

ONLINE ADAPTIVE PERSONALIZATION FOR FACE ANTI-SPOOFING

Davide Belli Debasmit Das Bence Major Fatih Porikli

Qualcomm AI Research*

{dbelli, debadas, bence, fporikli}@qti.qualcomm.com

ABSTRACT

Face authentication systems require a robust anti-spoofing module as they can be deceived by fabricating spoof images of authorized users. Most recent face anti-spoofing methods rely on optimized architectures and training objectives to alleviate the distribution shift between train and test users. However, in real online scenarios, past data from a user contains valuable information that could be used to alleviate the distribution shift. We thus introduce *OAP (Online Adaptive Personalization)*: a lightweight solution which can adapt the model online using unlabeled data. OAP can be applied on top of most anti-spoofing methods without the need to store original biometric images. Through experimental evaluation on the SiW dataset, we show that OAP improves recognition performance of existing methods on both single video setting and continual setting, where spoof videos are interleaved with live ones to simulate spoofing attacks. We also conduct ablation studies to confirm the design choices for our solution.

Index Terms— Face anti-spoofing, personalization, online learning, unsupervised adaptation

1. INTRODUCTION

Face authentication systems are widespread in everyday technology, but they can be easily spoofed by attackers when they have access to face images from the targeted user. Hence, face anti-spoofing models are an integral part of most modern face recognition systems. Face anti-spoofing research has received increased attention in recent years due to the availability of large-scale face image data, improvements in deep learning methods, and the potential for catastrophic data breaches. Nowadays, convolution neural networks [1, 2, 3, 4] are a standard backbone for face anti-spoofing models, with recent work exploring additional modalities [5, 6, 5, 7] or different task formulations like disentanglement [8] and temporal modelling [9]. However, most existing methods do not explicitly account for the distribution shift between training and testing face images.

While large face anti-spoofing datasets [6, 12, 13] have been collected in recent years to enable the development of deep learning solutions, it is infeasible to capture all the variations in the data that might appear at test time. Distribution shift between training and test data naturally occurs due to the presence of new users, sensors, and environmental conditions which were not captured in the training data. To address the distribution shift due to novel users and sensors, one could use live enrollment data from a user to personalize the face anti-spoofing model to that particular user. For example, in [14, 15, 16], the authors use statistics from the enrollment data to calibrate the classification threshold or person-specific coefficients

*Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

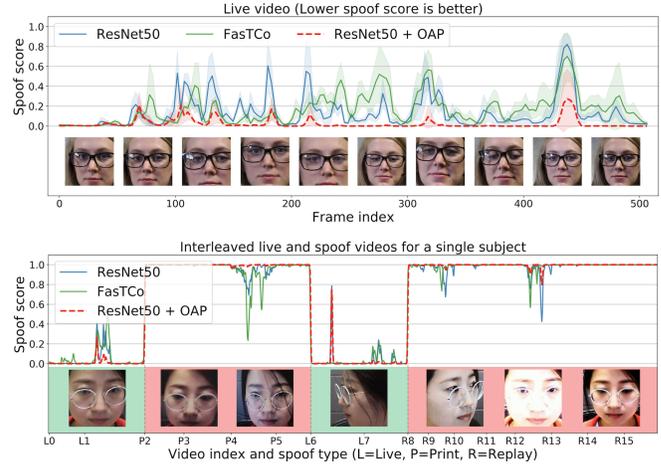


Fig. 1. Top: predictions from ResNet50 [10], FasTCo [11] and our proposed OAP in the single video scenario. **Bottom:** predictions in the continual scenario, where multiple live (in green) and spoof (in red) videos are interleaved and concatenated. In both scenarios the OAP solution adapts over time and outperforms the other methods.

in ensembles. Other recent work [17] proposes to obtain personalized predictions by directly conditioning the anti-spoofing model using enrollment images as examples of live data. However, since the enrollment data is fixed, this type of personalization does not allow continuous adaptation throughout the device’s lifetime.

Different approaches for personalization are instead based on domain adaptation and generalization approaches. For example, Shao et al. [18] propose adversarial learning of domain-invariant representations, while Yang et al. [19] suggest training subject-specific classifiers with synthesized spoofs for each user. Other works [20, 21, 22] consider defining specific training objectives to minimize the distribution shift. These personalized methods are also static, as the trained model does not adapt during test time.

Finally, some recent works investigate the adaptation of anti-spoofing models at test time. Quan et al. [23] propose temporal smoothing of predictions and a progressive pseudo-labeling approach where the thresholds are varied over time. Lv et al. [24] use predictions of model ensembles from different test epochs. However, both of these methods [23, 24] assume that all test data is received at once and that the model can be trained using this data before finally making its prediction. This is different from the real-world setting where test data arrives in an online streaming fashion and the system requires low-latency prediction for every incoming frame. FasTCo [11] proposes an uncertainty-based smoothing method to improve the consistency of predictions over time. While this solution can efficiently run in the online scenario, it does not allow for model adaptation at test time to handle the distribution shift.

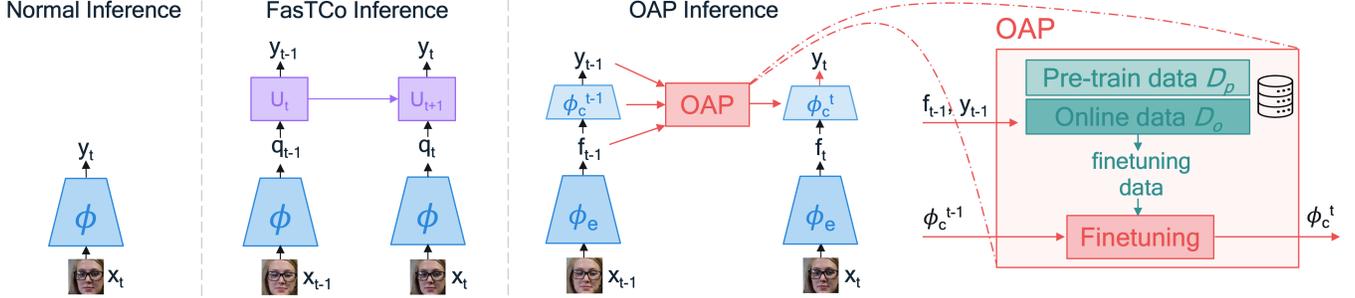


Fig. 2. Left: in standard online face anti-spoofing solutions, the prediction for each frame is independent from previous frames. **Center:** in FastCo [11], an uncertainty module U_t is used to smooth the logits scores \mathbf{q}_t over time. **Right:** in OAP (ours), latent features \mathbf{f}_t and predictions y_t are stored and used to adapt parts of the model ϕ_c^t over time.

To address the drawbacks found in existing approaches, we propose *OAP (Online Adaptive Personalization)*: a method to efficiently adapt the anti-spoofing model to specific users and conditions observed at test time. In particular, we develop a solution to address multiple challenges in the online adaptation, like fine-tuning a model with unlabeled data, preventing catastrophic forgetting, and continuously adapting to evolving scenarios. Our solution performs online inference with low latency and minimal compute and memory requirements. In Fig. 1, we show how the proposed OAP method produces more reliable predictions over time compared to ResNet50 baseline and FastCo in both single video and continual scenarios.

To summarize, our contributions are as follows: (a) We develop a lightweight personalization method for online face anti-spoofing which runs with negligible compute overhead. Our solution is compatible with most other solutions for face ASP, as it only redefines the behavior of the model at inference time, and does not require storing sensitive personal images on-device; (b) We show consistent improvements with respect to recent anti-spoofing solutions in the online Face ASP scenario for all SiW protocols; (c) We evaluate our approach in a realistic continual interaction setting and observe how our method can handle context switches between live user accesses and spoof attacks without catastrophic forgetting.

2. PROPOSED APPROACH

Differently from anti-spoofing via offline video classification, in the online face anti-spoofing scenario [11] the model is required to output a low-latency prediction for each frame in the incoming video stream. The task can be formulated as the sequential classification of the frames \mathbf{x}_t in a video $\mathbf{V} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T]$ of length T , with the label for all frames $l_t \in \{0 = \text{live}, 1 = \text{spoof}\}$ being unknown at test time. Each frame is evaluated only once to predict its spoof probability $y_t = p(l_t = 1 | \mathbf{x}_t)$. Commonly, the probability is modeled through a neural network ϕ as $y_t = \phi(\mathbf{x}_t)$ with parameters optimized on a training dataset and fixed at test time.

2.1. Pre-training

As our solution adapts the model online, it relies on a pre-trained anti-spoofing model as starting point. Different backbones, sources, and training objectives have been proposed in recent literature to optimize anti-spoofing performance. The proposed OAP solution is compatible with most existing models, as it only assumes that latent features can be obtained at some point in the network. In Fig. 2 we show how the proposed methods performs inference online.

2.2. Online Adaptive Personalization

Without loss of generality, we represent the anti-spoofing model ϕ as two consecutive components: the feature extractor ϕ_f and the

classifier ϕ_c , so that $\mathbf{f}_t = \phi_f(\mathbf{x}_t)$ and $y_t = \phi_c(\mathbf{f}_t) = \phi_c(\phi_f(\mathbf{x}_t))$. The features \mathbf{f}_t are low-dimensional latent representations of the input image at time step t . In our implementation, we define ϕ_f as the convolutional backbone and ϕ_c as the final dense layers. During pre-training, both ϕ_f and ϕ_c are updated, while during online evaluation ϕ_f remains fixed and only ϕ_c is fine-tuned. By choosing ϕ_c to be the last few layers in the network, we ensure that the compute time for updating ϕ_c is negligible with respect to time to obtain $\mathbf{f}_t = \phi_f(\mathbf{x}_t)$. In our implementation (see details in Sec. 3.1), less than 0.1% of the compute time during inference is used for OAP. In addition, we only need to store the compact latent features \mathbf{f}_t for fine-tuning instead of the raw full-resolution frames \mathbf{x}_t which contain sensitive biometric information. This design choice allows us to run Online Adaptive Personalization with minimal compute and memory overhead without storing sensitive personal images on the device.

Fine-tuning with unlabeled data The first challenge is how to fine-tune the model online using unlabeled data. Pseudo-labeling allows us to use the model predictions as a proxy for the correct ground-truth labels. A general implementation is: $\hat{l}_t = 1$ if $y_t > \tau$ else 0, with \hat{l}_t being the pseudo-label for frame \mathbf{x}_t and the threshold τ being calibrated during pre-training. We propose, instead, a more expressive formulation using two thresholds:

$$\hat{l}_t = \begin{cases} 1, & \text{if } y_t > \tau_{\text{spoof}} \\ 0, & \text{if } y_t < \tau_{\text{live}} \\ \text{discard}, & \text{otherwise} \end{cases} \quad (1)$$

with $\tau_{\text{spoof}} > \tau_{\text{live}}$ being two separate thresholds for the spoof and live classes. This formulation allows us to discard online samples for which the model predictions are uncertain. We investigate the impact of this formulation through ablation studies in Sec.3.

To further improve the quality of the pseudo-labels, we exploit once more the sequential aspect of the online data. We can reasonably assume that frames appearing in a small temporal window belong to the same class. Accordingly, a simple smoothing through majority with a sliding window is used to improve the pseudo-label consistency for samples currently stored in the online dataset:

$$\hat{l}_t = 1 \text{ if } \left(\frac{1}{W+1} \sum_{r \in [t-\frac{W}{2}, t+\frac{W}{2}]} \hat{l}_r \right) > 0.5 \text{ else } 0. \quad (2)$$

We select a time window $W = 30$ frames (equivalent to 1 second in SiW dataset), as we want our solution to work in scenarios with quick transitions between authentic accesses and spoofing attacks.

During online inference, the classifier is initialized with the pre-trained layers $\phi_c^0 = \phi_c$, and the online data is gradually added to an online dataset D_o which is initially empty: $D_o^0 = \emptyset$. For each frame \mathbf{x}_t in the video stream, its features \mathbf{f}_t and pseudo-label \hat{l}_t are used

to update the current online dataset $D_o^{t+1} = D_o^t \cup \{(\mathbf{f}_t, \hat{l}_t)\}$. Then, ϕ_c^t is fine-tuned using samples from D_o^{t+1} to get the updated model ϕ_c^{t+1} . We implement the fine-tuning by few iterations of mini-batch gradient descent with Cross-Entropy objective:

$$L_{CE}(\hat{l}_i, y_i) = \hat{l}_i \cdot \log y_i + (1 - \hat{l}_i) \cdot \log(1 - y_i), \quad (3)$$

where y_i are the model predictions and \hat{l}_i are the pseudo-labels for sample i in the online batch. Next, we describe the challenges and solutions to effectively run OAP online with limited unlabeled data.

Preventing catastrophic forgetting In a real scenario we might expect a user to rarely or never witness spoofing attacks. In this example, only live samples would be provided to the OAP module, and as a result, the model might forget the existence of spoofs. To avoid the catastrophic forgetting of one class or spoof types observed during pre-training, we regularize the adaptation by storing a small subset of the pre-training data D_p to balance the online data D_o during fine-tuning. We perform weighted sampling to balance the distribution of live, spoof, online and offline samples. Similar to the online samples, only compressed features \mathbf{f} and labels l from the pre-training samples are required for fine-tuning, which allows for minimal overhead in memory requirements.

Adapting to evolving scenarios Finally, we notice that not all past data from the user is needed for the online adaptation. In particular, we would like our anti-spoofing model to be optimized not only for the current user (online personalization) but also for the changing environmental and facial conditions like lighting, pose, and expression (online adaptation). To achieve this, we gradually discard old samples from the online dataset as new frames appear in the online stream. The model will thus gradually adapt to the current conditions. In our implementation, we discard online samples older than 4 seconds, where full videos in the SiW dataset are up to 30 seconds long. Notice that, even if older samples are discarded, their information will partially be retained in the model’s weights thanks to previous OAP iterations.

In Algorithm 1 we summarize the OAP method in pseudo-code.

Algorithm 1 OAP definition and inference over a sample \mathbf{x}_t .

```

1: function OAP( $\mathbf{f}, y, \phi_c, D_o, D_p$ )
2:    $\hat{l} \leftarrow \text{pseudo\_label}(y, \tau_{\text{spoof}}, \tau_{\text{live}})$  ▷ Eq.1
3:   if  $\hat{l} \in \{0, 1\}$  then
4:      $D_o \leftarrow D_o \cup \{(\mathbf{f}, \hat{l})\}$ 
5:      $D_o \leftarrow \text{drop\_old\_samples}(D_o)$ 
6:      $D_o \leftarrow \text{label\_smoothing}(D_o)$  ▷ Eq.2
7:      $\phi_c \leftarrow \text{finetune}(\phi_c, D_p, D_o)$  ▷ Eq.3
8:     return  $\phi_c, D_o$ 
9:
10:  $\mathbf{f}_t \leftarrow \phi_f(\mathbf{x}_t), y_t \leftarrow \phi_c^t(\mathbf{f}_t)$  ▷ inference
11:  $\phi_c^{t+1}, D_o^{t+1} \leftarrow \text{OAP}(\mathbf{f}_t, y_t, \phi_c^t, D_o^t, D_p)$  ▷ OAP

```

3. EXPERIMENTAL RESULTS

We compare the proposed method against recent anti-spoofing solutions in a single video and continual scenarios, and we conduct ablation studies on the main hyper-parameters, analysing their impact on performance, compute and memory requirements.

3.1. Experimental setup

Dataset We evaluate the proposed method on the SiW [6] dataset, which consists of 4620 live and spoof videos from 165 subjects collected under different poses and illumination conditions. The dataset defines 3 protocols to measure different generalization capabilities.

Implementation details To evaluate the efficacy of OAP, we apply it on top of the following baselines: (a) ResNet50 [10] pre-trained on ImageNet [26]; (b) FeatherNet [25], a lightweight architecture for anti-spoofing; (c) CDCN++ [4] which predicts facial depth maps; and (d) FasTCO [11], which uses non-adaptive uncertainty based smoothing for inference. We implement all backbones following the hyper-parameters suggested in the original papers, and we set ϕ_c to be a dense classifier with a single 64-neurons hidden layer. We train all models with a batch size of 128 over 30k iterations. We use Adam [27] optimizer with a weight decay of 1e-3, an initial learning rate of 1e-3 and an exponential decay with $\gamma = 0.8$ every 1000 iterations. For online adaptation, we use batches of size 16 and learning rate 1e-6. We evaluate two variants of the proposed methods: OAP, which we evaluate in the standard online scenario over single videos, and OAP-C, which we evaluate in the continual scenario, where live and spoof videos from the same subject are interleaved to simulate spoofing attacks.

Evaluation metrics We report the following metrics for evaluation: (a) APCER: Attack (spoof) Presentation Classification Error Rate; (b) BPCER: Bonafide (live) Presentation Classification Error Rate; (c) ACER: average of APCER and BPCER errors; (d) EER: error rate at which APCER is equal to BPCER. As the SiW dataset does not include a development set for calibration, we fix the evaluation threshold at 0.5. We also find that using a subset of the training data for threshold calibration yields inaccurate thresholds. While APCER, BPCER, and ACER depend on this threshold, EER allows measuring the discriminative power of the model independently of the evaluation threshold. During online inference, the evaluation of each frame happens before it is potentially employed for fine-tuning.

3.2. Quantitative evaluation

The proposed OAP method can be easily applied on top of existing anti-spoofing backbones. In Table 1, we evaluate OAP in combination and comparison with different backbones. We first observe how the best results across all protocols are obtained with models augmented through OAP, in particular using the ResNet50 backbone.

We also notice how the improvements obtained by applying OAP on top of ResNet50 and FeatherNet are more consistent and significant with respect to CDCN++ and FasTCO counterparts. We believe this is because the pre-training objective is identical for the first two methods to the one used for fine-tuning in OAP. This is not the case for CDCN++ and FasTCO, where multiple sources and different loss components are used during pre-training.

Finally, we verify that the adaptive model evaluated in the continual scenario (ResNet50 w/ OAP-C) achieves comparable results with the single video evaluation, albeit the hyper-parameters were tuned only in the single video scenario. This serves as proof that the OAP solution does not overfit the specific video on which it is fine-tuned, but it can instead continuously adapt and improve over multiple live and spoof videos without catastrophic forgetting. Next, we further analyse the model behavior in the two scenarios.

3.3. Model behavior over videos

In Fig. 1 we visualize the evolution of model predictions over randomly sampled test videos. In the top plot, we consider the single video scenario and report predictions from ResNet50 and FasTCO in comparison with ResNet50 w/ OAP (plotting average and standard deviation over 3 runs). We notice how the OAP solution quickly adapts the model to the current subject and outputs more confident and robust predictions over time in comparison to both baselines.

In the bottom plot, we consider the continual scenario, where live and spoof videos from the same subject are concatenated and

Table 1. Experimental results for SiW Protocols 1, 2, 3 averaged over 3 seeds. For all metrics, lower is better. Underlined implies the best result per backbone while bold implies the best result overall.

	Protocol 1				Protocol 2				Protocol 3			
	ACER	APCER	BPCER	EER	ACER	APCER	BPCER	EER	ACER	APCER	BPCER	EER
RN50 [10]	1.12	0.37	1.98	0.75	0.61 ± 0.51	1.10 ± 1.05	0.12 ± 0.06	0.34 ± 0.25	29.0 ± 12.8	57.9 ± 25.6	0.08 ± 0.08	13.8 ± 9.2
RN50 w/ OAP	0.73	<u>0.25</u>	1.22	0.49	0.35 ± 0.25	0.57 ± 0.55	0.13 ± 0.12	0.18 ± 0.10	22.9 ± 13.5	45.5 ± 27.3	0.34 ± 0.34	<u>13.6 ± 8.9</u>
RN50 w/ OAP-C	0.64	0.55	0.74	0.60	0.48 ± 0.18	0.91 ± 0.39	0.04 ± 0.03	0.30 ± 0.05	<u>22.8 ± 10.6</u>	<u>45.2 ± 21.5</u>	0.41 ± 0.41	17.4 ± 4.7
FeatherNet [25]	1.53	0.42	2.64	0.99	0.57 ± 0.35	0.91 ± 0.76	0.23 ± 0.09	0.36 ± 0.19	31.1 ± 11.4	62.1 ± 22.8	<u>0.10 ± 0.08</u>	14.0 ± 7.1
FeatherNet w/ OAP	<u>1.00</u>	<u>0.29</u>	<u>1.71</u>	<u>0.87</u>	<u>0.42 ± 0.25</u>	<u>0.67 ± 0.58</u>	<u>0.16 ± 0.10</u>	<u>0.24 ± 0.09</u>	<u>24.3 ± 14.2</u>	<u>48.0 ± 28.9</u>	<u>0.59 ± 0.57</u>	<u>13.8 ± 7.7</u>
CDCN++ [4]	<u>3.53</u>	0.38	6.68	<u>2.32</u>	0.84 ± 0.42	1.48 ± 0.82	0.20 ± 0.07	0.61 ± 0.29	40.2 ± 2.8	80.2 ± 5.6	<u>0.12 ± 0.05</u>	25.7 ± 4.3
CDCN++ w/ OAP	3.69	0.09	7.30	2.93	<u>0.45 ± 0.27</u>	<u>0.88 ± 0.53</u>	<u>0.03 ± 0.02</u>	<u>0.24 ± 0.14</u>	<u>28.7 ± 2.2</u>	<u>54.2 ± 4.1</u>	3.05 ± 0.21	<u>22.9 ± 1.3</u>
FasTCo-NA [11]	1.08	0.24	1.93	0.64	<u>0.56 ± 0.52</u>	1.00 ± 1.05	0.12 ± 0.05	<u>0.32 ± 0.27</u>	28.7 ± 13.2	57.3 ± 26.4	0.09 ± 0.08	14.0 ± 9.1
FasTCo [11]	<u>1.05</u>	0.23	<u>1.86</u>	<u>0.62</u>	0.57 ± 0.53	1.03 ± 1.08	<u>0.12 ± 0.05</u>	<u>0.32 ± 0.27</u>	28.7 ± 13.2	57.3 ± 26.5	0.08 ± 0.08	13.8 ± 9.0
FasTCo w/ OAP	4.40	<u>0.20</u>	8.61	1.77	0.85 ± 0.64	1.46 ± 1.30	0.24 ± 0.09	0.51 ± 0.34	<u>21.7 ± 13.0</u>	<u>42.5 ± 26.4</u>	0.88 ± 0.48	<u>12.0 ± 6.3</u>

interleaved. This allows us to simulate a challenging and realistic online scenario, where a user is interacting with the device and, eventually, spoofing attacks take place with multiple source videos from different spoof types. The FasTCo baseline is evaluated in the single video scenario since its uncertainty-based module requires resetting in-between videos from different spoof types. The OAP solution outperforms the baselines by significantly reducing the prediction error and uncertainty in challenging parts of the videos. More importantly, this study confirms that the OAP method is able to personalize and adapt the model to evolving scenarios without catastrophic forgetting of the live or spoof class. We don’t observe any prediction delay in the switch of regime between live and spoof videos.

3.4. Ablation studies and efficiency

In Table 2 we investigate the optimal choices for the main hyper-parameters defining our solution.

Table 2. Ablation study for the main OAP hyper-parameters on SiW Protocol 1. Here ν : fine-tuning frequency; $(\tau_{\text{spoof}}, \tau_{\text{live}})$: pseudo-labeling thresholds; α : online samples probability for weighted sampling; $|D_p|$: number of pre-training samples available during fine-tuning. Best ACER per parameter is highlighted in bold.

ν	1	0.5	0.2	0.05	0.01
ACER	0.73	0.83	0.88	0.92	0.96
KFLOPs/frame	960	480	192	48	9.6
$(\tau_{\text{spoof}}, \tau_{\text{live}})$	(.01, .99)	(.05, .95)	(.1, .9)	(.2, .8)	(.5, .5)
ACER	0.73	0.77	0.87	0.90	1.22
α	1	0.9	0.8	0.6	0.3
ACER	35.55	0.73	0.77	0.83	0.92
$ D_p $	100	500	1000	5000	10000
ACER	1.08	0.82	0.73	0.71	0.72
Memory (MB)	0.8	4	8	40	160

Fine-tuning frequency The fine-tuning frequency ν regulates how often to adapt the model in comparison to the video frame rate. In the case of SiW, we find that fine-tuning the model every 100 frames ($\nu = 0.01$) is enough to improve over the ResNet50 baseline. Running the OAP more frequently allows the model to adapt more quickly to environmental and pose changes throughout the video, further improving the model predictions. Since we only update a small part of the neural network ϕ_c , the compute cost to run OAP after each frame is negligible with respect to the 5 GFLOPs of the

feature extractor ϕ_f , which is required for the prediction. For this reason, we select the highest frequency $\nu = 1$ in all our experiments.

Pseudo-labeling thresholds Different choices for the pseudo-labeling thresholds $(\tau_{\text{spoof}}, \tau_{\text{live}})$ allow tuning the trade-off between the amount of data available for adaptation and the pseudo-label quality. We define the thresholds symmetrically around 0.5 to simplify their formulation. We find the optimal configuration to be $\tau_{\text{spoof}} = 0.01, \tau_{\text{live}} = 0.99$, which suggests pseudo-label quality is more important than sample quantity, at least for the high frame rate in SiW videos. Notice also how the approach based on a single threshold $\tau_{\text{spoof}} = \tau_{\text{live}} = 0.5$, which does not allow for discarding uncertain predictions, results in significantly worse performance.

Pre-training data To conclude, we consider the hyper-parameters regulating the impact of pre-training data. We define α as the probability to select online data points when randomly sampling batches for fine-tuning. Higher values of α allow for faster adaptation. However, when we completely exclude pre-training data by setting $\alpha = 1$, we observe catastrophic forgetting as the model overfits to the only class available in the online video stream. We select a sampling weight of $\alpha = 0.9$ in our implementation.

The number of pre-training samples available to regularize the online adaptation depends on the selected dataset size $|D_p|$. We randomly sub-sample the selected number of frames from the videos in the pre-training dataset. We find that providing enough variations in terms of spoof types, subjects, illumination, and pose conditions results in better OAP performance. Selecting $|D_p| \geq 500$ is enough to achieve this, with larger sizes providing gradually diminishing returns. Since the pre-training samples must be stored in memory, there is a trade-off between error rate and memory requirements. Considering that ResNet50 parameters require around 94MB of memory, we select $|D_p| = 1000$ as it provides a good balance between performance and memory requirements. In comparison, the size of the online dataset never exceeds $|D_o| = 120$ (less than 1MB) since we discard online samples older than 4 seconds.

4. CONCLUSION

In this paper, we proposed a lightweight adaptive method to personalize a pre-trained face anti-spoofing model to videos of a specific user. Our method does not require storing raw original images on the device and supports evaluation in the online anti-spoofing scenario. Empirical results confirm that our method can be applied on top of existing solutions to achieve a drop in error rates in both single video and continual settings. We described our solution in detail and included ablation studies for the main hyper-parameters and efficiency costs to validate our implementation choices.

5. REFERENCES

- [1] Litong Feng, Lai-Man Po, Yuming Li, Xuyuan Xu, Fang Yuan, Terence Chun-Ho Cheung, and Kwok-Wai Cheung, "Integration of image quality and motion cues for face anti-spoofing: A neural network approach," *Journal of Visual Communication and Image Representation*, vol. 38, pp. 451–460, 2016.
- [2] Lei Li, Xiaoyi Feng, Zinelabidine Boulkenafet, Zhaoqiang Xia, Mingming Li, and Abdenour Hadid, "An original face anti-spoofing approach using partial convolutional neural network," in *International Conference on Image Processing Theory, Tools and Applications*, 2016, pp. 1–6.
- [3] Haocheng Feng, Zhibin Hong, Haixiao Yue, Yang Chen, Keyao Wang, Junyu Han, Jingtuo Liu, and Errui Ding, "Learning generalized spoof cues for face anti-spoofing," *ArXiv Pre-print:2005.03922*, 2020.
- [4] Zitong Yu, Chenxu Zhao, Zezheng Wang, Yunxiao Qin, Zhuo Su, Xiaobai Li, Feng Zhou, and Guoying Zhao, "Searching central difference convolutional networks for face anti-spoofing," in *Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5295–5305.
- [5] Taewook Kim, YongHyun Kim, Inhan Kim, and Daijin Kim, "Basn: Enriching feature representation using bipartite auxiliary supervisions for face anti-spoofing," in *International Conference on Computer Vision Workshops*, 2019, pp. 494–503.
- [6] Yaojie Liu, Amin Jourabloo, and Xiaoming Liu, "Learning deep models for face anti-spoofing: Binary or auxiliary supervision," in *Conference on Computer Vision and Pattern Recognition*, 2018, pp. 389–398.
- [7] Si-Qi Liu, Xiangyuan Lan, and Pong C Yuen, "Remote photoplethysmography correspondence feature for 3d mask face presentation attack detection," in *European Conference on Computer Vision*, 2018, pp. 558–573.
- [8] Amin Jourabloo, Yaojie Liu, and Xiaoming Liu, "Face despoofing: Anti-spoofing via noise modeling," in *European Conference on Computer Vision*, 2018, pp. 290–306.
- [9] Xiao Yang, Wenhan Luo, Linchao Bao, Yuan Gao, Dihong Gong, Shibao Zheng, Zhifeng Li, and Wei Liu, "Face anti-spoofing: Model matters, so does data," in *Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3507–3516.
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [11] Xiang Xu, Yuanjun Xiong, and Wei Xia, "On improving temporal consistency for online face liveness detection system," in *International Conference on Computer Vision Workshops*, 2021, pp. 824–833.
- [12] Yuanhan Zhang, ZhenFei Yin, Yidong Li, Guojun Yin, Junjie Yan, Jing Shao, and Ziwei Liu, "Celeba-spoof: Large-scale face anti-spoofing dataset with rich annotations," in *European Conference on Computer Vision*, 2020, pp. 70–85.
- [13] Ajian Liu, Zichang Tan, Jun Wan, Sergio Escalera, Guodong Guo, and Stan Z. Li, "Casia-surf cefa: A benchmark for multimodal cross-ethnicity face anti-spoofing," in *Winter Conference on Applications of Computer Vision*, January 2021, pp. 1179–1187.
- [14] Waldir R Almeida, Fernanda A Andaló, Rafael Padilha, Gabriel Bertocco, William Dias, Ricardo da S Torres, Jacques Wainer, and Anderson Rocha, "Detecting face presentation attacks in mobile devices with a patch-based cnn and a sensor-aware loss function," *PLOS ONE*, vol. 15, no. 9, pp. e0238058, 2020.
- [15] Soroush Fatemifar, Muhammad Awais, Shervin Rahimzadeh Arashloo, and Josef Kittler, "Combining multiple one-class classifiers for anomaly based face spoofing attack detection," in *International Conference on Biometrics*, 2019, pp. 1–7.
- [16] Soroush Fatemifar, Shervin Rahimzadeh Arashloo, Muhammad Awais, and Josef Kittler, "Client-specific anomaly detection for face presentation attack detection," *Pattern Recognition*, vol. 112, pp. 107696, 2021.
- [17] Davide Belli, Debasmitt Das, Bence Major, and Fatih Porikli, "A personalized benchmark for face anti-spoofing," in *Winter Conference on Applications of Computer Vision Workshops*, 2022, pp. 338–348.
- [18] Rui Shao, Xiangyuan Lan, Jiawei Li, and Pong C Yuen, "Multi-adversarial discriminative deep domain generalization for face presentation attack detection," in *Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10023–10031.
- [19] Jianwei Yang, Zhen Lei, Dong Yi, and Stan Z Li, "Person-specific face antispoofing with subject domain adaptation," *Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 797–809, 2015.
- [20] Haoliang Li, Wen Li, Hong Cao, Shiqi Wang, Feiyue Huang, and Alex C Kot, "Unsupervised domain adaptation for face anti-spoofing," *Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1794–1809, 2018.
- [21] Jingjing Wang, Jingyi Zhang, Ying Bian, Youyi Cai, Chunmao Wang, and Shiliang Pu, "Self-domain adaptation for face anti-spoofing," in *Conference on Artificial Intelligence*, 2021, vol. 35, pp. 2746–2754.
- [22] Zhihong Chen, Taiping Yao, Kekai Sheng, Shouhong Ding, Ying Tai, Jilin Li, Feiyue Huang, and Xinyu Jin, "Generalizable representation learning for mixture domain face anti-spoofing," *ArXiv Pre-print:2105.02453*, 2021.
- [23] Ruijie Quan, Yu Wu, Xin Yu, and Yi Yang, "Progressive transfer learning for face anti-spoofing," *Transactions on Image Processing*, vol. 30, pp. 3946–3955, 2021.
- [24] Lingling Lv, Youjun Xiang, Xianfeng Li, Hanye Huang, Rongju Ruan, Xiaoyan Xu, and Yuli Fu, "Combining dynamic image and prediction ensemble for cross-domain face anti-spoofing," in *International Conference on Acoustics, Speech and Signal Processing*, 2021, pp. 2550–2554.
- [25] Peng Zhang, Fuhao Zou, Zhiwen Wu, Nengli Dai, Skarpness Mark, Michael Fu, Juan Zhao, and Kai Li, "Feathernets: convolutional neural networks as light as feather for face anti-spoofing," in *Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 1574–1583.
- [26] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.
- [27] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.