mToFNet: Object Anti-Spoofing with Mobile Time-of-Flight Data

Yonghyun Jeong[1†], Doyeon Kim[1†], Jaehyeon Lee[1], Minki Hong[1], Solbi Hwang[1], Jongwon Choi[2*]
[1]Samsung SDS, Seoul, Korea
yhyun.jeong, dy31.kim, jhreplay.lee, mkidea.hong, solbi2.hwang@samsung.com
[2]Dept. of Advanced Imaging, Chung-Ang University, Seoul, Korea
choijw@cau.ac.kr

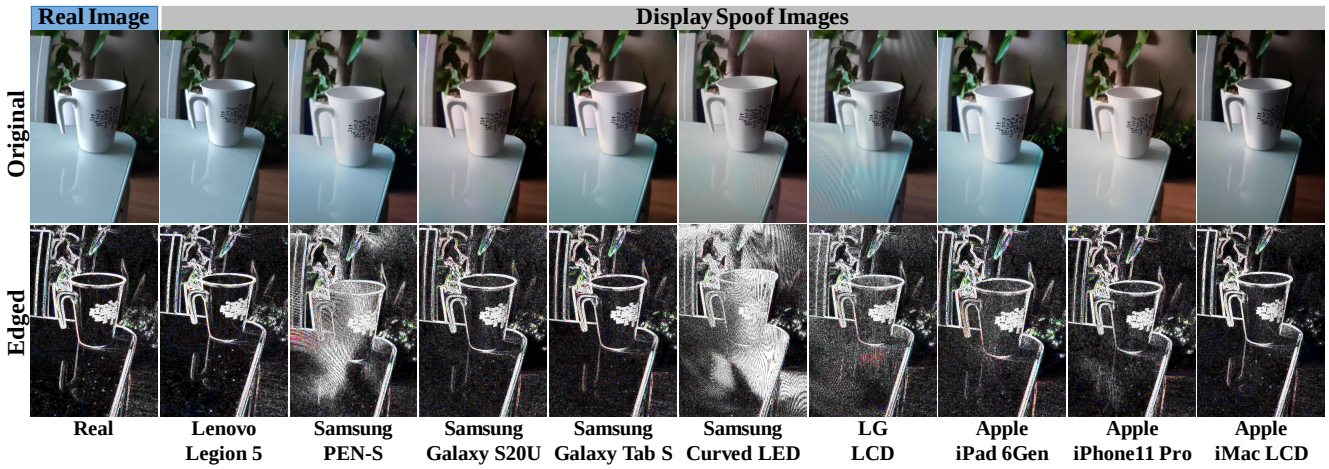


Figure 1: **Comparison between real and display images showing diverse moiré patterns in edged images.** The artifacts varying by the displays are known as moiré patterns, which can lead to overfitting of the model to the training data. The unique patterns per display makes it challenging to develop a generalized method to distinguish such spoof images. Since each display has a unique moiré pattern, it is challenging to develop a generalized model to distinguish such spoof images.

Abstract

In online markets, sellers can maliciously recapture others' images on display screens to utilize as spoof images, which can be challenging to distinguish in human eyes. To prevent such harm, we propose an anti-spoofing method using the pairs of RGB images and depth maps provided by the mobile camera with a time-of-fight sensor. When images are recaptured on display screens, various patterns differing by the screens as known as the moiré patterns can be also captured in spoof images. These patterns lead the anti-spoofing model to be overfitted and unable to detect spoof images recaptured on unseen media. To avoid the issue, we build a novel representation model composed of two embedding models, which can be trained without considering the recaptured images. Also, we newly introduce mToF dataset, the largest and most diverse object anti-spoofing dataset, and the first to utilize the time-of-flight (ToF) data. Experimental results confirm that our model achieves robust generalization even across unseen domains.

1. Introduction

As the volume of online transactions increases, the size of online person-to-person transactions is also on the rise (i.e. Craigslist). In unfortunate cases, sellers can maliciously use spoof images for scams, and buyers are forced to bear the risk of scams to proceed with transactions. To prevent such cases, many online services provide mobile applications specifically developed for secure verification with real-time capturing and direct transferring of users' images. However, such verification methods are still imperfect because the abusers can avoid such safeguards by recapturing others' images displayed on a screen. Thus, distinguishing

[†]These authors contributed equally to this work.
[*]Corresponding author

such spoof images has become one of the most important challenges to fostering reliable online transactions.

Unfortunately, however, most previous anti-spoofing studies have focused on the human face [1,7,8,20,22,25,28, 40,42]. While the common characteristics of the human face can be utilized in the face anti-spoofing detector, the object anti-spoofing detector cannot utilize the architectural properties of the target objects due to their variety in merchandise. As a result, the object anti-spoofing detector needs to focus on the difference between the real and display images, which originates from the capturing constraints.

As shown in the edged images located at the second row of Fig. 1, some display screens show distinct artifacts, which are known as moiré patterns. The moiré patterns can be utilized to train the anti-spoofing model; however, due to the uniqueness of the moiré patterns, the anti-spoofing model suffers from the overfitting issue to the various moiré patterns appearing in the training data. When overfitting occurs, the model's performance can dramatically decline when tested with the new display screens unseen during the training phase. Furthermore, the moiré patterns of some display screens appear in a subtle and almost indistinguishable manner in human eyes. Thus, for the generality across the various spoof media, a more advanced anti-spoofing detector is required to avoid the overfitting issue to the moiré patterns of various display screens.

To solve the issues, we propose a novel framework to utilize both the image and the depth map obtained with the Time-of-Flight (ToF) sensor. The proposed framework contains dual embedding models to learn the multi-sensor representation of the real images, which are trained without the display images. Since no display image is considered during training, the representation model can ignore the effect of moiré patterns entirely. Thus, our proposed model can achieve improved robustness across the various types of spoof media. To train and evaluate our model, we collect *mToF* dataset, which provides the largest amount of real and display images captured with the various objects on diverse spoof media, each paired with the ToF map. The ToF map is the depth map obtained with the ToF sensor, which estimates the depth based on the duration of the light emitted from the sensor to reach the object and return. *mToF dataset* is the largest and the most diverse object anti-spoofing dataset and the first to utilize ToF maps in this field of study. By using the mToF dataset, numerous experimental results confirm that our anti-spoofing method can outperform other models and achieve state-of-the-art performance on generalized detection in various combinations of objects and spoof media. [1]

- In the field of object anti-spoofing, our study is the first to employ the images with depth information gathered by the mobile ToF sensor.
- Using the RGB images with the depth maps, we propose a generalized anti-spoofing method to distinguish even the unseen display images during the training phase.
- We introduce a new dataset of $12,529$ pairs of RGB images and the corresponding ToF maps, all of which are labeled as either 'real' or 'display.'
- Numerous experiments validate the effectiveness of our method in object anti-spoofing.

[1] https://github.com/SamsungSDS-Team9/mToFNet

2. Related Work

2.1. Face Anti-Spoofing

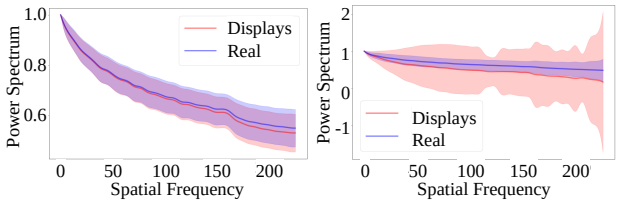
Most previous studies in anti-spoofing methods have been focused on the face category only, usually for the biometric recognition system to allow access to genuine users and prevent identity theft. For texture-based anti-spoofing, early studies focus on the hand-crafted feature descriptors, such as the local binary patterns [7,8,28], and the histogram of the oriented gradients [22,40]. The Convolutional Neural Networks (CNN) are also employed for face anti-spoofing, such as [1], which utilizes CNN and score fusion methods. Also, [25] employ a combination of CNN, principle component analysis, and support vector machine as a face anti-spoofing method. Then, [20] inversely decomposed a spoof face into a live face and a spoof noise for classification. Recently, [42] introduced a vast amount of face anti-spoofing dataset with rich annotations.

2.2. Object Anti-Spoofing

To expand the scope, we explore the literature on the object anti-spoofing for a deeper analysis. Recently, [35] take the issue of recaptured images on spoof media using the CNN-based framework consisting of GOGen, GODisc, and GOLab. [35] also provide GOSet, a new dataset consisting of 2,849 videos captured with 7 camera sensors, 7 spoof media, and 24 objects for Generic Object Anti-Spoofing (GOAS). [19] have tackled the issue by analyzing the difference in depth of field of two images, each with a different focal length. [19] also provides a unique paired dataset using various objects and three spoof media, and each pair consists of two images with the same viewpoint but different focal lengths. Our work improves upon the prior literature by utilizing various object categories and spoof media with mobile ToF data and providing the richest content and the largest amount of dataset.

2.3. Studies on RGB-D Images

Generally referred to colored images containing depth information, RGB-D has been employed in various research areas. [24] suggest placing 3D bounding boxes to detect



(a) Difference in the real and display in image domain.

(b) Difference in the real and display in ToF domain.

Figure 2: **Comparison in power spectra of real and display images.** The statistical difference between the real and display becomes magnified when ToF maps are utilized.

objects using 2D information in RGB-D images, in order to decrease the run-time and 3D search space. For RGB-D salient object detection, [3] propose a fusion method for the RGB and depth information using CNN. Also, RGB-D is employed for object classification, as [34] introduces a model that combined convolutional and recursive neural networks. Also, [2, 12] suggests utilizing the RGB-D images for object classification using dictionary learning and covariance descriptors. RGB-D is used in-depth estimation to improve performance, as many studies based on deep networks have suggested [11, 26, 32, 38]. Recently, with a spread of Time-of-Flight (ToF) sensors in mobile cameras, [31] suggest a joint alignment and refinement using deep learning for the ToF RGB-D module. We find that it is useful to detect the display images using the depth maps, since the conventional spoof media result in the flat regions in depth maps.

3. ToF-based Object Anti-spoofing

In this section, we observe the difference in ToF maps between the real pairs and display pairs, and propose a robust anti-spoofing method utilizing the ToF maps. The real pairs of images and ToF maps are obtained by capturing the actual objects, and the display pairs are acquired by recapturing the images displayed on the screens. First, we conduct a comparative analysis on the image level and the ToF level in Section 3.1, then we introduce our overall framework and its training method in detail in Section 3.2.

3.1. ToF Frequency Analysis

To compare the characteristics of the images and ToF maps, we conduct a frequency-level analysis. First, we transform the 2-Dimensional (2D) images and ToF maps into magnitude spectrum by applying Discrete Fourier Transform (DFT). DFT is a mathematical approach to disintegrate a discrete signal into the frequency-level components ranging from zero up to the maximum frequency that is proportional to the spatial resolution [16].

To reduce the dimension of the 2D spectrum from the images and ToF maps, the frequency-level 2D spectrum is transformed into 1D power spectrum by applying Azimuthal averaging [10], which is a computational approach to obtain a robust 1D representation of the power spectrum. Utilizing the method, we can scale down the number of features but maintain the relevant information. By using the training set of our mToF dataset, Fig. 2 shows the comparison between the 1D power spectrum of the images and the ToF maps. The detailed explanations on mToF dataset are given in Section 4. As illustrated, the ToF maps contain more 'artifacts' or 'patterns' from the display screens and thus show a greater difference in distributions. Based on the characteristics, we design the overall framework as described in the following section.

3.2. Overall Framework

Using the two types of modalities including the images and the ToF maps, we design a framework to distinguish between the real and display pair without using the moiré patterns. To overcome the overfitting issue by the moiré patterns of the training data, we need to entirely ignore the RGB images of the display pairs. Thus, we utilize the ToF maps for the display pairs, while both the images and the ToF maps are used for the real pairs. Since the conventional classification network cannot be trained by the inconsistent type of input data, we build a ToF representation network that can be trained even without the display images.

ToF representation network contains two separate embedding models trained to represent the data distributions of only the real pairs and both pairs, respectively. The embedding model with the real pairs is named as *multi-modal embedding model*, which receives both the images and the ToF maps and reconstructs the ToF maps. The other embedding model with both pairs is named as *ToF-modal embedding model*, and it only receives the ToF maps to recover the identity ToF maps. We let the two representation features of the embedding models be similar to each other upon the real pairs, which results in the abnormal distribution of the representation features for the display pairs. Then, the two representation features of the embedding models are concatenated to be inserted into the spoof classifier, which detects the display images by recognizing the dissimilarity of the two representation features. The overall framework is illustrated in Figure 3.

3.2.1 ToF Representation Network

We define the input image and the ToF map as $x_I \in \mathbb{R}^{w \times h \times 3}$ and $x_T \in \mathbb{R}^{w \times h}$, respectively, where w is the width and h means the height of data. We assume that the two types of data are resized to have the same size. The two embedding models respectively contain an encoder and the following generator. The encoders of the embedding models compress the input data into the representation feature,

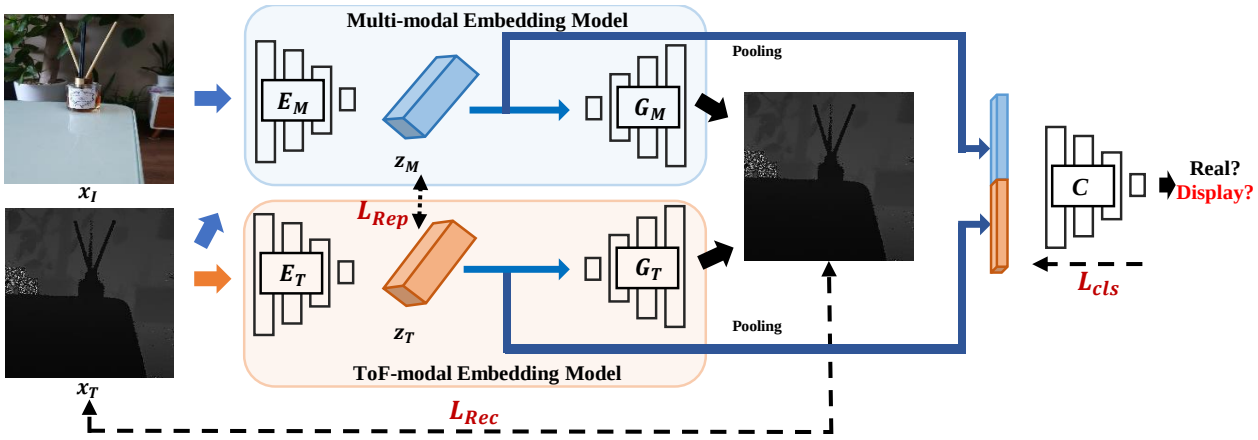


Figure 3: **Overall framework.** Using the real images and their paired ToF maps as the inputs, the first embedding model is trained to reconstruct the ToF maps. Then, using the sampled ToF maps, the second embedding model is trained to reconstruct the identical ToF maps. The representation features from the two embedding models are mapped into the identical feature space, from which the classifier makes the prediction.

which is used to reconstruct the input ToF map by the following generator. We define the encoder and generator of the multi-modal embedding model as E_M and G_M, respectively, and the encoder and generator of the ToF-modal embedding model are denoted as E_T and G_T, respectively.

The input of E_M is the 4-channel feature $c(x_I, x_T)$ concatenating x_I and x_T of the real pairs. Then, the generated latent code is defined by z_M (*i.e.* $z_M \equiv E_M(c(x_I, x_T))$). From the generated z_M, the following G_M reconstructs the input ToF map. Instead of the concatenated features, only the ToF map is reconstructed by G_M, which leads to stable training of the embedding model by reducing the information to be reconstructed. Unlike the multi-modal embedding model ($G_M(E_M(\bullet))$), the ToF-modal embedding model ignores the images as its input. Thus, E_T gets the 1-channel feature of x_T, and its generated latent code can be obtained as $z_T \equiv E_T(x_T)$. Finally, the generator G_T of the ToF-modal embedding model ($G_T(E_T(\bullet))$) reconstructs the input ToF map by estimating $G_T(z_T)$. The encoders contain three convolution layers of stride 2 and kernel size 3, which are respectively followed by a ReLU activation function [29] and batch normalization. In the generators, three transposed convolution layers are serially connected with stride 2 and kernel size 2. To allow the latent codes to implicitly represent the data distribution, we do not employ the U-Net architecture [18] nor the skip connection [15] for our embedding models.

The training loss for the ToF representation network consists of two reconstruction losses respectively for each embedding model and one representation loss. Using the reconstruction losses, we train the two embedding models to effectively represent the ToF information. The multi-modal embedding model is trained to always reconstruct the input ToF maps only from the real pairs. Thus, for the training phase of the multi-modal embedding model, the input images are sampled as $(x_I, x_T) \sim X_{real}$, where X_{real} is the set of real pairs. The reconstruction loss for the multi-modal embedding model can be defined as follows:

$$\mathcal{L}^M_{rec} = \mathbb{E}_{(x_I,x_T)\sim X_{real}}[||x_T - D_M(E_M(c(x_I, x_T)))||_2]. \quad (1)$$

In the case of the ToF-modal embedding model, its encoder receives the ToF maps from both the real and display pairs, and its generator reconstructs the ToF map similarly to the input ToF maps. Thus, the reconstruction loss for the ToF-modal embedding model can be defined as follows:

$$\mathcal{L}^T_{rec} = \mathbb{E}_{(x_I,x_T)\sim X}[||x_T - G_T(E_T(x_T))||_2], \quad (2)$$

where X presents the set of all pairs.

In addition, to detect the display pairs by using the latent codes, we design a representation loss to reduce the distance of the latent codes from the two embedding models when the real pairs are given. By applying the loss, only the latent codes of real pairs are similar to each other, so the display pairs can be easily detected based on their discrepancies. Then, the representation loss can be derived as follows:

$$\mathcal{L}_{rep} = \mathbb{E}_{(x_I,x_T)\sim X_{real}}[||E_M(c(x_I, x_T)) - E_T(x_T)||_1]. \quad (3)$$

3.2.2 Spoof Classifier

The spoof classifier compares the latent codes from the two embedding models to predict the real and display pairs. First, z_M and z_I are pooled by an average pooling and vectorized, which are defined by $\hat{z}_M$ and $\hat{z}_I$, respectively. Then, the spoof classifier is fed by the concatenated feature of $[\hat{z}_M, \hat{z}_I]$. The spoof classifier consists of three fully

Table 1: Dataset Comparison

Dataset	Category	Domain	Size	# of Spoof Subject	# of Spoof Medium	Paired	ToF
Celeb-A Spoof [42]	Face	Image	625,537	10,177	-	No	No
GOAS [35]	Object	Video	2,849	24	7	No	No
DoFNet [19]	Object	Image	2,757	6	3	Yes	No
Ours	Object	Image	12,529	27	16	Yes	Yes

Table 2: Types of Spoof Mediums for mToF Dataset

Monitor	**Laptop**	**Mobile Phone**	**Tablet PC**	**Projector**
Samsung LED (S27H850QF)	Samsung PEN-S (NT930)	Samsung S20 Ultra (SM-G988)	Samsung Tab S4 (SM-T830)	NEC LCD (M311XG)
Samsung Curved LED (C34J791WT)	Lenovo Legion 5 (Y540)	Samsung Note 8 (SM-N950N)	Apple iPad 6Gen (A1893)	LG CineBeam DLP (HF60LA)
Apple iMac LCD (A1419)	Apple Macbook Pro (A1398)	Apple iPhone12 Pro (A2407)	-	-
LG LCD (32GK850G)	LG Gram (15Z990)	Apple iPhone11 Pro (A2215)	-	-

connected layers to classify the concatenated features as either real or display pairs. To operate the spoof classifier, only the encoders of the two embedding models are necessary, so we remove their generators to improve the computational efficiency in the test phase. The spoof classifier is trained for the binary classification of 'real' or 'display', so we utilize the conventional cross-entropy loss with softmax function [5].

4. mToF Dataset

Recently, several mobile manufacturers have begun to equip ToF sensors on their mobile devices such as Samsung Galaxy S20+, Apple iPhone 12 Pro, and LG G8 ThinkQ. Due to its easy accessibility and simple operation, the ToF sensor is a great tool for the measurement of depth information. Our *mToF* dataset is collected to overcome the limitations in size and variety of the previous object anti-spoofing datasets and to provide additional ToF data for the first time in this area of research. With 12,529 images in 27 categories captured on 16 different spoof media, mToF dataset is the largest in size with the most variety, compared to other recent anti-spoofing datasets as shown in Table 1. Using the ToF map, we can effectively distinguish whether an image is taken of a real object, or recaptured on a display medium.

4.1. Data Composition

Our new mToF dataset can be divided into two major segments: the images taken of the real objects are defined as *real images*, and the recaptured images of the real images on the spoof media are defined as *display images*. Each real and display image is paired with its corresponding ToF map, which is a unique feature compared to other datasets. For data diversity, our mToF dataset is composed of 27 object categories, including book, bottle, bowl, bug spray, candle, cellphone holder, condiment, cosmetic, cup, diffuser, dish, food container, glasses case, household goods, humidifier, mouth wash, music album, ointment, pan, perfume, pot, snack, toy, vitamin, wallet, wet-wipe, and window cleaner.

Also, we use 16 different spoof media, including various monitors, laptops, mobile phones, tablet PCs, and projectors from diverse manufacturers, as listed in Table 2. For monitors, four types of display screens are used, including a Samsung Wide Quad High Definition (WQHD) Light-Emitting Diode (LED) monitor (S27H850QF, 27-inch, 2019), a Samsung WQHD curved quantum-dot LED monitor (C34J791WT, 34-inch, 2018), a retina 5K liquid-crystal display (LCD) of Apple iMac (A1419, 27-inch, 2017), and an LG anti-glare LCD monitor (32GK850G, 32-inch, 2017). Also, four types of laptops are used, including a Samsung PEN-S (NT930, 13.3-inch, 2019), a Lenovo Legion 5 (Y540, 15.6-inch, 2020), an Apple Macbook Pro (A1398, 15.4-inch, 2015), and an LG Gram (15Z990, 15.6-inch, 2019). For mobile phones, four types of devices are used, including a Samsung Galaxy S20 Ultra (SM-G988, 6.9-inch, 2020), a Samsung Galaxy Note 8 (SM-N950N, 6.3-inch, 2017), an Apple iPhone 12 Pro (A2407, 6.1-inch, 2020), an Apple iPhone 11 Pro (A2215, 5.8-inch, 2019). Lastly, two types of projectors are used, including a projector screen for NEC LCD projector (NP-M311XG, 2012) and an LG Cinebeam Digital Light Processing (DLP) projector (HF60LA, 2019).

4.2. Data Collection

All images are captured with a mobile application specifically developed to obtain a pair of RGB images and its corresponding ToF maps, using the mobile ToF sensor on a Samsung Galaxy Note 10. For data collection, we first take the real images by focusing on the subjects located in the middle and then recapture the display images by setting the mobile phone on a tripod in front of the screens displaying the real pictures. All display images are taken in the dark with the lights off to minimize unnecessary factors, such as light reflections and noises. Also, to imitate real-world settings, we include various backgrounds behind the subjects to provide a variety of shooting environments. The backgrounds include bookshelves, curtains, home appliances, kitchen cabinets, paintings, and more. For training, the images are randomly mixed to prevent data bias.

4.3. On-device Refinement of ToF Maps

The raw ToF maps captured with the ToF sensor require a refining process due to the numerous artifacts and noises included, which are affected by various factors, such as multi-path interference, motion artifacts, shot noises, and different camera response functions [14, 31]. After capturing the paired images with the ToF maps, we refine the data within the mobile application according to the API guide provided by Android[2]. For refinement, we acquire the ToF map with an even width and height, in which each pixel is 16-bit valued indicating the range of ToF measurements. When the ToF map is captured, the correctness of the depth value is saved along with the depth value at the three most signif-

[2]https://developer.android.com/reference/android/graphics/ImageFormat

icant bits of them. The correctness of the depth value is encoded in the three bits as follows: when a value of zero represents 100% confidence, one represents 0% confidence. Our collected images have a resolution of 1280×720, and their ToF maps have a resolution of 240×180.

4.4. Pre-processing of Paired Images

Without pre-processing, we cannot use the RGB image and the depth map simultaneously, since the depth maps from the ToF sensor is represented by 16 bits while a color channel of the RGB images is in 8 bits. To equalize the bit scales of the RGB images and the depth maps, we transform the bit scale of the depth map into 8 bits where the value ranges between 0 and 255. Thus, the bit scale of the depth map becomes equivalent to one color channel of the RGB image. Afterward, we resize all of the RGB images and the depth maps to 180×180 in resolution, which is the smallest image length among the captured data. Finally, when the samples captured with the same target object and the same type of displays are grouped by a sample set, we compose each sample set as follows: 80% as training set, 10% as validation set, and the last 10% as test set. The sample sets of real images are also organized in the same fashion.

5. Experimental Results

In this section, we conduct experiments to evaluate our model in various scenarios. The implementation details of our method are as follows. To align the resolutions of the image and ToF map, we resize the image resolution to 180×240 to match the resolution of ToF map. For data augmentation, we randomly crop the image size to 160×160. The networks are trained based on Adam optimizer [21] using the learning rate of 0.0001 and the batch size of 32 with 20 epochs. All experiments are conducted using a single NVIDIA Titan RTX GPU. To effectively evaluate the performance of our method, we employ the measurements commonly used in anti-spoofing and image forgery detection: Accuracy (Acc.), AUROC, and Average Precision (A.P.) [6, 37, 41].

5.1. Comparison models

In this section, we provide explanations for the compared models including the *PCA-based*, *frequency-based*, and *CNN-based naive models*.

5.1.1 PCA-based Model

To find a better representation method for detecting display images than the variance of ToF maps, we employ the Principal Component Analysis (PCA) [17], which is one of the most popular methods to find the representative features from raw data. Using PCA, we can obtain the projection axis maximizing the discrimination of the input samples, which is called *the principal component*. Also through PCA, we can get the multiple principal components that maximize the discrimination, however, the principal components are constrained to be orthogonal to each other. After reducing the feature size of the ToF maps by two, we integrate a linear Support Vector Machine (SVM) to detect the display pairs by using them.

5.1.2 Frequency-based Model

To show the effectiveness of the frequency-based detector with the ToF maps, we also employ a frequency-based detector [9] as one of the compared algorithms. In the detector, 2-D image or ToF map is transformed into the frequency domain by Fast Fourier Transform [16], first. Then, by utilizing the operation of Azimuthal average [10], the 2-D frequency domain is compressed into a 1-D power spectrum that compresses the frequency information of the ToF maps. The classification model is based on SVM [4] with linear kernel, which is also employed for the proposed framework.

5.1.3 CNN-based Naive Classifier

For the classification model of Convolution Neural Networks (CNN), we utilize ResNet [15], VGG [33], resnext [39], alexnet [23]. In a comparative analysis, we concatenate the ToF maps and the images to use them as the training input. Since CNN simultaneously works as the feature extractor, we utilize the raw ToF maps for CNN instead of the feature extraction methods. We replace the last fully connected layer of the models to reduce the number of classes by 2. Then, the network is fine-tuned by using the softmax cross-entropy loss for the binary classification. The network is updated by Adam optimizer [21], and the number of epochs, the learning rate, and the batch size are set to 20, 0.0001, and 32, respectively.

5.1.4 Face anti-spoofing model

To compare our method to the face anti-spoofing model, we utilize the methodology proposed by Moon *et al*. [27] and George and Marcel [13]. Moon *et al*. [27] detects the spoofing face by only using the RGB images, so the decline in performance is dramatic compared to our model. George and Marcel [13] utilize the RGB images and depth maps simultaneously like our method and show state-of-the-art performance in the face anti-spoofing problem. Since only the trained model and the loss are shared publicly for the method, we implement the code for training and testing, which is fully validated by producing the same performance with the uploaded model.

5.2. Unseen Display Performance

For real-world applications, the anti-spoofing task should be able to cover the unseen environment, because it is impractical to gather all necessary data to detect every

Table 3: **Unseen display performance.**

Models	Data Type	Target-display			Unseen-display			All-display		
		Acc.	A.P.	AUROC	Acc.	A.P.	AUROC	Acc.	A.P.	AUROC
AlexNet [23]	RGB, ToF	51.37	88.64	93.59	40.91	68.77	62.70	41.56	70.01	64.63
VGG [33]	RGB, ToF	98.63	100.00	100.00	87.12	92.97	88.16	87.84	93.41	88.90
ResNext [39]	RGB, ToF	100.00	100.00	100.00	86.26	86.17	74.64	87.12	87.03	76.23
ResNet [15]	RGB, ToF	100.00	100.00	100.00	86.67	88.95	80.53	87.50	89.64	81.75
PCA [17]	ToF	100.00	100.00	100.00	87.31	87.31	87.31	88.10	88.10	88.10
Moon *et al*. [27]	RGB	73.57	89.75	85.08	63.33	75.34	73.80	63.97	76.24	74.50
George & Marcel [13]	RGB, ToF	100.00	100.00	100.00	86.67	91.13	86.30	87.50	91.68	87.16
Durall *et al*. [9]	ToF	100.00	100.00	100.00	93.33	93.33	93.33	93.75	93.75	93.75
Durall *et al*. [9]	RGB, ToF	100.00	100.00	100.00	93.33	93.33	93.33	93.75	93.75	93.75
Ours	RGB, ToF	100.00	100.00	100.00	96.67	100.00	100.00	96.88	100.00	100.00

Table 4: **Ablation Study.**

	Target-display			Unseen-display			All-display		
	Acc.	A.P.	AUROC	Acc.	A.P.	AUROC	Acc.	A.P.	AUROC
w/o ToF	50.00	64.54	56.54	44.75	40.33	23.06	45.08	41.84	25.15
w/o Rep.	100.00	100.00	100.00	87.26	95.94	92.40	88.06	96.19	92.88
w/o L_{rep}	100.00	100.00	100.00	86.67	95.80	95.13	87.50	96.06	95.43
Ours	100.00	100.00	100.00	96.67	100.00	100.00	96.88	100.00	100.00

Table 5: **Moire-based Train performance.**

Number of Displays	Target-display			Unseen-display			All-display		
	Acc.	A.P.	AUROC	Acc.	A.P.	AUROC	Acc.	A.P.	AUROC
1	50.00	56.64	50.83	46.03	35.94	19.18	46.28	37.23	21.16
2	50.00	68.88	66.76	47.90	39.99	24.96	48.16	43.60	30.19
4	63.36	76.27	66.87	58.61	69.73	60.12	59.80	71.37	61.81
8	69.78	87.07	84.36	64.81	81.57	74.79	67.30	84.32	79.58
15	77.40	82.52	84.33	74.70	80.45	81.17	77.23	82.39	84.13
Ours	100.00	100.00	100.00	96.67	100.00	100.00	96.88	100.00	100.00

spoof medium [19, 35, 37]. Thus, for practical applications, we validate the proposed method by estimating the robustness in anti-spoofing when only a limited number of displays is considered during training. Table 3 presents the robustness to the unseen media when only one display is considered in the training phase. 'Target-display' experiments are of the models trained and tested using the same display types. Also, 'Unseen-display' experiments are of the models tested with unseen display types, while 'All-display' experiments are of the models tested with all display types.

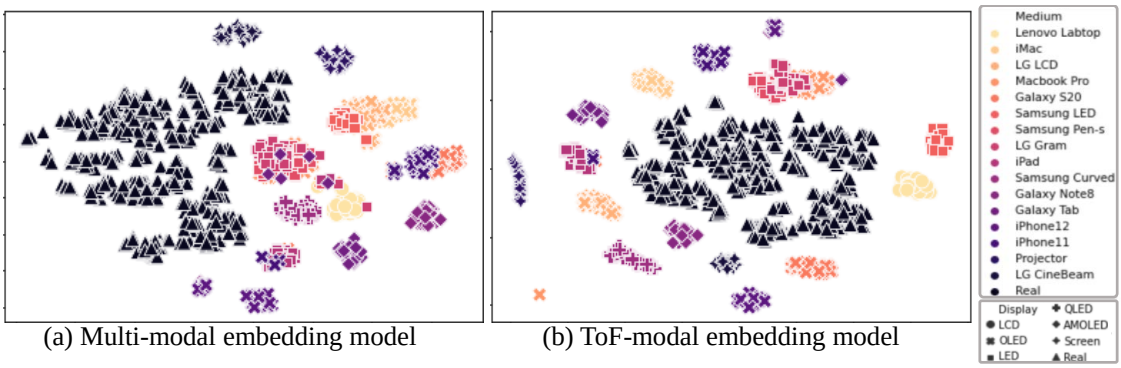
We compare with not only the CNN-based methods [15, 23, 33, 39], such as AlexNet, VGG, and ResNet, but also the PCA-based methods [25]. As employed in [9, 10], we additionally compare with the frequency-based detection methods. As shown in the experimental results, our method achieves the most robust performance in distinguishing the display images, using the ToF maps. Furthermore, the state-of-the-art method for face anti-spoofing detection [13] shows a decline in robustness for the unseen media, even though the method also considers the depth map. This result validates that the object anti-spoofing detection cannot be solved just by applying the face anti-spoofing detectors. Since the moiré pattern is easily distinguishable by every compared model except for AlexNet, the target-display performance is consistently superior among all models. However, in the case of unseen-display experiments, our proposed framework shows state-of-the-art performance by far.

5.3. Ablation

Table 4 indicates the experimental results of the ablation study to validate the individual component of our method. First, we conduct experiments of our model using the CNN classifier and the images only, without the ToF maps (*w/o ToF*). In this case, the model learns the moiré patterns [30] of the display images for classification, which results in a decline in performance. Such results demonstrate the importance of using the ToF maps for accurate classification of the real and display images. Second, we add the ToF maps along with the images as the input of the CNN classifier of our model but eliminate the representation network (*w/o Representation Network*). By considering the ToF maps, we can achieve improved performance in distinguishing the real and display images. Lastly, we conduct experiments of our model without the representation loss (*w/o* L_{rep}), which makes both the encoder and generator to exist in the same representation space (*w/o* L_{ref}). Although both of the embedding models are trained to reconstruct ToF maps, the results validate the representation loss is essential since each network is independent of the other.

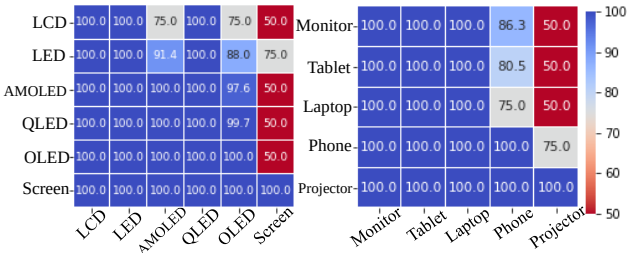
5.4. Effectiveness to Train Various Moiré Patterns

In this section, we conduct additional experiments verifying that the training of numerous moiré patterns is ineffective to improve the generality of the anti-spoof detector. For the experiments, we utilize Resnet-50 [15], and its hyperparameters for training are also the same as those in Section 5.2. For the compared models, to maintain the moiré patterns clearly on the display images, we randomly crop the display images to use as the input, instead of resizing as proceeded in the ToF maps. Also, for data augmentation, we apply random flips and random rotations of the images to expand the training data. By gradually increasing the number of training displays, we observe the model's performance when tested with the target-display and unseen-display. The experimental results are listed in Table 5, which indicates the improved performance as the number of training displays increases. However, our method using the ToF maps outperforms this method yet. Thus, the training of numerous types of moiré patterns is less effective than our proposed method ignoring the display images.



(a) Multi-modal embedding model (b) ToF-modal embedding model

Figure 4: **t-SNE Visualization for Real and Display Pairs.** The latent codes from real and display pairs are visualized by t-SNE. While the real pairs are gathered to form a stable distribution, the latent codes from the display images are grouped by the spoof media and scattered out of the distribution of real pairs.



(a)Display Type (b) Device Type

Figure 5: **Confusion Matrix.** Experiments are conducted with various combinations of training and test domains to assess our model by categorizing the spoof media by the (a) display type and (b) device type. The row and column indicate the training domain and the test domain, respectively.

5.5. Visualization of Latent Codes

Using t-Stochastic Neighbor Embedding (t-SNE) [36], we observe whether our method based on the representation training actually makes the real images and the real ToF maps exist in the same space, so that it can effectively distinguish between the real and display images. Fig. 4 illustrates the clear division between the latent codes from the real and display images. The features are well categorized according to the device type, which can provide interesting relations among the various spoofing media.

5.6. Analysis of Display and Device Types

To show the different characteristics of various displays, we perform additional experiments to evaluate the performance with various combinations of training and test domains. As shown in Fig. 5, we build a taxonomy of the various spoof media categorized into two categories: the display types and device types. As shown in the display types of the taxonomy, our method experiences difficulties with the screen type when trained with other media. Similarly, in the device type, the projector type is more challenging to detect using our method trained with other media. Also, while the characteristics of the monitor, tablet, and laptop types can be trained by the phone display, it is more challenging to detect the phone display using those characteristics only. This indicates that the robustness among the media can be related asymmetrically. From the results, we discover that the proposed algorithm can be improved for future investigations to enhance the robustness of the projector type and to analyze the asymmetric relations among the spoof media.

6. Conclusion

With the expansion of online commercial transactions, it becomes increasingly important to prevent image spoofing in various categories. Our newly proposed method achieves the most robust performance in distinguishing the real and display images by using the ToF maps, even when tested with unseen displays during the training phase. Numerous experiments confirm our model's robustness compared to others, and the individual components of our framework are evaluated through vigorous ablation study. Also, our *mToF* dataset is the largest and the most diverse dataset for object anti-spoofing and is composed of the real and display images paired with the ToF maps. We expect our mToF dataset to be utilized in various tasks, such as 3D reconstruction and object detection. Furthermore, we believe our work can make ethical impacts in society by recovering the digital mistrust in online markets. To enhance the applicability, we plan to extend the proposed framework to work with the camera sensor only by utilizing the depth map from the multiple-view geometry or single-view depth estimation.

Acknowledgements

This work was partly supported by Samsung SDS, Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2021-0-01341, Artificial Intelligence Graduate School Program(Chung-Ang University)), and (2021-0-02067, Next Generation AI for Multi-purpose Video Search).

References

[1] Yousef Atoum, Yaojie Liu, Amin Jourabloo, and Xiaoming Liu. Face anti-spoofing using patch and depth-based CNNs. In *IJCB*, 2017.

[2] W. J. Beksi and N. Papanikolopoulos. Object classification using dictionary learning and rgb-d covariance descriptors. In *ICRA*, 2015.

[3] Hao Chen and Youfu Li. Progressively complementarity-aware fusion network for rgb-d salient object detection. In *CVPR*, 2018.

[4] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 1995.

[5] David R Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232, 1958.

[6] Davide Cozzolino, Justus Thies, Andreas Rössler, Christian Riess, Matthias Nießner, and Luisa Verdoliva. Forensictransfer: Weakly-supervised domain adaptation for forgery detection. *arXiv*, 2018.

[7] Tiago de Freitas Pereira, André Anjos, José Mario De Martino, and Sébastien Marcel. LBP- TOP based countermeasure against face spoofing attacks. In *ACCV*, 2012.

[8] Tiago de Freitas Pereira, André Anjos, José Mario De Martino, and Sébastien Marcel. Can face anti-spoofing countermeasures work in a real world scenario? In *ICB*, 2013.

[9] Ricard Durall, Margret Keuper, and Janis Keuper. Watch your Up-Convolution: CNN Based Generative Deep Neural Networks are Failing to Reproduce Spectral Distributions. In *CVPR*, 2020.

[10] Ricard Durall, Margret Keuper, Franz-Josef Pfreundt, and Janis Keuper. Unmasking deepfakes with simple features. *arXiv*, 2019.

[11] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015.

[12] D. Fehr, W. J. Beksi, D. Zermas, and N. Papanikolopoulos. RGB-D object classification using covariance descriptors. In *ICRA*, 2014.

[13] Anjith George and Sebastien Marcel. Cross modal focal loss for rgbd face anti-spoofing. In *CVPR*, 2021.

[14] Qi Guo, Iuri Frosio, Orazio Gallo, Todd Zickler, and Jan Kautz. Tackling 3D ToF artifacts through learning and the FLAT dataset. In *ECCV*, 2018.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[16] Michael Heideman, Don Johnson, and Charles Burrus. Gauss and the history of the fast fourier transform. *IEEE ASSP Magazine*, 1984.

[17] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 1933.

[18] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[19] Yonghyun Jeong, Jongwon Choi, Doyeon Kim, Sehyeon Park, Minki Hong, Changhyun Park, Seungjai Min, and Youngjune Gwon. DoFNet: Depth of Field Difference Learning for Detecting Image Forgery. In *ACCV*, 2020.

[20] Amin Jourabloo, Yaojie Liu, and Xiaoming Liu. Face despoofing: Anti-spoofing via noise modeling. In *ECCV*, 2018.

[21] Diederik Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *ICLR*, 2014.

[22] Jukka Komulainen, Abdenour Hadid, and Matti Pietikäinen. Context based face anti-spoofing. In *BTAS*, 2013.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 2012.

[24] Jean Lahoud and Bernard Ghanem. 2d-driven 3d object detection in rgb-d images. In *ICCV*, 2017.

[25] Lei Li, Xiaoyi Feng, Zinelabidine Boulkenafet, Zhaoqiang Xia, Mingming Li, and Abdenour Hadid. An original face anti-spoofing approach using partial convolutional neural network. In *IPTA*, 2016.

[26] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *TPAMI*, 2015.

[27] Youngjun Moon, Intae Ryoo, and Seokhoon Kim. Face antispoofing method using color texture segmentation on fpga. *Security and Communication Networks*, 2021.

[28] J. Määttä, A. Hadid, and M. Pietikäinen. Face spoofing detection from single images using micro-texture analysis. In *IJCB*, 2011.

[29] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[30] Keyurkumar Patel, Hu Han, Anil K Jain, and Greg Ott. Live face video vs. spoof face video: Use of moiré patterns to detect replay video attacks. In *ICB*, 2015.

[31] Di Qiu, Jiahao Pang, Wenxiu Sun, and Chengxi Yang. Deep End-to-End Alignment and Refinement for Time-of-Flight RGB-D Module. In *ICCV*, 2019.

[32] Anirban Roy and Sinisa Todorovic. Monocular depth estimation using neural regression forest. In *CVPR*, 2016.

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[34] Richard Socher, Brody Huval, Bharath Bath, Christopher D Manning, and Andrew Y Ng. Convolutional-recursive deep learning for 3d object classification. In *NeurIPS*, 2012.

[35] Joel Stehouwer, Amin Jourabloo, Yaojie Liu, and Xiaoming Liu. Noise modeling, synthesis and classification for generic object anti-spoofing. In *CVPR*, 2020.

[36] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

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

[38] Junyuan Xie, Ross Girshick, and Ali Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *ECCV*, 2016.

[39] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[40] Jianwei Yang, Zhen Lei, Shengcai Liao, and Stan Z Li. Face liveness detection with component dependent descriptor. In *ICB*, 2013.

[41] X. Zhang, S. Karaman, and S. Chang. Detecting and simulating artifacts in gan fake images. In *WIFS*, 2019.

[42] Yuanhan Zhang, Zhenfei Yin, Yidong Li, Guojun Yin, Junjie Yan, Jing Shao, and Ziwei Liu. CelebA-Spoof: Large-Scale Face Anti-Spoofing Dataset with Rich Annotations. In *ECCV*, 2020.