A. Brömme, C. Busch, N. Damer, A. Dantcheva, M. Gomez-Barrero, K. Raja,
C. Rathgeb, A. Sequeira, and A. Uhl (Eds.): BIOSIG 2021,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2021

Shuffled Patch-Wise Supervision for Presentation Attack Detection

Alperen Kantarcı[1], Hasan Dertli[2], Hazım Kemal Ekenel[1]

Abstract: Face anti-spoofing is essential to prevent false facial verification by using a photo, video, mask, or a different substitute for an authorized person's face. Most of the state-of-the-art presentation attack detection (PAD) systems suffer from overfitting, where they achieve near-perfect scores on a single dataset but fail on a different dataset with more realistic data. This problem drives researchers to develop models that perform well under real-world conditions. This is an especially challenging problem for frame-based presentation attack detection systems that use convolutional neural networks (CNN). To this end, we propose a new PAD approach, which combines pixel-wise binary supervision with patch-based CNN. We believe that training a CNN with face patches allows the model to distinguish spoofs without learning background or dataset-specific traces. We tested the proposed method both on the standard benchmark datasets —Replay-Mobile, OULU-NPU— and on a real-world dataset. The proposed approach shows its superiority on challenging experimental setups. Namely, it achieves higher performance on OULU-NPU protocol 3, 4 and on inter-dataset real-world experiments.

Keywords: Face antispoofing, presentation attack detection, convolutional neural networks, real-world dataset.

1 Introduction

In recent years, facial recognition systems are widely used as they are robust and reliable for common usage. However, these recognition systems have to be careful about the authenticity of a given face input. If the given input is recorded from a video of an authorized user, the recognition system should not recognize the person in the video and give access to the system. Presentation attack detection (PAD) systems aim to prevent this problem by evaluating the liveness of the given person's image.

In recent years, PAD methods improved significantly with the progress in deep learning methods and publicly available large, representative datasets [Bo17, LJL18, Zh20, Co16]. Most of the significant progress has been achieved when researchers found different cues to decide liveness of a face [Li16, MHP11, At17]. These different cues used with complex deep neural networks to create PAD systems that are very successful in intra-dataset benchmark results. However, the real challenge in PAD still remains as an inter-dataset benchmark which shows the real performance of the PAD systems in real-world like scenario. Most of the systems that use CNNs overfit the data easily by memorizing reflection

[1] Department of Computer Engineering, Istanbul Technical University, Turkey, {kantarcia,ekenel}@itu.edu.tr
[2] Sodec Technologies, Istanbul, Turkey, hasan.dertli@sodecapps.com

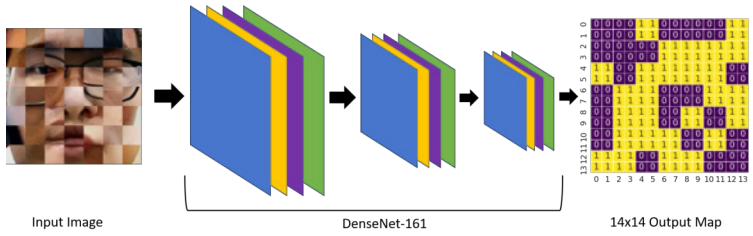

Fig. 1: Overview of the proposed method

and illumination effects. To address this problem, in this paper, we propose a new training procedure for face PAD systems. We show that our training method utilizes the pixel-wise binary loss in a better way. Moreover, we show that our proposed method improves model performances on real-world experiments.

2 Related Work

PAD approaches are mainly grouped into two categories; video-based and frame level. While video-based methods use temporal consistency and temporal cues, frame-based methods use subtle cues from the given face images. These cues can be summarized as liveness, texture, and 3D geometry [Mi20]. Among these cue-based methods, liveness cues are applicable for video-based PAD. Therefore, texture and 3D geometry cue-based methods are more popular for detecting liveness from a frame. 3D geometry cue-based methods either use depth or pseudo-depth signals to distinguish attacks from real attempts. Even though devices with depth sensors, time-of-flight cameras, Lidar sensors, etc., are getting popular in daily usage, most mobile phones or video cameras do not have depth sensors. Therefore, methods mostly rely on pseudo-depth maps which are not real data and may not reflect real-world data distribution very well. Therefore, most of these methods might get good results on specific datasets but fail to generalize.

As initial work with deep neural networks, [YLL14] proposes to use a face alignment network for preparing face images to train an AlexNet [KSH12] model. They use the model for extracting features of the face and use an SVM classifier to classify images as artefact or bona fide.

In order to improve PAD performances, researchers search different supervisions, along with the binary classification objective, for training their models. [GM19] proposes an effective model for frame-level PAD. They add additional supervision, which they call pixel-wise binary supervision, to simplify the necessity of complex depth maps and temporal information. Their model creates a 14x14 score map which helps to perform pixel-wise binary supervision. On top of this supervision, their model is guided with binary cross-entropy. We build our model on top of [GM19] by using their pixel-wise binary supervision and model architecture. Instead of using binary cross-entropy, we propose to use only pixel-wise binary supervision. Moreover, we train our models with shuffled face images that are created by multiple patches of different face parts of different subjects.

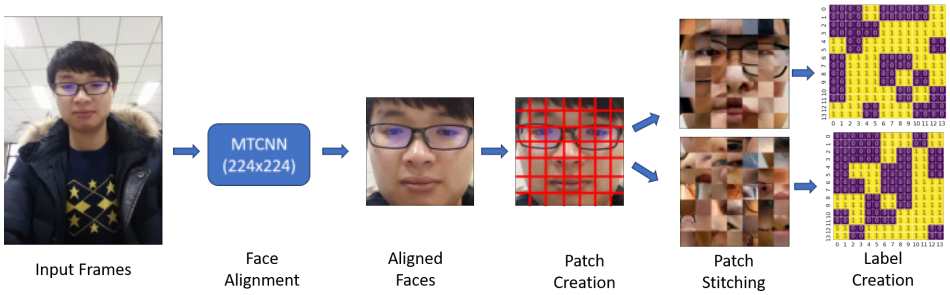

Fig. 2: Our overall pipeline of input and ground truth label creation.

[At17] proposes a two-stream CNN that uses patches and depth maps. They claim that patch-based CNN learns to discriminate artefact patches independent of the spatial face areas whereas depth-based CNN allows the model to learn how a face-like holistic depth map should look like. Our work, which combines pixel-wise binary supervision with patch-based CNN, is inspired by [GM19, At17]. As [At17] showed, we believe that training a CNN with face patches allows the model to distinguish artefacts without learning background or dataset-specific traces. Therefore, it prevents the model to overfit. Effectiveness of similar patch shuffling method is presented in [Ka17]. In [Ka17] it is shown that shuffling pixels within a patch can increase the generalization of the models. Our approach tries to mimic this behaviour from a higher level of view.

Currently, the state-of-the-art frame-level PAD method [Yu20] leverages a novel convolution operation. Authors propose Central Difference Convolution (CDC) for detecting detailed artefact traces. CDC specifically focuses on artefact traces. Their model is trained with pseudo-depth maps which require additional pseudo-depth map creation steps for ground truth. They also use computationally expensive Neural Architecture Search (NAS) to find a better and more efficient model which they call CDCN++. They report the lowest error rates on the OULU-NPU [Bo17] dataset.

3 Methodology

Preprocessing of the images is an important task in PAD systems. The preprocessing pipelines are very similar among different methods. We first detect faces and face landmarks in the given frame with MTCNN [Zh16] face detector. Then by using Bob framework [An12, An17] we align the detected faces according to the eye coordinates and crop these aligned faces in 224x224 resolution. After that, we create patches from each face image by dividing the face into 7x7. Therefore, we get 49 face patches with 32x32 resolutions for each face image. We then combine these patches to create a new 224x224 image. Each face patch corresponds to a 2x2 location in the ground truth 14x14 label map. Bona fide patches have 1 as the label and fake patches have 0 as the label. We call this process patch stitching. We use two different strategies while stitching the patches. In the first approach, which we call random stitching, we completely randomize the patches and do not care

about facial structure while combining different patches. Therefore, in this method, we get completely shuffled face images. As we randomly choose patches, the same parts of faces can be found in stitched images. For example, multiple noses or eyes can be seen in the bottom part of the Figure 2. In the second approach, which we call controlled-stitching, we combine patches of different faces while keeping the facial structure as much as possible. Therefore, we actually create an input that resembles a face and consists of multiple subjects' face parts. The input contains both bona fide and attack patches, therefore stitched images must have labels for each patch separately. In our experiments we use the former approach which gives better performance in inter-dataset experiments. The overview of the proposed method is illustrated in the Figure 1.

3.1 Model Architecture and Training

We use a deep CNN network that takes 224x224 input images and creates a 14x14 output map. We use the model that was proposed in [GM19] which is based on DenseNet-161 [Hu17] architecture. However, we do not have the final linear and sigmoid layers. The model contains the first eight layers of the DenseNet [Hu17] and we use pretrained weights. At the end of eight layers, we add a 1x1 convolution to produce a 14x14x1 feature map which is the model output.We use input images that contain multiple patches together, therefore, we do not have a binary label. In training time, we use 14x14 pixel-wise binary labels to train the model with Binary Cross Entropy (BCE) loss. We assign ground truth y=0 for patches that come from attack images and y=1 for patches that come from bona fide images. The equation for pixel-wise loss is shown below.

$$\mathscr{L}_{pixel-wiseBCE} = -(y(log(p)) + (1-y)log(1-p)) \tag{1}$$

In Equation 1 p is the 14x14 model output that contains probability values between 0 and 1. We minimize this loss with Adam [KB15] optimizer. We use 0.001 as the initial learning rate and halve this value at each 10th epoch. We use 32 as batch size and we generally train our methods for 30 epochs. We use horizontal flip and color jitter as data augmentation. In the test time, we give the aligned face image to the model and our model gives 14x14 output. We calculate the mean probability score by using the 14x14 output, then we use this probability as our liveness score. If the score is higher than the predefined threshold, we classify the given input as bona fide, else we classify it as an attack.

4 Experimental Results

This section presents the datasets that have been used to assess the performance of the proposed approach, the experiments carried out, and the obtained results.

(a) Samples from Sodec Real-World dataset (b) Samples from OULU-NPU dataset

Fig. 3: Sample images from OULU-NPU and Sodec Real-World datasets.

4.1 Datasets and Metrics

Replay Mobile: Replay Mobile dataset [Co16] consists of 1190 video clips of 40 subjects. It contains paper and replay presentation attacks under five different lighting conditions. Each subject has 10 videos under different lighting and background conditions. There are mainly two kinds of attacks; matte screen and print. Matte screen attacks are the replay attack scenario where subject videos are displayed on a 1080p monitor, then recorded off of it. In print attacks, the faces of the subjects are printed on A4 paper then put on a stationary surface. The dataset has *"grandtest"* protocol for evaluating the global performance of an algorithm. We report our results on the *"grandtest"* protocol.

OULU-NPU: OULU-NPU dataset [Bo17] is a high resolution antispoofing dataset. It has over 5900 videos of 55 subjects. The dataset has both print and replay attacks with two printers and two display devices. There are 4 protocols for evaluating the methods' generalization capabilities. The protocol names were set based on the level of difficulty.

SiW: SiW dataset [LJL18] is one of the largest high-quality antispoofing datasets. It has over 4400 videos of 165 subjects collected over 4 different sessions. All videos have 1080p resolution and contain variations of distance, pose, illumination, and expression. It contains both print and replay attacks. The dataset has 3 protocols to test the generalization of the models. We utilize this dataset for inter-dataset experiments because of the following reasons: it contains one of the highest numbers of subjects among PAD datasets, it contains different poses and expressions, and the acquisition device changes its distance to subjects which is a common behavior in the real world PAD attempts.

Sodec Real-World Dataset: Sodec Real-World dataset has been collected to simulate the real-world presentation attack scenarios. It contains more than 51k frames of 31 different subjects. There are 16 male and 15 female subjects. It contains only replay attacks over 3 different presentation attack instruments (PAI), namely mobile phone, computer display, and television. Unlike controlled datasets, every subject recorded real videos with

their mobile phone in their own home. Moreover, subjects were asked to rotate the phone vertically on the spot themselves while holding the phone in their outstretched arms and recording the video of themselves. It allows us to capture different backgrounds, illumination conditions, and pose variations with different input sensors. We show some examples from the dataset in Figure 3. We utilize this dataset in inter-dataset experiments to test models on real-world attacks.

In all of our experiments, we used ISO/IEC 30107-3 [IS16] standards which are standard metrics for the PAD. Attack Presentation Classification Error Rate (APCER), Bona fide Presentation Classification Error Rate (BPCER), Half Total Error Rate (HTER) along with the Average Classification Error Rate (ACER) in the test set are reported in the experiments. For all of our experiments, we use the threshold according to the equal error rate (EER) criterion. Average of False Recognition Rate (FRR) and False Acceptance Rate (FAR) is equal to HTER. We show the calculation of these metrics in Equation 2 and Equation 3.

$$ACER = \frac{APCER + BPCER}{2} \tag{2}$$

$$HTER = \frac{FRR + FAR}{2} \tag{3}$$

4.2 Experiments and Results

We use OULU-NPU and Replay Mobile datasets for our intra-dataset experiments. As explained above, on OULU-NPU we report APCER, BPCER, and ACER performances of our and other models. On Replay Mobile dataset we report EER and HTER performances. We compare our method with CDCN++ [Yu20] and DeepPixBis [GM19] models. CDCN++ is a state-of-the-art method that uses pseudo-depth maps and NAS. DeepPixBis has the same CNN architecture as our model. We differ from DeepPixBis in only our input and label creation where we utilize patch-wise labels. Therefore, our proposed method is directly comparable with DeepPixBis [GM19]. Similar to us, in the Replay-Mobile dataset, most of the newest methods report 0% error. Table 1 shows that our proposed method also achieves 0% error on the dataset. In Table 2 we report our intra-dataset results on the

Model	**EER(%)**	**HTER(%)**
CDCN	**0.0**	**0.0**
DeepPixBis	**0.0**	**0.0**
Ours	**0.0**	**0.0**

Tab. 1: Intra-dataset test results of Replay-Mobile *"grandtest"* protocol. The first column represents the Equal Error Rate (EER) in percentage and the second represents the Half Total Error Rate (HTER) in percentage.

Protocol	Model	APCER(%)	BPCER(%)	ACER(%)
1	CDCN	**0.4**	1.7	1.0
	CDCN++	**0.4**	**0.0**	**0.2**
	DeepPixBis	0.83	**0.0**	0.42
	Ours	2.14	2.14	2.14
2	CDCN	**1.5**	1.4	1.5
	CDCN++	1.8	**0.8**	**1.3**
	DeepPixBis	11.39	0.56	5.97
	Ours	6.22	6.26	6.24
3	CDCN	2.4 ± 1.3	2.2 ± 2.0	2.3 ± 1.4
	CDCN++	$\mathbf{1.7 \pm 1.5}$	$\mathbf{2.0 \pm 1.2}$	$\mathbf{1.8 \pm 0.7}$
	DeepPixBis	11.67 ± 19.57	10.56 ± 14.06	11.11 ± 9.4
	Ours	6.10 ± 2.57	6.30 ± 2.55	6.20 ± 2.55
4	CDCN	4.6 ± 4.6	9.2 ± 8.0	6.9 ± 2.9
	CDCN++	$\mathbf{4.2 \pm 3.4}$	$\mathbf{5.8 \pm 4.9}$	$\mathbf{5.0 \pm 2.9}$
	DeepPixBis	36.67 ± 29.67	13.33 ± 16.75	25.0 ± 12.0
	Ours	11.51 ± 7.86	11.58 ± 7.84	11.54 ± 7.84

Tab. 2: Intra-dataset test results of OULU-NPU dataset.

OULU-NPU dataset. Table 2 shows that our method outperforms DeepPixBis on Protocol 3 and Protocol 4 which are the hardest protocols in the dataset. These protocols have smaller training data and their test data is not very similar to the training data in terms of environment, PAI, PA acquisition device. Our method has a higher error rate on Protocol 1 and Protocol 2, but according to the our experiment results we show that our method is well suited for generalization whereas other methods gain an advantage of similar training and testing sets in these protocols. Our result does not outperform the state-of-the-art PAD model, however, CDCN and CDCN++ use pseudo-depth maps. In the case of CDCN++, it employs computationally expensive NAS operations.

4.3 Real-World Experiments

There are many face antispoofing datasets that are collected in controlled environments. Most of these datasets only consider two or three backgrounds with controlled illumination changes. However, in real-world applications, there are various backgrounds, illumination, poses, and expression changes. Moreover, attackers are more careful when creating a face presentation attack. We have collected a dataset to simulate the real-world use case of the presentation attacks. We trained CDCN [Yu20], DeepPixBis [GM19], and our model on the SiW dataset Protocol 1. We choose the SiW dataset because it is one of the most representative datasets in terms of distance, pose, illumination, and expression changes. From Table 3 we see that state-of-the art CDCN model has achieved the lowest error rate in the SiW dataset, but gets a higher error rate on Sodec Real-World Dataset. Our method is able to outperform DeepPixBis on both the SiW dataset and Sodec Real-World Dataset.

	Trained on SiW	
Model	tested on SiW	tested on Sodec Real-World Dataset
CDCN	**0.12**	12.52
DeepPixBis	3.68	12.05
Ours	2.15	**5.24**

Tab. 3: Experiment results of SiW (Protocol 1) and inter-dataset test results on Sodec Real-World Dataset. Reported metrics in the table represents the ACER values in percentage (%)

The results show that our proposed training method is useful for real-world inter-dataset scenarios which is the hardest task to perform.

5 Conclusion and Future Work

In this paper, we propose a new training method for the PAD models. Our proposed method uses combined face patches instead of one single face image in training time. We show that training models with pixel-wise binary loss and shuffled face patches can improve PAD performance. Our proposed method improves DeepPixBis' [GM19] performance on OULU-NPU Protocol 3 and 4 which are the hardest protocols. Moreover, the proposed method performs much better when we test the models on a real-world dataset. For future work, we are planning to extend the Sodec Real World dataset to print attacks. Furthermore, we are planning to modify the backbone architecture of the model and analyze the effects of different patch creation methods on model behavior.

6 Acknowledgements

This work was partially supported by a Sodec Technologies research grant. We would like to thank Sodec Technologies for their data collection efforts and support for this work.

References

[An12] Anjos, A.; Shafey, L. El; Wallace, R.; Günther, M.; McCool, C.; Marcel, S.: Bob: a free signal processing and machine learning toolbox for researchers. In: 20th ACM Conference on Multimedia Systems (ACMMM), Nara, Japan. October 2012.

[An17] Anjos, A.; Günther, M.; de Freitas Pereira, T.; Korshunov, P.; Mohammadi, A.; Marcel, S.: Continuously Reproducing Toolchains in Pattern Recognition and Machine Learning Experiments. In: Intl. Conference on Machine Learning (ICML). August 2017.

[At17] Atoum, Y.; Liu, Y.; Jourabloo, A.; Liu, X.: Face anti-spoofing using patch and depth-based CNNs. In: IEEE Intl. Joint Conference on Biometrics. 2017.

[Bo17] Boulkenafet, Z.; Komulainen, J.; Li, Lei.; Feng, X.; Hadid, A.: OULU-NPU: A mobile face presentation attack database with real-world variations. In: IEEE Intl. Conference on Automatic Face and Gesture Recognition. May 2017.

[Co16] Costa-Pazo, A.; Bhattacharjee, S.; Vazquez-Fernandez, E.; Marcel, S.: The REPLAY-MOBILE Face Presentation-Attack Database. In: Proceedings of the Intl. Conference on Biometrics Special Interests Group. September 2016.

[GM19] George, A.; Marcel, S.: Deep Pixel-wise Binary Supervision for Face Presentation Attack Detection. In: 12th IAPR Intl. Conference on Biometrics (ICB). 2019.

[Hu17] Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K. Q.: Densely Connected Convolutional Networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2261–2269, 2017.

[IS16] ISO/IEC JTC1 SC37: Information Technology - Biometric presentation attack detection - Part 3: Testingand Reporting. Technical report, International Organization for Standardization, Geneva, CH, February 2016.

[Ka17] Kang, Guoliang; Dong, Xuanyi; Zheng, Liang; Yang, Yi: Patchshuffle regularization. 2017.

[KB15] Kingma, Diederik P.; Ba, Jimmy: , Adam: A Method for Stochastic Optimization, 2015.

[KSH12] Krizhevsky, A.; Sutskever, I.; Hinton, G.: ImageNet Classification with Deep Convolutional Neural Networks. In: Advances in Neural Information Processing Systems. volume 25, 2012.

[Li16] Li, X.; K., Jukka; Z., G.; Y., Pong-Chi; Pietikainen, M.: Generalized face anti-spoofing by detecting pulse from face videos. In: 2016 23rd International Conference on Pattern Recognition (ICPR). pp. 4244–4249, 2016.

[LJL18] Liu, Y.; Jourabloo, A.; Liu, X.: Learning Deep Models for Face Anti-Spoofing: Binary or Auxiliary Supervision. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 389–398, 2018.

[MHP11] Maatta, J.; Hadid, A.; Pietikainen, M.: Face spoofing detection from single images using micro-texture analysis. In: 2011 Intl. Joint Conference on Biometrics. pp. 1–7, 2011.

[Mi20] Ming, Z.; Visani, M.; Luqman, M. M.; Burie, J.C.: A Survey On Anti-Spoofing Methods For Face Recognition with RGB Cameras of Generic Consumer Devices. In: arxiv.org/abs/2010.04145. 2020.

[YLL14] Yang, J.; Lei, Z.; Li, S. Z.: Learn Convolutional Neural Network for Face Anti-Spoofing. In: arxiv.org/abs/1408.5601. 2014.

[Yu20] Yu, Z.; Zhao, C.; Wang, Z.; Qin, Y.; Su, Z.; Li, X.; Zhou, F.; Zhao, G.: Searching Central Difference Convolutional Networks for Face Anti-Spoofing. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5294–5304, 2020.

[Zh16] Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y.: Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. IEEE Signal Processing Letters, 23(10):1499–1503, 2016.

[Zh20] Zhang, S.; Liu, A.; Wan, J.; L., Yanyan; Guo, G.; Escalera, S.; Escalante, H. J.; Li, S. Z.: CASIA-SURF: A Large-Scale Multi-Modal Benchmark for Face Anti-Spoofing. IEEE Transactions on Biometrics, Behavior, and Identity Science, 2(2):182–193, 2020.