

# Universal Material Translator: Towards Spoof Fingerprint Generalization

Rohit Gajawada<sup>\*†</sup>, Additya Popli<sup>\*†</sup>, Tarang Chugh<sup>‡</sup>, Anoop Namboodiri<sup>†</sup> and Anil K. Jain<sup>‡</sup>

<sup>†</sup>IIT Hyderabad, India

<sup>‡</sup>Michigan State University, USA

{rohit.gajawada@students.iiit.ac.in, additya.popli@research.iiit.ac.in,  
chughtar@cse.msu.edu, anoop@iiit.ac.in, jain@cse.msu.edu}

## Abstract

*Spoof detectors are classifiers that are trained to distinguish spoof fingerprints from bonafide ones. However, state of the art spoof detectors do not generalize well on unseen spoof materials. This study proposes a style transfer based augmentation wrapper that can be used on any existing spoof detector and can dynamically improve the robustness of the spoof detection system on spoof materials for which we have very low data. Our method is an approach for synthesizing new spoof images from a few spoof examples that transfers the style or material properties of the spoof examples to the content of bonafide fingerprints to generate a larger number of examples to train the classifier on. We demonstrate the effectiveness of our approach on materials in the publicly available LivDet 2015 dataset and show that the proposed approach leads to robustness to fingerprint spoofs of the target material.*

## 1. Introduction

Fingerprints have been widely used for identification of individuals for more than a century. Their uniqueness and persistence makes it one of the most reliable forms of biometric traits that have been used to augment security in a variety of applications. However, with the widespread use of fingerprint recognition systems, they are now a prime target of attackers. Fingerprint spoofs (i.e. fake/gummy fingers) can be fabricated using commonly available materials such as PlayDoh, Gelatin etc. with high fidelity, making them hard to distinguish from bonafide fingerprints. These materials can easily fool the existing spoof detectors especially if they were not included in the training. With a plethora of available materials that could be used to fabricate spoofs, it is important to be able to learn the concept of what a specific spoof looks like from just a few samples of it.

<sup>\*</sup>These authors have contributed equally.

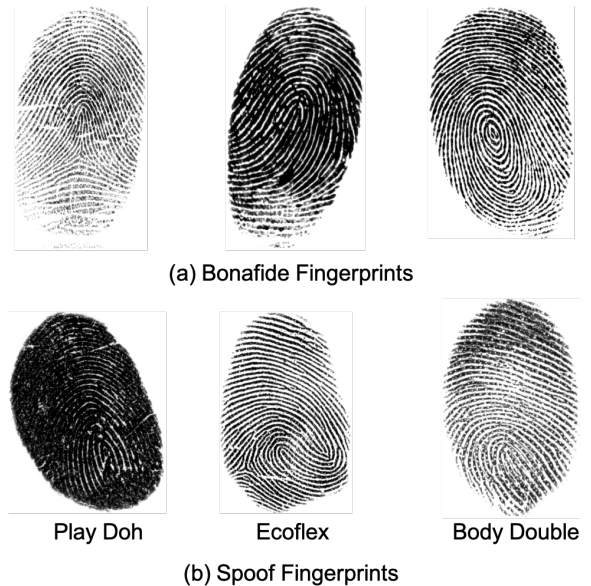


Figure 1. (a) Bonafide fingerprint samples and (b) Acquired spoof fingerprint samples fabricated using different materials.

One of the major limitations of many published anti-spoofing methods is their poor generalization performance across novel spoof materials, that were not seen during training. To generalize an algorithm's effectiveness across spoof fabrication materials, called *cross-material* performance, some studies have approached spoof detection as an *open-set problem*<sup>1</sup>. [31] applied the Weibull-calibrated SVM (W-SVM) to detect spoofs made of new materials. [5] trained an ensemble of multiple one-class SVMs using textural features extracted from only bonafide fingerprint images.

Although, CNN-based approaches outperform the earlier mentioned spoof detection approaches utilizing hand-crafted features on publicly available datasets [14], these require large amount of training data to avoid over-fitting.

<sup>1</sup>Open-set problems address the possibility of 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.

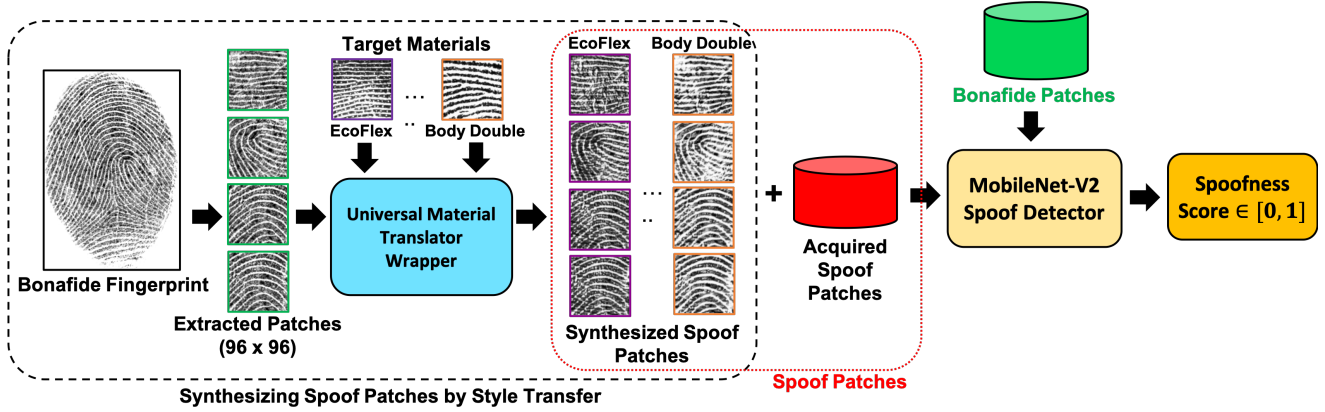


Figure 2. An overview of the proposed approach utilizing Universal Material Transfer (UMT) wrapper for synthesizing spoof patches.

With the advent of new fabrication techniques and novel materials, it is increasingly challenging to include all spoof fabrication materials and generate large datasets for each new material for training. In this study, we show that as few as 5 sample spoof fingerprint images can go a long way to improve the robustness of spoof detectors towards a novel fabrication material. It enables a framework, called Universal Material Translator (UMT) that can be used for cross-material spoof fingerprint style transfer. Moreover, 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.

## 2. Related Work

Fingerprint spoof detection has been approached from both hardware-based and software-based solutions. Hardware solutions usually rely upon detecting physical liveness characteristics of a human finger, such as blood flow [20], human odor [1] and multi-perspective images [9]. Software approaches do not require specialized sensors as they work with features extracted from captured fingerprint images to perform spoof detection.

Conventional software-based solutions [25, 24, 13, 12] extract hand-crafted features based on texture, physiology or anatomy of the input fingerprints to detect spoofs. Recently, learning-based approaches (CNNs in particular) have been utilized resulting in a higher degree of accuracy than previous works. Nogueira et al. [28] uses transfer learning from object-recognition CNNs pre-trained on ImageNet [33], and fine-tuned with fingerprints for spoof detection. Pala et al. [30] uses randomly selected patches to train a CNN architecture with triplet loss. Chugh et al. [3] utilizes local patches extracted and aligned using fingerprint minutiae to train a two-class CNN to detect spoof fingerprints. It is based on the premise that noise related to spoof fabrication introduces artifacts, that is pronounced around anomalous points in a fingerprint image i.e., *minutiae points*. It has

performed better with novel spoof materials than previous approaches noted above.

Realistic image synthesis is a difficult problem. Early non-parametric methods [32, 2, 7] find difficulty in generating images with textures that are unseen during training. Machine learning has been very effective in this regard both in terms of realism and generality. [11] and [10] perform artistic 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 pretrained convolutional network at the same time. Their method produces high quality results, but is computationally expensive since each step of the optimization problem requires a forward and backward pass through the pretrained network. [18, 38, 36, 21] train a feed-forward network to quickly approximate solutions to this optimization problem. [37, 6, 15] use methods based on feature statistics to perform style transfer. [8] applied GANs to generate artistic images. [16] used conditional adversarial networks to learn the loss for image to image translation. [39] learnt to synthesize objects consistent with texture suggestions. Our approach utilizes the approach in [15] for designing the proposed universal material translator that is capable of producing realistic fingerprint patches containing friction ridge information (content) of bonafide patches and style information from target material spoof patches.

## 3. Universal Material Translator

Studies in [28, 23, 31] have shown that when a spoof detector is evaluated on spoofs fabricated using materials that were not seen during training, there can be up to a three-fold increase in the spoof detection error rates. [4] identified a subset of spoof fabrication materials based on their optical and mechanical properties that are crucial to train a robust spoof detector. However, with the advent of new fabrication techniques and novel materials, it is prohibitively expensive and time-consuming to generate large-scale datasets

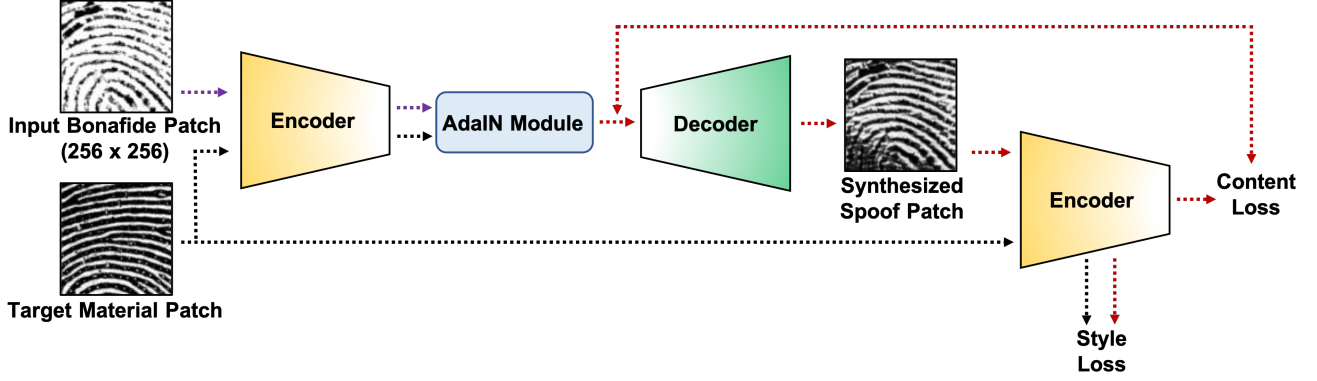


Figure 3. Training pipeline of the Universal Material Translator (UMT) wrapper. An AdaIN module is used for performing the style transfer in the encoded feature space. The same VGG-19 encoder is used for computing content loss and style loss.

for each new material for training CNN-based approaches. Therefore, it is crucial to generate high quality synthetic fingerprint spoof images to address the lack of spoof training data.

In order to move towards this goal, we desire to generate synthetic spoof images that are similar in material style to the spoofs for which we only have very limited number of samples. It is possible that we might have only a few spoof images of a particular material due to expensive fabrication process or use of rare materials, and we cannot simply use these few images to train a deep CNN-based spoof classifier.

Towards this goal, we propose **Universal Material Translator (UMT)** wrapper, a setup that extracts the material style characteristics from a few acquired spoof fingerprint images<sup>2</sup> and transfers it to the available large-scale bonafide image database to generate synthetic yet realistic spoof images in the desired target material style while retaining the friction ridge information of the source fingerprint. In our experiments, we utilize local patches of size  $96 \times 96$  for style transfer instead of whole images. Figure 2 presents an overview of the proposed approach utilizing Universal Material Translator wrapper for synthesizing spoof patches in target material style. From here onwards, we refer spoofs acquired using fingerprint readers as spoofs and spoofs fabricated using the proposed approach as synthesized spoofs.

### 3.1. Material Style Transfer

[37] used an InstanceNorm layer to normalize feature statistics across spatial dimensions given a style image, per channel per sample. The equation for InstanceNorm is given below:

$$IN(x) = \alpha \left( \frac{x - \mu(x)}{\sigma(x)} \right) + \beta \quad (1)$$

<sup>2</sup>Acquired spoof fingerprint images refer to the fingerprint impressions of physical spoof specimens that are captured using a fingerprint reader, as opposed to *synthesized* spoof fingerprint images which are generated images using the proposed approach.

In Equation 1,  $x$  is the input feature space,  $\mu(x)$  and  $\sigma(x)$  are the mean and variance parameters of that feature space, respectively. It was observed that changing the affine parameters  $\alpha$  and  $\beta$  (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 [6], which learns alpha and beta values for each feature space and style pair. However, this required retraining of the network for each new style. [15] replaced the InstanceNorm layer with an 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 style image to the content image, through the affine parameters. We follow the same approach as described in [15] in our paper, by transferring feature statistics from the target material spoof to the input content bonafide in the feature space. As described in AdaIN, we apply instance normalization on the input content bonafide feature space however not with learnable affine parameters. The channel-wise mean and variance of the content bonafide’s feature space is aligned to match those of the target material spoof’s feature space. This is done by computing the affine parameters from the target material spoof feature space in the following manner:

$$AdaIN(x, y) = \sigma(y) \left( \frac{x - \mu(x)}{\sigma(x)} \right) + \mu(y) \quad (2)$$

In Equation 2, the content bonafide patch’s feature space is  $x$  and the target material spoof patch’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  $Enc$  and a decoder  $Dec$ . For the encoder, we use the first few layers of a pre-trained VGG network similar to [17]. The weights of this network are frozen throughout all stages of the setup. Given the content bonafide patch as  $c$  and the target material spoof patch as  $m$ , then  $x$  is  $Enc(c)$  and  $y$  is  $Enc(m)$ . The desired feature space is obtained as:

$$t = AdaIN(Enc(c), Enc(m)) \quad (3)$$

We use  $Dec$  to take  $t$  as input to produce  $Dec(t)$  which is the final synthesized spoof stylized in the desired material.

In order to ensure that our synthesized spoof patches i.e  $Dec(t)$  are matching the style statistics of the target material spoof, we apply a loss  $L_s$  similar to [17] given as:

$$\mathcal{L}_s = \sum_{i=1}^L \|\mu(\phi_i(g(t))) - \mu(\phi_i(s))\|_2 + \sum_{i=1}^L \|\sigma(\phi_i(g(t))) - \sigma(\phi_i(s))\|_2 \quad (4)$$

where each  $\phi_i$  denotes a layer in the VGG-19 network we use as encoder. We pass  $Dec(t)$  and  $m$  through  $Enc$  and extract the outputs of  $relu1\_1$ ,  $relu2\_1$ ,  $relu3\_1$  and  $relu4\_1$  layers for computing  $L_s$ . To ensure that the synthesized spoof patches retain fingerprint (friction ridge) content from bonafide patches, we use a content loss  $L_c$  which is computed as the euclidean distance between the features of the synthesized spoof patches i.e.  $Enc(Dec(t))$  and the features of the bonafide patches with the target style ( $t$ ). The effective style loss is given as:

$$L = \lambda_c L_c + \lambda_s L_s \quad (5)$$

[22] goes into the theory behind why neural style transfer works so well, by interpreting it as a domain adaptation problem. Although we do not follow the gram-matching approach, our style transfer still involves second-order feature statistics transfer, similar to what is achieved by gram matrix matching.

#### 4. UMT-Wrapper for Spoof Robustness

Once we obtain a spoof synthesizer  $G$ , we use the existing dataset of bonafide patches and generate a dataset of synthesized spoof patches conditioned on the material  $M$ . In our experiments, the bonafide fingerprints used for generating new spoofs in the target material style are the materials that we use in our training set in that particular experiment. We combine this synthetic data with the existing training data of LivDet15 [26] for training our classifier. Note when evaluating the performance on an unseen material, spoofs of that particular material are removed from the training dataset of the LivDet15 [26] split. Our approach acts like a wrapper on top of any existing spoof detector. If our approach is applied to a spoof detector, it can successfully make it more robust to spoofs of particular target materials just seen during training.

For our classification experiments, we use the Mobilenet V2 [34] pretrained on ImageNet [33] as it is faster than other architectures like Inception V3 [35] used in published studies. We follow a training procedure similar to [3].

### 5. Experiments and Results

#### 5.1. Datasets

We use the Crossmatch sensor of the LivDet15 [26] dataset in our experiments. For each image in the training

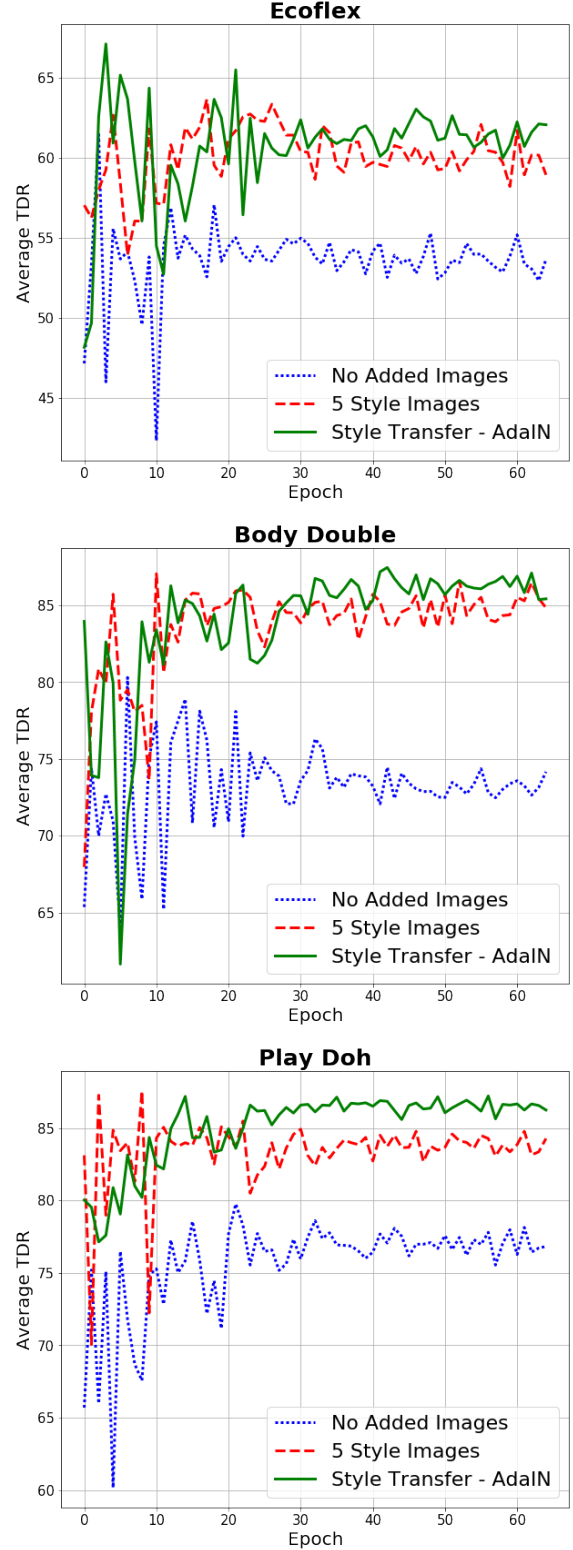


Figure 4. TDR% on unknown materials at FDR = 0.1%

and testing dataset, we perform Otsu's Global Thresholding [29] followed by a set of morphological operations to obtain a segmentation mask for the fingerprint similar to how



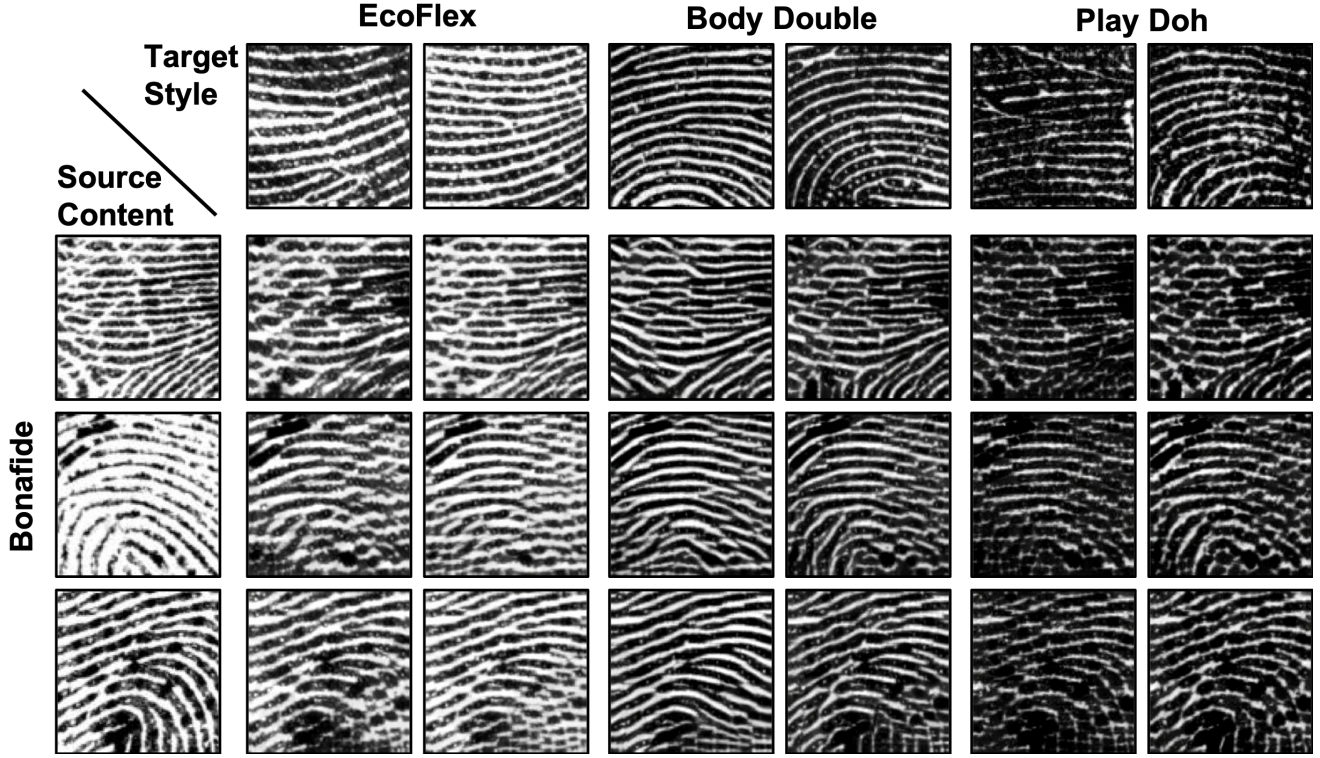


Figure 5. Spoof patches synthesized from bonafide patches (first column) by our approach conditioned on a target material (first row).

it is done in [27]. To further remove any background noise, we find the largest connected component in the foreground part of this resultant image. We generate 30 patches of size  $150 \times 150$  which lie within the foreground of this resultant image. For all the patches, we compute the orientation similar to how it is done in [27] with a window of 64 and stride of 32. We perform alignment on each patch by taking the trimmed mean of the orientations, rotating the patch by this mean orientation and doing a center crop of  $96 \times 96$  from the resultant image.

## 5.2. Training procedure

For the Crossmatch sensor in LivDet15 [26], spoof training dataset is fabricated using three spoof materials, namely Ecoflex, Play Doh, and Body Double. To simulate testing our method against a novel target spoof material, we train an AdaIN spoof generator on fingerprint spoofs fabricated using the other two (known) materials to learn an effective style extractor and translator amongst spoofs in those two materials. After this, we take  $k$  fingerprints of the target material (material not used in training) which we want to use for generating a synthetic spoof dataset. 30 patches of size  $96 \times 96$  are extracted and aligned from each fingerprint image and these patches become the target material patch set for our UMT-Wrapper.

We take a large number of bonafide fingerprint patches ( $\sim 15000$ ) such that the number of patches is roughly equal

to the number of patches of any one of the materials in the LivDet trainset of the CrossMatch sensor. These content fingerprint patches are fed into our UMT-Wrapper for generating synthetic spoof fingerprint patches that retain the content of the bonafide patches with the style of the target material patch set. We add these synthetic fingerprint patches along with the target material patch set to our existing training data of that classifier and test on a combined set of the LivDet training and the testing splits of that target material.

For each type of experiment, we run 5 different runs where each run has a different set of style patches and different bonafide images which are used for generating synthetic spoofs. We report the means at each training epoch in Figure 4. We report the mean TDR and its standard deviation of the last 15 epochs (from 50 to 65) in Table 1 and Table 2.

## 5.3. Implementation Details

For our experiments, we use  $k = 5$ . For the encoder of synthetic spoof generator, we use weights pretrained on ImageNet [33]. The encoder is the first few layers of a VGG-19 network. The weights of this encoder are frozen during training of the synthetic spoof generator. For the training of our synthetic spoof generator, we use  $\lambda_c = 1.0$  and  $\lambda_s = 10.0$ . We use the Adam optimizer [19] and a batch size of 1. The learning rate of the generator and the discriminator are both  $1e - 5$ . For the orientation calcula-

Table 1. Cross-material performance (TDR (%) @ FDR = 0.1%) using EcoFlex, Body Double, and PlayDoh spoof materials, without and with using synthesized spoof patches in training.

Training Set	Testing Set	TDR (%) @ FDR = 0.1%
Bonafide vs. [Play Doh + Body Double]	Bonafide vs. EcoFlex	53.56 ± 3.05
Bonafide vs. [Play Doh + Body Double + 150 Spoof EcoFlex Patches*]	Bonafide vs. EcoFlex	59.99 ± 6.23
Bonafide vs. [Play Doh + Body Double + 150 Spoof EcoFlex Patches + 15K Synthesized EcoFlex Patches]	Bonafide vs. EcoFlex	<b>61.40 ± 3.73</b>
Bonafide vs. [Play Doh + EcoFlex]	Bonafide vs. Body Double	73.20 ± 8.59
Bonafide vs. [Play Doh + EcoFlex + 150 Spoof Body Double Patches*]	Bonafide vs. Body Double	85.01 ± 1.96
Bonafide vs. [Play Doh + EcoFlex + 150 Spoof Body Double Patches + 15K Synthesized Body Double Patches]	Bonafide vs. Body Double	<b>86.24 ± 2.28</b>
Bonafide vs. [EcoFlex + Body Double]	Bonafide vs. Play Doh	76.99 ± 4.08
Bonafide vs. [EcoFlex + Body Double + 150 Spoof Play Doh Patches*]	Bonafide vs. Play Doh	83.90 ± 4.64
Bonafide vs. [EcoFlex + Body Double + 150 Spoof Play Doh Patches + 15K Synthesized Play Doh Patches]	Bonafide vs. Play Doh	<b>86.49 ± 1.56</b>

\* 150 Spoof EcoFlex / Body Double / Play Doh Patches are generated from only 5 Spoof EcoFlex / Body Double / Play Doh images, respectively.

Table 2. Known-material performance (TDR (%) @ FDR = 0.1%) using EcoFlex, Body Double, and PlayDoh spoof materials, without and with using synthesized spoof patches in training.

Training Set	Testing Set	TDR (%) @ FDR = 0.1%
Bonafide vs. [Play Doh + Body Double]	Bonafide vs. [Play Doh + Body Double]	86.11 ± 1.95
Bonafide vs. [Play Doh + Body Double + 150 Spoof EcoFlex Patches*]	Bonafide vs. [Play Doh + Body Double]	84.68 ± 2.66
Bonafide vs. [Play Doh + Body Double + 150 Spoof EcoFlex Patches + 15K Synthesized EcoFlex Patches]	Bonafide vs. [Play Doh + Body Double]	<b>87.08 ± 1.89</b>
Bonafide vs. [Play Doh + EcoFlex]	Bonafide vs. [Play Doh + EcoFlex]	93.61 ± 1.31
Bonafide vs. [Play Doh + EcoFlex + 150 Spoof Body Double Patches*]	Bonafide vs. [Play Doh + EcoFlex]	93.59 ± 0.77
Bonafide vs. [Play Doh + EcoFlex + 150 Spoof Body Double Patches + 15K Synthesized Body Double Patches]	Bonafide vs. [Play Doh + EcoFlex]	<b>94.17 ± 1.03</b>
Bonafide vs. [EcoFlex + Body Double]	Bonafide vs. [EcoFlex + Body Double]	98.6 ± 0.26
Bonafide vs. [EcoFlex + Body Double + 150 Spoof Play Doh Patches*]	Bonafide vs. [EcoFlex + Body Double]	<b>98.88 ± 0.57</b>
Bonafide vs. [EcoFlex + Body Double + 150 Spoof Play Doh Patches + 15K Synthesized Play Doh Patches]	Bonafide vs. [EcoFlex + Body Double]	98.55 ± 0.42

\* 150 Spoof EcoFlex / Body Double / Play Doh Patches are generated from only 5 Spoof EcoFlex / Body Double / Play Doh images, respectively.

tion for alignment of fingerprint images, we use a window of size 64 and stride of size 32. All our experiments are performed in the PyTorch framework.

For the MobileNet V2 classifier, we also used pretrained ImageNet [33] weights and cut off the last fully connected layer and replace with a fully connected layer for two class outputs. We also train this with an Adam optimizer with a learning rate of  $1e - 4$  and a batch size of 64.

## 5.4. Results

In order to demonstrate the contribution of our proposed UMT-Wrapper for making a particular spoof detector more robust towards an unknown material  $M \in \{Eco\ Flex, Body\ Double, PlayDoh\}$ , we conduct three sets of cross-material experiments for each material  $M$ , by adopting the leave-one-out method.

In the first set of experiment, we test on the unknown

material  $M$  which is the material completely left-out from training and we desire to make our spoof detector more robust to. Note that for testing on material  $M$ , we combine the train and the test set of that material and use this as a test set in order to increase the size of the test set with respect to the live test set. In the second set, we include 150 spoof patches extracted from only 5 images from training set of unknown material  $M$  in the training. And in the third set, we also include 15,000 synthesized patches generated using the proposed UMT-wrapper in the training. Table 1 presents the cross-material results for each of the three materials. It can be observed that including the 15,000 synthesized spoof patches in training results in an improvement in TDR @ FDR = 0.1%.

It is important to ensure that adding the synthesized spoof patches, does not degrade the performance of spoof detector against known materials. We conduct a set of three

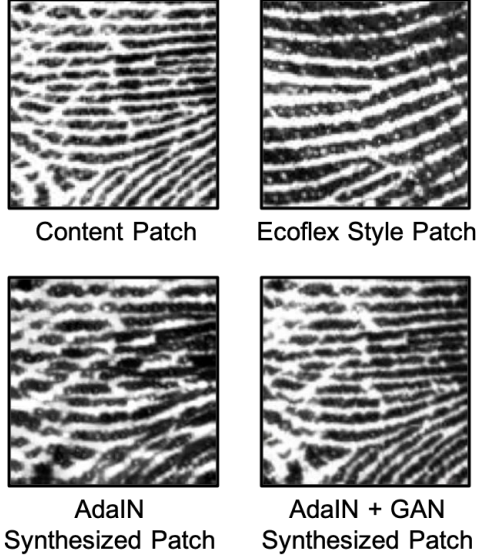


Figure 6. Effect of GAN loss on synthesized spoof patches

known-material experiments similar to the cross-material experiments. Table 2 presents the known-material results for each of the three subsets of 2 materials. It can be observed that the known-material performance is sustained by adding synthesized spoof patches in training. This indicates the positive contributions of utilizing UMT-wrapper for improving the generalization of a spoof detector while ensuring robustness to known materials.

## 6. Addition of Adversarial Loss

In our work, we also experimented with the addition of an adversarial loss in our UMT-Wrapper training. This led to better visual quality of the generated spoof patches. A typical GAN setup consists of a generator  $G$  and a discriminator  $D$ , where  $D$  learns to distinguish between fake and real images and  $G$  tries to fool  $D$ . We use a similar GAN setup to [16] in our approach where bonafide patches are used to generate synthesized spoof patches similar to existing spoof patches via a discriminator. We believe that UMT-Wrapper + GAN supervision generates spoof fingerprint patches which are superior than normal UMT-Wrapper in terms of Level 1 and Level 2 features (see Figure 6). However, there was a reduction in the TDRs obtained when adding an adversarial loss when compared to the experiments where adversarial loss wasn't added. We reason that GANs introduce certain artefacts and noise in the generated images which can be easily picked up by detectors and since we pose spoof detection as a binary problem, it turns out to be a problem to the current setup.

## 7. Conclusions and Future Work

This study proposes a framework for Universal Material Translator to generate large synthetic spoof datasets of new

materials using only bonafide images and a few samples of target material type. It is shown that the proposed approach improves the generalization performance of spoof detector on unknown materials while maintaining high performance on known materials. We believe our work can provide a direction to make spoof detection systems generalize the understanding of what a spoof is. In our future work, we would like to connect our approach to adversarial machine learning and lifelong learning methods.

## 8. Acknowledgement

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] D. Baldissera, A. Franco, D. Maio, and D. Maltoni, "Fake fingerprint detection by odor analysis," in *Proc. ICB*. Springer, 2006.
- [2] T. Chen, M. ming Cheng, P. Tan, A. Shamir, and S. min Hu, "Sketch2photo: internet image montage", *ACM SIGGRAPH Asia*, 2009.
- [3] T. Chugh, K. Cao, and A. K. Jain, "Fingerprint Spoof Buster: Use of Minutiae-centered Patches", in the *IEEE TIFS*, 2018.
- [4] T. Chugh and A. K. Jain. "Fingerprint Presentation Attack Detection: Generalization and Efficiency," arXiv preprint arXiv:1812.11574 (2018).
- [5] Y. Ding and A. Ross, "An ensemble of one-class SVMs for fingerprint spoof detection across different fabrication materials," in *Proc. IEEE WIFS*, 2016.
- [6] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," in *ICLR*, 2017.
- [7] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *ICCV*, 1999.
- [8] 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.
- [9] J. Engelsma, K. Cao, and A.K. Jain, "RaspiReader: Open Source Fingerprint Reader," in *IEEE TPAMI*, 2018.
- [10] L.A. Gatys, A.S. Ecker, M. Bethge, "Texture synthesis using convolutional neural networks," in *NIPS*, 2015.
- [11] L.A. Gatys, A.S. Ecker, M. Bethge, "A neural algorithm of artistic style," arXiv preprint arXiv:1508.06576, 2015.

- [12] L. Ghiani, A. Hadid, G. L. Marcialis, and F. Roli, "Fingerprint liveness detection using Binarized Statistical Image Features," in *BTAS*, 2013.
- [13] L. Ghiani, G. L. Marcialis, and F. Roli, "Fingerprint liveness detection by local phase quantization," in *ICPR*, 2012.
- [14] L. Ghiani, D. A. Yambay, V. Mura, G. L. Marcialis, F. Roli, and S. A. Schuckers, "Review of the Fingerprint Liveness Detection (LivDet) competition series: 2009 to 2015," *Image and Vision Computing*, 2017.
- [15] X. Huang and S. Belongie, "Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization," in *ICCV*, 2017.
- [16] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017.
- [17] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *ECCV*, 2016.
- [18] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *ECCV*, 2016.
- [19] D.P. Kingma, J. Ba, "Adam: A Method for Stochastic Optimization," in *ICLR*, 2015.
- [20] P. D. Lapsley, J. A. Lee, D. F. Pare Jr, and N. Hoffman, "Anti-fraud biometric scanner that accurately detects blood flow," US Patent 5,737,439, 1998.
- [21] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in *ECCV*, 2016.
- [22] Y. Li, N. Wang, J. Liu, X. Hou, "Demystifying Neural Style Transfer," in *IJCAI*, 2017.
- [23] E. Marasco and C. Sansone, "On the Robustness of Fingerprint Liveness Detection Algorithms against New Materials used for Spoofing," in *Proc. Intl. Conf. Bio-Insp. Syst. Sign. Process.*, 2011
- [24] E. Marasco and C. Sansone, "Combining perspiration-and morphology-based static features for fingerprint liveness detection," *Pattern. Reco. Letters*, 2012
- [25] G. L. Marcialis, F. Roli, and A. Tidu, "Analysis of fingerprint pores for vitality detection," in *Proc. 20th ICPR*, 2010.
- [26] V. Mura, L. Ghiani, G. L. Marcialis, F. Roli, D. A. Yambay, and S. A. Schuckers, "LivDet 2015 - Fingerprint liveness detection competition 2015," in *Proc. IEEE 7th Intl. Conf. BTAS*, 2015
- [27] D. Nguyen, K. Cao and A.K. Jain, "Robust Minutiae Extractor: Integrating Deep Networks and Fingerprint Domain Knowledge," in *ICB*, 2018.
- [28] R. F. Nogueira, R. de Alencar Lotufo, and R. C. Machado, "Fingerprint Liveness Detection Using Convolutional Neural Networks," *IEEE TIFS*, 2016.
- [29] N. Otsu, "A threshold selection method from gray-level histogram," *IEEE Transactions on System Man Cybernetics*, 1979.
- [30] F. Pala and B. Bhanu, "Deep Triplet Embedding Representations for Liveness Detection," in *Deep Learning for Biometrics. Advances in Computer Vision and Pattern Recognition*.
- [31] A. Rattani, W. J. Scheirer, and A. Ross, "Open set fingerprint spoof detection across novel fabrication materials," *IEEE TIFS*, 2015.
- [32] A. Roy, N. Memon, and A. Ross, "Masterprint: Exploring the vulnerability of partial fingerprint-based authentication systems," *IEEE TIFS*, 2017.
- [33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," in *IJCV*, 2015.
- [34] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *CVPR*, 2018.
- [35] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *CVPR*, 2016.
- [36] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky, "Texture networks: Feed-forward synthesis of textures and stylized images," in *ICML*, 2016.
- [37] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis," in *CVPR*, 2017.
- [38] X. Wang, G. Oxholm, D. Zhang, and Y.-F. Wang, "Multimodal transfer: A hierarchical deep convolutional neural network for fast artistic style transfer," arXiv preprint arXiv:1612.01895, 2016.
- [39] 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 *CVPR*, 2017.