manufacture a spoof image $\mathbf{I}$ from $\hat{\mathbf{I}}$. Considering no non-rigid deformation between $\mathbf{I}$ and $\hat{\mathbf{I}}$, we summarize the degradation from $\hat{\mathbf{I}}$ to $\mathbf{I}$ as the following steps:

1. **Color distortion:** Color distortion is due to a narrower color gamut of the spoof medium (e.g. LCD screen or Toner Cartridge). It is a projection from the original color space to a tinier color subspace. This noise is dependent on the color intensity of the subject, and hence it may apply as a degradation matrix to the genuine face $\mathbf{I}$ during the degradation.
2. **Display artifacts:** Spoof mediums often use several nearby dots/sensors to approximate one pixel's color, and they may also display the face differently than the original size. Approximation and down-sampling procedure would cause a certain degree of high-frequency information loss, blurring, and pixel perturbation. This noise may also apply as a degradation matrix due to its subject dependence.

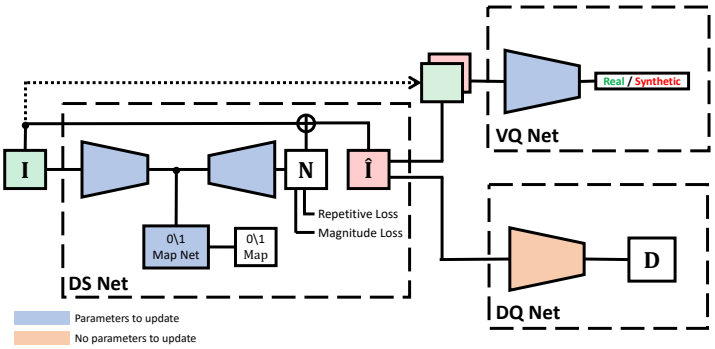

Fig. 3. The proposed network architecture.

3. **Presenting artifacts:** When presenting the spoof medium to the camera, the medium interacts with the environment and brings several artifacts, including reflection and transparency of the surface. This noise may apply as an additive noise.
4. **Imaging artifacts:** Imaging lattice patterns such as screen pixels on the camera's sensor array (e.g. CMOS and CCD) would cause interference of light. This effect leads to aliasing and creates moiré pattern, which appears in replay attack and some print attack with strong lattice artifacts. This noise may apply as an additive noise.

These four steps show that the spoof image $\mathbf{I}$ can be generated via applying degradation matrices and additive noises to $\hat{\mathbf{I}}$, which is basically conveyed by Eqn. 1. As expressed by Eqn. 1, the spoof image is the summation of the live image and image-dependent noise. To further validate this model, we show an example in Fig. 2. Given a high-quality live image, we carefully produce two spoof images via print and replay attack, with minimal non-rigid deformation. After each spoof image is registered with the live image, the live image becomes the *ground truth* live image if we would perform despoofing on the spoof image. This allows us to compute the difference between the live and spoof images, which is the noise pattern $N(\hat{\mathbf{I}})$. To analyze its frequency properties, we perform FFT on the spoof noise and show the 2D shifted magnitude response.

In both spoof cases, we observe a high response in the low-frequency domain, which is related to color distortion and display artifacts. In print attack, *repetitive* noise in Step 3 leads to a few "peak" responses in the high-frequency domain. Similarly, in the replay attack, visible moiré pattern reflects as several spurs in the low-frequency domain, and the lattice pattern that causes the moiré pattern is represented as peaks in the high-frequency domain. Moreover, spoof patterns are uniformly distributed in the image domain due to the uniform texture of the spoof mediums. And the high response of the repetitive pattern in the frequency domain exactly demonstrates that it appears widely in the image and thus can be viewed as ubiquitous.

Under this ideal registration, the comparison between live and spoof images provides us a basic understanding of the spoof noise pattern. It is a type of texture that has the characteristics of **repetitive** and **ubiquitous**. Based on this modeling and noise characteristics, we design a network to estimate the noise *without* the access to the precisely registered ground truth live image, as this case study has.

Table 1. The network structure of DS Net, DQ Net and VQ Net. Each convolutional layer is followed by an exponential linear unit (ELU) and batch normalization layer. The input image size for DS Net is $256 \times 256 \times 6$. All the convolutional filters are 3×3. $0\backslash 1$ Map Net is the bottom-left part, i.e., conv1-10, conv1-11, and conv1-12.

DS Net (Encoder Part)			DS Net (Decoder Part)			DQ Net			VQ Net		
Layer	Chan./Stri.	Outp. Size	Layer	Chan./Stri.	Outp. Size	Layer	Chan./Stri.	Outp. Size	Layer	Chan./Stri.	Outp. Size
Input			**Input**			**Input**			**Input**		
image			pool1-1+pool1-2+pool1-3			{image,live}			{image,live}		
conv1-0	24/1	256	resize	-/-	256	conv3-0	64/1	256			
conv1-1	20/1	256	conv2-1	28/1	256	conv3-1	128/1	256	conv4-1	24/2	256
conv1-2	25/1	256	conv2-2	24/1	256	conv3-2	196/1	256	conv4-2	20/2	256
conv1-3	20/1	256				conv3-3	128/1	256	pool4-1	-/2	128
pool1-1	-/2	128				pool3-1	-/2	128			
conv1-4	20/1	128	conv2-3	20/1	256	conv3-4	128/1	128	conv4-3	20/1	128
conv1-5	25/1	128	conv2-4	20/1	256	conv3-5	196/1	128	conv4-4	16/1	128
conv1-6	20/1	128				conv3-6	128/1	128	pool4-2	-/2	64
pool1-2	-/2	64				pool3-2	-/2	64			
conv1-7	20/1	64	conv2-5	20/1	256	conv3-7	128/1	64	conv4-5	12/1	64
conv1-8	25/1	64	conv2-6	16/1	256	conv3-8	196/1	64	conv4-6	6/1	64
conv1-9	20/1	64				conv3-9	128/1	64	pool4-3	-/2	32
pool1-3	-/2	32				pool3-3	-/2	32			
short-cut connection						**short-cut connection**			**vectorize**		
pool1-1+pool1-2+pool1-3						pool3-1+pool3-2+pool3-3					1024
conv1-10	28/1	32	conv2-7	16/1	256	conv3-10	128/1	32	fc4-1	1/1	100
conv1-11	16/1	32	conv2-8	6/1	256	conv3-11	64/1	32	dropout	-	0.2%
conv1-12	1/1	32	live	(image - conv2-8)		conv3-12	1/1	32	fc4-2	1/1	2

3.2 De-Spoof Network

Network Overview: Figure 3 shows the overall network architecture of our proposed method. It consists of three parts: De-Spoof Net (DS Net), Discriminative Quality Net (DQ Net), and Visual Quality Net (VQ Net). DS Net is designed to estimate the spoof noise pattern $\mathbf{N}$ (i.e. the output of $N(\hat{\mathbf{I}})$) from the input image $\mathbf{I}$. The live face $\hat{\mathbf{I}}$ then can be reconstructed by subtracting the estimated noise $\mathbf{N}$ from the input image $\mathbf{I}$. This reconstructed image $\hat{\mathbf{I}}$ should be both visually appealing and indeed live, which will be safeguarded by the DQ Net and VQ Net respectively. All networks can be trained in an end-to-end fashion. The details of the network structure are shown in Tab. 1.

As the core part, DS Net is designed as an encoder-decoder structure with the input $\mathbf{I} \in \mathbb{R}^{256\times256\times6}$. Here the 6 channels are RGB + HSV color space, following the suggestion in [31]. In the encoder part, we first stack 10 convolutional layers with 3 pooling layers. Inspired by the residual network [40], we follow by a short-cut connection: concatenating the responses from $pool$1-1, $pool$1-2 with $pool$1-3, and then sending them to $conv$1-10. This operation helps us to pass the feature responses from different scales to the later stages and ease the training procedure. Going through 3 more convolution layers, the responses $\mathbf{F} \in \mathbb{R}^{32\times32\times32}$ from $conv$1-12 are the feature representation of the spoof noise patterns. The higher magnitudes the responses have, the more spoofing-perceptible the input is.

Out from the encoder, the feature representation $\mathbf{F}$ is fed into the decoder to reconstruct the spoof noise pattern. $\mathbf{F}$ is directly resized to the input spatial size 256×256. It introduces no extra grid artifacts, which exist in the alternative approach of using a deconvolutional layer. Then, we pass the resized $\mathbf{F}$ to several convolutional layers to reconstruct the noise pattern $\mathbf{N}$. According to Eqn. 1, the reconstructed live image can be retrieved by: $\hat{\mathbf{x}} = \mathbf{x} - N(\hat{\mathbf{x}}) = \mathbf{I} - \mathbf{N}$.

Each convolutional layer in the DS Net is equipped with exponential linear unit (ELU) and batch normalization layers. To supervise the training of DS Net, we design

multiple loss functions: losses from DQ Net and VQ Net for the image quality, $0\backslash 1$ map loss, and noise property losses. We introduce these loss functions in Sec. 3.3-3.4.

3.3 DQ Net and VQ Net

While we do not have the ground truth to supervise the estimated spoof noise pattern, it is possible to supervise the reconstructed live image, which implicitly guides the noise estimation. To estimate a good-quality spoof noise, the reconstructed live image should be quantitatively and visually recognized as live. For this purpose, we propose two networks in our architecture: Discriminative Quality Net (DQ Net) and Visual Quality Net (VQ Net). The VQ Net aims to guarantee the reconstructed live face is photorealistic. The DQ Net is proposed to guarantee the reconstructed face would indeed be considered as live, based on the judgment of a pre-trained face anti-spoofing network. The details of our proposed architecture are shown in Tab. 1.

Discriminative Quality Net: We follow the state-of-the-art network architecture of face anti-spoofing [18] to build our DQ Net. It is a fully convolutional network with three filter blocks and three additional convolutional layers. Each block consists of three convolutional layers and one pooling layer. The feature maps after each pooling layer are resized and stacked to feed into the following convolutional layers. Finally, DQ Net is supervised to estimate the pseudo-depth $\mathbf{D}$ of an input face, where $\mathbf{D}$ for the live face is the depth of the face shape and $\mathbf{D}$ for the spoof face is a zero map as a flat surface. We adopt the 3D face alignment algorithm in [41] to estimate the face shape and render the depth via Z-Buffering.

Similar to the previous work [42], DQ Net is pre-trained to obtain the semantic knowledge of live faces and spoofing faces. And during the training of DS Net, the parameters of DQ Net are fixed. Since the reconstructed images $\hat{\mathbf{I}}$ are live images, the corresponding pseudo-depth $\mathbf{D}$ should be the depth of the face shape. The backpropagation of the error from DQ Net guides the DS Net to estimate the spoof noise pattern which should be subtracted from the input image,

$$J_{DQ} = \left\| \text{CNN}_{DQ}(\hat{\mathbf{I}}) - \mathbf{D} \right\|_1, \tag{2}$$

where CNN_{DQ} is a fixed network and $\mathbf{D}$ is the depth of the face shape.

Visual Quality Net: We deploy a GAN to verify the visual quality of the estimated live image $\hat{\mathbf{I}}$. Given both the real live image $\mathbf{I}_{\mathbf{live}}$ and the synthesized live image $\hat{\mathbf{I}}$, VQ Net is trained to distinguish between $\mathbf{I}_{\mathbf{live}}$ and $\hat{\mathbf{I}}$. Meanwhile, DS Net tries to reconstruct photorealistic live images where the VQ Net would classify them as non-synthetic (or real) images. The VQ Net consists of 6 convolutional layers and a fully connected layer with a 2D vector as the output, which represents the probability of the input image to be real or synthetic. In each iteration during the training, the VQ Net is evaluated with two batches, in the first one, the DS Net is fixed and we update the VQ Net,

$$J_{VQ_{train}} = -\mathbb{E}_{\mathbf{I}\in\mathcal{R}} \log(\text{CNN}_{VQ}(\mathbf{I})) - \mathbb{E}_{\mathbf{I}\in\mathcal{S}} \log(1 - \text{CNN}_{VQ}(\text{CNN}_{DS}(\mathbf{I}))), \tag{3}$$

where $\mathcal{R}$ and $\mathcal{S}$ are the sets of real and synthetic images respectively. In the second batch, the VQ Net is fixed and the DS Net is updated,

$$J_{VQ_{test}} = -\mathbb{E}_{\mathbf{I}\in\mathcal{S}} \log(\text{CNN}_{VQ}(\text{CNN}_{DS}(\mathbf{I}))). \tag{4}$$

3.4 Loss functions

The main challenge for spoof modeling is the lack of the ground truth for the spoof noise pattern. Since we have concluded some properties about the spoof noise in Sec. 3.1, we can leverage them to design several novel loss functions to constrain the convergence space. First, we introduce magnitude loss to enforce the spoof noise of the live image to be zero. Second, zero\one map loss is used to demonstrate the ubiquitousness of the spoof noise. Third, we encourage the repetitiveness property of spoof noise via repetitive loss. We describe three loss functions as the following:

Magnitude Loss: The spoof noise pattern for the live images is zero. The magnitude loss can be utilized to impose the constraint for the estimated noise. Given the estimated noise $\mathbf{N}$ and reconstructed live image $\hat{\mathbf{I}} = \mathbf{I} - \mathbf{N}$ of an original live image $\mathbf{I}$, we have,

$$J_m = \|\mathbf{N}\|_1 . \tag{5}$$

Zero\One Map Loss: To learn discriminative features in the encoder layers, we define a sub-task in the DS Net to estimate a zero-map for the live faces and an one-map for the spoof. Since this is a per *pixel* supervision, it is also a constraint of ubiquitousness on the noise. Moreover, $0\backslash 1$ map enables the receptive field of each pixel to cover a local area, which helps to learn generalizable features for this problem. Formally, given the extracted features $\mathbf{F}$ from an input face image $\mathbf{I}$ in the encoder, we have,

$$J_z = \|\text{CNN}_{01map}(\mathbf{F};\Theta) - \mathbf{M}\|_1 , \tag{6}$$

where $\mathbf{M} \in \mathbf{0}^{32\times 32}$ or $\mathbf{M} \in \mathbf{1}^{32\times 32}$ is the zero\one map label.

Repetitive Loss: Based on the previous discussion, we assume the spoof noise pattern to be repetitive, because it is generated from the repetitive spoof medium. To encourage the repetitiveness, we convert the estimated noise $\mathbf{N}$ to the Fourier domain and compute the maximum value in the high-frequency band. The existence of high peak is indicative of the repetitive pattern. We would like to maximize this peak for spoof images, but minimize it for live images, as the following loss function:

$$J_r = \begin{cases} -\max(H(\mathcal{F}(\mathbf{N}),k)), & \mathbf{I}\in Spoof \\ \|\max(H(\mathcal{F}(\mathbf{N}),k))\|_1, & \mathbf{I}\in Live \end{cases},$$

where $\mathcal{F}$ is the Fourier transform operator, H is an operator for masking the low-frequency domain of an image, i.e., setting a $k \times k$ region in the center of the shifted 2D Fourier response to zero.

Finally, the total loss function in our training is the weighted summation of the aforementioned loss functions and the supervisions for the image qualities,

$$J_T = J_z + \lambda_1 J_m + \lambda_2 J_r + \lambda_3 J_{DQ} + \lambda_4 J_{VQ_{test}}, \tag{7}$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are the weights. During the training, we alternate between optimizing Eqn. 7 and Eqn. 3.

Table 2. The accuracy of different outputs of the proposed architecture and their fusions.

Method	$0\backslash 1$ map	Spoof noise	Depth map	Fusion (Spoof noise, Depth map)		Fusion of all three outputs	
				Maximum	Average	Maximum	Average
APCER	2.50	1.70	1.66	1.70	1.27	1.70	1.27
BPCER	2.52	1.70	1.68	1.73	1.73	1.73	1.73
ACER	2.51	1.70	1.67	1.72	1.50	1.72	1.50

4 Experimental Results

4.1 Experimental Setup

Databases We evaluate our work on three face anti-spoofing databases, with print and replay attacks: Oulu-NPU [43], CASIA-MFSD [44] and Replay-Attack [45]. Oulu-NPU [43] is a high-resolution database, considering many real-world variations. Oulu-NPU also includes 4 testing protocols: Protocol 1 evaluates on the illumination variation, Protocol 2 examines the influence of different spoof medium, Protocol 3 inspects the effect of different camera devices and Protocol 4 contains all the challenges above, which is close to the scenario of cross testing. CASIA-MFSD [44] contains videos with resolution 640×480 and 1280×720. Replay-Attack [45] includes videos of 320×240. These two databases are often used for cross testing [2].

Parameter setting We implement our method in Tensorflow [46]. Models are trained with the batch size of 6 and the learning rate of $3\mathrm{e}{-5}$. We set the $k = 64$ in the repetitive loss and set λ_1 to λ_4 in Eqn. 7 as $3, 0.005, 0.1$ and 0.016, respectively. DQ Net is trained separately and remains fixed during the update of DS Net and VQ Net, but all sub-networks are trained with the same and respective data in each protocol.

Evaluation metrics To compare with previous methods, we use Attack Presentation Classification Error Rate (*APCER*) [47], Bona Fide Presentation Classification Error Rate (*BPCER*) [47] and, $ACER = (APCER + BPCER)/2$ [47] for the intra testing on Oulu-NPU, and Half Total Error Rate (*HTER*) [48], half of the summation of FAR and FRR, for the cross testing between CASIA-MFSD and Replay-Attack.

4.2 Ablation Study

Using Oulu-NPU Protocol 1, we perform three studies on the effect of score fusing, the importance of each loss function, and the influence of image resolution and blurriness.

Different fusion methods In the proposed architecture, three outputs can be utilized for classification: the norms of either the $0\backslash 1$ map, the spoof noise pattern or the depth map. Because of the discriminativeness enabled by our learning, we can simply use a rudimentary classifier like L-1 norm. Note that a more advance classifier is applicable and would likely lead to higher performance. Table 2 shows the performance of each output and their fusion with maximum and average. It shows that the fusion of spoof noise and depth map achieves the best performance. However, adding the $0\backslash 1$ map scores do not improve the accuracy since it contains the same information as the spoof noise. Hence, for the rest of experiments, we report performance from the average fusion of the spoof noise $\mathbf{N}$ and the depth map $\hat{\mathbf{D}}$, i.e., $score = (\|\mathbf{N}\|_1 + \left\|\hat{\mathbf{D}}\right\|_1)/2$.

Advantage of each loss function We have three main loss functions in our proposed architecture. To shows the effect of each loss function, we train a network with each loss

Table 3. ACER of the proposed method with different image resolutions and blurriness. To create blurry images, we apply Gaussian filters with different kernel sizes to the input images.

Metric \ Resolution	256×256	128×128	64×64
APCER	1.27	2.27	5.24
BPCER	1.73	3.36	5.30
ACER	1.50	3.07	5.27

Metric \ Blurriness	1×1	3×3	5×5	7×7	9×9
APCER	1.27	2.29	3.12	3.95	4.79
BPCER	1.73	2.50	3.33	4.16	5.00
ACER	1.50	2.39	3.22	4.06	4.89

Table 4. The intra testing results on 4 protocols of Oulu-NPU.

Protocol	Method	APCER (%)	BPCER (%)	ACER (%)
1	CPqD[49]	2.9	10.8	6.9
	GRADIANT[49]	1.3	12.5	6.9
	Auxiliary[18]	1.6	**1.6**	1.6
	Ours	**1.2**	1.7	**1.5**
2	MixedFASNet[49]	9.7	2.5	6.1
	Ours	4.2	4.4	4.3
	Auxiliary[18]	2.7	2.7	2.7
	GRADIANT	**3.1**	**1.9**	**2.5**
3	MixedFASNet	5.3 ± 6.7	7.8 ± 5.5	6.5 ± 4.6
	GRADIANT	$\mathbf{2.6 \pm 3.9}$	5.0 ± 5.3	3.8 ± 2.4
	Ours	4.0 ± 1.8	3.8 ± 1.2	3.6 ± 1.6
	Auxiliary[18]	2.7 ± 1.3	$\mathbf{3.1 \pm 1.7}$	$\mathbf{2.9 \pm 1.5}$
4	Massy_HNU [49]	35.8 ± 35.3	8.3 ± 4.1	22.1 ± 17.6
	GRADIANT	$\mathbf{5.0 \pm 4.5}$	15.0 ± 7.1	10.0 ± 5.0
	Auxiliary[18]	9.3 ± 5.6	10.4 ± 6.0	9.5 ± 6.0
	Ours	5.1 ± 6.3	$\mathbf{6.1 \pm 5.1}$	$\mathbf{5.6 \pm 5.7}$

excluded one by one. By disabling the magnitude loss, the $0\backslash 1$ map loss and the repetitive loss, we obtain the ACERs 5.24, 2.34 and 1.50, respectively. To further validate the repetitive loss, we perform an experiment on high-resolution images by changing the network input to the cheek region of the original 1080P resolution. The ACER of the network with the repetitive loss is 2.92 but the network without cannot converge.

Resolution and blurriness As shown in the ablation study of repetitive loss, the image quality is critical for achieving a high accuracy. The spoof noise pattern may not be detected in the low-resolution or motion-blurred images. The testing results on different image resolutions and blurriness are shown in Tab. 3. These results validate that the spoof noise pattern is less discriminative for the lower-resolution or blurry images, as the high-frequency part of the input images contains most of the spoof noise pattern.

4.3 Experimental Comparison

To show the performance of our proposed method, we present our accuracy in the intra testing of Oulu-NPU and the cross testing on CASIA and Replay-Attack.

Intra Testing We compare our intra testing performance on all 4 protocols of Oulu-NPU. Table 4 shows the comparison of our method and the best 3 out of 18 previous methods [18, 49]. Our proposed method achieves promising results on all protocols. Specifically, we outperform the previous state of the art by a large margin in Protocol 4, which is the most challenging protocol, and similar to cross testing.

Table 5. The HTER of different methods for the cross testing between the CASIA-MFSD and the Replay-Attack databases. We mark the top-2 performances in bold.

Method	Train CASIA MFSD	Test Replay Attack	Train Replay Attack	Test CASIA MFSD
Motion [4]	50.2%		47.9%	
LBP-TOP [4]	49.7%		60.6%	
Motion-Mag [50]	50.1%		47.0%	
Spectral cubes [51]	34.4%		50.0%	
CNN [8]	48.5%		45.5%	
LBP [24]	47.0%		39.6%	
Colour Texture [25]	30.3%		**37.7**%	
Auxiliary[18]	**27.6**%		**28.4**%	
Ours	**28.5**%		41.1%	

Cross Testing We perform cross testing between CASIA-MFSD [44] and Replay-Attack [45]. As shown in Tab. 5, our method achieves the competitive performance on the cross testing from CASIA-MFSD to Replay-Attack. However, we achieve a worse HTER compared to the best performing methods from Replay Attack to CASIA-MFSD. We hypothesize the reason is that images of CASIA-MFSD are of much higher resolution than those of Replay Attack. This shows that the model trained with higher-resolution data can generalize well on lower-resolution testing data, but not the other way around. This is one limitation of the proposed method, and worthy further research.

4.4 Qualitative Experiments

Spoof medium classification The estimated spoof noise pattern of the test images can be used for clustering them into different groups and each group represents one spoof medium. To visualize the results, we use t-SNE [52] for dimension reduction. The t-SNE projects the noise $\mathbf{N} \in \mathbb{R}^{256\times256\times6}$ to 2 dimensions by best preserving the KL divergence distance. Fig. 4 shows the distributions of the testing videos on Oulu-NPU Protocol 1. The left image shows that the noise of live is well-clustered, and the noise of spoof is subject dependent, which is consistent with our noise assumption. To obtain a better visualization, we utilize the high pass filter to extract the high-frequency information of noise pattern for dimension reduction. The right image shows that the high frequency part has more subject independent information about the spoof type and can be utilized for classification of the spoof medium.

To further show the discriminative power of the estimated spoof noise, we divide the testing set of Protocol 1 to training and testing parts and train an SVM classifier for spoof medium classification. We train two models, a three-class classifier (live, print and display) and a five-class classifier (live, print1, print2, display1 and display2), and they achieve the classification accuracy of 82.0% and 54.3% respectively, shown in Tab. 6. Most classification errors of the five-class model are within the same spoof medium. This result is noteworthy given that no label of spoof medium type is provided during the learning of the spoof noise model. Yet the estimated noise actually carries appreciable information regarding the medium type; hence we can observe reasonable results of

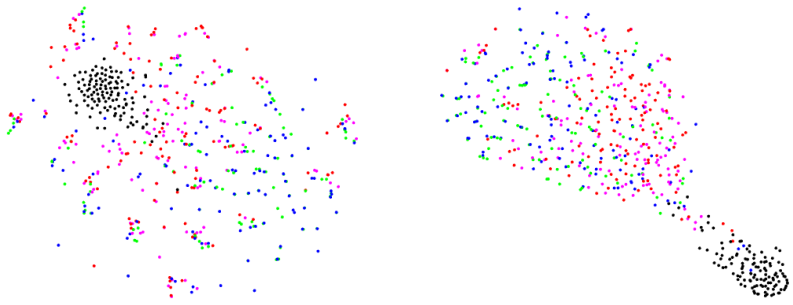

Fig. 4. The 2D visualization of the estimated spoof noise for test videos on Oulu-NPU Protocol 1. Left: the estimated noise, Right: the high-frequency band of the estimated noise, *Color code* used: *black*=live, *green*=printer1, *blue*=printer2, *magenta*=display1, *red*=display2.

Table 6. The confusion matrices of spoof mediums classification based on spoof noise pattern.

Actual \ Predicted	live	print	display
live	59	1	0
print	0	88	32
display	13	8	99

Actual \ Predicted	live	print1	print2	display1	display2
live	59	0	1	0	0
print1	0	41	2	11	6
print2	0	34	11	9	6
display1	10	6	0	13	31
display2	8	7	0	6	39

spoof medium classification. This demonstrates that the estimated noise contains spoof medium information and indeed we are moving toward estimating the faithful spoof noise residing in each spoof image. In the future, if the performance of spoof medium classification improves, this could bring new impact to applications such as forensic.

Successful and failure cases We show several success and failure cases in Fig. 5-6. Fig. 5 shows that the estimated spoof noises are similar within each medium but different from the other mediums. We suspect that the yellowish color in the first four columns is due to the stronger color distortion in the paper attack. The fifth row shows that the estimated noise for the live images is nearly zero. For the failure cases, we only have a few false positive cases. The failures are due to undesired noise estimation which will motivate us for further research.

5 Conclusions

This paper introduces a new perspective for solving the face anti-spoofing by inversely decomposing a spoof face into the live face and the spoof noise pattern. A novel CNN architecture with multiple appropriate supervisions is proposed. We design loss functions to encourage the pattern of the spoof images to be ubiquitous and repetitive, while the noise of the live images should be zero. We visualize the spoof noise pattern which can help to have a deeper understanding of the added noise by each spoof medium. We evaluate the proposed method on multiple widely-used face anti-spoofing databases.

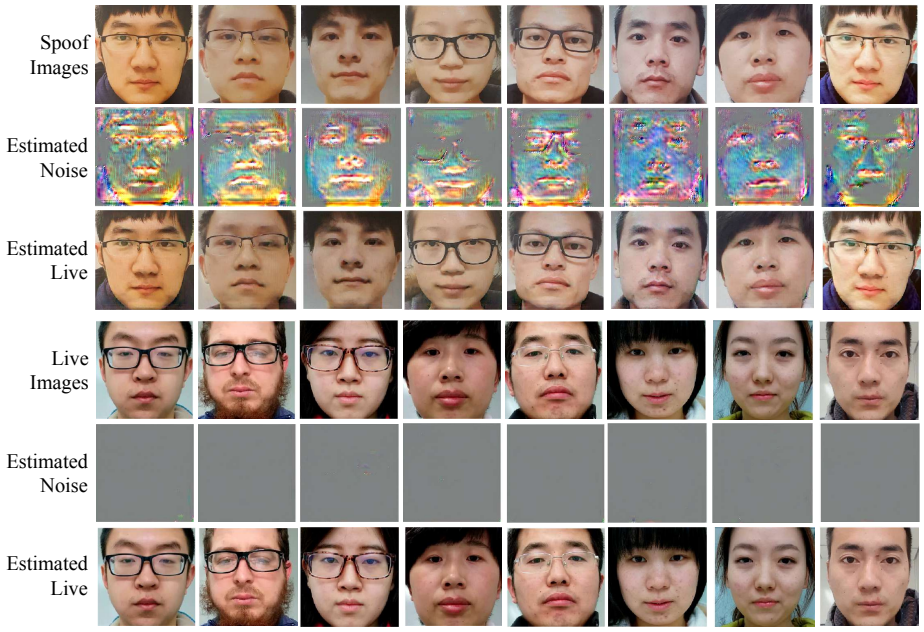

Fig. 5. The visualization of input images, estimated spoof noises and estimated live images for test videos of Protocol 1 of Oulu-NPU database. The first four columns in the first row are paper attacks and the second four are the replay attacks. For a better visualization, we magnify the noise by 5 times and add the value with 128, to show both positive and negative noise.

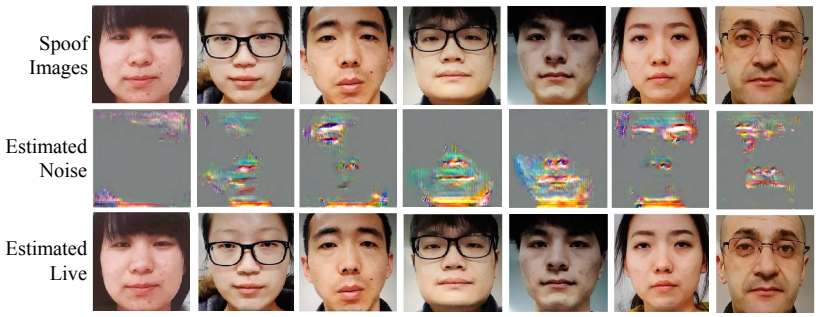

Fig. 6. The failure cases for converting the spoof images to the live ones.

Acknowledgment This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. 2017-17020200004. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

References

1. Pan, G., Sun, L., Wu, Z., Lao, S.: Eyeblink-based anti-spoofing in face recognition from a generic webcamera. In: ICCV, IEEE (2007)
2. Patel, K., Han, H., Jain, A.K.: Cross-database face antispoofing with robust feature representation. In: Chinese Conference on Biometric Recognition, Springer (2016)
3. de Freitas Pereira, T., Anjos, A., De Martino, J.M., Marcel, S.: LBP-TOP based countermeasure against face spoofing attacks. In: ACCV, Springer (2012)
4. de Freitas Pereira, T., Anjos, A., De Martino, J.M., Marcel, S.: Can face anti-spoofing countermeasures work in a real world scenario? In: ICB, IEEE (2013)
5. Komulainen, J., Hadid, A., Pietikainen, M.: Context based face anti-spoofing. In: BTAS, IEEE (2013)
6. Yang, J., Lei, Z., Liao, S., Li, S.Z.: Face liveness detection with component dependent descriptor. In: ICB, IEEE (2013)
7. Li, L., Feng, X., Boulkenafet, Z., Xia, Z., Li, M., Hadid, A.: An original face anti-spoofing approach using partial convolutional neural network. In: Image Processing Theory Tools and Applications (IPTA), 2016 6th International Conference on, IEEE (2016)
8. Yang, J., Lei, Z., Li, S.Z.: Learn convolutional neural network for face anti-spoofing. arXiv preprint arXiv:1408.5601 (2014)
9. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: ECCV, Springer (2014)
10. Jourabloo, A., Feghahati, A., Jamzad, M.: New algorithms for recovering highly corrupted images with impulse noise. Scientia Iranica **19**(6) (2012) 1738–1745
11. Kulkarni, K., Lohit, S., Turaga, P., Kerviche, R., Ashok, A.: Reconnet: Non-iterative reconstruction of images from compressively sensed measurements. In: CVPR, IEEE (2016)
12. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: CVPR, IEEE (2016)
13. Lefkimmiatis, S.: Non-local color image denoising with convolutional neural networks. In: CVPR, IEEE (2017)
14. Tai, Y., Yang, J., Liu, X., Xu, C.: Memnet: A persistent memory network for image restoration. In: ICCV, IEEE (2017)
15. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: CVPR, IEEE (2017)
16. Zhou, R., Achanta, R., Süsstrunk, S.: Deep residual network for joint demosaicing and super-resolution. arXiv preprint arXiv:1802.06573 (2018)
17. Gharbi, M., Chaurasia, G., Paris, S., Durand, F.: Deep joint demosaicking and denoising. ACM Trans. Graph. **35**(6) (2016) 191
18. Liu, Y., Jourabloo, A., Liu, X.: Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In: CVPR, IEEE (2018)
19. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. (2012)
20. Jourabloo, A., Liu, X.: Pose-invariant face alignment via CNN-based dense 3D model fitting. Int. J. Comput. Vision **124**(2) (April 2017) 187–203
21. Määttä, J., Hadid, A., Pietikäinen, M.: Face spoofing detection from single images using micro-texture analysis. In: ICJB, IEEE (2011)
22. Patel, K., Han, H., Jain, A.K.: Secure face unlock: Spoof detection on smartphones. IEEE Trans. Inf. Forens. Security **11**(10) (2016) 2268–2283
23. Boulkenafet, Z., Komulainen, J., Hadid, A.: Face antispoofing using speeded-up robust features and fisher vector encoding. IEEE Signal Processing Letters **24**(2) (2017) 141–145

24. Boulkenafet, Z., Komulainen, J., Hadid, A.: Face anti-spoofing based on color texture analysis. In: ICIP, IEEE (2015)
25. Boulkenafet, Z., Komulainen, J., Hadid, A.: Face spoofing detection using colour texture analysis. IEEE Trans. Inf. Forens. Security **11**(8) (2016) 1818–1830
26. Siddiqui, T.A., Bharadwaj, S., Dhamecha, T.I., Agarwal, A., Vatsa, M., Singh, R., Ratha, N.: Face anti-spoofing with multifeature videolet aggregation. In: ICPR, IEEE (2016)
27. Bao, W., Li, H., Li, N., Jiang, W.: A liveness detection method for face recognition based on optical flow field. In: IASP, IEEE (2009)
28. Feng, L., Po, L.M., Li, Y., Xu, X., Yuan, F., Cheung, T.C.H., Cheung, K.W.: Integration of image quality and motion cues for face anti-spoofing: a neural network approach. Journal of Visual Communication and Image Representation **38** (2016) 451–460
29. Xu, Z., Li, S., Deng, W.: Learning temporal features using LSTM-CNN architecture for face anti-spoofing. In: IAPR Asian Conference, IEEE (2015)
30. Li, J., Wang, Y., Tan, T., Jain, A.K.: Live face detection based on the analysis of fourier spectra. In: Biometric Technology for Human Identification, SPIE (2004)
31. Atoum, Y., Liu, Y., Jourabloo, A., Liu, X.: Face anti-spoofing using patch and depth-based cnns. In: ICJB, IEEE (2017)
32. Tran, L., Yin, X., Liu, X.: Disentangled representation learning GAN for pose-invariant face recognition. In: CVPR, IEEE (2017)
33. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: CVPR, IEEE (2005)
34. Zhang, H., Sindagi, V., Patel, V.M.: Image de-raining using a conditional generative adversarial network. arXiv preprint arXiv:1701.05957 (2017)
35. Zhang, H., Patel, V.M.: Densely connected pyramid dehazing network. In: CVPR, IEEE (2018)
36. Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. IEEE Trans. Image Process. **13**(9) (2004) 1200–1212
37. Bertalmio, M., Vese, L., Sapiro, G., Osher, S.: Simultaneous structure and texture image inpainting. IEEE Trans. Image Process. **12**(8) (2003) 882–889
38. Chen, Y., Tai, Y., Liu, X., Shen, C., Yang, J.: FSRNet: End-to-end learning face super-resolution with facial priors. In: CVPR, IEEE (2018)
39. Liu, Y., Shu, C.: A comparison of image inpainting techniques. In: Sixth International Conference on Graphic and Image Processing (ICGIP 2014), SPIE (2015)
40. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, IEEE (2016)
41. Liu, Y., Jourabloo, A., Ren, W., Liu, X.: Dense face alignment. In: ICCVW, IEEE (2017)
42. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV, Springer (2016)
43. Boulkenafet, Z., Komulainen, J., Li, L., Feng, X., Hadid, A.: OULU-NPU: A mobile face presentation attack database with real-world variations. In: FG, IEEE (2017)
44. Zhang, Z., Yan, J., Liu, S., Lei, Z., Yi, D., Li, S.Z.: A face antispoofing database with diverse attacks. In: ICB, IEEE (2012)
45. Chingovska, I., Anjos, A., Marcel, S.: On the effectiveness of local binary patterns in face anti-spoofing, IEEE (2012)
46. Abadi, M., Agarwal, A., et al: TensorFlow: Large-scale machine learning on heterogeneous systems (2015)
47. ISO/IEC JTC 1/SC 37 Biometrics: Information technology biometric presentation attack detection part 1: Framework. international organization for standardization. https://www.iso.org/obp/ui/iso (2016)
48. Bengio, S., Mariéthoz, J.: A statistical significance test for person authentication. In: Proceedings of Odyssey 2004: The Speaker and Language Recognition Workshop. (2004)

49. Boulkenafet, Z.: A competition on generalized software-based face presentation attack detection in mobile scenarios. In: ICJB, IEEE (2017)
50. Bharadwaj, S., Dhamecha, T.I., Vatsa, M., Singh, R.: Computationally efficient face spoofing detection with motion magnification. In: CVPRW, IEEE (2013)
51. Pinto, A., Pedrini, H., Schwartz, W.R., Rocha, A.: Face spoofing detection through visual codebooks of spectral temporal cubes. IEEE Trans. Image Process. **24**(12) (2015) 4726–4740
52. Maaten, L.v.d., Hinton, G.: Visualizing data using t-SNE. Journal of machine learning research **9**(Nov) (2008) 2579–2605