Noise Modeling, Synthesis and Classification for Generic Object Anti-Spoofing

Joel Stehouwer, Amin Jourabloo, Yaojie Liu, Xiaoming Liu
Department of Computer Science and Engineering
Michigan State University, East Lansing MI 48824
{stehouw7, liuyaoj1, jourablo, liuxm}@msu.edu

Abstract

Using printed photograph and replaying videos of biometric modalities, such as iris, fingerprint and face, are common attacks to fool the recognition systems for granting access as the genuine user. With the growing online person-to-person shopping (e.g., Ebay and Craigslist), such attacks also threaten those services, where the online photo illustration might not be captured from real items but from paper or digital screen. Thus, the study of anti-spoofing should be extended from modality-specific solutions to generic-object-based ones. In this work, we define and tackle the problem of Generic Object Anti-Spoofing (GOAS) for the first time. One significant cue to detect these attacks is the noise patterns introduced by the capture sensors and spoof mediums. Different sensor/medium combinations can result in diverse noise patterns. We propose a GAN-based architecture to synthesize and identify the noise patterns from seen and unseen medium/sensor combinations. We show that the procedure of synthesis and identification are mutually beneficial. We further demonstrate the learned GOAS models can directly contribute to modality-specific anti-spoofing without domain transfer. The code and GOSet dataset are available at cvlab.cse.msu.edu/project-goas.html.

1. Introduction

Anti-spoofing (i.e., spoof detection) is a long-standing topic in the biometrics field that empowers recognition systems to detect samples from spoofing mediums, e.g., printed paper or digital screen [2, 6, 8, 26]. A similar concern may appear in online commerce websites, e.g., Ebay, Craigslist, which provide services to enable direct user-to-user buying and selling. For instance, when purchasing, a customer may wonder, “is that a picture of a real item he owns?” This scenario motivates a broader problem of anti-spoofing:

Given an image of a generic object, such as a cup or a desk, can we automatically classify if this was captured from the real object, or through a medium, such as digital screen or printed paper?

We define this problem as Generic Object Anti-Spoofing (GOAS). With the wider variety of objects, there are richer appearance variations and greater challenges in GOAS, as shown in Fig. 1, compared to individual biometric modalities. Successful solutions [2, 8, 23, 26, 27, 30, 31, 33] for modality-specific anti-spoofing are likely ineffective for GOAS. We find that capture sensors and spoofing mediums bring certain texture patterns (e.g., Moiré pattern [32]) to all captured images, regardless of the content. These patterns are often low-energy and regarded as “noise”. However, they are ubiquitous and consistent, since they result from the physical properties of the sensors/mediums and environmental conditions, such as light reflection. We believe a proper modeling of such noise patterns will lead to effective solutions for GOAS and may contribute to modality-specific anti-spoofing tasks. In this work, we study the fundamental low-level vision problem of modeling, synthesizing, and classifying the noise patterns for tackling GOAS.

Modeling noise patterns is a promising, yet challenging, approach for GOAS. In [9, 10, 39], the camera model identi-
fication problem is studied for the purpose of digital forensics. The properties of different capture sensors are examined thanks to the assistance of databases, such as PRNU-PAR Dataset [22] and Dresden Image Database [20]. Related topics such as noise pattern removal [1] and noise pattern modeling for face modality [23] are also investigated. The authors of [42] show that simple synthesis methods for data augmentation are beneficial for the anti-spoofing task. These prior works provide a solid base to begin the study of GOAS. Meanwhile, we still face three major challenges:

**Complexity of spoof noise patterns:** The noise patterns in GOAS are related to both sensor and medium, as well as their interaction with the environment. First, it’s hard to model the interaction mathematically. Second, the noises are “hidden” under large appearance variations, thus even more untraceable. Additionally, each physical device has a unique fingerprint, though these fingerprints are similar within the same device models as shown in [19,21].

**Insufficient data and lack of strong labels:** Unlike many other computer vision tasks, spoof data for anti-spoofing cannot be collected from the Internet. Moreover, strong labels, e.g., pixel-wise correspondence between spoof images and ground truth live images, is extremely difficult to obtain. The constant development of new sensors and spoof mediums further complicates the data collection, and increases the difficulty of learning a CNN that is robust to these small, but significant variations [5].

**Modality dependency:** Current anti-spoofing methods are designed for a specific modality, e.g., face, iris, or fingerprint. These solutions cannot be applied to a different modality. Thus, it is desirable to have a single anti-spoofing model applicable to multiple modalities or applications.

To address these challenges, we propose a novel Generative Adversarial Network (GAN)-based approach for GOAS, consisting of three parts: GOGen, GOLab, and GoPad. GOGen is a generator network, which learns to convert a live image to a spoof one given a target known or unknown sensor/medium combination. GOGen allows for synthesis of new images with specific combinations, which helps to remedy insufficiency and imbalance issues in training data, such as the long tail problem [43]. GOLab serves as a multi-class classifier to identify the type of sensor and medium as well as live vs. spoof. GoPad is a binary classifier for GOAS. The three parts in this design, including the synthesis procedure and multi-class identification, contribute to our final goal of GOAS. To properly train such a network, three novel loss functions are proposed to model the noise pattern and supervise the training. Furthermore, we collect the first generic object dataset (GOSet) to conduct this study. GOSet involves 7 camera sensors, 7 spoof mediums, and other image variations.

To summarize, the contributions of this work include:

- We propose a novel network architecture to synthesize unseen noise patterns that are shown to benefit GOAS.
- A generic object dataset (GOSet) is collected and contains live and spoof videos of 24 objects.
- We demonstrate SOTA generalization performance when applying GOSet trained models to face anti-spoofing.

### 2. Prior Work

While there is no prior work on GOAS, we review relevant prior work from three perspectives.

**Modality-specific anti-spoofing:** Early works [6,8] perform texture analysis via hand-crafted features for anti-spoofing. [2] utilizes a patch-based CNN and score fusion to show that spoof noise can be detected in small image patches. Similarly, [15] uses minutiae to guide patch selection for fingerprint anti-spoofing. Rather than detecting spoof noise, [23] attempts to estimate and remove the spoof noise from images. Cue-based methods incorporate domain knowledge into anti-spoofing, e.g., rPPG [25,26], eyetracker [30], visual rhythms [3,16,18], paired audio cues [12], and pulse oximetry [34]. A significant limitation is that each modality is domain specific; an algorithm developed for one modality cannot be applied to the others. The closest approach to cross domain is [28], via transfer learning to fine-tune on the face modality. Our work improves upon these by utilizing generic objects, and therefore is forced to be content independent. Further, we learn a deep representation for the spoof noise of multiple spoof mediums, and show that these noises can be convolved with a live image to synthesize new spoof images.

**Noise patterns modeling:** Modeling or extracting noise from images is challenging, since there is no canonical ground truth. Hence some works attempt to estimate the noise via assumptions about the physical properties of the sensors and software post-processing of captured images [38,39]. With these assumptions, ensemble classifiers [9], hand-crafted feature based classifiers [38,39], and deep learning approaches [22] are proposed to address camera model identification. Following these, we assume that the sensor noise is image content independent. However, we not only classify the noise in an image, but also learn a noise prototype for each sensor that can be convolved with any image to modify its “noise footprint”. We also address the challenge of spoof medium noise modeling and classification. [23] estimates the spoof noise on an image, but is limited to face images and estimates the noise per image. Hence we extend both camera model identification and spoof noise estimation works by combining both tasks within a single CNN, and by modeling a generalized representation of both the sensor and medium noises.

**Image manipulation and synthesis:** GANs have gained increasing interest for style transfer and image synthesis tasks. Star-GAN [14] utilizes images from multiple do-
Figure 2. The overall framework of training GOGen. Live images are given to the generator to modify either the sensor or spoof noise. The resulting image is classified by the GOLab discriminator to supervise the generated images. An additional discriminator is used to ensure the generated images remained visually appealing and realistic. In each section of the figure, only the solid-colored network is updated in that training step. We alternate between training GOGen in one step and GOLab and GODisc in the next step. Input one-hot vectors are used as a mask to select the appropriately learned noise map, which is then concatenated to the input image.

3. Proposed Methods

In this section, we present the details of the proposed methods, including GOGen, GODisc, and GOLab. As shown in Fig. 2, the overall framework adopts a GAN architecture, which is composed of a generator (GOGen) and two discriminators (GODisc and GOLab). GOGen synthesizes additional spoof videos of any combination of sensor and medium, even unseen combinations. GODisc is the discriminator network to guide images from GOGen to be visually plausible. GOLab performs sensor and medium identification. In addition, GOLab serves as the module to produce a final spoof detection score. We also present GOPad, which is adapted from a traditional binary classifier used by previous anti-spoofing works, to compare with the proposed method. To prevent overfitting and increase the quantity of training data, the input for the networks are image patches extracted from the original images.

3.1. GOGen: Spoof Synthesis

In anti-spoofing, the increasing variety of sensors and spoof mediums creates a large challenge for data collection and generalization. It is increasingly expensive to collect additional data from every combination of camera and spoof medium. Meanwhile, the quantity, quality, and diversity of training data determine the performance and impact the generalization of deep learning algorithms. Hence, we develop GOGen to address this need for continual data collection via synthesis of unseen combinations.

We train GOGen to synthesize new images of unseen sensor/medium combinations using knowledge learned from known combinations. When introducing a new device, GOGen can be trained with minimal data from the new device while utilizing all previously collected data from other devices. The generator, CNN_{Gen}(), converts a live image into a targeted spoof image of a specified spoof medium captured by a specified sensor. Specifically, the inputs of the generator are a live image \( I \in \mathbb{R}^{H \times W} \) and two one-hot vectors specifying the sensor of the output image \( a_c \in \mathbb{R}^{n_c} \), and the medium through which the output would be captured \( a_m \in \mathbb{R}^{n_m} \). The output is a synthetic image \( I' \).
Then, we concatenate $I$, $N_c$ and $N_m$ as $T = [I, N_c, N_m]$ and feed $T$ to the generator. With this concatenated input, through convolution the generator mimics the interplay between the image content $I$, and the learnt $N_c$ and $N_m$, to generate a device-specific, image-dependent, synthetic image. By manipulating only the sensor or the medium at a time, we are able to supervise either of $M_c$ or $M_m$ independently. In this manner, any combination from the learned $N_c$ and $N_m$ are used together to produce the noise for a synthetic image, even from unseen combinations.

We hypothesize that by integrating the noise representation as part of the GOGen, via backpropagation, we should be able to learn latent noise prototypes that are specific to the device but universal across all images captured by that device. Such representations will enable GOGen to better synthesize images under many $(n_c \times n_m)$ combinations of sensors and mediums. We show the learned sensor and medium noise prototypes in Fig. 3. After the input image and noise prototypes are concatenated, they are fed to 8 convolution layers to synthesize spoof images. The detailed network architecture of GOGen is shown in Tab. 1.

Since the additional spoof noise should be low-energy, an $L_2$ loss is employed to minimize the difference between the live image and the image synthesized by the generator. This loss helps to limit the magnitude of the noise:

$$J_{\text{Vis}} = \|I - \text{CNN}_\text{Gen}(T)\|^2_2.$$  

### 3.2. GODisc: Discriminator and GAN Losses

Next, the discriminator GODisc ensures that $I$ is visually appealing. The GODisc network includes 10 convolution layers and 2 fully connected layers, shown in Tab. 1. It outputs the Softmax probability for the two classes, real spoof images vs. synthesized spoof images.

The training of the GAN follows an alternating training procedure. During the training of CNN$_{\text{Disc}}()$, we fix the parameters of CNN$_{\text{Gen}}()$ and use the following loss:

$$J_{\text{Disc}_{\text{gan}}} = -\mathbb{E}_{I \in \mathcal{R}} \log(\text{CNN}_{\text{Disc}}(I))$$

$$- \mathbb{E}_{T \in \mathcal{L}} \log(||1 - \text{CNN}_{\text{Disc}}(\text{CNN}_{\text{Gen}}(T))||),$$  

where $\mathcal{R}$ represents the real spoof images and $\mathcal{L}$ the real live images. During the training of GOGen, we fix the parameters of CNN$_{\text{Disc}}()$ and use the following loss:

$$J_{\text{Disc}_{\text{gan}}} = -\mathbb{E}_{I \in \mathcal{R}} \log(||\text{CNN}_{\text{Disc}}(\text{CNN}_{\text{Gen}}(T))||).$$

### 3.3. GOLab: Sensor and Medium Identification

GOLab is designed to classify noises from certain sensors and spoof mediums. It serves as the discriminator to guide GOGen to generate accurate spoof images as well as the final module to produce scores for GOAS. Shown in Tab. 1, the input for GOLab is an RGB image with the size of $64 \times 64$. The input images can be either the original images or the GOGen synthesized images. It uses 11 convolution layers and 3 max pooling layers to extract features, and
then two fully connected layers to generate \(n_{c^-} \) and \(n_{m^-}\)-dim vectors for sensor and medium classification. Each comes from an independent stack of fully connected layers.

We use the cross entropy loss to supervise the training of GOLab. Given the input image \(I\), the ground truth one-hot label \(a_c\) and the softmax normalized prediction \(\hat{a}_c\) for the spoof medium; and \(a_m\), \(\hat{a}_m\) for the sensor, the loss functions are defined as:

\[
S_c = -\sum_i a_c^i \log(\hat{a}_c^i), \quad S_m = -\sum_i a_m^i \log(\hat{a}_m^i),
\]

where \(i\) is the class index of the sensors and spoof mediums. Then, the final loss to supervise GOLab is:

\[
J_{\text{Lab}} = S_c(I) + S_m(I),
\]

The GOLab network provides supervision for the generator and guides it via backpropagation from the sensor and spoof medium loss functions. Specifically, we define a normalized loss for updating the generator network:

\[
J_{\text{Labnet}} = \frac{S_m(\text{CNN}_{\text{Gen}}(T))}{1 + S_m(I)} + \frac{S_c(\text{CNN}_{\text{Gen}}(T))}{1 + S_c(I)},
\]

where the numerator shows the classification losses, \(i.e., S_m(\cdot)\) and \(S_c(\cdot)\), for the synthesized images, and \(S_m(I)\) and \(S_c(I)\) are the loss of the live images during the updating of GOLab. By using the normalized loss, GOGen will not be penalized when GOLab has high classification error on the real data, \(i.e.,\) a large denominator leads to a small quotient regardless of the numerator.

### 3.4. GOPad: Binary Classification

To show the benefits of the proposed method, we follow the baseline algorithm [26], specifically the pseudo-depth map branch, to implement a binary classification of GOAS, termed as GOPad. To demonstrate strong generalization ability later, we limit the size of the GOPad algorithm by dramatically reducing the number of convolution kernels in each layer to approximately one-third compared to the baseline algorithm. The GOPad network takes an RGB image as input, and produces a 0-1 map \(\text{CNN}_{\text{Pad}}(I) \in \mathbb{R}^{H \times W}\) in the final layer, where it is 0 for live and 1 for spoof. The network activates where the spoof noise is detected. During the training process, this map allows the CNN model to make live/spoof labeling at the pixel level. When converged, the 0-1 map should be uniformly 0 or 1, representing a confident classification of live vs. spoof. Formally, the loss function is defined as:

\[
J_{\text{Pad}} = \|\text{CNN}_{\text{Pad}}(I) - G\|_2^2,
\]

where \(G\) is the ground truth 0-1 map.

#### 3.5. Implementation Details

We show all of the three proposed CNN networks in Fig. 2. We use an alternating training scheme for updating the networks during the training. We train the GOGen while the GOLab and GODisc are fixed. In the next step, we keep the GOGen fixed and train the other two networks. We alternate between these two steps until all networks converge. To train the GOGen and GOLab, we use batch sizes of 40. Patch sizes of \(64 \times 64\) are used for the GOGen, GODisc, and GOLab. Patch sizes of \(256 \times 256\) are used for the GOPad, following the setting of previous works. The final loss for training the generator of GOGen can be summarized as:

\[
J = J_{\text{Disc}_{\text{gan}}} + \lambda_0 J_{\text{Vis}} + \lambda_1 J_{\text{Labnet}},
\]

where \(\lambda_0\) and \(\lambda_1\) are weighting factors. And the final loss for training GODisc and GOLab can be denoted as:

\[
J = J_{\text{Disc}_{\text{gan}}} + \lambda_1 J_{\text{Labnet}},
\]

and \(\lambda_0\) and \(\lambda_1\) were set to 0.5 and 0.1, for all experiments.

### 4. Generic Object Dataset for Anti-Spoofing

To enable the study of GOAS, we consider a total of 24 objects, 7 backgrounds, 7 commonly used camera sensors, and 7 spoofing mediums (including live as a blank medium) while collecting the Generic Object Dataset (GOSet). If fully enumerated, this would require a prohibitory collection of 8,232 videos. Due to constraints, we selectively collect 2,849 videos to cover most combinations of backgrounds, camera sensors and spoof mediums.

The objects we collect are: squeezer, mouse, multi-pen, sunglasses, water bottle, keyboard, pencils, calculator, stapler, flash drive, cord, hard drive disk, keys, shoe (red), shoe (white), shoe (black), Airpods, remote, PS4 (color),...
PS4 (black), Kleenex, blow torch, lighter, and energy bar, shown in Fig. 4. Generic objects are more easily available for data collection and are unencumbered by privacy or security concerns, as opposed to human biometrics. The objects are placed in front of 7 backgrounds, which are desk wood, carpet speckled, carpet flowered, floor wood, bed sheet (white), blanket (blue), and desk (black). The spoof mediums include 3 common computer screens, (Acer Desktop, Dell Desktop, and Acer Laptop), and 3 mobile device screens, (iPad Pro, Samsung Tab, and Google Pixel), which are of varying size and display quality.

The videos were collected using 7 commercial devices, (Moto X, Samsung S8, iPad Pro, iPod Touch, Google Pixel, Logitech Webcam, and Canon EOS Rebel). Except for videos from the iPod Touch at 720P resolution, all videos are captured at 1, 080P resolution, with average length of 12.5 seconds. We first capture the live videos of all objects while varying the distance and viewing angle, and then collect the spoof videos via directly viewing a spoof medium while the live video is displayed on it. During the collection of spoof videos, care is taken to prevent unnecessary spoofing artifacts (light reflection, screen bezels), as well as data bias (differences in distance, brightness, and orientation).

To leverage the GOSet, we split it into a train and test set. The train set is composed of the first 13 objects and corresponds to the first 2 backgrounds. The test set is composed of the rest of the objects and backgrounds. This split prevents overlap and presents a real-world testing scenario.

### 5. Experiments

In all experiments, we use the training/testing partition mentioned above to train and evaluate the proposed method. For evaluation metrics, we report Area Under the Curve (AUC), Half Total Error Rate (HTER) [4], and Equal Error Rate (EER) [41]. Performance is video-based, which is computed via majority voting of patch scores. For each video, we use all frames; and for each frame, we randomly select 20 patches.

#### 5.1. Generic Object Anti-Spoofing

**Baseline Performance:** To demonstrate the superiority of our proposed method, we compare our method with our implementation of the recent methods [2, 7, 8, 13] on the GOSet test set. These recent methods are modality specific algorithms that perform anti-spoofing based on color and texture information. From Tab. 2, it is shown that GOLab outperforms the other anti-spoofing methods by a large margin for the GOAS task.

**Benefits of GOLab:** Tab. 3 (a) and (b) show the confusion matrices of GOLab on sensor and spoof medium classification. The classification performance for sensors is noticeably better than that of mediums, with the overall accuracy of 87.6% vs. 43.2%. Although Fig. 3 indicates the noises among medium have distinct patterns, it is worth noting that the medium noises can be "hidden" in the image by the sensor noises, which causes the lower accuracy. The accuracy for detecting live videos is 92.4% which exhibits its promising ability for the anti-spoofing task.

We compute the ROC curves of GOLab on GOSet testing data. Fig. 5 (a) and (b) show the ROC curves of different objects and different backgrounds respectively. We can see the AUCs for different objects are similar. But AUCs for different backgrounds have larger variation, which denotes that the GOLab is more sensitive to surfaces with rich texture, e.g., Carpet Flowered in (b). By comparing the ROCs for different sensors in Fig. 5 (c), we observe that the “Google Pixel” and “iPad Touch” are the hardest sensors to detect, because they are the highest and lowest quality, respectively. This causes images from the iPod to appear more spoof-like, and images from the Pixel less so, while their respective noise patterns are most distinguishable in Tab. 3. Similarly, the “Acer Laptop” is the most challenging spoof medium for anti-spoofing, shown in Fig. 5 (d).
shows the learned noise representation on the two face databases. The results are shown in Table 4.

Table 4. Performance of GOLab when trained on varying amounts of live, real spoof, and synthetic spoof data. Live data was randomly selected. For each live video, 1 or 2 (out of 6 possible) spoof videos were then selected. We randomly select from the generated data to increase the training data by 10%.

<table>
<thead>
<tr>
<th>Data</th>
<th>GOLab</th>
<th>GOLab + GOGen</th>
</tr>
</thead>
<tbody>
<tr>
<td>Live, Spoof</td>
<td>AUC HTER EER</td>
<td>AUC HTER EER</td>
</tr>
<tr>
<td>1/4, 1/6</td>
<td>79.7 26.8 27.7 79.7 27.2 27.6</td>
<td></td>
</tr>
<tr>
<td>1/4, 1/3</td>
<td>85.1 24.0 25.7 86.5 22.3 22.8</td>
<td></td>
</tr>
<tr>
<td>1/2, 1/3</td>
<td>81.8 24.7 26.7 86.0 22.2 22.8</td>
<td></td>
</tr>
<tr>
<td>1/2, 1/3</td>
<td>87.6 19.6 21.0 92.5 14.9 16.2</td>
<td></td>
</tr>
</tbody>
</table>

**Benefits of GOGen:** GOGen generates synthetic spoof images and performs data augmentation to improve the training of GOLab. It can synthesize spoof images which may be under-represented or missing in the training data. To present the advantage of GOGen, we train the GOLab with different compositions of training data. The data compositions and corresponding results are shown in Table 4. Comparing the relative performance, we see that more spoof data is more important than more live data because additional spoof data contains sensor and medium noise, whereas live data only has sensor noise. Comparing the performance of the GOLab without GOGen to those of the GOLab with GOGen, the inclusion of synthetic data during training has significant benefit for the anti-spoofing performance of GOLab. As additional sensors/mediums are introduced, GOGen can reduce the cost of future data collection by appropriately generating images for the new sensor/medium combinations.

5.2. Face Anti-Spoofing Performance

We also evaluate the generalization performance of the proposed method on face anti-spoofing tasks. We present cross-database testing between two face anti-spoofing databases, SiW and OULU-NPU. The testing on OULU-NPU follows the Protocol 1 and the testing on SiW is executed on all test data. The evaluation and comparison include two parts: firstly, we train the previous methods on either OULU-NPU or SiW, and test on the other; secondly, we train the previous methods and ours on GOSet, and test on the two face databases. The results are shown in Table 5.

Table 5. Performance of GOPad and GOLab algorithms along with SOTA face anti-spoofing algorithms on face anti-spoofing datasets. The algorithms trained on face data are cross-tested between OULU and MSU-SiW. The rest are trained on GOSet. [Key: Best, Second best]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Train</th>
<th>OULU P1</th>
<th>MSU SiW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chingovska LBP [13]</td>
<td>Face</td>
<td>38.5 44.2 30.5 31.7</td>
<td></td>
</tr>
<tr>
<td>Boulkenafet Texture [7]</td>
<td>Face</td>
<td>40.8 43.3 28.6 29.9</td>
<td></td>
</tr>
<tr>
<td>Boulkenafet SURF [8]</td>
<td>Face</td>
<td>38.2 40.8 36.0 36.7</td>
<td></td>
</tr>
<tr>
<td>Atoum et al. [2]</td>
<td>Face</td>
<td>11.8 13.3 11.0 11.2</td>
<td></td>
</tr>
<tr>
<td>Chingovska LBP [13]</td>
<td>GOPad</td>
<td>44.1 46.1 42.2 42.4</td>
<td></td>
</tr>
<tr>
<td>Boulkenafet Texture [7]</td>
<td>GOPad</td>
<td>34.6 36.7 44.1 44.9</td>
<td></td>
</tr>
<tr>
<td>Boulkenafet SURF [8]</td>
<td>GOPad</td>
<td>45.3 45.8 47.7 48.6</td>
<td></td>
</tr>
<tr>
<td>Atoum et al. [2]</td>
<td>GOPad</td>
<td>32.9 35.0 8.2 8.8</td>
<td></td>
</tr>
<tr>
<td>GOPad (Ours)</td>
<td>GOPad</td>
<td>33.4 34.2 9.5 10.2</td>
<td></td>
</tr>
<tr>
<td>GOLab (Ours)</td>
<td>GOPad</td>
<td>41.2 42.5 15.6 16.0</td>
<td></td>
</tr>
</tbody>
</table>

GOPad is structurally very similar to the Atoum et al. algorithm [2], however, [2] uses more than 10X the number of network parameters. The similar performance between these two methods implies that the leaner and faster GOPad was able to learn strong discriminative ability, regardless of its smaller size. The SOTA performance of both Atoum et al. and GOPad on SiW when trained on GOSet demonstrates the generalization ability from generic objects to face data. The lack of such performance when tested on OULU shows that the generalization of current methods to unseen sensors/mediums is poor, providing future incentive for GOGen to synthesize data that represents these devices.

We train Atoum et al. [2] using MSU SiW face dataset and test on the GOSet dataset, resulting in an AUC of 62.3, HTER of 37.0, and EER of 41.4. Comparing to Tab. 4, Atoum et al. [2] has the lowest performance, even worse than GOLab trained with the smallest amount of data. This shows that models trained only on faces are domain specific and can not model or detect the true noise in spoof images.

5.3. Ablation Study

**Noise representation:** Fig. 3 shows the learned noise prototypes for the sensors and mediums. In the last row of Fig. 3, the distinctive high frequency information is evident.
Table 6. Anti-spoofing performance of GOPad and GOLab on the GOSet dataset with varying amounts of training data.

<table>
<thead>
<tr>
<th>Data</th>
<th>GOLab</th>
<th>GOPad</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC</td>
<td>HTER</td>
</tr>
<tr>
<td>(Live, Spoof)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1/4, 1/6)</td>
<td>79.7</td>
<td>26.8</td>
</tr>
<tr>
<td>(1/4, 4/6)</td>
<td>86.0</td>
<td>21.6</td>
</tr>
<tr>
<td>(All, 4/6)</td>
<td>94.6</td>
<td>12.5</td>
</tr>
<tr>
<td>(All, All)</td>
<td>97.5</td>
<td>6.3</td>
</tr>
</tbody>
</table>

Figure 6. Illustration of GOLab-based anti-spoofing, with 2 success (left) and 2 failure (right) cases for live (top row) and real spoof (bottom row) using 20 patches per image. The color bar shows the output range of the network: 1 is spoof and 0 is live. The score at the top left corner is the average of all patches.

in the FFT of the spoof medium prototypes. In contrast, the FFT for the sensor prototypes are similar. To evaluate the advantage of modeling noise prototypes, we train the GOGen network without noise prototypes by constructing \( T = [I, M'_s, M'_m] \). \( M'_s \) and \( M'_m \) are of the same size of as \( M_s \) and \( M_m \), and with all elements being zeros except the prototypes of selected spoof and medium being all 1. The Rank-1 accuracy for sensor and spoof medium identification of the related GOLab on the synthesized data is 11.0% and 19.7%, respectively. However, by learning noise prototypes, as shown in Fig. 2, the accuracy is 56.0% and 26.3%.

**Binary or N-ary Classification:** We train the GOPad on the GOSet dataset, and we find that GOPad performs better than GOLab when only a small amount of data is utilized for training. However, GOLab is better than GOPad when using a larger training set. The training data to be used is chosen by randomly sampling of the GOSet training set. We attribute this improvement to the auxiliary information (classification between multiple sensors and mediums) that is learned by GOLab for the sensor and spoof medium identification. The detailed comparison is shown in Tab. 6.

**GOLab Loss Functions:** To demonstrate the benefit of both sensor and medium classification in the GOLab algorithm, experiments were run using each independently. Using only \( S_s(I) \) in Eq. 6, we obtain a Rank-1 accuracy of 84.7%. Similarly using only \( S_m(I) \), we obtain an accuracy of 42.0% with anti-spoofing performance AUC of 85.9, HTER of 22.1 and EER of 22.8. By fusing tasks, we improve accuracy for sensor and medium to 87.6% and 43.2%, respectively. This also improves anti-spoofing performance to AUC of 97.5, HTER of 6.3, and EER of 6.7.

5.4. Visualization and Qualitative Analysis

Fig. 6 shows success and failure cases of the GOLab model on the GOSet dataset. This suggests that the smooth, reflective background is classified disproportionately as live and the textured carpet/cloth backgrounds are inversely classified as spoof. Hence, it is crucial that GOAS and biometric anti-spoofing be possible over the entire image, because no singular patch in the image can provide an accurate and confident score for the entire image.

We show some examples of the generated synthetic spoof images in Fig. 7. We can compare the visual quality with their corresponding live and real spoof images. The GOGen network is trained to change the high frequency information in the images which are related to the sensor and spoof medium noises. GOGen is successfully able to alter the high frequency information in these patches to be more similar to the associated spoof than the input live.

6. Conclusion

We present our proposed generic object anti-spoofing method which consists of multiple CNNs designed for modeling the sensor and spoof medium noises. It generates synthetic images which are helpful for increasing anti-spoofing performance. We show that by modeling the spoof noise properly, the anti-spoofing methods are domain independent and can be utilized in other modalities. We propose the first generic object anti-spoofing dataset which contains live and spoof videos from 7 sensors and 7 spoof mediums.

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


7302


