Fingerprint Spoof Generalization

Tarang Chugh*, *Student Member, IEEE,* and Anil K. Jain, *Life Fellow, IEEE*

***Abstract*—We present a style-transfer based wrapper, called Universal Material Generator (UMG), to improve the generalization performance of any fingerprint spoof detector against spoof materials not seen during training. Specifically, we transfer the style (texture) characteristics between fingerprint images of known materials with the goal of synthesizing fingerprint images corresponding to unknown materials, that may occupy the space between the known materials in the deep feature space. Synthetic live fingerprint images are also added to the training dataset to force the CNN to learn generative-noise invariant features to discriminate between lives and spoofs. The proposed approach is shown to improve the generalization performance of a state-of-the-art spoof detector, namely Fingerprint Spoof Buster, from TDR of 75.24% to 91.78% @ FDR = 0.2%. These results are based on a large-scale dataset of 5,743 live and 4,912 spoof images fabricated using 12 known materials. Additionally, UMG wrapper is shown to improve the average cross-sensor spoof detection performance from 67.60% to 80.63% when tested on LivDet 2017 dataset. Only a set of 100 live fingerprint images from the target sensor are used to train the UMG wrapper, alleviating the time and resources required to generate large-scale live and spoof datasets for a new sensor. We also fabricate physical spoof artifacts using a mixture of known spoof materials to explore the role of cross-material style transfer in improving generalization performance.**

***Index Terms*—Fingerprint spoof detection, presentation attack detection, liveness detection, generalization, style transfer, fingerprint spoof buster**

I. Introduction

WITH the increasing proliferation of automated fingerprint recognition systems in many diverse applications, including mobile payments, access control, international border security, and national ID, they are now a prime target for fingerprint spoof attacks [3], [4]. Fingerprint spoof attacks, one of the most common forms of presentation attacks[1], include the use of (i) *gummy fingers* [6], *i.e.* fabricated finger-like objects with accurate imitation of another individual's fingerprint ridge-valley structures, and (ii) *2D or 3D printed fingerprint targets* [7], [8], [9], [10]. Other forms of presentation attacks include use of *altered fingerprints* [11], [12], *i.e.* intentionally tampered or damaged real fingerprint patterns to avoid identification, and *cadaver fingers* [13].

Fingerprint spoof attacks can be realized using a multitude of fabrication processes ranging from basic *molding and casting* to utilizing sophisticated 2D and 3D printing techniques [6], [7], [9]. Readily available and inexpensive materials such as gelatin, play doh, and wood glue, have been utilized to fabricate high fidelity fingerprint spoofs which are capable of bypassing a fingerprint recognition system. For instance, in March 2013, a Brazilian doctor was arrested for using spoof fingers made of silicone to fool the biometric

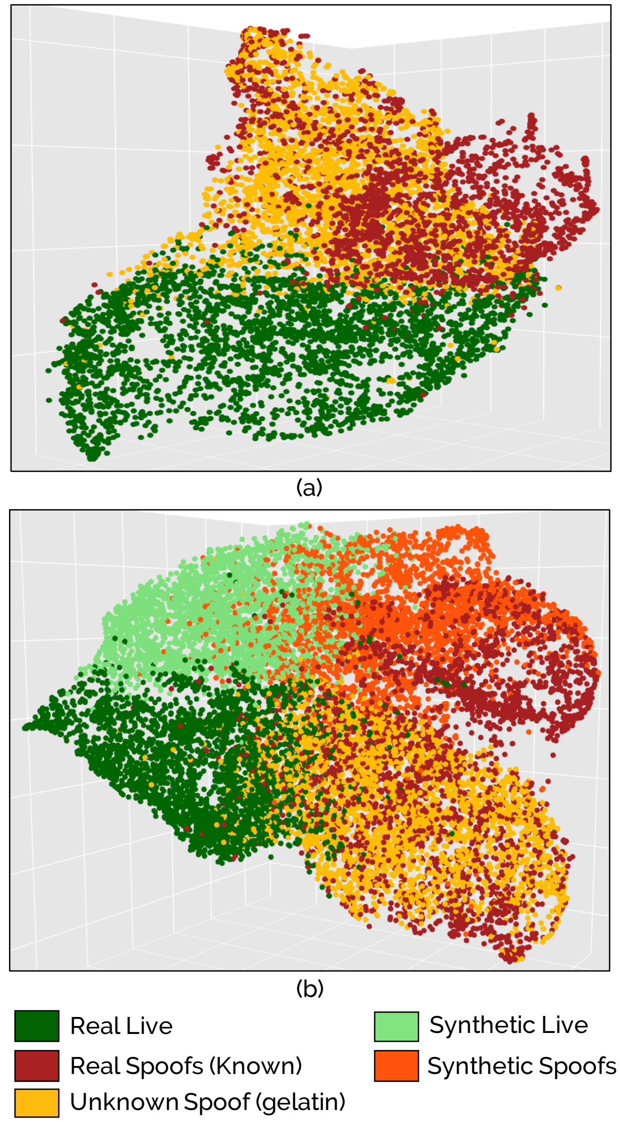

Fig. 1. 3D t-SNE visualization of feature embeddings learned by Fingerprint Spoof Buster [2] of (a) live (dark green) and eleven known spoof materials (red) *(2D printed paper, 3D universal targets, conductive ink on paper, Dragon Skin, gold fingers, latex body paint, Monster liquid latex, play doh, silicone, transparency, and wood glue)* used in training, and unknown spoof, gelatin (yellow). A large overlap between unknown spoof (gelatin) and live feature embeddings indicate poor generalization performance of state of the art spoof detector. (b) Synthetic live (bright green) and synthetic spoof (orange) images generated by the proposed Universal Material Generator (UMG) wrapper improve the separation between real live and real spoof. 3D t-SNE visualizations are available at http://tarangchugh.me/posts/umg/index.html

T. Chugh and A. K. Jain are with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, 48824. E-mail: {chughtar, jain}@cse.msu.edu
*Corresponding Author

A preliminary version of this paper was presented at the International Conference on Biometrics (ICB), Greece, June 4-7, 2019 [1].

[1]The ISO standard *IEC 30107-1:2016(E)* [5] defines presentation attacks as the *"presentation to the biometric data capture subsystem with the goal of interfering with the operation of the biometric system"*.

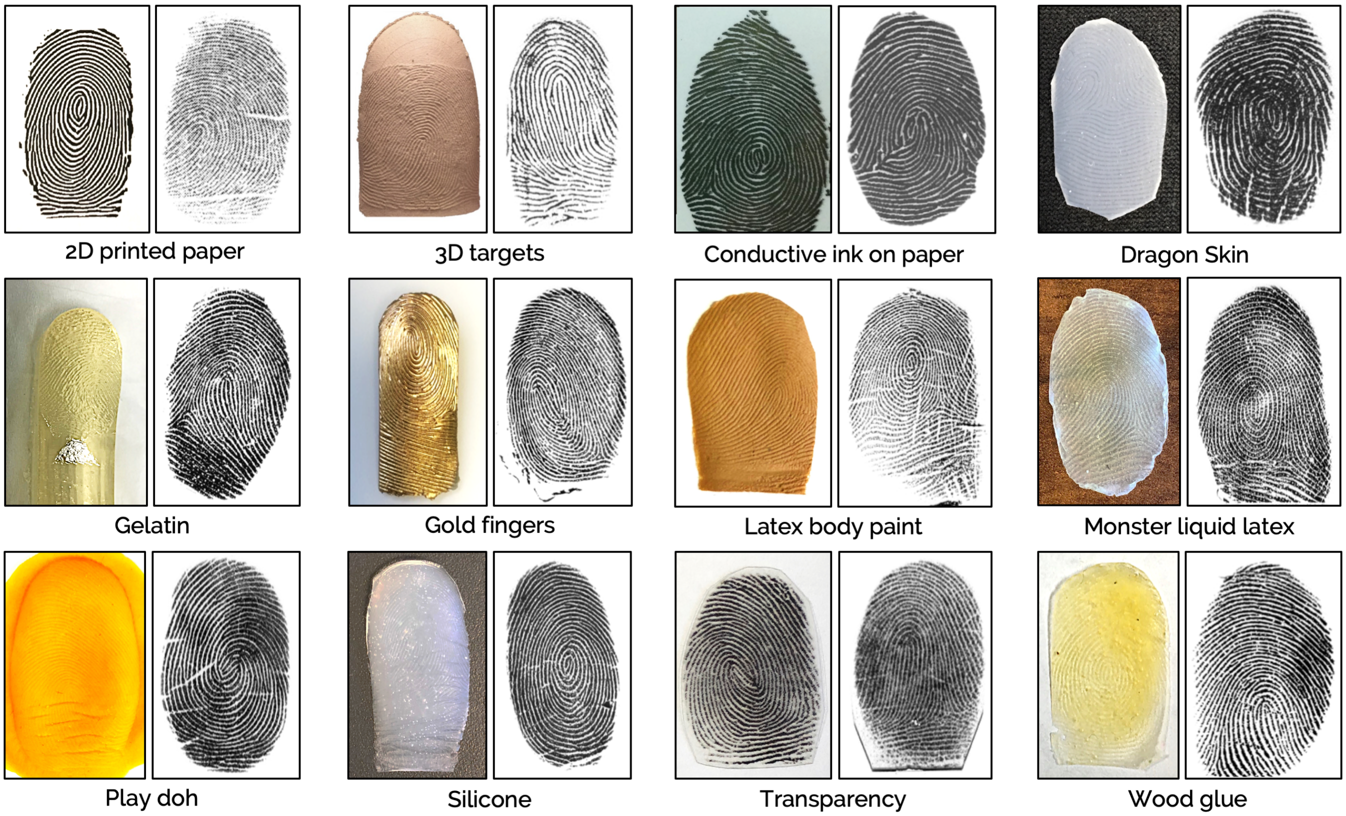

Fig. 2. Illustration of physical spoof artifacts and the corresponding images. Spoofs are fabricated using twelve different readily available and inexpensive spoof materials. The physical artifacts and their fingerprint images do not necessarily correspond to the same finger.

attendance system at a hospital in Sao Paulo[2]. In July 2016, researchers at Michigan State University unlocked a fingerprint secure-smartphone using a 2D printed fingerprint spoof to help police with a homicide case[3], using a technique proposed in [7]. In March 2018, a gang was arrested for spoofing the biometric attendance system, using glue casted in wax molds, to provide proxies for a police entrance exam[4] in Rajasthan, India. As early as in April 2019, a Galaxy S10 owner with a 3D printer and a photo of his own fingerprint was able to spoof the ultrasonic in-display fingerprint sensor on his smartphone[5]. Other similar successful spoof attacks have been reported showing the vulnerabilities of fingerprint biometric systems[6,7]. A large number of these attacks are never detected and hence not reported.

Given this vulnerability, there is an urgent need to develop accurate and generalizable spoof detection approaches. By generalizable spoof detector, we mean that it should be able to detect new spoof types not used to train the detector. In response to this growing threat, a series of fingerprint Liveness Detection (LivDet) competitions [14] have been held since 2009 to benchmark various spoof detection solutions. See [15] for results of the LivDet 2019. Another initiative is the IARPA ODIN Program [4] with the goal of developing robust spoof detection systems for fingerprints, face, and iris biometric modalities.

Generally, fingerprint spoofs can be detected by either (i) hardware-based, or (ii) software-based approaches [3], [13]. In the case of hardware-based approaches, the fingerprint readers are, typically, augmented with sensor(s) with the goal of detecting the characteristics of vitality, such as blood flow, thermal output, heartbeat, skin, distortion, odor, and so on [16], [17]. Additionally, special types of fingerprint sensing technologies have been developed for imaging sub-dermal friction ridge surface based on multi-spectral [18], short-wave infrared [19] and optical coherent tomography (OCT) [20], [21]. An open-source fingerprint reader, called RaspiReader, with a two-camera design provides two complementary streams (direct-view and FTIR) of images for spoof detection [22]. Ultrasound-based in-display fingerprint readers developed for smartphones by Qualcomm Inc. [23] utilizes acoustic response characteristics for spoof detection.

In contrast, software-based solutions can work with any commodity fingerprint reader. These solutions extract salient features from the captured fingerprint image (or a sequence of frames) for separating live and spoof images. The software-based approaches in the literature are typically based on (i) anatomical features (e.g. pore locations and their distribu-

[2]https://www.bbc.com/news/world-latin-america-21756709

[3]http://statenews.com/article/2016/08/how-msu-researchers-unlocked-a-fingerprint-secure-smartphone-to-help-police-with-homicide-case

[4]https://www.medianama.com/2018/03/223-cloned-thumb-prints-used-to-spoof-biometrics-and-allow-proxies-to-answer-online-rajasthan-police-exam/

[5]https://imgur.com/gallery/8aGqsSu

[6]http://fortune.com/2016/04/07/guy-unlocked-iphone-play-doh/

[7]https://srlabs.de/bites/spoofing-fingerprints/

TABLE I
SUMMARY OF THE STUDIES PRIMARILY FOCUSED ON FINGERPRINT SPOOF GENERALIZATION.

Study	Approach	Database	Performance
Rattani et al. [24]	Weibull-calibrated SVM	LivDet 2011	EER = 19.70%
Ding & Ross [25]	Ensemble of multiple one-class SVMs	LivDet 2011	EER = 17.60%
Chugh & Jain [2]	MobileNet trained on minutiae-centered local patches	LivDet 2011-2015	ACE = 1.48% (LivDet 2015), 2.93% (LivDet 2011, 2013)
Chugh & Jain [26]	Identify a representative set of spoof materials to cover the deep feature space	MSU-FPAD v2.0, 12 spoof materials	TDR = 75.24% @ FDR = 0.2%
Engelsma & Jain [27]	Ensemble of generative adversarial networks (GANs)	Custom database with live and 12 spoof materials	TDR = 49.80% @ FDR = 0.2%
Gonzlez-Soler et al. [28]	Feature encoding of dense-SIFT features	LivDet 2011-2015	TDR = 7.03% @ FDR = 1% (LivDet 2015), 1.01% (LivDet 2011, 2013)
Tolosana et al. [29]	Fusion of two CNN architectures trained on SWIR images	Custom database with live and 8 spoof materials	EER = 1.35%
Gajawada et al. [1] (preliminary work)	Style transfer from spoof to live images to improve generalization; requires few samples of target material	LivDet 2015, CrossMatch sensor	TDR = 78.04% @ FDR = 0.1%
Proposed Approach	Style transfer between known spoof materials to improve generalizability against completely unknown materials	MSU-FPAD v2.0, 12 spoof materials & LivDet 2017	TDR = **91.78%** @ FDR = 0.2% (MSU-FPAD v2.0); Avg. Acc. = **95.88%** (LivDet 2017)

tion [30]), (ii) physiological features (e.g. perspiration [31]), and (iii) texture-based features (e.g. Weber Local Binary Descriptor (WLBD) [32], SIFT [28]. Current approaches are learning-based, where the representation is based on training convolutional neural networks (CNN) [33], [34], [35], [36], [19], [2].

One of the major limitations of a myriad of anti-spoofing methods is their poor generalization performance across "unknown" spoof materials, that were not used during training the spoof detector. To generalize an algorithm's effectiveness across spoof fabrication materials, called *cross-material* performance, spoof detection has been referred to as an *open-set problem*[8]. Table I presents a summary of the studies primarily focused on generalization. Engelsma and Jain [39], [27] proposed using an ensemble of generative adversarial networks (GANs) on live fingerprint images with the hypotheses that features learned by a discriminator to distinguish between real live and synthesized live fingerprints can be used to separate live fingerprints from spoof fingerprints as well. One limitation of this approach is that the discriminator in the GAN architecture may only learn features that are salient in distinguishing real and synthesized live images, such as structural noise added by the generative process. It is possible that such features may not be present in the spoofs fabricated with unknown materials.

It has been shown that the selection of spoof materials used in training (known spoofs) directly impacts the performance against unknown spoofs [24], [26]. However, with the increasing popularity of fingerprint authentication systems, hackers are constantly devising new fabrication techniques and novel materials to attack them. It is prohibitively expensive to include all spoof fabrication materials in training a spoof detector. Chugh and Jain [26] analyzed the material characteristics (two optical and two physical) of 12 different spoof materials to identify a representative set of six materials that cover most of the spoof feature space. Although, this approach can be used to identify if including a new spoof material in training dataset would be beneficial, it does not improve the generalization performance against materials that are unknown during training.

We hypothesize that the texture (style) information from the known spoof fingerprint images can be transferred from one spoof type to another type to synthetically generate spoof images with potentially new styles not present in the training set. It enables a framework, called Universal Material Generator (UMG) that can be used for cross-material style transfer. We show that these style transferred images can be used effectively for augmenting CNN-based spoof detectors, significantly improving their performance against a novel material, while retaining its performance on known materials. See Figure 5 for examples of some of the style transferred images.

Realistic image synthesis is a challenging problem. Early non-parametric methods faced difficulty in generating images with textures that are not known during training [40]. Machine learning has been very effective in this regard, both in terms of realism and generality. Gatys et al. [41] perform artistic

[8]Open-set problems address the possibility of new spoof classes during testing, that were not seen during training. Closed-set problems, on the other hand, evaluate only those spoof classes that the system was trained on.

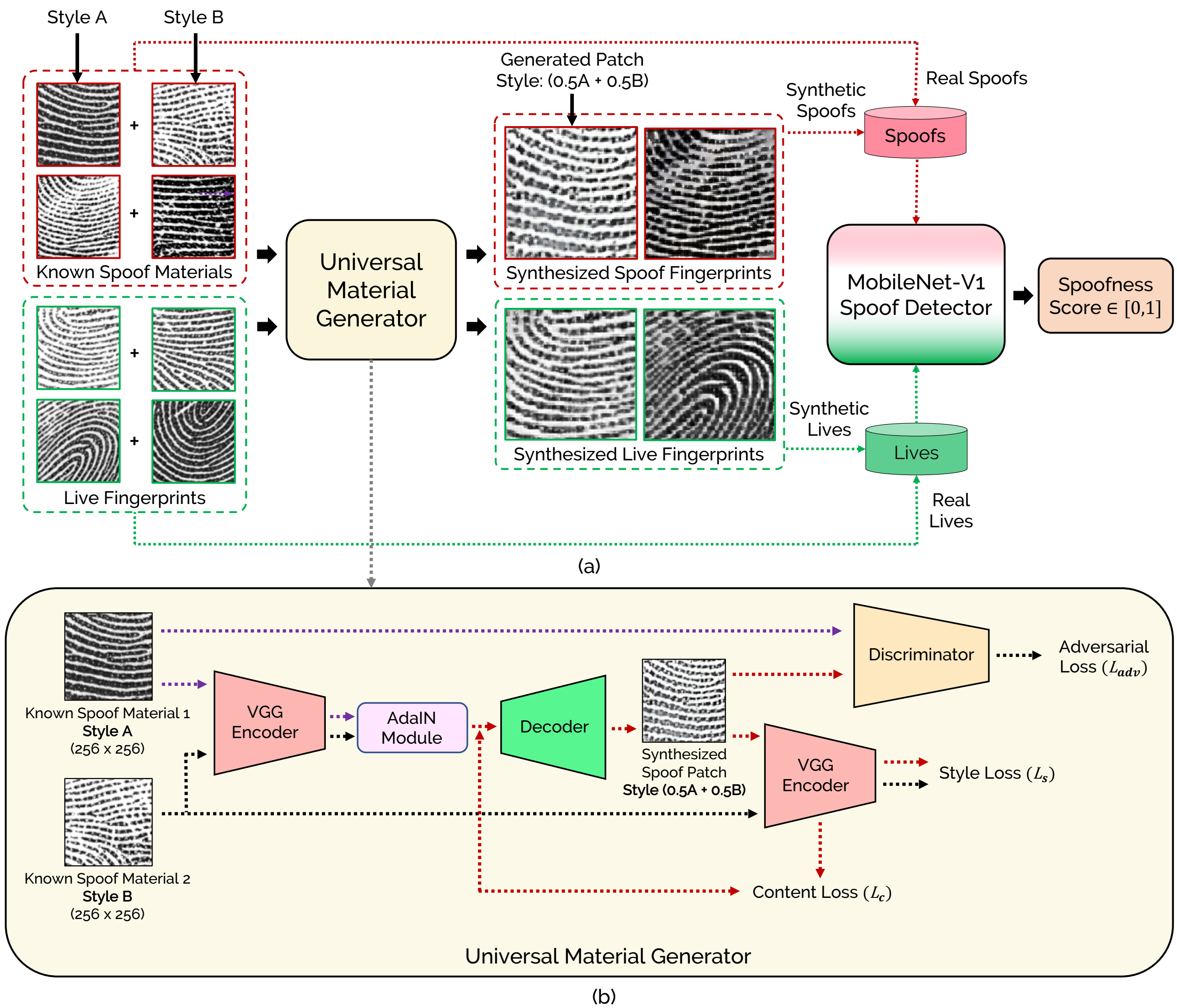

Fig. 3. Proposed approach for (a) synthesizing spoof and live fingerprint patches, and (b) design of the proposed Universal Material Generator (UMG) wrapper. An AdaIN module is used for performing the style transfer in the encoded feature space. The same VGG-19 [37] encoder is used for computing content loss and style loss. A discriminator similar to the one used in DC-GAN [38] is used for computing the adversarial loss. The synthesized patches can be used to train any fingerprint spoof detector hence our approach is referred to as a wrapper which can be used in conjunction with any spoof detector.

style transfer, combining the content of an image with the style of any other by minimizing the feature reconstruction loss and a style reconstruction loss which are based on features extracted from a pre-trained CNN at the same time. While this approach generates realistic looking images, it is computationally expensive since each step of the optimization requires a forward and backward pass through the pre-trained network. Other studies [42], [43], [44] have explored training a feed-forward network to approximate solutions to this optimization problem. There are other methods based on feature statistics to perform style transfer [45], [46]. Elgammal et al. [47] applied GANs to generate artistic images. Isola et al. [48] used conditional adversarial networks to learn the loss for image-to-image translation. Xian et al. [49] learnt to synthesize objects consistent with texture suggestions. Our approach is based on [46] for designing the Universal Material Generator that is capable of producing realistic fingerprint images containing style (texture) information from images of two different spoof materials. While simply performing style transfer like previous approaches conditions the source image with target material style, it also results in a loss in fingerprint ridge-valley information. In order to preserve both style and content, we use adversarial supervision to ensure that the synthesized images appear similar to the real fingerprint images.

The main contributions of this study are enumerated below.

- A style-transfer based wrapper, called Universal Material Generator (UMG), to extract material style (texture) characteristics from randomly selected fingerprint images of different spoof materials. This results in synthesized images potentially corresponding to unknown spoof materials.

- Experiments on a database of $5,743$ live and $4,912$ spoof images of 12 different materials to demonstrate that the proposed approach improves the generalization performance of a state-of-the-art spoof detector from TDR of 75.4% to TDR of 91.78% @ FDR = 0.2%. Additionally, experimental results on LivDet 2017 datasets show that our approach achieves state-of-the-art performance.
- Improved the cross-sensor spoof detection performance by synthesizing large-scale live and spoof datasets for a new target sensor. Our approach is shown to improve the average cross-sensor spoof detection performance from 67.60% to 80.63% on LivDet 2017 dataset.
- Used 3D t-SNE visualization to interpret the performance improvement against unknown spoof materials.
- Fabricated physical spoof artifacts using mixtures of known spoof materials to show that the synthetically generated images using fingerprint images of the same set of spoof materials correspond to an unknown material that has similar style (texture) characteristics.

II. Proposed Approach

The proposed approach includes three stages: (i) training the style-transfer based Universal Material Generator (UMG) wrapper using the spoof images of known materials (with one material left-out from training), (ii) generating synthetic spoof images using randomly selected image pairs of different but known materials, and (iii) training any spoof detector on the augmented dataset to evaluate its performance on the "unknown" material left out from training. In all our experiments, we utilize local image patches (96×96) centered and aligned using minutiae location and orientation, respectively [2]. During the evaluation stage, the spoof detection decision is made based on the average of spoofness scores for individual patches output from the CNN model. An overview of the proposed approach is presented in Fig. 3.

A. Universal Material Generator (UMG) Wrapper

The primary goal of the UMG wrapper is to generate synthetic spoof images corresponding to unknown spoof materials, by transferring the style (texture) characteristics between fingerprint images of known spoof materials. Gatys et al. [50] were the first to show that deep neural networks (DNNs) could encode not only content but also the style information. They proposed an optimization-based style-transfer approach, although prohibitively slow, for arbitrary images. Ulyanov et al. [51] trained a feed-forward network to perform style transfer with a single pass, however with a limitation of each network being restricted to a single style. In [45], Ulyanov et al. proposed use of an InstanceNorm layer to normalize feature statistics across spatial dimensions. An InstanceNorm layer is designed to perform the following operation:

$$IN(x) = \gamma\Big(\frac{x - \mu(x)}{\sigma(x)}\Big) + \beta \tag{1}$$

where, x is the input feature space, $\mu(x)$ and $\sigma(x)$ are the mean and standard deviation parameters, respectively, computed across spatial dimensions independently for each channel and each sample. It was observed that changing the affine parameters γ and β (while keeping convolutional parameters fixed) leads to variations in the style of the image, and the affine parameters could be learned for each particular style. This motivated an approach for artistic style transfer [52], which learns γ and β values for each feature space and style pair. However, this required retraining of the network for each new style.

Huang and Belongie [46] replaced the InstanceNorm layer with an Adaptive Instance Norm (AdaIN) layer, which can directly compute affine parameters from the style image, instead of learning them – effectively transferring style by imparting second-order statistics from the target style image to the source content image, through the affine parameters. We follow the same approach as described in [46] in UMG wrapper for fusing feature statistics of one known (source) spoof material image (c) providing friction ridge (content) information and source style, with another known, but different (target style) spoof material (s) in the feature space. As described in AdaIN, we apply instance normalization on the input source image feature space however not with learnable affine parameters. The channel-wise mean and variance of the source image's feature space is aligned to match those of the target image's feature space. This is done by computing the affine parameters from the target material spoof feature space in the following manner:

$$AdaIN(x, y) = \sigma(y)\Big(\frac{x - \mu(x)}{\sigma(x)}\Big) + \mu(y) \tag{2}$$

where the source (c) feature space is x and the target (s) feature space is y. In this manner, x is normalized with $\sigma(y)$ and shifted by $\mu(y)$. Our synthetic spoof generator G is composed of an encoder $f(\cdot)$ and a decoder $g(\cdot)$. For the encoder, $f(\cdot)$, we use the first few layers of a pre-trained VGG-19 network similar to [42]. The weights of this network are frozen throughout all stages of the setup. For source image (c) and the target image (s), x is $f(c)$ and y is $f(s)$. The desired feature space is obtained as:

$$t = AdaIN(f(c), f(s)) \tag{3}$$

We use the decoder, $g(\cdot)$, to take t as input to produce $T(c, s) = g(t)$ which is the final synthesized image conditioned on the style from the target image. In order to ensure that our synthesized spoof patches i.e $g(t)$ do match the style statistics of the target material spoof, we apply a style loss L_s similar to [42], [53] given as:

$$\begin{aligned}\mathcal{L}_s = \textstyle\sum_{i=1}^{L} \|\mu\left(\phi_i(g(t))\right) - \mu\left(\phi_i(s)\right)\|_2 + \\ \textstyle\sum_{i=1}^{L} \|\sigma\left(\phi_i(g(t))\right) - \sigma\left(\phi_i(s)\right)\|_2\end{aligned} \tag{4}$$

where each ϕ_i denotes a layer in the VGG-19 network we use as encoder. We pass $g(t)$ and s through $f(\cdot)$ and extract the outputs of $relu1\_1$, $relu2\_1$, $relu3\_1$ and $relu4\_1$ layers for computing $\mathcal{L}_s$.

The extent of style transfer can be controlled by interpolating between feature maps that are:

$$T(c, s, \alpha) = g((1 - \alpha) \cdot f(c) + \alpha \cdot t) \tag{5}$$

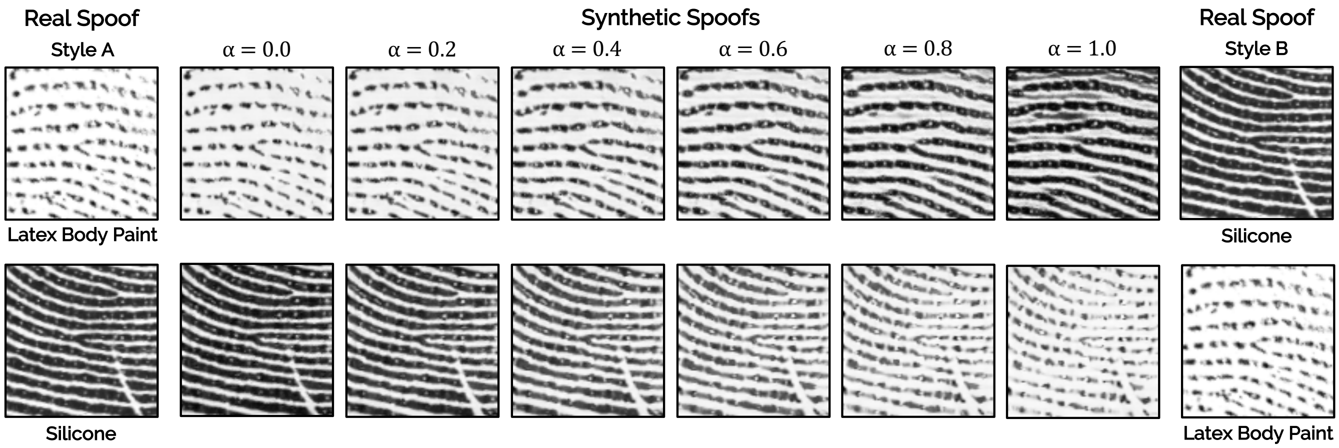

Fig. 4. Style transfer between real spoof patches fabricated with latex body paint and silicone to generate synthetic spoof patches using the proposed Universal Material Generator (UMG) wrapper. The extent of style transfer can be controlled by the parameter $\alpha \in [0, 1]$.

where setting $\alpha = 0$ will reconstruct the original content image and $\alpha = 1$ will construct the most stylized image. To combine the two known styles, we preserve the style of source spoof material while conditioning it with target spoof material by setting the value of α to 0.5.

To ensure that the synthesized images retain friction ridge (fingerprint) content from the real image, we use a content loss, $\mathcal{L}_c$, which is computed as the euclidean distance between the features of the synthesized image *i.e.* $f(g(t))$ and the target features (t) from the real image.

$$\mathcal{L}_c = \|f(g(t)) - t\|_2 \tag{6}$$

Doing the style transfer, simply using a content loss ($\mathcal{L}_c$) to ensure that content is retained is not enough to ensure that the synthesized images look like real images. Fingerprints have many details in terms of structure due to the presence of certain landmarks e.g. minutiae, ridges, and pores. With the aim of synthesizing fingerprints that look indistinguishable from the real fingerprints, we use adversarial supervision. A typical generative adversarial network (GAN) setup consists of a generator G and a discriminator D playing a *minimax game*, where D tries to distinguish between synthesized and real images, and G tries to fool D by generating realistic looking images. The adversarial objective functions for the generator ($\mathcal{L}^G_{adv}$) and discriminator ($\mathcal{L}^D_{adv}$) are given as:

$$\mathcal{L}^G_{adv} = \mathbb{E}_{t \in X_{synth}}[\log(1 - D(G(t)))] \tag{7}$$

$$\mathcal{L}^D_{adv} = \mathbb{E}_{x \in X_{real}}[\log D(x)] + \mathbb{E}_{t \in X_{synth}}[\log(1 - D(G(t)))] \tag{8}$$

In our approach, we use a discriminator as used in [38] and the generator is the decoder function $g(\cdot)$. We optimize the UMG wrapper in an end-to-end manner with the following objective functions:

$$\min_G \mathcal{L}_G = \lambda_c \cdot \mathcal{L}_c + \lambda_s \cdot \mathcal{L}_s + \mathcal{L}^G_{adv} \tag{9}$$

$$\max_D \mathcal{L}_D = \mathcal{L}^D_{adv} \tag{10}$$

Algorithm 1 Training UMG wrapper

procedure
input
  x: source image providing friction ridge content and known style A
  y: target image providing known style B
  $f(\cdot)$: encoder network; first 4 layers of VGG-19 network pre-trained on ImageNet with weights frozen during training
  $g(\cdot)$: decoder network; mirrors $f(\cdot)$ with pooling layers replaced with nearest up-sampling layers
  $D(\cdot)$: discriminator function similar to [38]
  $A(x, y)$: AdaIN operation; transfer style from x to y (using Eq. 2)
  $\alpha = 0.5$
  $\lambda_c = 0.001$, $\lambda_s = 0.002$
output
  $UMG(\cdot)$: UMG wrapper trained on known materials
begin:
  Encoding: $f_x = f(x)$ and $f_y = f(y)$
  Style transfer: $t = A(f_x, f_y)$
  Stylized image: $T(c, s, \alpha) = g((1 - \alpha) \cdot f_c + \alpha \cdot t)$
  Style Loss: $\mathcal{L}_s$ using Eq. 4
  Content Loss: $\mathcal{L}_c$ using Eq. 6
  Adversarial Loss (generator): $\mathcal{L}^G_{adv}$ using Eq. 7
  Adversarial Loss (discriminator): $\mathcal{L}^D_{adv}$ using Eq. 8
  Objective functions for training UMG wrapper
  $\min_G \mathcal{L}_G = \lambda_c \cdot \mathcal{L}_c + \lambda_s \cdot \mathcal{L}_s + \mathcal{L}^G_{adv}$
  $\max_D \mathcal{L}_D = \mathcal{L}^D_{adv}$
end

where λ_c and λ_s are the weight parameters for content loss ($\mathcal{L}_c$) and style loss ($\mathcal{L}_s$), respectively. Algorithm 1 summarizes the steps involved in training a UMG wrapper.

B. UMG-Wrapper for Spoof Generalization

Given a spoof dataset of real images, S^m_{real}, fabricated using a set of m spoof materials, we adopt a leave-one-out protocol

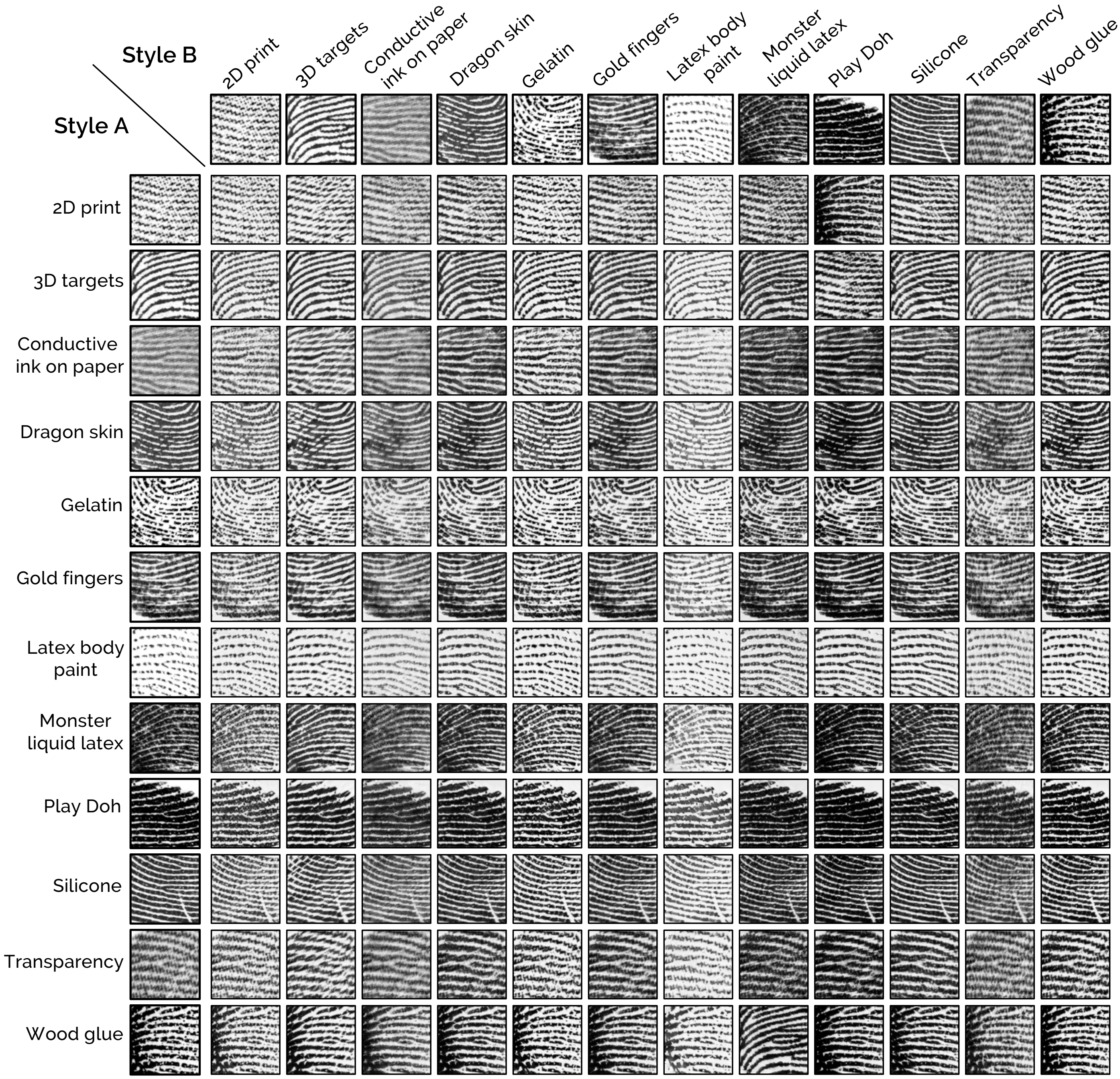

Fig. 5. Synthesized spoof patches (96 x 96) by the proposed Universal Material Generator using patches of a known (source) material (first column) conditioned on style ($\alpha = 0.5$) of another (target) known material (first row).

to split the dataset such that spoof images fabricated using $m-1$ materials are considered as "known" and used for training. And the images fabricated using the left-out m^{th} material are considered as "unknown" and used for computing the generalization performance. The fingerprint images of known materials ($k = m-1$) are used to train the UMG wrapper (UMG$_{spoof}$) described in section II-A.

After we train the UMG$_{spoof}$, we utilize a total of N_{synth} randomly selected pairs of images $\{I^i_{m_a},\ I^i_{m_b}\}$ s.t. $i \in \{1, ..., N_{synth}\}$ from known but different materials $m_a, m_b \in \{m_1, ...m_k\}$, $a \neq b$, to generate a dataset of synthesized spoof images S^k_{synth}. For each synthesized image, the friction ridge (content) information and the source material (style) characteristics are provided by the first image, I_{m_a}, and the target material (style) characteristics are provided by the second image, I_{m_b}. See Figures 4 and 5. The real spoof dataset is augmented with the synthesized spoof data to create a dataset that is used for training the fingerprint spoof detector. Additionally, we also augment the real live dataset with a total of N_{synth} synthesized live images using another UMG wrapper (UMG$_{live}$) trained on only live images. Adding synthesized live data balances the data distribution and forces the spoof detector to learn generative-noise invariant features to distinguish between lives and spoofs. Figure 6 presents

TABLE II
SUMMARY OF THE MSU-FPAD V2 AND LIVDET 2017 DATASETS.

Dataset	**MSU-FPAD v2 [26]**	**LivDet 2017 [55]**		
Fingerprint Reader	CrossMatch Guardian 200	GreenBit Dacty Scan 84C	Orcanthus Certis2 Image	Digital Persona U.are.U 5160
Image Size	800×750	500×500	$300 \times n$	252×324
Resolution (dpi)	500	569	500	500
#Live Images (Train / Test)	$4,743$ / $1,000$	$1,000$ / $1,700$	$1,000$ / $1,700$	999 / $1,692$
#Spoof Images (Train / Test)	$4,912$ (leave-one-out)	$1,200$ / $2,040$	$1,180$* / $2,018$	$1,199$ / $2,028$
Known Spoof Materials (Training)	**Leave-one-out**: 2D Printed Paper, 3D Universal Targets, Conductive Ink on Paper, Dragon Skin, Gelatin, Gold Fingers, Latex Body Paint, Monster Liquid Latex, Play Doh, Silicone, Transparency, Wood Glue	Wood Glue, Ecoflex, Body Double		
Unknown Spoof Materials (Testing)		Gelatine, Latex, Liquid Ecoflex		

*A set of 20 Latex spoof fingerprints found in the training set of Orcanthus fingerprint reader were excluded in our experiments. Only Wood Glue, Ecoflex, and Body Double are expected to be in the training dataset.

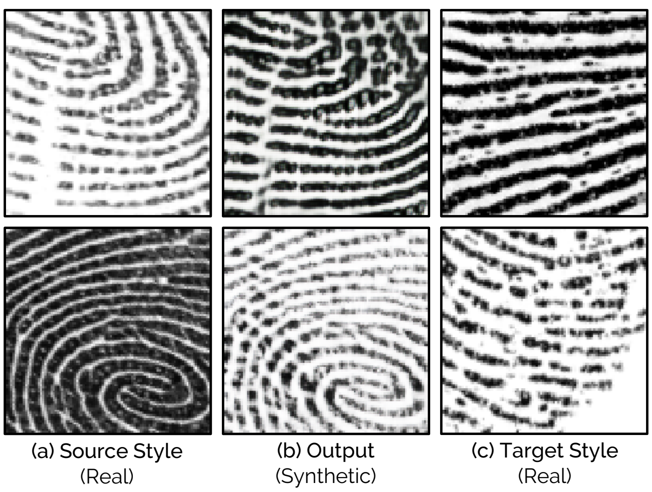

Fig. 6. Synthetic live images (96 x 96) generated by the proposed Universal Material Generator. (a) Source style images, (c) target style images, and (b) synthesized live images.

examples of the synthesized live images.

C. Fingerprint Spoof Detection

The proposed Universal Material Generator approach acts like a wrapper on top of any existing spoof detector to make it more robust to spoofs not seen during training. In this study, we employ Fingerprint Spoof Buster [2], a state-of-the-art CNN-based approach, that utilizes local patches (96×96) centered and aligned around fingerprint minutiae to train MobileNet-v1 [54] architecture. It achieved state-of-the-art performance on publicly available LivDet databases [14] and exceeded the IARPA Odin Project [4] requirement of True Detection Rate (TDR) of 97.0% @ False Detection Rate (FDR) = 0.2%.

III. EXPERIMENTS AND RESULTS

A. Datasets

The following datasets have been utilized in this study:

1) MSU Fingerprint Presentation Attack Database (FPAD) v2.0: A database of $5,743$ live and $4,912$ spoof images captured on CrossMatch Guardian 200[9], one of the most popular slap readers. The database is constructed by combining the publicly available [2] MSU Fingerprint Presentation Attack Dataset v1.0 (MSU-FPAD v1.0) and Precise Biometrics Spoof-Kit Dataset (PBSKD). Tables II and III presents the details of this database including the sensors used, 12 spoof materials, total number of fingerprint impressions, and the number of minutiae-based local patches for each material type. Fig. 2 presents sample fingerprint spoof images fabricated using the 12 materials.

2) LivDet Datasets: LivDet 2017 [55] dataset is one of the most recent[10] publicly-available LivDet datasets, containing over $17,500$ fingerprint images. These images are acquired using three different fingerprint readers, namely Green Bit, Orcanthus, and Digital Persona. Unlike other LivDet datasets, spoof fingerprint images included in the test set are fabricated using new materials (Wood Glue, Ecoflex, and Body Double), that are not used in the training set (Wood Glue, Ecoflex, and Body Double). Table II presents a summary of the LivDet 2017 dataset.

B. Minutiae Detection and Patch Extraction

The proposed UMG wrapper is trained on local patches of size 96×96 centered and aligned using minutiae points. We extract fingerprint minutiae using the algorithm proposed in [56]. For a given fingerprint image I with k detected minutiae points, $M = \{m_1, m_2, \ldots, m_k\}$, where $m_i = \{x_i, y_i, \theta_i\}$, *i.e.* the minutiae m_i is defined in terms of spatial coordinates (x_i, y_i) and orientation (θ_i), a corresponding set of k local patches $L = \{l_1, l_2, \ldots, l_k\}$, each of size $[96 \times 96]$, centered and aligned using minutiae location (x_i, y_i) and orientation (θ_i), are extracted as proposed in [2].

[9]https://www.crossmatch.com/wp-content/uploads/2017/05/20160726-DS-En-Guardian-200.pdf
[10]The testing set of LivDet 2019 database has not yet been made public.

TABLE III
GENERALIZATION PERFORMANCE (TDR (%) @ FDR = 0.2%) WITH LEAVE-ONE-OUT METHOD ON MSU-FPAD V2 DATASET. A TOTAL OF TWELVE MODELS ARE TRAINED WHERE THE MATERIAL LEFT-OUT FROM TRAINING IS TAKEN AS THE "UNKNOWN" MATERIAL FOR EVALUATING THE MODEL.

Unknown Spoof Material	**# Images**	**# Local Patches**	**Generalization Performance (TDR (%) @ FDR = 0.2%)**	
			Fingerprint Spoof Buster [26]	**Fingerprint Spoof Buster + UMG wrapper**
Silicone	1,160	38,145	67.62	**98.64**
Monster Liquid Latex	882	27,458	94.77	**96.24**
Play Doh	715	17,602	58.42	**72.36**
2D Printed Paper	481	7,381	55.44	**80.22**
Wood Glue	397	12,681	86.38	**98.97**
Gold Fingers	295	9,402	88.22	**88.59**
Gelatin	294	10,508	54.95	**97.96**
Dragon Skin	285	7,700	97.48	**100.00**
Latex Body Paint	176	6,366	76.35	**89.72**
Transparency	137	3,846	95.83	**100.00**
Conductive Ink on Paper	50	2,205	90.00	**100.00**
3D Universal Targets	40	1,085	95.00	**100.00**
Total Spoofs	**4,912**	**144,379**	Weighted mean* (± weighted s.d.)	
Total Lives	**5,743**	**228,143**	**75.24 ± 15.21**	**91.78 ± 9.43**

*The generalization performance for each spoof material is weighted by the number of images to produce the weighted mean and standard deviation.

TABLE IV
PERFORMANCE COMPARISON BETWEEN THE PROPOSED APPROACH AND STATE-OF-THE-ART RESULTS [55] REPORTED ON LIVDET 2017 DATASET FOR **CROSS-MATERIAL EXPERIMENTS** IN TERMS OF AVERAGE CLASSIFICATION ERROR (ACE) AND TDR @ FDR = 1.0%.

LivDet 2017	**LivDet 2017 Winner [55]**	**Fingerprint Spoof Buster [26]**		**Fingerprint Spoof Buster + UMG wrapper**	
	ACE (%)	**ACE (%)**	**TDR @ FDR = 1.0%**	**ACE (%)**	**TDR @ FDR = 1.0%**
Green Bit	96.44	96.68	91.07	**97.42**	**92.29**
Orcanthus	**95.59**	94.51	66.59	95.01	74.45
Digital Persona	93.71	95.12	62.29	**95.20**	**75.47**
Mean ± s.d.	95.25 ± 1.40	95.44 ± 1.12	73.32 ± 15.52	**95.88 ± 1.34**	**80.74 ± 10.02**

C. Implementation Details

The encoder of the UMG wrapper is the first four convolutional layers ($conv1\_1$, $conv2\_1$, $conv3\_1$, and $conv4\_1$) of a VGG-19 network [37] as discussed in section II-A. We use weights pre-trained on ImageNet [57] database which are frozen during training of the UMG wrapper. The decoder mirrors the encoder with pooling layers replaced with nearest up-sampling layers, and without use of any normalization layers as suggested in [46]. Both encoder and decoder utilize reflection padding to avoid border artifacts. The discriminator for computing the adversarial loss is similar to the one used in [38]. The weights for style loss and content loss are set to $\lambda_s = 0.002$ and $\lambda_c = 0.001$. We use the Adam optimizer [58] with a batch size of 8 and a learning rate of $(1e-4)$ for both generator (decoder) and discriminator objective functions. The input local patches are resized from 96×96 to 256×256. All experiments are performed in the Tensorflow framework.

For the spoof detector, we train a MobileNet-V1 [54] classifier from scratch similar to [2] using the augmented dataset. The last layer of the architecture, a 1000-unit softmax layer (originally designed to predict the $1,000$ classes of ImageNet dataset), was replaced with a 2-unit softmax layer for the two-class problem, i.e. live vs. spoof. The optimizer used to train the network is RMSProp with asynchronous gradient descent and a batch size of 100.

TABLE V
CROSS-SENSOR FINGERPRINT SPOOF GENERALIZATION PERFORMANCE ON LIVDET 2017 DATASET IN TERMS OF AVERAGE CLASSIFICATION ERROR (ACE) AND TDR @ FDR = 1.0%.

LivDet 2017		Fingerprint Spoof Buster [26]		Fingerprint Spoof Buster + UMG wrapper	
Sensor in Training	**Sensor in Testing**	**ACE (%)**	**TDR @ FDR = 1.0%**	**ACE (%)**	**TDR @ FDR = 1.0%**
Green Bit	Orcanthus	49.43	0.00	**66.05**	**21.52**
Green Bit	Digital Persona	89.37	57.48	**94.81**	**72.91**
Orcanthus	GreenBit	69.93	8.00	**81.75**	**30.91**
Orcanthus	Digital Persona	57.99	4.97	**76.36**	**28.46**
Digital Persona	GreenBit	89.54	57.06	**96.35**	**85.21**
Digital Persona	Orcanthus	49.32	0.00	**68.44**	**20.38**
Mean $\pm$ s.d.		67.60 ± 18.53	21.25 ± 28.07	**80.63 $\pm$ 12.88**	**43.23 $\pm$ 28.31**

D. Experimental Protocol

The fingerprint spoof generalization performance against unknown materials is evaluated by adopting a leave-one-out protocol [26]. In the case of MSU FPAD v2.0 dataset, one out of the twelve known spoof materials is left-out and the remaining eleven materials are used to train the proposed UMG wrapper. The real spoof data (of eleven known materials) is augmented with the synthesized spoof data generated using the trained UMG wrapper, which is then used to train the fingerprint spoof detector *i.e.* Fingerprint Spoof Buster [2]. This requires training a total of twelve different UMG wrappers and spoof detection models each time leaving out one of the twelve different spoof materials. The $5,743$ live images in MSUFPAD v2.0 are partitioned into training and testing such that there are $1,000$ randomly selected live images in testing set and the remaining $4,743$ images in training. The real live data is also augmented with synthesized live data generated using another UMG wrapper trained on real live data.

In the case of LivDet 2017 dataset, the spoof materials available in the test set (Gelatin, Latex, and Liquid Ecoflex) are deemed as "unknown" materials because these are different from the materials included in the training set (Wood Glue, Ecoflex, and Body Double). To evaluate the generalization performance, we evaluate the performance of Fingerprint Spoof Buster with and without using the UMG wrapper and compare with the state-of-the-art published results. As the LivDet 2017 dataset contains fingerprint images from three different readers, we train two UMG wrappers per sensor, one for each of the live and the spoof training datasets.

We adopt the leave-one-out protocol to simulate the scenario of encountering unknown materials to evaluate the generalization ability of Fingerprint Spoof Buster. One spoof material out of the known materials is left-out from the training set which is then utilized during testing. This requires training a separate model for each subset of known materials. In the case of MSU Fingerprint Presentation Attack Database v2.0, a total of 12 different MobileNet-v1 models each time leaving out one of the 12 spoof materials. The $5,743$ live images are partitioned into training and testing such that there are $1,000$ randomly selected live images in testing set and the remaining $4,743$ images in training.

E. Cross-Material Fingerprint Spoof Generalization

Table III presents the generalization performance of the proposed approach on the MSU FPAD v2.0 dataset. The mean generalization performance of the spoof detector against unknown spoof materials improved from TDR of 75.24% to TDR of 91.78% @ FDR = 0.2%, resulting in a 67% decrease in the error rate, when the spoof detector is trained in conjunction with the proposed UMG wrapper. Table IV presents a performance comparison of the proposed approach and the state-of-the-art approach when tested on the publicly available LivDet 2017 dataset. The proposed UMG wrapper improves the state-of-the-art average cross-material spoof detection performance from 95.44% to 95.88%. However, a much higher performance gain is observed, from TDR of 73.32% to 80.74%, at a strict operating point of FDR = 1%.

F. Cross-Sensor Fingerprint Spoof Generalization

Fingerprint images captured using different fingerprint sensors, typically, have unique characteristics due to different sensing technologies, sensor noise, and varying resolution. As a result, fingerprint spoof detectors, especially CNN-based, are known to suffer from poor generalization performance in cross-sensor scenario, where the spoof detector is trained on images captured using one sensor and tested on images from another. Improving cross-sensor spoof detection performance is important in order to alleviate the time and resources involved in collecting large-scale datasets with introduction of new sensors. To improve the cross-sensor performance, we employ the proposed UMG wrapper to synthetically generate large-scale live and spoof dataset to train a spoof detector for the target sensor.

Given a real fingerprint database, D^A_{real}, collected on a source fingerprint sensor, F^A, containing real live, L^A_{real}, and real spoof S^A_{real} datasets, s.t. $D^A_{real} = \{L^A_{real} \cup S^A_{real}\}$, the proposed UMG wrapper is used to generate $50,000$ synthetic live images, L^B_{synth}, and $50,000$ synthetic spoof images, S^B_{synth}, for a target sensor, F_B. The UMG wrapper is trained only on

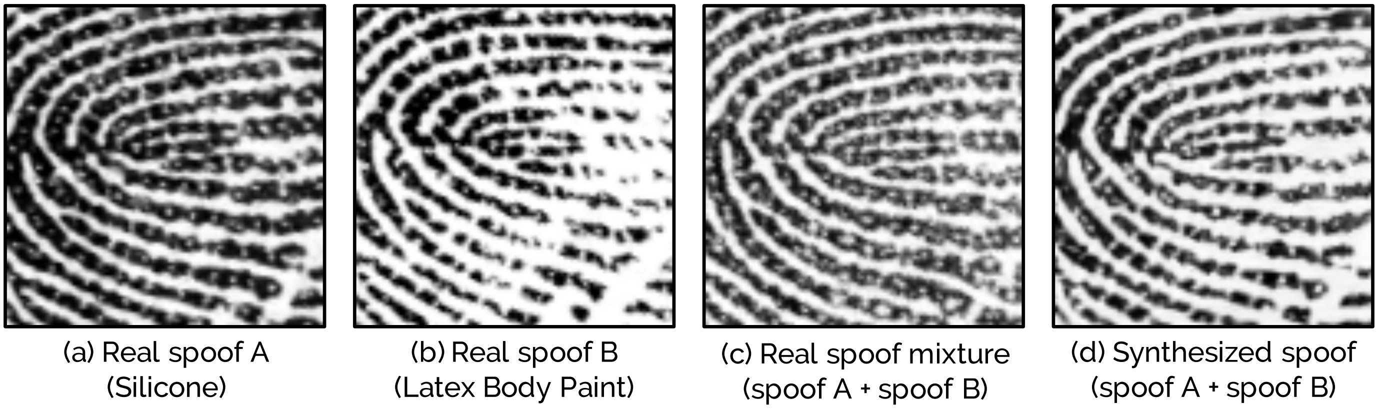

Fig. 7. Fingerprint patches fabricated with real spoofs (a) silicone, (b) latex body paint, (c) their mixture (in 1:1 ratio), and (d) synthesized using UMG wrapper with style transfer between silicone and latex body paint.

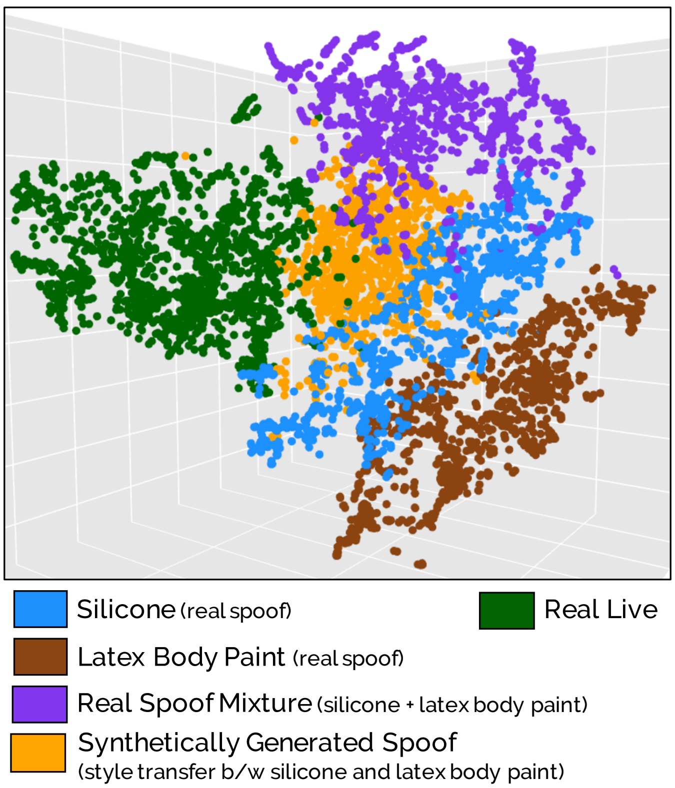

Fig. 8. 3D t-SNE visualization of feature embeddings of real live fingerprints, spoof fingerprints fabricated using silicone, latex body paint, and their mixture (1:1 ratio), and synthesized spoof fingerprints using style-transfer between silicone and latex body paint spoof fingerprints. The embeddings are available at http://tarangchugh.me/posts/umg/index.html

the live images collected on S_B, and used for style transfer on L^A_{real} and S^A_{real} to generate L^B_{synth}, and S^B_{synth}, respectively. We evaluate the cross-sensor generalization performance using LivDet 2017 dataset where the UMG wrapper trained on source sensor, say Green Bit, is used to generate synthetic data for a target sensor, say Orcanthus, using only a small set of 100 live fingerprint images from the target sensor[11]. The spoof detector is trained from scratch only on the synthetic dataset created for target sensor using UMG wrapper and tested on the real test set of the target sensor. Table V presents the cross-sensor fingerprint spoof generalization performance of the spoof detector in terms of average classification accuracy and TDR (%) @ FDR = 1%. We note that the proposed UMG wrapper improves the average cross-sensor spoof detection performance from 67.60% to 80.63%.

G. Computational Requirements

The proposed approach includes an offline stage of training the UMG wrapper and synthesis of fingerprints for augmenting the training dataset. Therefore, once the spoof detector is trained on the augmented data, the proposed approach has no impact on the computational requirements in the online spoof detection test stage. The proposed UMG wrapper takes under 2 hours to train, and around 1 hour to generate 100,000 local fingerprint patches on a Nvidia GTX 1080Ti GPU.

IV. Fabricating Unknown Spoofs

To explore the role of cross-material style transfer in improving generalization performance, we fabricate physical spoof specimens using two spoof materials, namely silicone and latex body paint, and their mixture in a 1:1 ratio by volume[12]. We fabricate a total of 24 physical specimens, including 8 specimens for each of the two materials, and 8 specimens using their mixture. A total of 72 spoof fingerprints, 3 impressions/specimen, are captured using a CrossMatch Guardian 200 fingerprint reader. Fingerprint Spoof Buster, trained on twelve known spoof materials including silicone and latex body paint, achieves TDR of 100% @ FDR = 0.2% on the two known spoof materials, and TDR of 83.33 @ FDR = 0.2% against the mixture. We utilize the testing dataset of 1,000 live fingerprint images from MSU FPAD v2.0 for these experiments.

We utilize the proposed UMG wrapper to generate a dataset of 5,000 synthesized spoof patches[13] using cross-material

[11]Note that an average of $\sim$ 3100 local patches are extracted from 100 live fingerprint images in LivDet 2017 experiments.

[12]Note that not all spoof materials can be physically combined and may result in mixtures with poor physical properties for them to be used to fabricate any good quality spoof artifacts.

[13]Around 1,100 minutiae-based local patches are extracted from 24 fingerprint images corresponding to each material.

style transfer between spoof fingerprints of silicone and latex body paint. Fingerprint Spoof Buster, fine-tuned using the synthesized dataset, improves the TDR from 83.33% to 95.83% @ FDR = 0.2% when tested on the silicone and latex body paint mixture, highlighting the role of the style-transferred synthesized data in improving generalization performance. Figure 7 presents sample fingerprint patches of the two spoof materials, silicone and latex body paint, their physical mixture, and synthesized using style-transfer. Figure 8 presents the 3D t-SNE visualization of feature embeddings of live fingerprints (green), two materials, silicone (blue) and latex body paint (brown), their mixture (purple), and synthetically generated images (orange). Although the mixture embeddings are not exactly in between the embeddings for the two known materials, possibly due to low-dimensional t-SNE representation, they are close to the embeddings of the synthetically generated spoof images. This explains the improvement in performance against mixture when synthesized spoofs are used in training. Therefore, the proposed UMG wrapper is able to generate spoof images that are potentially similar to the unknown spoofs.

V. Conclusions

Automatic fingerprint spoof detection is critical for secure operation of a fingerprint recognition system. Introduction of new spoof materials and fabrication techniques poses a continuous threat and requires design of robust and generalizable spoof detectors. To address that, we propose a style-transfer based wrapper, Universal Material Generator (UMG), to improve the generalization performance of any spoof detector against novel spoof fabrication materials that are unknown to the system during training. The proposed approach is shown to improve the average generalization performance of a state-of-the-art spoof detector from TDR of 75.24% to 91.78% @ FDR = 0.2% when evaluated on a large-scale dataset of 5,743 live and 4,912 spoof images fabricated using 12 materials. It is also shown to improve the average cross-sensor performance from 67.60% to 80.63% when tested on LivDet 2017 dataset, alleviating the time and resources required to generate large-scale spoof datasets for every new sensor. We have also fabricated physical spoof specimens using a mixture of known spoof materials to explore the role of cross-material style-transfer in improving generalization performance.

Acknowledgment

This research is based upon work supported in part 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, either expressed or implied, of 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 therein.

References

[1] R. Gajawada, A. Popli, T. Chugh, A. Namboodiri, and A. K. Jain, "Universal Material Translator: Towards Spoof Fingerprint Generalization," in *IEEE International Conference on Biometrics (ICB)*, 2019.

[2] T. Chugh, K. Cao, and A. K. Jain, "Fingerprint Spoof Buster: Use of Minutiae-centered Patches," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2190–2202, 2018.

[3] S. Marcel, M. S. Nixon, J. Fierrez, and N. Evans, Eds., *"Handbook of Biometric Anti-Spoofing: Presentation Attack Detection"*, 2nd ed. Springer, 2019.

[4] ODNI, IARPA, "IARPA-BAA-16-04 (Thor)," https://www.iarpa.gov/index.php/research-programs/odin/odin-baa, 2016.

[5] International Standards Organization, "ISO/IEC 30107-1:2016, Information Technology—Biometric Presentation Attack Detection—Part 1: Framework," https://www.iso.org/standard/53227.html, 2016.

[6] T. Matsumoto, H. Matsumoto, K. Yamada, and S. Hoshino, "Impact of artificial gummy fingers on fingerprint systems," in *Proc. SPIE*, vol. 4677, 2012, pp. 275–289.

[7] K. Cao and A. K. Jain, "Hacking mobile phones using 2D Printed Fingerprints," MSU Tech. report, MSU-CSE-16-2 https://www.youtube.com/watch?v=fZJI_BrMZXU, 2016.

[8] S. S. Arora, K. Cao, A. K. Jain, and N. G. Paulter, "Design and Fabrication of 3D Fingerprint Targets," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 10, pp. 2284–2297, 2016.

[9] S. S. Arora, A. K. Jain, and N. G. Paulter, "Gold Fingers: 3D Targets for Evaluating Capacitive Readers," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 9, pp. 2067–2077, 2017.

[10] J. J. Engelsma, S. S. Arora, A. K. Jain, and N. G. Paulter, "Universal 3D wearable Fingerprint Targets: Advancing Fingerprint Reader Evaluations," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 6, pp. 1564–1578, 2018.

[11] S. Yoon, J. Feng, and A. K. Jain, "Altered fingerprints: Analysis and detection," *IEEE TPAMI*, vol. 34, no. 3, pp. 451–464, 2012.

[12] E. Tabassi, T. Chugh, D. Deb, and A. K. Jain, "Altered Fingerprints: Detection and Localization," in *IEEE International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2018.

[13] E. Marasco and A. Ross, "A survey on antispoofing schemes for fingerprint recognition systems," *ACM Computing Surveys*, vol. 47, no. 2, p. 28, 2015.

[14] D. Yambay, L. Ghiani, G. L. Marcialis, F. Roli, and S. Schuckers, "Review of Fingerprint Presentation Attack Detection Competitions," in *Handbook of Biometric Anti-Spoofing*, S. Marcel, M. S. Nixon, J. Fierrez, and N. Evans, Eds. Springer, 2019.

[15] G. Orrù, R. Casula, P. Tuveri, C. Bazzoni, G. Dessalvi, M. Micheletto, L. Ghiani, and G. L. Marcialis, "LivDet in Action-Fingerprint Liveness Detection Competition 2019," *arXiv preprint arXiv:1905.00639*, 2019.

[16] A. Antonelli, R. Cappelli, D. Maio, and D. Maltoni, "Fake finger detection by skin distortion analysis," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 3, pp. 360–373, 2006.

[17] D. Baldisserra, A. Franco, D. Maio, and D. Maltoni, "Fake fingerprint detection by odor analysis," in *Proc. ICB*. Springer, 2006, pp. 265–272.

[18] C. D. Robison and M. S. Andrews, "System and method of fingerprint anti-spoofing protection using multi-spectral optical sensor array," Mar. 26 2019, US Patent 10,242,245.

[19] R. Tolosana, M. Gomez-Barrero, J. Kolberg, A. Morales, C. Busch, and J. Ortega-Garcia, "Towards fingerprint presentation attack detection based on convolutional neural networks and short wave infrared imaging," in *2018 International Conference of the Biometrics Special Interest Group (BIOSIG)*. IEEE, 2018, pp. 1–5.

[20] Y. Moolla, L. Darlow, A. Sharma, A. Singh, and J. Van Der Merwe, "Optical coherence tomography for fingerprint presentation attack detection," in *Handbook of Biometric Anti-Spoofing*. Springer, 2019, pp. 49–70.

[21] T. Chugh and A. K. Jain, "OCT Fingerprints: Resilience to Presentation Attacks," *arXiv preprint arXiv:1908.00102*, 2019.

[22] J. J. Engelsma, K. Cao, and A. K. Jain, "Raspireader: An open source fingerprint reader facilitating spoof detection," *arXiv preprint arXiv:1708.07887*, 2017.

[23] M. Agassy, B. CASTRO, A. Lerner, G. Rotem, L. Galili, and N. Altman, "Liveness and spoof detection for ultrasonic fingerprint sensors," Apr. 16 2019, US Patent 10,262,188.

[24] A. Rattani, W. J. Scheirer, and A. Ross, "Open set fingerprint spoof detection across novel fabrication materials," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 11, pp. 2447–2460, 2015.

[25] Y. Ding and A. Ross, "An ensemble of one-class SVMs for fingerprint spoof detection across different fabrication materials," in *Proc. IEEE WIFS*, 2016, pp. 1–6.
[26] T. Chugh and A. K. Jain, "Fingerprint Presentation Attack Detection: Generalization and Efficiency," *IEEE International Conference on Biometrics (ICB)*, 2019.
[27] J. J. Engelsma and A. K. Jain, "Generalizing Fingerprint Spoof Detector: Learning a One-Class Classifier," *IEEE International Conference on Biometrics (ICB)*, 2019.
[28] L. J. González-Soler, M. Gomez-Barrero, L. Chang, A. Pérez-Suárez, and C. Busch, "Fingerprint Presentation Attack Detection Based on Local Features Encoding for Unknown Attacks," *arXiv preprint arXiv:1908.10163*, 2019.
[29] R. Tolosana, M. Gomez-Barrero, C. Busch, and J. Ortega-Garcia, "Biometric Presentation Attack Detection: Beyond the Visible Spectrum," *IEEE Transactions on Information Forensics and Security*, 2019.
[30] S. Schuckers and P. Johnson, "Fingerprint Pore Analysis for Liveness Detection," Nov. 14 2017, US Patent 9,818,020.
[31] E. Marasco and C. Sansone, "Combining perspiration-and morphology-based static features for fingerprint liveness detection," *Pattern Recognition Letters*, vol. 33, no. 9, pp. 1148–1156, 2012.
[32] Z. Xia, C. Yuan, R. Lv, X. Sun, N. N. Xiong, and Y.-Q. Shi, "A Novel Weber Local Binary Descriptor for Fingerprint Liveness Detection," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.
[33] R. F. Nogueira, R. de Alencar Lotufo, and R. C. Machado, "Fingerprint Liveness Detection Using Convolutional Neural Networks," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1206–1213, 2016.
[34] H.-U. Jang, H.-Y. Choi, D. Kim, J. Son, and H.-K. Lee, "Fingerprint Spoof Detection using Contrast Enhancement and Convolutional Neural Networks," in *International Conference on Information Science and Applications*. Springer, 2017, pp. 331–338.
[35] T. Chugh, K. Cao, and A. K. Jain, "Fingerprint Spoof Detection using Minutiae-based Local Patches," in *IEEE International Joint Conference on Biometrics (IJCB)*, 2017.
[36] F. Pala and B. Bhanu, "Deep Triplet Embedding Representations for Liveness Detection," in *Deep Learning for Biometrics. Advances in Computer Vision and Pattern Recognition.* Springer, 2017, pp. 287–307.
[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[38] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.
[39] J. J. Engelsma, K. Cao, and A. K. Jain, "Raspireader: Open Source Fingerprint Reader," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.
[40] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu, "Sketch2photo: Internet Image Montage," *ACM Transactions on Graphics*, vol. 28, no. 5, p. 124, 2009.
[41] L. A. Gatys, A. S. Ecker, and M. Bethge, "A Neural Algorithm of Artistic Style," *arXiv preprint arXiv:1508.06576*, 2015.
[42] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual Losses for Real-time Style Transfer and Super-resolution," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 694–711.
[43] X. Wang, G. Oxholm, D. Zhang, and Y.-F. Wang, "Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5239–5247.
[44] C. Li and M. Wand, "Precomputed Real-time Texture Synthesis with Markovian Generative Adversarial Networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 702–716.
[45] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Improved Texture Networks: Maximizing Quality and Diversity in Feed-forward Stylization and Texture Synthesis," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6924–6932.
[46] X. Huang and S. Belongie, "Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1501–1510.
[47] A. Elgammal, B. Liu, M. Elhoseiny, and M. Mazzone, "CAN: Creative Adversarial Networks, Generating"art" by Learning about Styles and Deviating from Style Norms," *arXiv preprint arXiv:1706.07068*, 2017.
[48] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1125–1134.
[49] W. Xian, P. Sangkloy, V. Agrawal, A. Raj, J. Lu, C. Fang, F. Yu, and J. Hays, "TextureGAN: Controlling Deep Image Synthesis with Texture Patches," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8456–8465.
[50] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image Style Transfer using Convolutional Neural Networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2414–2423.
[51] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. S. Lempitsky, "Texture Networks: Feed-forward Synthesis of Textures and Stylized Images," *International Conference on Machine Learning, JMLR*, vol. 1, no. 2, p. 4, 2016.
[52] V. Dumoulin, J. Shlens, and M. Kudlur, "A Learned Representation for Artistic Style," *arXiv preprint arXiv:1610.07629*, 2016.
[53] Y. Li, N. Wang, J. Liu, and X. Hou, "Demystifying neural style transfer," *arXiv preprint arXiv:1701.01036*, 2017.
[54] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv preprint arXiv:1704.04861*, 2017.
[55] V. Mura, G. Orrù, R. Casula, A. Sibiriu, G. Loi, P. Tuveri, L. Ghiani, and G. L. Marcialis, "LivDet 2017 Fingerprint Liveness Detection Competition 2017," in *IEEE International Conference on Biometrics (ICB)*, 2018, pp. 297–302.
[56] K. Cao, D.-L. Nguyen, C. Tymoszek, and A. K. Jain, "End-to-end latent fingerprint search," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 880–894, 2019.
[57] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *Proc. International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.
[58] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

Tarang Chugh received the B. Tech. (Hons.) degree in Computer Science and Engineering from the Indraprastha Institute of Information Technology, Delhi (IIIT-D) in 2013. He was affiliated with IBM Research Lab, New Delhi, India as a research engineer during 2013-2015. He is currently a doctoral student in the Department of Computer Science and Engineering at Michigan State University. His research interests include biometrics, pattern recognition, and machine learning.

Anil K. Jain is a University distinguished professor in the Department of Computer Science and Engineering at Michigan State University. His research interests include pattern recognition and biometric authentication. He served as the editor-in-chief of the IEEE Transactions on Pattern Analysis and Machine Intelligence and was a member of the United States Defense Science Board. He has received Fulbright, Guggenheim, Alexander von Humboldt, and IAPR King Sun Fu awards. He is a member of the National Academy of Engineering and foreign fellow of the Indian National Academy of Engineering and Chinese Academy of Sciences.