Fingerprint Presentation Attack Generation using Generative Adversarial Networks

Leonardo Capozzi and Tiago Gonçalves
Faculdade de Engenharia
Universidade do Porto
Porto, Portugal
{up201503708, up201607753}@fe.up.pt

Abstract

Biometric recognition has been widely used in diverse government and civilian applications. Most of the available systems rely on fingerprint recognition and have shown to be reliable in terms of accuracy, speed and purported security. However, they also present several vulnerabilities against spoof attacks. To overcome this flaw, several automated spoofing detection models have been developed, but they end up assuming that spoof detection is a binary closed-set problem, which is not realistic. Recent works have proposed the use of adversarial methodologies to improve the model's generalisation capacity to unseen spoof attacks. Following this research line, we performed a study on the application of generative adversarial neural networks (GANs) for the generation of synthetic data to be employed during the model's training. We hypothesise that by using GANs, one could learn a distribution that could contain all possible spoofing attacks, thus opening the possibility to learn classifiers that could be more robust. In this work, we optimised a GAN and conditional GAN (cGAN) to generate synthetic images of real and fake fingerprints and used this data in the training of classifiers for the detection of single spoofing attacks. The results obtained suggest that this strategy does not jeopardise the predictive performance of the model's which may unveil an interesting opportunity for the detection of multi-spoofing attacks and, consequently, unseen attacks.

1. Introduction

Biometric recognition has been widely used in diverse government and civilian applications such as e-passports, ID cards, border control, and in most unlock/authentication systems present in handheld/mobile devices. Although there are several biometric traits [16], most of the available recognition systems rely on fingerprint recognition. Fingerprints are small lines/ridges and valleys in the skin of fingertips that are formed during the development of the fetus due to genetic and environmental factors. This results in a high variation of configurations such that it is considered impossible to have two fingerprints looking exactly alike [9]. These systems have shown to be reliable in terms of accuracy, speed, and purported security. However, several studies revealed that such systems are highly vulnerable to spoof attacks (*e.g.*, a 2D printing with conductive ink to replicate the fingerprint of a victim left behind on their keyboard) [11, 2]. Therefore, intending to overcome this flaw, several automated spoofing detection systems have been developed, to detect and flag spoof attacks before performing biometric authentication [2]. The main drawback about these systems is that they assume that spoof detection is a binary closed-set problem, *i.e.*, *live* or *single-material spoof* [2]. For this reason, most of the methods may be overestimated regarding their performance metrics, since they end up using only one type of spoof samples to train and test the models [16]. If we take into account the diversity of materials (with different mechanical and optical properties) that can be employed to produce different spoofs, we may conclude that this may be, in fact, an open-set problem, where we should assume *a priori* that we do not know all the existing spoofs [2]. This premise endorsed the creation of new research lines regarding the study of unseen spoofs, pioneered by [10]. Later, [19] proposed the idea of regularising the training of the models to enforce the knowledge *bona fide* (BF) presentations over the presentation attack (PA), to increase the robustness of the models against unseen PAs. The increase of the available computational power and the democratised access to large datasets facilitated the use of deep learning in several pattern recognition algorithms. As shown in [12] and [17], biometric applications were no exception, although these works revisited the binary approach. The LivDet competition series in 2015 [5] included evaluation with unseen attacks, but

this paradigm was discontinued in the following editions. The problem of presentation attack detection (PAD) generalisation is not new. Besides, [21] stated that every time we present a new PA species in the test phase, the performance of the classifier drops. Several methodologies based on one-class classification for the BF class or adversarial training [2] have been proposed to increase the robustness of the classifier. However, the current paradigm has not changed: most of the recent proposals either rely on binary classification approaches or use part of the data available at training time to design the models. These two possibilities still make very optimistic assumptions regarding the attacker [15]. Following the algorithm proposed in [3] for sign language recognition, and later applied in iris PAD [4], [15] addressed the problem of fingerprint iris recognition using this regularisation technique. Since the main goal is to improve the model's generalisation capacity to unseen PAs, this technique promotes the joint learning of both the representation and classifier from the data and explicitly imposes "PA species invariance" in the high-level representations for the PAD method [15]. Following the work proposed in [15], we intend to extend the adversarial methodology and use generative adversarial networks (GANs) to 1) increase the quality of the generated PA species; and 2) increase the robustness of the PAD method against unseen attacks. The intuition behind this approach is supported by the following hypothesis: if one could learn a distribution that contains all possible PAs, then it should be possible to learn a function that could correctly discriminate against BF and unseen PAs, without the need of explicitly using them during the training phase.

Besides the Introduction, this work is organised as follows: section 2 provides an overview of the fundamentals on GANs and related work on PAD methodologies, section 3 presents the datasets used in our experiments, section 4 explains the details of the implementation of the data processing pipeline and the deep learning algorithms, section 5 presents the results and contemplates our discussion, and finally, section 6 concludes this paper and provides our view for future work directions. Code and pre-trained models are hosted in a GitLab repository [1].

2. Related Work

2.1. Generative Adversarial Networks

Generative Adversarial Networks (GANs) were first proposed by [6]. These types of networks have become very popular for image generation, and some variations to the original methodology have been proposed. It consists of two networks, a generator, and a discriminator. The goal of a GAN is to train a generator that creates images that are indistinguishable from images in the train set. The generator is constantly trying to generate images that "fool" the discriminator into thinking they are real, and the discriminator is constantly trying to distinguish between real images and generated images. This competition between the two networks is what makes the training adversarial (each network is constantly trying to become better than the other). The input of the generator is a noise vector, and the output is a generated image. The input of the discriminator is an image, either generated or real, and its output is a classification of the input image as either real or generated. In other words, the generator creates images that have a realistic appearance and the discriminator tries to classify the images as real or generated. The loss function of a GAN can be written as:

$$\mathcal{L} = \mathbb{E}_x[\log D(x)] + \mathbb{E}_y[\log(1 - D(G(y)))] \quad (1)$$

where G is the generator, D is the discriminator, x is a real image and y is a noise vector. The generator is optimised to minimise this loss function, while the discriminator is optimised to maximise it. With the correct training, the generator will create increasingly realistic images, until they are indistinguishable from the real images. One of the problems of training these networks is mode collapse, which is when the generator fails to create a large variety of outputs and generates the same images despite the input that it is given [1].

2.2. Conditional Generative Adversarial Network

As seen in 2.1, generative models can be very useful for generating images that have the same distribution as the original data. Since the input of the network is a random vector we have limited control over the generated images, which would be very useful for some problems. Conditional Generative Adversarial Networks (cGANS) [13] aim

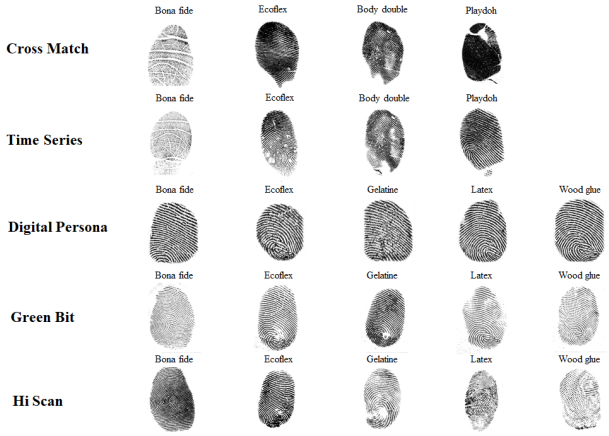

Figure 1. Example images from the LivDet2015 dataset, adapted from [16].

[1] https://gitlab.inesctec.pt/leonardo.g.capozzi/image-to-image-generative-adversarial-network

Table 1. LivDet2015 dataset description, adapted from [16].

Sensor	Attack Images	*Bona fide* Images	PAI Materials
Cross match	1473	1510	Ecoflex, Body double, Playdoh
Time Series	4495	4440	Ecoflex, Body double, Playdoh
Digital Persona	1000	1000	Ecoflex, Gelatine, Latex, Wood glue
Green Bit	1000	1000	Ecoflex, Gelatine, Latex, Wood glue
Hi Scan	1000	1000	Ecoflex, Gelatine, Latex, Wood glue

to solve this problem by giving the generator and the discriminator an additional input, a label that allows us to condition the generator to output an image belonging to a certain class or having a specific attribute present. The noise vector is still used to allow for some variation in the generated images belonging to the same class. The loss function of a cGAN can be written as:

$$\mathcal{L}_c = \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x,z)))] \quad (2)$$

where x is a class label, y is a "real" image from class x and z is a random noise vector. The generator (G) tries to minimise this loss, and the discriminator (D) tries to maximise it. As training progresses the generator will output increasingly realistic images until they are practically indistinguishable from the images belonging to the train set.

3. Data

The LivDet databases are large-scale datasets that were created for the Liveness Detection Competition Series, which takes place every two years since 2009. The main goal of the competition is the development of Presentation Attack Detection (PAD) methods and is open to all institutions. The large amounts of data allows for standardised tests that can be used to compare different methodologies [5, 16]. More specifically, the LivDet2015 dataset, which was developed for the 2015 edition of the competition is composed of five sub-datasets: Cross Match, Digital Persona, Green Bit, Hi Scan, and Time Series. Figure 1 shows sample images taken from the dataset. Table 1 shows more details about the datasets [5, 16]. We use the original train and test split from LivDet2015 dataset. Please note that the test split we used does not contain the Green Bit and Time Series sub-datasets.

4. Methodology

4.1. GAN

4.1.1 Model Architecture

The model we use in this work is based on the class of Deep Convolutional GANs (DCGANs) proposed by [18]. This class of models was a pioneer in providing guidelines to build stable models, thus contributing to the proper use of Convolutional Neural Networks (CNNs) into GANs to model images [18]. The guidelines read as follows: 1) replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator); 2) use batch-normalisation in both the generator and the discriminator; 3) remove fully connected hidden layers for deeper architectures; 4) use rectified linear unit (ReLU) activation in the generator for all layers except for the output, which uses hyperbolic tangent; 5) use leaky ReLU activation in the discriminator for all layers.

4.1.2 Data Pre-processing

Our data pre-processing pipeline is employed as follows: 1) images are converted into the grey-scale format; 2) images are then re-scaled into 110% of their original size through the addition of padding, with pixel values $= 255$; 3) a centre-crop operation is then applied to assure that images keep their final size as 64×64 and that the fingerprint is in the centre of the image as well; 4) the pixel values of the images are then re-scaled to be in the closed interval of $[-1, 1]$.

4.1.3 Training Pipeline

We start the training of the GAN with the generation of a latent-space vector nz of size 64 filled with random numbers sampled from a normal distribution with mean 0 and variance 1, *i.e.*, $nz \sim \mathcal{N}(0, 1)$. This vector is given as input for the generator $G(\cdot)$, which will generate a synthetic fingerprint image $G(nz)$. We then feed the discriminator $D(\cdot)$ with $G(nz)$ to obtain its prediction $D(G(nz))$. The weights of both $G(\cdot)$ and $D(\cdot)$ are optimised using the Adam optimiser [8], with learning rate 2×10^{-4} and the momentum term β in range $[-0.5, 0.999]$. The loss function is given by the binary cross-entropy. We train the GAN for 3000 epochs, with batch-size $= 128$. Besides, both the $G(\cdot)$ and $D(\cdot)$ weights w are randomly initialised from a normal distribution with mean 0 and standard deviation 0.02, *i.e.*, $w \sim \mathcal{N}(0, (0.02)^2)$.

Table 2. Accuracy results for the FPAD classifier trained without data augmentation.

Material / Dataset	Cross Match	Digital Persona	Green Bit	Hi Scan	Time Series
Body Double	-	-	-	-	-
Ecoflex 00 50	-	-	-	-	-
Ecoflex	0.9130	-	-	-	-
Playdoh	0.9568	-	-	-	-
Latex	-	0.9648	-	0.9752	-
Gelatine	-	0.9024	-	0.9144	-
WoodGlue	-	0.9384	-	0.8600	-

4.2. cGAN

4.2.1 Model Architecture

The architecture used for the generator is an auto-encoder based on the architecture of a U-Net [20]. The network was modified to remove the skip connections between the layers. This is not an image-to-image translation problem, therefore we only wanted to compress the inputs (keeping only the relevant information) and then upscale them to get the output.

4.2.2 Data Pre-processing

We follow the same data pre-processing as the GAN model (see 4.1.2).

4.2.3 Training Pipeline

The training of the cGAN starts with the generation of a noise vector nz with a dimension of $64 \times 64 \times 1$, the same dimensions as the input image x (*bona fide* fingerprint). The noise vector is filled with random numbers sampled from a normal distribution with mean 0 and variance 1, *i.e.*, $nz \sim \mathcal{N}(0, 1)$. Then it is concatenated to the input image in the channels dimension and given to the generator $G(\cdot)$, which will generate a synthetic fingerprint image $G(x, nz)$. We feed the discriminator $D(\cdot)$ with x and $G(x, nz)$ to obtain its prediction $D(x, G(x, nz))$. The discriminator is trained to classify the image y in $D(x, y)$ as being real or generated. The weights of both $G(\cdot)$ and $D(\cdot)$ are optimised using the Adam optimiser [8], with learning rate 2×10^{-4} and the momentum term β in range $[-0.5, 0.999]$. The loss function is given by the binary cross-entropy. We train the cGAN for 3000 epochs, with batch-size $= 128$. Besides, both the $G(\cdot)$ and $D(\cdot)$ weights w are randomly initialized from a normal distribution with mean 0 and standard deviation 0.02, *i.e.*, $w \sim \mathcal{N}(0, (0.02)^2)$.

4.3. Fingerprint Presentation Attack Detection

4.3.1 Model Architecture

For the fingerprint presentation attack detection (FPAD) task, we used as backbone (*i.e.*, feature extractor) the state-of-the-art architecture DenseNet121 [7]. This architecture connects all its layers in a feed-forwarding fashion, which, according to its authors, has several advantages: 1) alleviating the vanishing-gradient problem; 2) strengthening feature propagation; 3) encouraging feature reuse; 4) reducing the number of parameters. We added a single neuron (linear layer) with the sigmoid activation at the end of the architecture as the classifier.

4.3.2 Data Pre-processing

In this task, we follow the same data pre-processing as the GAN model (see 4.1.2). At the end of the data pre-processing pipeline, we create 3 copies of the resulting image and concatenate them by the channel dimensions, thus achieving an image with the following dimensions $64 \times 64 \times 3$ (the DenseNet121 backbone requires that inputs have 3 channels).

4.3.3 Training Pipeline

Images are given as inputs to the model in batches with size 128 and the model is trained for 100 epochs with the Adam optimiser and learning-rate 1×10^{-4}. The loss function is given by the binary cross-entropy. We save the best model weights based on loss value in training. Regarding data augmentation, we employ three different strategies: 1) Baseline training without data augmentation, in which no data augmentation strategy is employed; 2) Training with synthetic images generated with the GAN model, in which we add new images (*bona fide* and PA) generated with the different GAN models to the train set; 3) Training with synthetic images generated with the cGAN model, in which we add new images (PA only) generated with the cGAN models to the train set.

5. Results and Discussion

5.1. GAN Results

Figure 2 shows example images generated by our GAN model in different training epochs. It is possible to observe that in the first 1000 epochs, the generator suffers from the "checker-board effect", which suggests that the training

Figure 2. Examples of images generated by our GAN model per epoch (real fingerprints from the Cross Match dataset).

Table 3. Accuracy results for the FPAD classifier trained with data augmentation based on images generated by the GAN model.

Material / Dataset	Cross Match	Digital Persona	Green Bit	Hi Scan	Time Series
Body Double	-	-	-	-	-
Ecoflex 00 50	-	-	-	-	-
Ecoflex	0.8859	-	-	-	-
Playdoh	0.8787	-	-	-	-
Latex	-	0.9640	-	0.9744	-
Gelatine	-	0.9112	-	0.9008	-
Wood Glue	-	0.9392	-	0.8704	-

was still unstable in that time-point [14]. When the GAN reaches equilibrium, we can observe that the synthetic image is now more realistic, whether it is from the *bona fide* or the PA classes. It is also important to acknowledge that we are dealing with low resolution images (*i.e.,* 64×64); therefore, it is not reasonable to expect images with fine detail (*e.g.*, texture features).

5.2. cGAN Results

Figure 3 shows example images generated by our cGAN model in different training epochs. This model is conditional, therefore it receives a *bona fide* image and a noise vector from which the output is created. The same input image and noise vector was used at every training epoch to visualise the progress of the training. Due to the low resolution of the images (64×64), some details could not be captured, which is one of the limitations of this work. Nevertheless, the process of generating new samples helped the training of the FPAD classifier in some cases, as can be seen in 5.3.

5.3. FPAD Results

Table 2, Table 3 and Table 4 present the results for the FPAD classifier trained without data augmentation, the FPAD classifier trained with data augmentation based on images generated by the GAN model and the FPAD classifier trained with data augmentation based on images generated by the cGAN model, respectively. Regarding the performance of the FPAD models with or without data augmentation strategies based on a GAN or a cGAN, it is still not possible to conclude whether or not these strategies improve the model's predictive performance, however, one may observe that there is no clear evidence that their performance gets worse. Hence, this may suggest that the synthetic data we add during training may be part of the same distribution as the original data.

6. Conclusions and Future Work

This work presented an exploratory study of the use of the application of generative models to create synthetic *bona fide* and PA fingerprint images as a potential mean to increase the robustness of the models against unseen attacks. In this work, we optimised two generative models (GAN and cGAN) to generate *bona fide* and PA fingerprint images and trained a classifier with these synthetic images in the train set. Current results suggest that the predictive performance of the classifier is not jeopardised, which means that the synthetic data may be part of the same distribution as the original data. It is important to note that these experiments do not take into account unseen attacks (*i.e.*, models were just tested with the PAs that they were trained with). Future work should be devoted to improving the architectures of both the GAN and cGAN in the sense that they could be trained with higher resolutions. This is relevant because there are several features of the fingerprints that can only be detected at higher resolutions, however, we also noted that optimising a GAN or a cGAN with higher-resolution images (*e.g.* 128×128) is fairly difficult and leads to high training instability. Since this work was focused on the generative models, we did not address the unseen attack setting. Therefore, future work should be devoted to testing the predictive performance of models that were trained with synthetic data against unseen attacks and optimising generative models to generate samples from more than one type of PA.

References

[1] M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint*

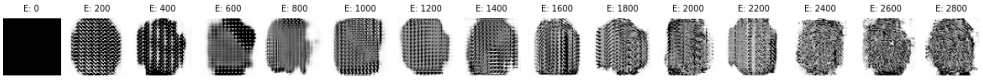

Figure 3. Examples of images generated by our cGAN model per epoch (wood glue FPA from the Hi Scan dataset).

Table 4. Accuracy results for the FPAD classifier trained with data augmentation based on images generated by the cGAN model.

Material / Dataset	Cross Match	Digital Persona	Green Bit	Hi Scan	Time Series
Body Double	-	-	-	-	-
Ecoflex 00 50	-	-	-	-	-
Ecoflex	0.8232	-	-	-	-
Playdoh	0.9062	-	-	-	-
Latex	-	0.9496	-	0.9712	-
Gelatine	-	0.9160	-	0.9064	-
Wood Glue	-	0.9384	-	0.8776	-

arXiv:1701.04862, 2017. 2

[2] J. J. Engelsma and A. K. Jain. Generalizing fingerprint spoof detector: Learning a one-class classifier. In *2019 International Conference on Biometrics (ICB)*, pages 1–8. IEEE, 2019. 1, 2

[3] P. M. Ferreira, D. Pernes, A. Rebelo, and J. S. Cardoso. Learning signer-invariant representations with adversarial training. In *Twelfth International Conference on Machine Vision (ICMV 2019)*, volume 11433, page 114333D. International Society for Optics and Photonics, 2020. 2

[4] P. M. Ferreira, A. F. Sequeira, D. Pernes, A. Rebelo, and J. S. Cardoso. Adversarial learning for a robust iris presentation attack detection method against unseen attack presentations. In *2019 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–7. IEEE, 2019. 2

[5] 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*, 58:110–128, 2017. 1, 3

[6] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014. 2

[7] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 4

[8] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3, 4

[9] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar. *Handbook of fingerprint recognition*. Springer Science & Business Media, 2009. 1

[10] E. Marasco and C. Sansone. On the robustness of fingerprint liveness detection algorithms against new materials used for spoofing. In *BIOSIGNALS*, volume 8, pages 553–555, 2011. 1

[11] T. Matsumoto, H. Matsumoto, K. Yamada, and S. Hoshino. Impact of artificial" gummy" fingers on fingerprint systems. In *Optical Security and Counterfeit Deterrence Techniques IV*, volume 4677, pages 275–289. International Society for Optics and Photonics, 2002. 1

[12] D. Menotti, G. Chiachia, A. Pinto, W. R. Schwartz, H. Pedrini, A. X. Falcao, and A. Rocha. Deep representations for iris, face, and fingerprint spoofing detection. *IEEE Transactions on Information Forensics and Security*, 10(4):864–879, 2015. 1

[13] M. Mirza and S. Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. 2

[14] A. Odena, V. Dumoulin, and C. Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016. 5

[15] J. A. Pereira, A. F. Sequeira, D. Pernes, and J. S. Cardoso. A robust fingerprint presentation attack detection method against unseen attacks through adversarial learning. In *2020 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–5. IEEE, 2020. 2

[16] J. A. P. Pereira. Fingerprint anti spoofing-domain adaptation and adversarial learning. 2020. 1, 2, 3

[17] A. Pinto, H. Pedrini, M. Krumdick, B. Becker, A. Czajka, K. W. Bowyer, and A. Rocha. Counteracting presentation attacks in face, fingerprint, and iris recognition. *Deep learning in biometrics*, 245, 2018. 1

[18] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 3

[19] A. Rattani, W. J. Scheirer, and A. Ross. Open set fingerprint spoof detection across novel fabrication materials. *IEEE Transactions on Information Forensics and Security*, 10(11):2447–2460, 2015. 1

[20] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. 4

[21] A. F. Sequeira, S. Thavalengal, J. Ferryman, P. Corcoran, and J. S. Cardoso. A realistic evaluation of iris presentation attack detection. In *2016 39th International Conference on Telecommunications and Signal Processing (TSP)*, pages 660–664. IEEE, 2016. 2