

On Improving Temporal Consistency for Online Face Liveness Detection System

Xiang Xu Yuanjun Xiong Wei Xia
 Amazon/AWS AI
 {xiangx, yuanjx, wxia}@amazon.com

Abstract

In this paper, we focus on improving the online face liveness detection system to enhance the security of the downstream face recognition system. Most of the existing frame-based methods are suffering from the prediction inconsistency across time. To address the issue, a simple yet effective solution based on temporal consistency is proposed. Specifically, in the training stage, to integrate the temporal consistency constraint, a temporal self-supervision loss and a class consistency loss are proposed in addition to the softmax cross-entropy loss. In the deployment stage, a training-free non-parametric uncertainty estimation module is developed to smooth the predictions adaptively. Beyond the common evaluation approach, a video segment-based evaluation is proposed to accommodate more practical scenarios. Extensive experiments demonstrated that our solution is more robust against several presentation attacks in various scenarios, and significantly outperformed the state-of-the-art on multiple public datasets by at least 40% in terms of ACER.

1. Introduction

In this paper, we focus on **online face liveness detection** for common real-world use cases such as face authorization. Unlike offline video analysis [27], which can observe an entire video to make a final prediction, online processing requires low-latency prediction for each incoming frame. In this setting, the most common approach is to predict the liveness probability per frame [17, 18]. However, as depicted in Fig. 1a, such a frame-based model has larger prediction variance within a short period (the standard deviation of predictions is 0.2). By further analyzing the false positives and false negatives, we noticed that it tends to make unstable predictions when the subject undergoes large motion or illumination changes. Therefore, we hypothesize that one common underlying issue for the frame-based liveness detection systems is temporal inconsistency.

To address this issue, a simple yet effective Face anti-spoofing system using Temporal Consistency (FasTCo) is

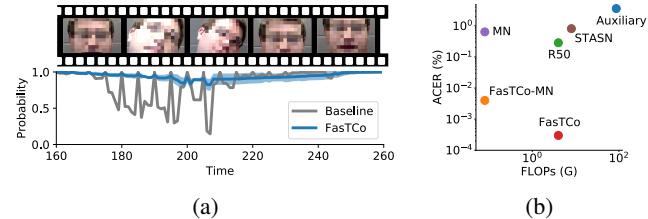


Figure 1: Depiction of (a) temporal inconsistency of the predictions on one video clip from a frame-based baseline model (gray curve) and the predictions with uncertainty estimations from our model - FasTCo (the blue curve represents prediction probability and the shade represents confidence levels, the face is blurred to hide identity); (b) model comparison of previous methods and two variations of FasTCo using ACER (%) on SiW with protocol one and FLOPs (G). Bottom left is the best (Best view in color).

proposed with temporal-aware model training and adaptive model predictions. Specifically, in the training stage, beyond the softmax cross-entropy loss for multi-class classification, to enforce consistency of video sequences in the embedding space, two additional loss functions are proposed to improve the training, aiming to minimize the intra-class embedding distances for video sequences and presentation attacks, respectively. In the deployment stage, based on temporal consistency, a training-free uncertainty estimation module is developed to adaptively update the liveness probability, which results in much more consistent predictions. For instance, as depicted in Fig. 1a, the liveness probabilities predicted by FasTCo on the same video clip have a much lower variation (standard deviation is 0.04) compared with the baseline model. This approach can be considered as a special way to select informative past frames in the online setting. Additionally, it is a generic approach that can deploy a more lightweight backbone (FasTCo-MN) and still achieve better performance than state-of-the-arts, as shown in Fig. 1b. Such lightweight model provides great potential for low-latency applications, especially on edge deployment environment such as mobile phone or IoT devices.

To evaluate the online models, besides the commonly used frame-based evaluation, a video segment evaluation approach is introduced to provide metrics for different ap-

plication scenarios. Extensive experiments including the ablation studies were conducted, and have demonstrated that our method significantly outperforms the state-of-the-art on several publicly available datasets with at least 33% fewer FLOPs. On the SiW [17] dataset, FasTCo obtained an ACER of 3×10^{-6} , almost 0.1% of state-of-the-art under the protocol one, while achieving at least 50% relative improvement using other protocols. Meanwhile, the proposed solution exceeds the state-of-the-arts by 40%+ on OULU-NPU [4], SiW-M [18], and cross-domain datasets. In summary, our method is more robust against multiple factors in practical use cases such as unseen presentation attacks, illumination change, and acquisition devices.

With the temporal consistency, the following benefits can be expected: (i) Simple thresholding: because the prediction of the model is unstable in adjacent continuous frames, it is difficult to determine an appropriate threshold for liveness classification. The system would either have less security if the threshold is too low (APCER \uparrow) or bad user experience from false rejects when the threshold is too high (BPCER \uparrow). However, with temporal consistency, the system outputs a more consistent prediction, leading to a much easier balance of APCER and BPCER in real applications. (ii) Uncertainty estimation: with the proposed uncertainty module, in addition to the liveness score, the system outputs the uncertainty estimation, which can be used to filter out frames with highly uncertain predictions. This greatly enhances the robustness of the system. In summary, the contributions of this paper are: (i) Temporal inconsistency was identified as a common issue of the current face liveness detection systems; (ii) A simple yet effective solution, including two additional losses and a training-free uncertainty estimation module, was proposed to significantly improve the model performance without extra complexity and latency; (iii) In addition to the common frame-based evaluation approach, a video segment-based evaluation was proposed to measure both the latency and accuracy of the model for different application scenarios.

2. Related Work

The common presentation attacks [9] to face recognition systems include using print photos, video replay, and 3D masks. The recent face liveness detection methods to identify these attacks can be classified into two major streams in general: (i) Static approaches: Some image clues from color space and frequency domain [3, 14] were used to detect artifacts. In addition, some human-crafted features such as LBP [3] and the features learned by CNN [1, 19, 33, 22] were extracted to train a binary classifier. Domain generalization and meta-learning techniques [29, 21, 23] have also been used to learn generalized feature representations [16, 15, 22, 26, 13] from multiple domains to improve the generalization of the model. Liu *et al.* [18] developed a deep

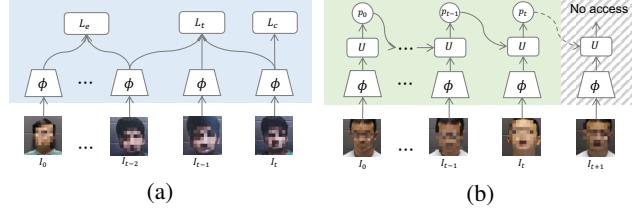


Figure 2: Overview of (a) the training stage: in addition to the multi-class classification loss L_c , we propose a temporal self-supervision (L_t) loss on the features extracted from the same video sequence and a class consistency loss (L_e) to enforce the intra-class distances; (b) the deployment stage: a training-free uncertainty module U is proposed to estimate the uncertainty based on the temporal consistency to smooth the liveness probabilities in the online setting.

tree-structured learning process to learn homogeneous features of presentation attacks in the upper nodes of the tree and distinct features to classify each specific attacks in the leaf nodes. However, such static methods do not consider the relationship across the temporal dimension, and thus lacking the temporal consistency in predictions. (ii) Dynamic methods: The motion of the face, either part or as a whole, was used to predict the liveness. Multiple features extracted from video frames were aggregated and the predictions were fused by Siddiqui *et al.* [25] to generate a liveness score. Similar to the common approach in action recognition [27, 24, 28, 10], both spatial and temporal information [30, 31] of a video clip have been explored to make the final decision based on a CNN-LSTM network [8]. Nevertheless, it is hard to learn the temporal information by jointly learning CNN and LSTM networks. In this work, we improved the model training by introducing new loss functions and inference consistency with an uncertainty estimation module.

3. Temporal Inconsistency

Compared with the sequential models [31], the frame-based model is easier to implement and can be directly used to process online untrimmed videos. To find the root cause of why the model fails in some cases, the ResNet-50, a commonly used network structure serving as the baseline in the literature [31, 33], was used to train a frame-based online liveness detection model with a softmax loss on SiW [17] dataset. The evaluation was performed following the protocol one of the dataset. The videos were ranked according to the number of predictions errors. After analyzing the errors, there are two observations: (i) Time-wise, one common pattern across the false positive and false negative samples is the prediction inconsistency. The predictions are not stable and there are many spikes in the estimated probabilities. By analyzing the frames corresponding to the false predictions,

we made a hypothesis that such sudden prediction change was due to the movement of the subject or the environment changes such as reflection. (ii) Prediction-wise, the probability outputs are either extremely high for false positives or low for false negatives at some time stamps, indicating that the model is over-confident of its predictions on outliers. Based on these observations, to improve the baseline model, we need to answer the following questions: (i) How to use temporal information to improve the training of a single network model? (ii) How to use temporal consistency to increase the robustness of the model inference? Therefore, this paper will focus on solving the temporal inconsistency issue from both training and inference stages.

4. Online Face Liveness Detection System

We first define two key properties for the online face liveness detection system, and then propose two strategies to improve the robustness of the model based on temporal consistency.

4.1. Formulation

Mathematically, a live video can be represented as a sequence of frames $V = \{I_0, \dots, I_t, \dots, I_T\}$, where t is the current time stamp and T is the total number of frames. To ensure the input sequence of faces belongs to the same identity, a face tracker [2] can be deployed instead of a naive face detector to provide the temporal-spatial information for a sequence of face bounding boxes b_t .

Input: The input of the system is a cropped face from the video frame I_i using its bounding box b_i from tracking. From now on, we will use I_i to represent the face region frame for simplicity.

Output: The output of the system is a liveness probability p_t , where a binary decision y (live or attack) can be determined with a threshold.

Model: Usually, a face liveness detection model $\Phi(\cdot)$ consists of a feature extractor $\phi(\cdot)$ and a classifier $C(\cdot)$. The liveness logit q_t can be obtained by forwarding the current face frame I_t to the network denoted as $q_t = C(\phi(I_t)) = C(x_t)$, where x_t is the feature representations of the frame I_t . The liveness probability p_t can be generated by applying a normalization activation function such as softmax or sigmoid to the logit q_t . As an additional constraint due to the online setting, when making a prediction on the frame I_t , the model $\Phi(\cdot)$ can only use the information from $[I_0, I_t]$ but is forbidden to access $[I_{t+1}, I_T]$ (Fig. 2b).

4.2. Temporal Consistency Properties

By deploying the face tracker, the temporal consistency comes with the following two properties:

Property 1 (Identity Consistency) *There is only one subject identity in the input stream.*

Property 2 (Prediction Consistency) *The model should have consistent predictions on the frames within the same video tracklet.*

4.3. Improving Consistency

Applying the properties of temporal consistency, the following loss functions are used to train the network in an end-to-end manner (Fig. 2a) and a uncertainty module is proposed to keep prediction consistency in the deployment stage (Fig. 2b):

Classification Supervision: Unlike the previous methods [17, 14, 31, 18] that trained a binary classifier, the labels of video types (e.g., print, video replay) can be used as the supervision to train a multi-class classifier using a softmax cross-entropy loss:

$$L_c = -\frac{1}{m} \sum_{i=0}^m \log p_{y_i}, \quad (1)$$

where m is the batch size. The benefits of converting a binary classification problem into a multi-class classification setting are in three-folds: (i) The discriminative features to distinguish different types of presentation attacks can be learned; (ii) The embedding space of liveness class can be squeezed into a more compact space than using a binary classification, which helps decrease the false positive rate; (iii) Fine-grained analysis can be conducted when the model makes mistakes.

Temporal Consistency Self-Supervision: To keep temporal consistency across multiple frames (e.g., frames within the same video tracklet), a self-supervision loss is proposed to regularize the intra-video consistency in the embedding space, denoted as:

$$L_t = \frac{1}{m} \sum_{i=0}^m \max_{i,j \in v} \|x_i - x_j\|_2^2, \quad (2)$$

where x_i and x_j are the feature representations of two frames from the same video clip v within the batch.

Class Consistency Loss: Similar to the temporal consistency self-supervision, the embedding learned from the same class but from different videos should be as close as possible. Therefore, a class consistency loss function can be defined as follows:

$$L_e = \frac{1}{m} \sum_{i=0}^m \max_{i,j} y_{ij} \|x_i - x_j\|_2^2, \quad (3)$$

where y_{ij} is equal to 1 when x_i and x_j belong to the same class within the batch, otherwise y_{ij} is 0. In the end, the final loss can be formulated as:

$$L = L_c + \beta L_t + \gamma L_e. \quad (4)$$

4.3.1 Filtering with Uncertainty Estimation

During the deployment stage, to keep the prediction consistent within the same tracklet, a simple yet effective solution is proposed to estimate the model uncertainty and smooth the model predictions adaptively.

Due to the online setting, the uncertainty module can only observe the historical logit outputs $\{q_0, \dots, q_t\}$ from the model. Based on the temporal consistency, we can assume that: (i) The random variable of the liveness score π_t at the time step t follows a Gaussian distribution $\mathcal{N}(\hat{\mu}_t, \hat{\delta}_t^2)$, where $\hat{\mu}_t$ and $\hat{\delta}_t$ denote the moving average and standard deviation of π ; (ii) The single logit observation q_t follows another Gaussian distribution $\mathcal{N}(\pi_t, \delta^2)$. According to the Bayesian rule, the posterior can be written:

$$\begin{aligned} p(\pi_t | \pi_0, \dots, \pi_{t-1}; q_t) &= \frac{p(\pi_0, \dots, \pi_{t-1}; q_t | \pi_t) \cdot p(\pi_t)}{p(\pi_0, \dots, \pi_{t-1}; q_t)} \\ &= \alpha \cdot p(q_t | \pi_t) \cdot p(\pi_t | \pi_0, \dots, \pi_{t-1}), \end{aligned} \quad (5)$$

where α is a normalization constant. Based on the assumptions of temporal consistency, the Equ. (5) can be derived into the following equation:

$$\frac{(\pi_t - \hat{\mu}_t)^2}{2\hat{\delta}_t^2} = \frac{(\pi_t - q_t)^2}{2\delta_t^2} + \frac{(\pi_t - \hat{\mu}_{t-1})^2}{2\hat{\delta}_{t-1}^2}. \quad (6)$$

Therefore, the best estimate of the current liveness $\hat{\mu}_t$ and its uncertainty $\hat{\delta}_t$ can be derived as:

$$\begin{aligned} \hat{\mu}_t &= \frac{\hat{\delta}_{t-1}^2 q_t + \delta_t^2 \hat{\mu}_{t-1}}{\delta_t^2 + \hat{\delta}_{t-1}^2} = \frac{\hat{\delta}_{t-1}^2}{\delta_t^2 + \hat{\delta}_{t-1}^2} q_t + \frac{\delta_t^2}{\delta_t^2 + \hat{\delta}_{t-1}^2} \hat{\mu}_{t-1}, \\ \hat{\delta}_t &= \frac{\delta_t^2 \hat{\delta}_{t-1}^2}{\delta_t^2 + \hat{\delta}_{t-1}^2}. \end{aligned} \quad (7)$$

Since $\hat{\mu}_{t-1}$ can be considered as the best measurement of the current state, we can compute $\delta_t^2 = (q_t - \hat{\mu}_{t-1})^2$. To adapt the probability with the uncertainty estimation, the current moving average $\hat{\mu}_t$ is used as the updated probability and the $\hat{\delta}_t$ as the estimated uncertainty. In the end, an activation function such as sigmoid can be applied to normalize the liveness estimation.

In practice, we can only keep recent liveness predictions to compute the moving average $\hat{\mu}_{t-1}$ and the standard deviation $\hat{\delta}_{t-1}$ as a relaxation. Then, the whole inference process in the deployment stage is depicted in Alg. 1. Interestingly, if we assume that $\theta = \hat{\delta}_{t-1}^2 / (\delta_t^2 + \hat{\delta}_{t-1}^2)$ is a constant value and discard the uncertainty, the Eq. (7) would degrade to the Exponential Moving Average (EMA) [6], which is a common technique used in the finance domain.

Algorithm 1: Training-Free Uncertainty Module

Input : Current video stream I_t , window size w
Output: Calibrated probability $\hat{\mu}_t$ and estimated uncertainty $\hat{\delta}_t$

- 1 Obtain current liveness logit q_t ;
- 2 Compute $\hat{\mu}_{t-1}$ and $\hat{\delta}_{t-1}$ with a window size w ;
- 3 Compute the weight $\theta = \frac{\hat{\delta}_{t-1}^2}{\delta_t^2 + \hat{\delta}_{t-1}^2}$;
- 4 Compute $\hat{\mu}_t = \theta q_t + (1 - \theta) \hat{\mu}_{t-1}$ and $\hat{\delta}_t = \theta \cdot \delta_t^2$;

5. Experiments

5.1. Experiments Settings

Datasets: Several public datasets were used to benchmark the proposed method: (i) SiW [17] dataset consists of 165 subjects with 4,620 videos in total to evaluate the robustness of the model with various poses, data sources, and unknown attacks. (ii) OULU-NLP [4] dataset contains 4,950 real and attack videos, recorded using six different phone cameras. (iii) SiW-M [18] dataset consists of 493 subjects with up to 13 types of spoofing attacks. The protocols in this dataset were designed for open scenario evaluation. It adopts a leave-one-out setting, using twelve “attack” videos as the training set and the remaining one as the testing set.

Baselines: Several baselines were implemented: (i) R50: This model was trained using the ResNet-50 network [11] to extract the features of each frame. It served as a baseline of the frame-based face liveness detection system. (ii) R50-LSTM: This model was trained using LSTM to learn the temporal information with the features generated by R50. (iii) STASN [31]: This model explored both spatial and temporal information to make a final prediction. The original paper only reported ACER, so we implemented our own version (denoted as STASN*), which obtained slightly better performance than the original work.

Model Variants: FasTCo was developed on R50 network by default. Another lightweight extension using MobileNetv2 [20] backbone with a growth rate of 0.5 was also developed, denoted by FasTCo-MN, to show the potential of low-power deployment like edge devices.

Evaluation Metrics: In addition to some widely used metrics (e.g., APCER, BPCER, ACER, and ROC) suggested by ISO [12] and Zhang *et al.* [33] that measures on the frame level, we proposed to report the evaluation metrics based on the video segment level. The videos were divided by the sequence of video segment with length K , and each video segment is treated independently, where the previous mentioned metrics (e.g., ACER and ROC) can be applied. For 30 FPS video, we suggest the maximum latency K of less than 30 (1 second) to have a good user experience. Such evaluation approach have the following advan-

Table 1: Comparison of different model implementations.

Method	Spatial	Temporal	Uncertainty	ACER (%)
R50	✓			0.2849
R50-SMA	✓	✓		0.0927
R50-LSTM	✓	✓		0.0794
FasTCo-NA	✓	✓		0.0632
FasTCo-EMA	✓	✓		0.0028
FasTCo	✓	✓	✓	0.0003

Table 2: Ablation study on the hyper-parameters of losses.

Hp	Binary	Multi-class						
		0	1	0	1	1	1	5
β	-	0	0	0.5	0.1	0.5	1.0	0.1
ACER (%)	3.3	2.5	2.1	1.7	1.3	0.8	2.5	1.7

ages: (i) Compare to frame-based evaluation, it provides two-dimensional metrics, which fit better to the practical scenario that cares more about the performance with a specific latency (video segment length); (ii) It allows the use of temporal information to some extent, which could provide a fair comparison with video segment-based models (e.g., 3D Convolutions) in the future.

For all the experiments, to fairly compare with the previous methods and the baseline models, we strictly followed the evaluation protocols provided in each dataset. In the cross-domain experiments, in addition to the frame-based metrics, we also compared the model performance with state-of-the-art on the video-segment level.

5.2. Ablation Study

Module: The SiW dataset with protocol one was used to evaluate the effectiveness of various components in the proposed framework. To fully evaluate the components, different model implementations were configured as follows: (i) R50-SMA: a simple moving average with a window size of five was used to smooth the R50’s predictions; (ii) FasTCo-NA: a R50 model trained with the proposed temporal consistency loss functions; (iii) FasTCo-EMA: an EMA with a smoothing factor of 0.1 and a window size of five was used to smooth the predictions of the model FasTCo-NA.

Table 1 presents the experimental comparisons on the baselines and proposed modules: (i) Comparing R50 with R50-SMA, a simple moving average can help smooth the predictions and reduce the ACER by 3 times; (ii) Comparing R50 with R50-LSTM, the temporal information encoded in LSTM does improve the accuracy by 4 times; (iii) Comparing FasTCo-NA with R50 and R50-LSTM, it can be observed that, rather than using more complex LSTM, the temporal consistency introduced by the proposed loss functions further increases the accuracy of the model; (iv) The uncertainty module does help to improve the robustness of the predictions. Even compared with the

Table 3: Different pre-trained weights as initializations.

Datasets	APCER (%)	BPCER (%)	ACER (%)
NA	2.49	3.13	2.81
VGGFace-2	0.76	0.64	0.70
ImageNet	0.30	0.26	0.28

Table 4: Runtime comparison with the recent methods.

Metrics	STASN [31]	DTN [18]	FasTCo	FasTCo-MN
FLOPs (G)	8*	6	4	0.08
Size (M)	208	-	90	2.8
Time (ms)	11.8	-	3.8	0.9

EMA, it still achieved much better performance. If comparing the full model with the R50 model, FasTCo improved ACER by 1000 times approximately.

Weights of the Losses: OULU-NPU dataset with the protocol one was used to evaluate the hyper-parameters to weight the different loss functions. The results are summarized in the Table 2: (i) Compared with the binary classification, the multi-class training ($\beta = 0, \gamma = 0$) reduced the error rate from 3.3% to 2.5%. (ii) To compare the two proposed loss functions, ACER would decrease to 2.1 if the temporal consistency loss L_t ($\beta = 1, \gamma = 0$) was applied. By applying the class consistency loss L_e , the ACER would further drop to 1.7%, showing that L_e had a larger impact than L_t . (iii) To balance the loss of the three losses (the weight for multi-classification loss is 1), the final ACER could be further reduced from 2.5 to 0.8 when $\beta = 1$ and $\gamma = 0.5$, demonstrating the necessity and effectiveness of both loss functions.

Pre-trained Models: To analyze the impact of using different pre-trained weights as initialization of the backbone R50 model, we tried three different settings: (i) Random initialized weights; (ii) Initialization using the pre-trained R50 weights trained on VGGFace-2 dataset. We used cropped face images from the VGGFace-2 dataset [5] to train a R50 model for face recognition task, whose weights of the convolutional layers were then used to initialize and fine-tune the new model for the face liveliness detection task; (iii) Initialization using the pre-trained R50 weights trained on ImageNet [7]. Evaluation results on the SiW dataset are depicted in Table 3. Surprisingly, the model initialized from the ImageNet dataset achieved a better performance than the model trained from the VGGFace-2 dataset. One possible reason is that the ImageNet dataset provides a wider distribution of the features, which can better capture the clue of presentation attacks rather than a face recognition dataset. Though all models are based on the R50 architecture, we believe this conclusion is generalizable to other model architectures as well.

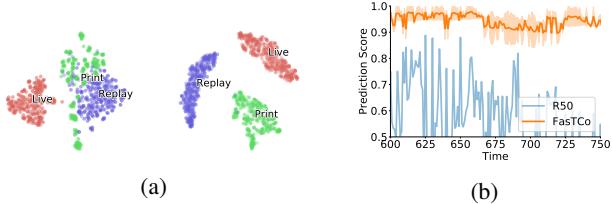


Figure 3: Depiction of (a) 2D t-SNE visualization of the representations generated by R50 (Left) and FasTCo (Right) on the SiW dataset; and (b) one sample video liveness predictions across time from R50 and FasTCo models.

5.3. Comparison with the State-of-the-Arts

Runtime Complexity: Inference efficiency is very critical to the low-latency online applications. The total number of floating-point operations (FLOPs) of the model was used to measure the runtime complexity. Note that “*” denotes the estimated FLOPs based on our implementation because this information cannot be found in the literature. The lower the FLOPs are, the fewer operations are performed and thus the faster of the inference speed. As summarized in Table 4, our model has the least operations, demonstrating better runtime efficiency (FasTCo only takes 3.8 ms to infer on a single frame, which is approximately three times faster than STASN [31]). If switching to a lightweight backbone, like FasTCo-MN, FLOPs and inference time reduced dramatically, showing great potential for low-latency low-power applications.

In-the-Wild Scenario: The SiW dataset [17] was used to evaluate the face liveness detection system in the presence of variances of subject pose, environment illumination, and unseen presentation attacks. The comparison with the current state-of-the-art methods on this benchmark is summarized in Table 5: (i) The baseline model (R50) initialized with ImageNet pre-trained weights, without any additional data, has already achieved comparable performance to the current state-of-the-art, STASN⁺, which used additional synthetic augmented data during training. One possible reason is that it is easier for the optimizer to find a better local minimum when using this single CNN network rather than training CNN and RNN jointly. (ii) FasTCo achieved significantly better performance on protocol one, even using the lightweight backbone. One possible explanation is as follows: the subjects in the training set are frontal faces only, leading to slight overfitting to frontal faces for the trained model. However, the subjects in the test videos have more pose changes. Due to the temporal consistency introduced in our uncertainty module, the large variation on the predictions due to the pose change was extremely suppressed, which results in predictions of higher confidence. (iii) FasTCo outperformed the state-of-the-art methods by at least 65% using the last two protocols. Even in protocol three for the open-set scenario, our method achieved better performance, demonstrating the effectiveness of the

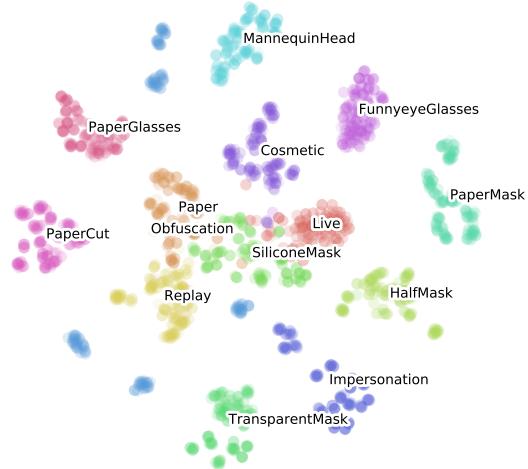


Figure 4: 2D t-SNE Visualization of the representations of FasTCo in the open-set liveness detection scenario (Best view in color and zoom in).

proposed model for liveness detection in the wild. To visually understand the learned model, the feature representations generated by our baseline R50 and FasTCo on the testing set are plotted in Fig. 3a. Compared with the baseline model, the representations of each class produced by our method are more compactly clustered and clearly separated, which could possibly explain the better classification performance. As a qualitative comparison on the prediction scores with the baseline depicted in Fig. 3b, our proposed loss functions improved the temporal consistency compared with the baseline while the uncertainty estimation module can further improve the prediction quality during inference.

Mobile Scenario: The comparison with the state-of-the-art on the OULU-NPU dataset [4] is depicted in Table 6. Similarly, our single network method outperformed the state-of-the-art on this benchmark on three out of four protocols. Note that STASN⁺ consisted of multiple networks (R50+LSTM for extracting temporal information and R50 for local spatial information). Besides it was trained with additional synthetic data, while FasTCo only used the provided training set. However, it obtained comparable performance using protocol one and two and achieved at least 40% improvement using protocol three and four, indicating more robustness to acquisition device changes, unseen illumination conditions, and unseen presentation attacks.

Open-world Scenario: The SiW-M dataset [18] was used to evaluate the performance of the model when it encounters unseen presentation attacks in the open-world scenario. Two state-of-the-art methods [17, 18] were reported on this dataset. Due to the lack of validation set to choose a threshold in this zero-shot scenario, a high threshold of 0.99 was set to reduce the false positive alarms. The comparison with the recent methods, depicted in Table 7, demon-

Table 5: Comparisons on the SiW dataset with three protocols (Pr.). The best results are marked in gray.

Pr.	Metrics (%)	TD-SF-CS [33]	STASN* [31]	STASN ⁺ [31]	R50	R50-LSTM	FasTCo	FasTCo-MN
1	APCER	1.27	0.72	-	0.30	0.13	0.06×10^{-2}	0.70×10^{-2}
	BPCER	0.33	0.89	-	0.26	0.02	0.00×10^{-2}	0.15×10^{-2}
	ACER	0.80	0.81	0.30	0.28	0.08	0.03×10^{-2}	0.43×10^{-2}
2	APCER	0.08 ± 0.17	0.29 ± 0.16	-	0.08 ± 0.05	0.03 ± 0.02	0.02 ± 0.02	0.02 ± 0.02
	BPCER	0.25 ± 0.22	0.27 ± 0.14	-	0.07 ± 0.03	0.03 ± 0.03	0.00 ± 0.00	0.01 ± 0.01
	ACER	0.17 ± 0.16	0.28 ± 0.15	0.15 ± 0.05	0.03 ± 0.03	0.03 ± 0.03	0.01 ± 0.01	0.01 ± 0.02
3	APCER	6.27 ± 4.36	11.05 ± 3.30	-	9.18 ± 4.32	4.45 ± 0.51	2.73 ± 0.91	3.36 ± 1.94
	BPCER	6.43 ± 4.42	7.74 ± 3.08	-	8.41 ± 0.94	3.61 ± 0.67	1.28 ± 0.21	5.00 ± 0.36
	ACER	6.35 ± 4.39	9.39 ± 3.19	5.85 ± 0.85	8.80 ± 2.62	4.03 ± 0.08	2.00 ± 0.56	4.18 ± 1.15

Table 6: Comparison on the OULU-NPU dataset with four protocols (Pr.). The best performance is marked in gray.

Pr.	Method	APCER (%)	BPCER (%)	ACER (%)	Pr.	Method	APCER (%)	BPCER (%)	ACER (%)
1	Auxiliary [17]	1.6	1.6	1.6	3	Auxiliary [17]	2.7 ± 1.3	3.1 ± 1.7	2.9 ± 1.5
	De-Spoofing [14]	1.2	1.7	1.5		De-Spoofing [14]	4.0 ± 1.8	3.8 ± 1.2	3.6 ± 1.5
	STASN ⁺ [31]	1.2	0.8	1.0		STASN ⁺ [31]	1.4 ± 1.4	3.6 ± 4.6	2.5 ± 2.2
	CDCN++ [32]	0.4	0.0	0.2		CDCN++ [32]	1.7 ± 1.5	2.0 ± 1.2	1.8 ± 0.7
	R50	2.3	4.7	3.5		R50	3.4 ± 3.0	0.7 ± 1.0	2.0 ± 1.9
	R50-LSTM	3.3	0.8	2.1		R50-LSTM	4.7 ± 1.4	2.6 ± 4.2	3.7 ± 2.7
	FasTCo	0.8	0.8	0.8		FasTCo	1.2 ± 1.3	1.0 ± 1.0	1.1 ± 0.8
	Auxiliary [17]	2.7	2.7	2.7		Auxiliary [17]	9.3 ± 5.6	10.4 ± 6.0	9.5 ± 6.0
2	De-Spoofing [14]	4.2	4.4	4.3		De-Spoofing [14]	5.1 ± 6.3	6.1 ± 5.1	5.6 ± 5.7
	STASN ⁺ [31]	1.4	0.8	1.1		STASN ⁺ [31]	0.9 ± 1.8	4.2 ± 5.3	2.6 ± 2.8
	CDCN++ [32]	1.8	0.8	1.3		CDCN++ [32]	4.2 ± 3.4	5.8 ± 4.9	5.0 ± 2.9
	R50	2.0	1.1	1.6		R50	5.1 ± 3.9	4.1 ± 2.4	4.6 ± 2.1
	R50-LSTM	3.4	1.3	2.3		R50-LSTM	8.9 ± 7.6	4.6 ± 3.7	6.7 ± 3.6
	FasTCo	1.0	1.3	1.1		FasTCo	1.0 ± 2.0	2.0 ± 4.1	1.5 ± 1.2

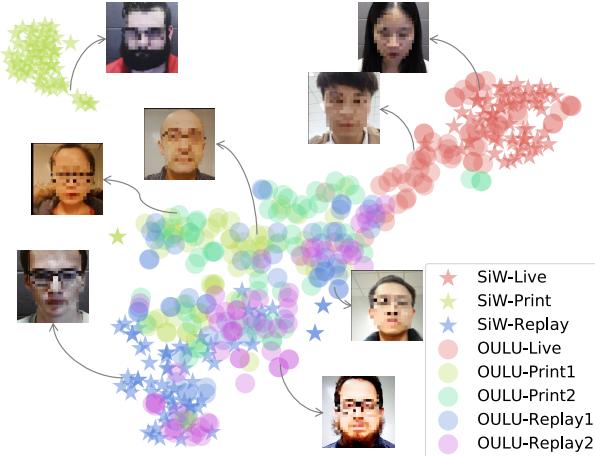


Figure 5: 2D t-SNE visualization of the feature representations generated from SiW and OULU-NPU datasets by FasTCo trained on SiW dataset only.

strated that our method outperformed the previous methods by at least 40% in terms of APCER, ACER, and EER. Diving deep into the details of the unseen attack scenarios, the following observations can be summarized: (i) In general, our method performed well on detecting paper mask,

mannequin head, impersonation, and partial paper or paper cut attacks. (ii) Compared with DTN [18], FasTCo obtained a worse result on predicting video replay attacks and achieved comparable performance on detecting print attacks. (iii) Compared with DTN [18], a significant improvement was achieved by FasTCo in detecting various masks, makeup (especially obfuscation attack), and partial occlusion attacks. It reveals that our uncertainty estimation module using temporal consistency also works well on detecting most of the unseen presentation attacks.

To visually understand the performance in seen and unseen attack scenarios, we separated out the video with silicone attacks, a hard case to 2D face liveness detection system, as the unseen presentation attacks. Then, a randomly selected 80% of the other videos were selected as the training set and the rest was used as the seen attacks. We re-trained the model and depicted the representations generated from the testing set in Fig. 4: (i) Most samples belong to the same attack types are clustered, even for the unseen silicone mask attack samples. It demonstrates the generalization of our model to this unseen attacks. (ii) There is a small overlap between silicone mask samples and live samples, which explains why this attack is more difficult to de-

Table 7: Comparison on SiW-M dataset with open-set evaluation protocols. The best overall performance is marked in gray.

Method	Metrics (%)	Replay	Print	Mask Attacks					Makeup Attacks			Partial Attacks			Average
				Half	Sili.	Trans.	Paper	Manne.	Obf.	Imp.	Cos.	F. Eye	P. Glass	P. Paper	
Auxiliary [17]	APCER	23.7	7.3	27.7	18.2	97.8	8.3	16.2	100.0	18.0	16.3	91.8	72.2	0.4	38.3 \pm 37.4
	BPCER	10.1	6.5	10.9	11.6	6.2	7.8	9.3	11.6	9.3	7.1	6.2	8.8	10.3	8.9 \pm 2.0
	ACER	16.8	6.9	19.3	14.9	52.1	8.0	12.8	55.8	13.7	11.7	49.0	40.5	5.3	23.6 \pm 18.5
	EER	14.0	4.3	11.6	12.4	24.6	7.8	10.0	72.3	10.1	9.4	21.4	18.6	4.0	17.0 \pm 17.7
DTN [18]	APCER	1.0	0.0	0.7	24.5	58.6	0.5	3.8	73.2	13.2	12.4	17.0	17.0	0.2	17.1 \pm 23.2
	BPCER	18.6	11.9	29.3	12.8	13.4	8.5	23.0	11.5	9.6	16.0	21.5	22.6	16.8	16.6 \pm 6.2
	ACER	9.8	6.0	15.0	18.7	36.0	4.5	7.7	48.1	11.4	14.2	19.3	19.8	8.5	16.8 \pm 11.1
	EER	10.0	2.1	14.4	18.6	26.5	5.7	9.6	50.2	10.1	13.2	19.8	20.5	8.8	16.1 \pm 12.2
FasTCo	APCER	1.7	0.0	2.8	9.9	3.4	0.0	0.0	22.9	0.0	10.1	40.9	12.0	0.0	8.0 \pm 12.0
	BPCER	20.6	12.0	10.2	9.4	20.7	9.2	14.6	10.8	8.1	9.9	8.5	9.5	14.8	12.2 \pm 4.3
	ACER	11.2	6.0	6.5	9.7	12.1	4.6	7.3	16.8	4.0	10.0	24.7	10.7	7.4	10.1 \pm 5.6
	EER	7.1	5.6	7.7	9.8	14.8	0.0	2.3	14.0	1.0	10.0	14.9	10.3	1.8	7.6 \pm 5.3

Table 8: Comparison with the state-of-the-art methods in a cross-domain setting (Best results are marked in gray).

Method	APCER (%)	BPCER (%)	ACER (%)	EER (%)	FNR(%)@FPR=				
					1E-5	1E-4	1E-3	1E-2	1E-1
Auxiliary [17]	26.82	14.17	20.50	17.07	98.07	96.91	94.70	79.26	34.45
STASN* [31]	13.24	5.47	9.35	5.42	67.13	62.46	52.22	20.67	3.13
R50	8.32	4.48	6.40	4.28	84.38	81.93	77.15	15.60	1.64
FasTCo	5.14	2.44	3.79	2.57	49.69	46.25	44.68	12.94	0.10

tect. (iii) The features from obfuscation are more scattered, indicating that the network suffers from learning a unique representation for this attack.

Cross-domain Scenario: To verify the generalization of the model, the following experiment was designed: Because both the SiW and OULU-NPU datasets contain the print and video attacks yet there is a large domain gap between these two, the SiW dataset was selected as the training set and the OULU-NPU dataset was used as the testing set. The state-of-the-art models such as Auxiliary [17] and STASN [31] and R50 model were selected as the baselines in this experiment. The model of Auxiliary generously provided by the authors was directly used to test its performance. Table 8 shows that FasTCo achieved significantly better performance (40%+ in ACER) compared with all baselines across all metrics. The video-segment level evaluation was performed and the results are depicted in Table 9. FasTCo consistently outperformed all baselines across different video segment lengths. Considering both model performance and latency, the video segment length between 5 to 15 were highly recommended in practice. Figure 5 depicts the t-SNE visualization of feature representations extracted from two different domains. (i) The model correctly learned the liveness features directly from videos since the features from two different domains were highly clustered. (ii) The feature representations from SiW-Print were isolated to the other features while the features from SiW-Replay were cluttered with the attack features from the OULU-NPU dataset, which indicates that the clue to distin-

Table 9: Comparison with the state-of-the-art in a cross-domain setting using the proposed segment level evaluation (Best results are marked in gray).

Method	Metrics (%)	video segment length $K =$				
		1	3	5	10	15
STASN* [31]	ACER	9.35	9.11	8.87	8.74	9.17
	FNR@FPR=1E-2	20.67	20.86	19.50	17.53	18.77
R50	ACER	6.40	6.08	5.98	5.87	5.61
	FNR@FPR=1E-2	15.60	16.70	16.65	15.60	16.21
FasTCo	ACER	3.96	3.92	3.87	3.77	3.70
	FNR@FPR=1E-2	9.72	8.94	8.55	8.60	8.28

guish the presentation attacks from the OULU-NPU dataset was mostly learned from Replay attacks in the SiW dataset. In summary, FasTCo has better generalization for cross-domain applications than current state-of-the-arts, and the actual performance will be even better if the target domain has higher overlap with the source domain.

6. Conclusion

In this paper, the temporal inconsistency was identified as a common underlying issue that undermines the model performance in the face liveness detection task. To address this issue, in addition to the classification loss, temporal consistency and class consistency losses were proposed for training the model. Moreover, a training-free uncertainty estimation module was developed to update the prediction adaptively in a smooth manner. Extensive experiments have demonstrated that, by applying two proposed strategies based on temporal consistency, the model outperformed the current state-of-the-art by a significant margin.

References

[1] Yousef Atoum, Yaojie Liu, Amin Jourabloo, and Xiaoming Liu. Face anti-spoofing using patch and depth-based CNNs. In *IEEE International Joint Conference on Biometrics*, pages 319–328, Denver, CO, Oct. 1- 4 2017. [2](#)

[2] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *Proc. IEEE International Conference on Computer Vision*, Oct. 27–Nov.2 2019. [3](#)

[3] Zinelabidine Boulkenafet, Jukka Komulainen, and Abdenour Hadid. Face Spoofing Detection Using Colour Texture Analysis. *IEEE Transactions on Information Forensics and Security*, 11(8):1818–1830, 2016. [2](#)

[4] Zinelabidine Boulkenafet, Jukka Komulainen, Lei Li, Xiaoyi Feng, and Abdenour Hadid. OULU-NPU: A mobile face presentation attack database with real-world variations. In *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, pages 612–618, Washington, DC, May 30-Jun. 3 2017. [2, 4, 6](#)

[5] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. VGGFace2: A dataset for recognising faces across pose and age. In *Proc. IEEE Conference on Automatic Face and Gesture Recognition*, Xi'an, China, May 15–19 2018. [5](#)

[6] Wikipedia Contributors. Moving average: https://en.wikipedia.org/wiki/Moving_average, 2019. [4](#)

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, Miami Beach, FL, Jun. 20–25 2009. [5](#)

[8] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015. [2](#)

[9] Javier Galbally, Sébastien Marcel, and Julian Fierrez. Biometric antispoofing methods: A survey in face recognition. *IEEE Access*, 2:1530–1552, 2015. [2](#)

[10] Mingfei Gao, Mingze Xu, Larry S. Davis, Richard Socher, and Caiming Xiong. StartNet: Online Detection of Action Start in Untrimmed Videos. In *Proc. IEEE International Conference on Computer Vision*, pages 5542–5551, Seoul, Korea, Oct. 27–Nov. 2 2019. [2](#)

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, Las Vegas, NV, jun 2016. [4](#)

[12] International Organization for Standardization. Information technology - Biometric presentation attack detection - Part 3: Testing and reporting. Technical Report ISO/IEC FDIS 30107-3:2017(E), Geneva, CH, 2017. [4](#)

[13] Yunpei Jia, Jie Zhang, Shiguang Shan, and Xilin Chen. Single-Side Domain Generalization for Face Anti-Spoofing. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Seattle, WA, Jun. 14-19 2020. [2](#)

[14] Amin Jourabloo, Yaojie Liu, and Xiaoming Liu. Face de-spoofing: Anti-spoofing via noise modeling. In *European Conference on Computer Vision*, pages 297–315, Munich, Germany, sep 2018. [2, 3, 7](#)

[15] Haoliang Li, Peisong He, Shiqi Wang, Anderson Rocha, Xinghao Jiang, and Alex C. Kot. Learning Generalized Deep Feature Representation for Face Anti-Spoofing. *IEEE Transactions on Information Forensics and Security*, 13(10):2639–2652, 2018. [2](#)

[16] Haoliang Li, Student Member, Wen Li, Hong Cao, Senior Member, Shiqi Wang, Feiyue Huang, Alex C Kot, Haoliang H Li Li, A C Kot, and W Li. Unsupervised Domain Adaptation for Face Anti-Spoofing. *IEEE Transactions on Information Forensics and Security*, 13(7):1794–1809, 2018. [2](#)

[17] Yaojie Liu, Amin Jourabloo, and Xiaoming Liu. Learning Deep Models for Face Anti-Spoofing: Binary or Auxiliary Supervision. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 389–398, Salt Lake City, UT, Jun. 18-22 2018. [1, 2, 3, 4, 6, 7, 8](#)

[18] Yaojie Liu, Joel Stehouwer, Amin Jourabloo, and Xiaoming Liu. Deep Tree Learning for Zero-shot Face Anti-Spoofing. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4680–4689, Long Beach, CA, Jun. 16-20 2019. [1, 2, 3, 4, 5, 6, 7, 8](#)

[19] Suril Mehta, Anannya Uberoi, Akshay Agarwal, Mayank Vatsa, and Richa Singh. Crafting A Panoptic Face Presentation Attack Detector. In *Proc. IEEE International Conference on Biometrics*, Crete, Greece, Jun. 4-7 2019. [2](#)

[20] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, Salt Lake City, UT, Jun. 18-23 2018. [4](#)

[21] Nikolaos Sarafianos, Xiang Xu, and Ioannis A. Kakadiaris. Adversarial Representation Learning for Text-to-Image Matching. In *Proc. IEEE International Conference on Computer Vision*, pages 5814–5824, Seoul, Korea, Oct. 27 - Nov. 2 2019. [2](#)

[22] Rui Shao, Xiangyuan Lan, Jiawei Li, and Pong C. Yuen. Multi-adversarial Discriminative Deep Domain Generalization for Face Presentation Attack Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10023–10031, Long Beach, CA, Jun. 16-20 2019. [2](#)

[23] Rui Shao, Xiangyuan Lan, and Pong C. Yuen. Regularized Fine-grained Meta Face Anti-spoofing. In *Proc. AAAI Conference on Artificial Intelligence*, New York, NY, Feb. 7-12 2020. [2](#)

[24] Zheng Shou, Junting Pan, Jonathan Chan, Kazuyuki Miyazawa, Hassan Mansour, Anthony Vetro, Xavier Giro-I-Nieto, and Shih Fu Chang. Online Detection of Action Start in Untrimmed, Streaming Videos - Modeling and Evaluation. In *Proc. European Conference in Computer Vision*, pages 551–568, Munich, Germany, Sept. 8–14 2018. [2](#)

[25] Talha Ahmad Siddiqui, Samarth Bharadwaj, Tejas I. Dhamecha, Akshay Agarwal, Mayank Vatsa, Richa Singh, and Nalini Ratha. Face Anti-spoofing with Multifeature Videolet Aggregation. In *Proc. IAPR International Conference on Pattern Recognition*, pages 1035–1040, Poznan, Poland, July 30 - Aug. 3 2017. 2

[26] Guoqing Wang, Hu Han, Shiguang Shan, and Xilin Chen. Cross-domain Face Presentation Attack Detection via Multi-domain Disentangled Representation Learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, Jun. 14-19 2020. 2

[27] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal Segment Networks for Action Recognition in Videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–14, 2018. 1, 2

[28] Mingze Xu, Mingfei Gao, Yi-Ting Chen, Larry S. Davis, and David J. Crandall. Temporal Recurrent Networks for Online Action Detection. In *Proc. IEEE International Conference on Computer Vision*, pages 5532–5541, Seoul, Korea, Oct. 27–Nov. 2 2019. 2

[29] Xiang Xu, Zhou Xiong, Ragav Venkatesan, Gurumurthy Swaminathan, and Orchid Majumder. d -SNE: Domain Adaptation using Stochastic Neighborhood Embedding. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2497–2506, Long Beach, CA, Jun. 16–20 2019. 2

[30] Zhenqi Xu, Shan Li, and Weihong Deng. Learning Temporal Features using LSTM-CNN Architecture for Face Anti-spoofing. In *Proc. IAPR Asian Conference on Pattern Recognition*, pages 141–145, Kuala Lumpur, Malaysia, Nov 2015. 2

[31] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3507–3516, Long Beach, CA, Jun. 16-20 2019. 2, 3, 4, 5, 6, 7, 8

[32] 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 *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, Jun. 14–19 2020. 7

[33] Shifeng Zhang, Xiaobo Wang, Ajian Liu, Chenxu Zhao, Jun Wan, Sergio Escalera, Hailin Shi, Zezheng Wang, and Stan Z. Li. A Dataset and Benchmark for Large-scale Multi-modal Face Anti-spoofing. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 919–928, Long Beach, CA, Jun. 16-20 2019. 2, 4, 7