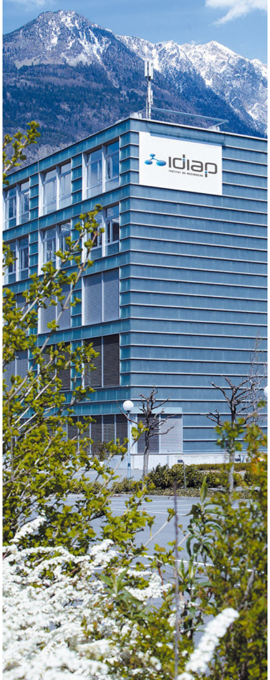
Improving Cross-dataset Performance of Face Presentation Attack Detection Systems Using Face Recognition Datasets

ICASSP 2020

Amir Mohammadi, Sushil Bhattacharjee, Sébastien Marcel

May 8, 2020

idiap RESEARCH INSTITUTE EPFL



Outline

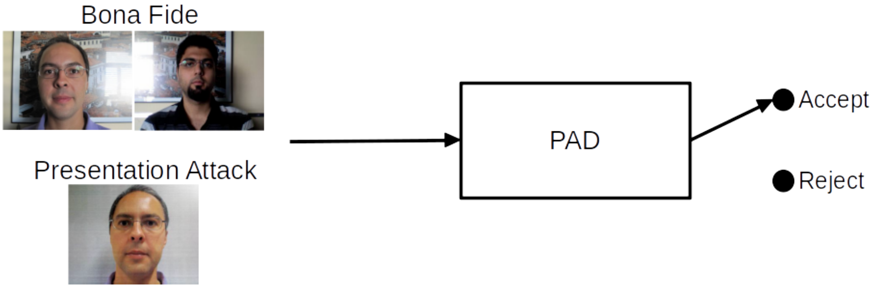
1. Introduction
2. Related Work
3. Proposed Method
4. Experiments
5. Conclusions

Face Recognition (FR)

FR systems are vulnerable to presentation attacks.

Examples of presentation attacks (PA): print attack (left) and 3D rigid mask attack (right).

Presentation Attack Detection (PAD)



PAD systems are binary classification systems.

Presentation Attack Detection (PAD)

Bona Fide

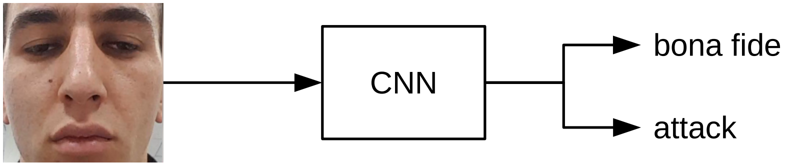
Replay PA

Print Mask PA

Print PA

PAD systems rely on artifacts present in presentation attacks to detect them.

PAD Using Deep Learning



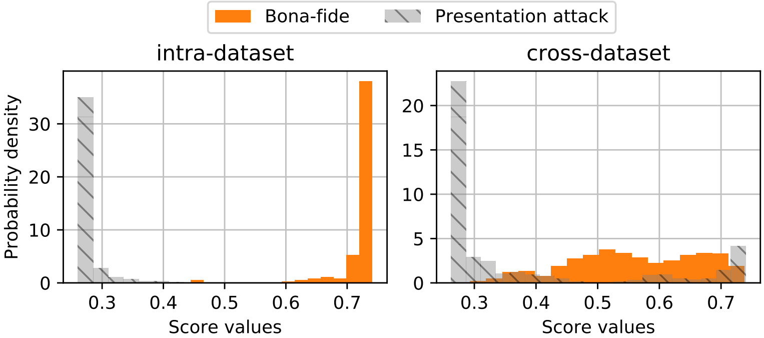
CNN based PAD approaches outperform previous methods which use hand-crafted features[1,2,3].

[1] J. Yang, Z. Lei, and S. Z. Li. "Learn Convolutional Neural Network for Face Anti-Spoofing". In: *arXiv:1408.5601 [cs]* (Aug. 2014).

[2] K. Patel, H. Han, and A. Jain. "Cross-Database Face Antispoofing with Robust Feature Representation". In: *Chinese Conference on Biometric Recognition*. 2016.

[3] Z. Boulkenafet, J. Komulainen, Z. Akhtar, et al. "A Competition on Generalized Software-Based Face Presentation Attack Detection in Mobile Scenarios". In: *Proceedings of the International Joint Conference on Biometrics, 2017*. Oct. 2017.

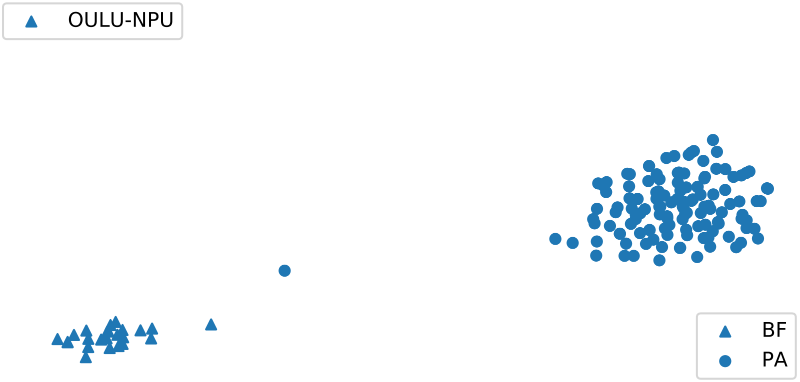
Problem: Generalization in PAD



Intra-dataset vs cross-dataset PAD evaluation.

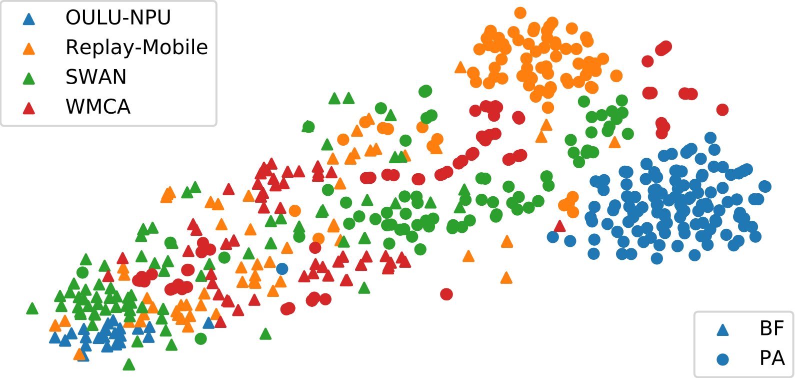
- Cross-dataset evaluations represent real-world scenarios.

Domain Shift



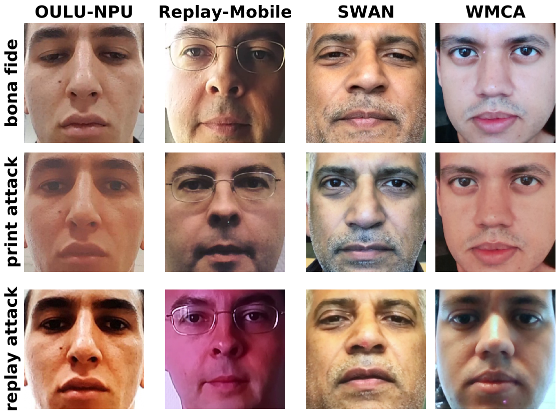
Visualization of learned features of a PAD CNN using TSNE.
The PAD CNN is trained on the OULU-NPU dataset.

Domain Shift



Visualization of learned features of a PAD CNN using TSNE.
The PAD CNN is trained on the OULU-NPU dataset.

Nuisance Factors[4]



Domain shift is caused by variations of nuisance factors

[4] https://en.wikipedia.org/wiki/Nuisance_variable

Nuisance Factors

Nuisance factors include:

- camera device
- distance of the subject from the camera
- instrument used to create the attack
- lighting conditions
- identity, pose, etc.

Nuisance Factors

Nuisance factors include:

- camera device
- distance of the subject from the camera
- instrument used to create the attack
- lighting conditions
- identity, pose, etc.

Current face PAD datasets contain limited variations of nuisance factors.

- Less than 10 camera devices
- 50 to 150 identities
- Limited variations in lighting conditions and pose

Related Work: Domain Generalization Methods

Most methods account for domain shift by learning features that are domain invariant:

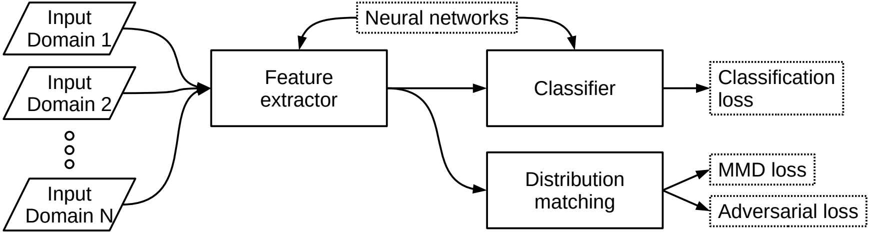


Diagram of a typical domain generalization method.

MMD: Maximum Mean Discrepancy[5]

[5] A. Gretton et al. "A Kernel Two-Sample Test". In: *Journal of Machine Learning Research* 13.Mar (2012).

Related Work

What is a domain?

1. Each camera device is a domain and MMD is used[6].
2. Each PAD dataset is a domain and an adversarial loss is used[7].

[6] H. Li et al. "Learning Generalized Deep Feature Representation for Face Anti-Spoofing". In: *IEEE Transactions on Information Forensics and Security* 13.10 (Oct. 2018).

[7] R. Shao et al. "Multi-Adversarial Discriminative Deep Domain Generalization for Face Presentation Attack Detection". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

Related Work

Downsides of most domain generalization methods:

- Domain needs to be defined.
- Data from each domain is needed.

Proposed Method: Motivation

How can we account for nuisance factors?

Some nuisance factors are common between *bona fide* and presentation attacks, such as:

- identities
- camera devices
- lighting conditions

Some nuisance factors are specific to presentation attacks, such as:

- presentation attack instruments

Nuisance Factors

Current face PAD datasets contain limited variations of nuisance factors.

- Less than 10 camera devices
- 50 to 150 identities
- Limited variations in lighting conditions and pose

Nuisance Factors

Current face PAD datasets contain limited variations of nuisance factors.

- Less than 10 camera devices
- 50 to 150 identities
- Limited variations in lighting conditions and pose

Face recognition datasets contain large variations of many of those nuisance factors.

- Hundreds of different camera devices
- More than 100,000 identities
- Faces captured in the *wild* with a variety of lighting conditions and pose

Proposed Method

Hypothesis

All the underlying factors that explain the data in a face recognition dataset (which contains only *bona fide* samples) are nuisance factors in a face PAD system.

- Face PAD datasets contain limited variations of nuisance factors.
- Face recognition datasets are much larger and more varied and can help us model the common nuisance factors.

Proposed Method

$$\text{Assume: } \mathbf{I} = \mathrm{f}(\mathbf{y}, \mathbf{z}_1, \mathbf{z}_2) + \epsilon$$

- $\mathbf{I}$ is a face image.
- f is a function.
- $\mathbf{y}$ is the variable that we want to predict – whether $\mathbf{I}$ is a PA.
- $\mathbf{z}_1$ is the variable that represents nuisance factors **common** between two classes.
- $\mathbf{z}_2$ is the variable that represents nuisance factors exclusive to presentation attacks.
- ϵ is noise.

Proposed Method

$$\text{Assume: } \mathrm{f}(\mathbf{y}, \mathbf{z}_1, \mathbf{z}_2) = \mathrm{g}(\mathbf{z}_1) + \mathrm{h}(\mathbf{y}, \mathbf{z}_2)$$

- g and h are functions that produce images given their respective latent variables.

Proposed Method

$$\text{Assume: } \mathrm{f}(\mathbf{y}, \mathbf{z}_1, \mathbf{z}_2) = \mathrm{g}(\mathbf{z}_1) + \mathrm{h}(\mathbf{y}, \mathbf{z}_2)$$

- g and h are functions that produce images given their respective latent variables.

$$\text{Assume: } \mathbf{z}_1 = \mathrm{e}(\mathbf{I}),$$

- e and g are the functions that we want to model.

Proposed Method

$$\text{Assume: } \mathrm{f}(\mathbf{y}, \mathbf{z}_1, \mathbf{z}_2) = \mathrm{g}(\mathbf{z}_1) + \mathrm{h}(\mathbf{y}, \mathbf{z}_2)$$

- g and h are functions that produce images given their respective latent variables.

$$\begin{aligned} \text{Assume: } \mathbf{z}_1 &= \mathrm{e}(\mathbf{I}), \\ \mathbf{I}_{z_1} &= \mathrm{g}(\mathbf{z}_1) = \mathrm{g}(\mathrm{e}(\mathbf{I})), \end{aligned}$$

- e and g are the functions that we want to model.

Proposed Method

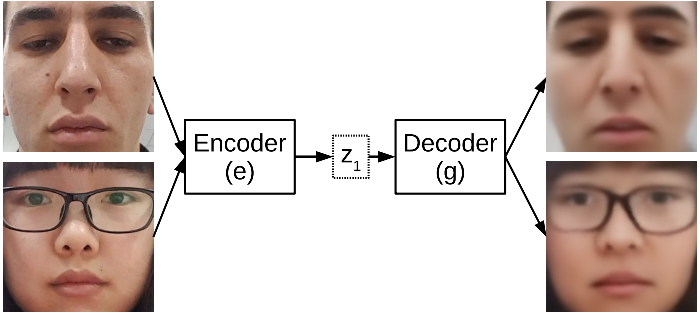
$$\text{Assume: } \mathrm{f}(\mathbf{y}, \mathbf{z}_1, \mathbf{z}_2) = \mathrm{g}(\mathbf{z}_1) + \mathrm{h}(\mathbf{y}, \mathbf{z}_2)$$

- g and h are functions that produce images given their respective latent variables.

$$\begin{aligned} \text{Assume: } \mathbf{z}_1 &= \mathrm{e}(\mathbf{I}), \\ \mathbf{I}_{z_1} &= \mathrm{g}(\mathbf{z}_1) = \mathrm{g}(\mathrm{e}(\mathbf{I})), \\ \mathrm{h}(\mathbf{y}, \mathbf{z}_2) &\approxeq \mathbf{I} - \mathbf{I}_{z_1} = \mathbf{I}_{y,z_2} \end{aligned}$$

- e and g are the functions that we want to model.

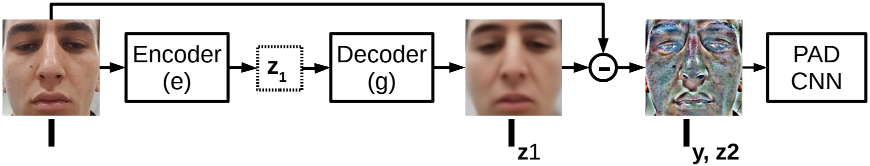
Proposed Method Using Deep Autoencoders



Autoencoders can model the factors present in data.

- Using a face recognition dataset to train an autoencoder allows us to accurately model $\mathbf{z}_1$ nuisance factors.

Proposed Method Using Deep Autoencoders



- The proposed method adds a pre-processing step to traditional methods.

$$\mathbf{I}_{z_1} = \mathrm{g}(\mathbf{z}_1) = \mathrm{g}(\mathrm{e}(\mathbf{I}))$$
$$\mathrm{h}(\mathbf{y}, \mathbf{z}_2) \approx \mathbf{I} - \mathbf{I}_{z_1} = \mathbf{I}_{y,z_2}$$

Autoencoder Details

- InfoVAE (a variational autoencoder) was used in the experiments.
- Encoder: DenseNet-161[8]
- Decoder: 7 layer deep CNN[9]
- Dimension of $\mathbf{z}_1$: 256
- Prior distribution: $\mathcal{N}(0, 3)$ (diagonal covariance matrix)
- Face recognition datasets: cleaned versions of Microsoft Celeb (MS-Celeb-1M)[10] and the Celeb-A[11]

[8] G. Huang et al. "Densely Connected Convolutional Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

[9] T. Miyato et al. "Spectral Normalization for Generative Adversarial Networks". In: *International Conference on Learning Representations*. 2018.

[10] Y. Guo et al. "MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition". In: *arXiv preprint arXiv:1607.08221* (2016).

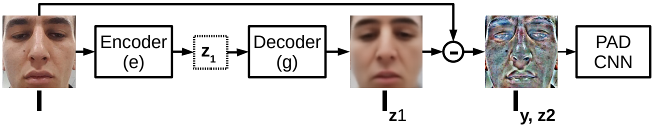
[11] Z. Liu et al. "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)*. Dec. 2015.

Experiments

Evaluation of 3 PAD systems:

- DeepPixBiS[12] as a baseline PAD CNN.

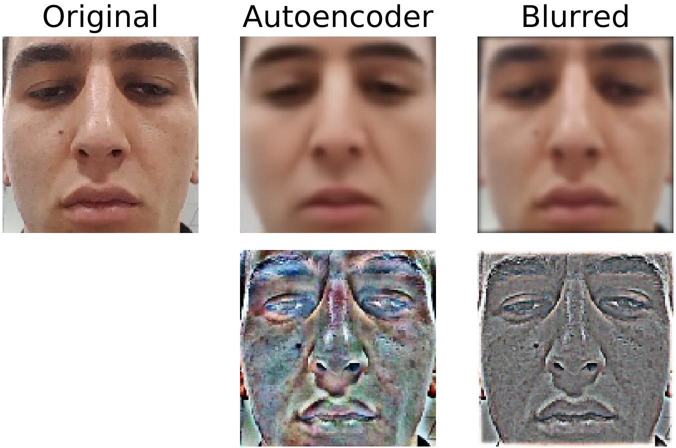
- Autoencoder Error (AE, proposed method) based on DeepPixBiS.



- Blur Error (BE) – Similar to AE but a Gaussian blur filter is used instead of an autoencoder.
  - $\mathbf{I}_{BE} = \mathbf{I} - \mathbf{I}_{blurred}$

[12] A. George and S. Marcel. "Deep Pixel-Wise Binary Supervision for Face Presentation Attack Detection". In: *International Conference on Biometrics*. 2019.

Input images of different systems



Datasets

Experiments are done using 4 recent face PAD datasets

- OULU-NPU[13]
- Replay-Mobile[14]
- SWAN[15]
- WMCA[16]

All PAD methods are trained on OULU-NPU and tested on all datasets.

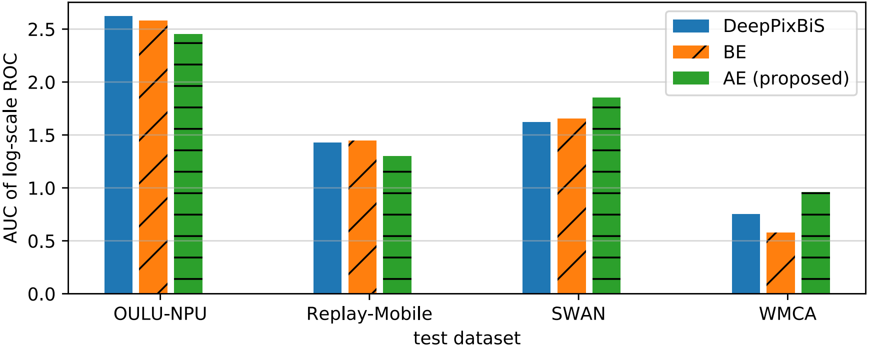
[13] Z. Boulkenafet, J. Komulainen, L. Li, et al. "OULU-NPU: A Mobile Face Presentation Attack Database with Real-World Variations". In: *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference On*. 2017

[14] A. Costa-Pazo et al. "The REPLAY-MOBILE Face Presentation-Attack Database". In: *Biometrics Special Interest Group (BIOSIG), 2016 International Conference of The*. 2016

[15] R. Ramachandra et al. "Smartphone Multi-Modal Biometric Authentication: Database and Evaluation". In: *arXiv:1912.02487 [cs]* (Dec. 2019)

[16] A. George et al. "Biometric Face Presentation Attack Detection with Multi-Channel Convolutional Neural Network". In: *IEEE Transactions on Information Forensics and Security* (2019)

Intra-dataset and Cross-dataset Evaluations



- Area under the curve (AUC) of the ROC plots is reported.
- Comparison of intra-dataset (OULU-NPU) versus cross-dataset (Replay-Mobile, SWAN, WMCA) evaluations.

Conclusions

- All the factors present in face recognition datasets can be seen as nuisance factors for face PAD.
- Autoencoders can be used to explicitly model these nuisance factors.

The proposed method:

- Decreased the intra-dataset performance.
- Increased the cross-dataset performance.

Code and models available at: `https://gitlab.idiap.ch/bob/bob.paper.icassp2020_facepad_generalization_infovae`