Pattern Recognition Letters
journal homepage: www.elsevier.com

Self-supervised 2D face presentation attack detection via temporal sequence sampling

Usman Muhammad[a], Zitong Yu[a], Jukka Komulainen[a,b,**]

[a]Center for Machine Vision and Signal Analysis (CMVS), University of Oulu, Finland
[b]Visidon Ltd., Oulu, Finland

Article history:

Face recognition, presentation attack detection, spoofing, liveness detection, self-supervised learning, motion compensation

ABSTRACT

Conventional 2D face biometric systems are vulnerable to presentation attacks performed with different face artefacts, e.g., printouts, video-replays and wearable 3D masks. The research focus in face presentation attack detection (PAD) has been recently shifting towards end-to-end learning of deep representations directly from annotated data rather than designing hand-crafted (low-level) features. However, even the state-of-the-art deep learning based face PAD models have shown unsatisfying generalization performance when facing unknown attacks or acquisition conditions due to lack of representative training and tuning data available in the existing public benchmarks. To alleviate this issue, we propose a video pre-processing technique called Temporal Sequence Sampling (TSS) for 2D face PAD by removing the estimated inter-frame 2D affine motion in the view and encoding the appearance and dynamics of the resulting smoothed video sequence into a single RGB image. Furthermore, we leverage the features of a Convolutional Neural Network (CNN) by introducing a self-supervised representation learning scheme, where the labels are automatically generated by the TSS method as the stabilized frames accumulated over video clips of different temporal lengths provide the supervision. The learnt feature representations are then fine-tuned for the downstream task using labelled face PAD data. Our extensive experiments on four public benchmarks, namely Replay-Attack, MSU-MFSD, CASIA-FASD and OULU-NPU, demonstrate that the proposed framework provides promising generalization capability and encourage further study in this domain.

1. Introduction

Face recognition (FR) has become an indispensable component in numerous real-world application domains requiring reliable person verification or identification, such as device unlocking, online banking, smart homes, airports, video surveillance and law enforcement. The threat of presentation attacks (spoofing) is one of the main issues with FR systems as conventional FR techniques are vulnerable to direct sensor-level attacks, where an artificial biometric sample is presented to confuse the recognition system using different presentation attack instruments (PAI), e.g., printouts, displays, paper masks and wearable 3D masks. Compared with other biometric traits, such as iris and fingerprint, face biometric samples of the targeted person are much easier to obtain. For instance, people are sharing their pictures openly on the Internet using different social media platforms, from which attackers can easily acquire photographs to create face artefacts. Since conventional FR algorithms are not inherently capable of discriminating attacks from bona fide faces, dedicated presentation attack detection (PAD) methods are needed to mitigate the vulnerabilities to spoofing.

The accuracy of automatic FR is no longer a major concern in numerous real-world applications, thus the focus in FR research community has shifted towards mitigating the threat posed by presentation attacks. Traditionally, software-based face PAD techniques have been founded on hand-crafted (low-level) features describing liveness and motion cues, like eye blinking and lip movements [18], and facial texture [5, 22] and image quality

**Corresponding author
e-mail: muhammad.usman@oulu.fi (Usman Muhammad), zitong.yu@oulu.fi (Zitong Yu), jukka.komulainen@oulu.fi (Jukka Komulainen)

[11, 34] properties of bona fide and artificial faces, for instance. However, low-level features rely heavily on human experience to extract detailed information and the designed feature spaces might not be able to distinguish subtle differences between bona fide samples and various face artefacts. In the past few years, end-to-end learning of deep features, e.g., Convolutional Neural Networks (CNNs) with different loss functions, have been successfully utilized to overcome some limitations of hand-crafted descriptors [15, 17, 21, 26, 27, 32, 33, 36, 37, 39]. A comprehensive overview of the recent advances in deep learning based face PAD can be found in [38].

Although promising results have been achieved, even the state-of-the-art deep learning based face PAD techniques have shown unsatisfying generalization performance when facing unknown operating conditions of unconstrained real-world applications. The lack of generalization is largely due to the domain shift between source (train) and target (test) data as the existing public face PAD benchmarks suffer from severe bias across different covariates, including user demographics, PAIs, sensors, image/video resolution, frame rate, illumination conditions and stand-off distance between face and sensor. The domain generalization issues of software-based face PAD methods have been widely acknowledged and the recent trend in face PAD research has been increasingly on improving the performance in: 1) cross-database studies where a method is trained and tested with different datasets [9], and 2) specific intra-database cross-test evaluation protocols where pre-defined subsets of a dataset are used to introduce unseen test conditions, e.g., cameras, PAIs, illumination and environments [7, 21].

Leveraging the potential of the state-of-the-art deep learning architectures and tuning well-generalizing face PAD models are still very difficult problems due to the huge number of parameters and the limited amount of representative training data available in the existing public datasets. The approaches proposed in the context of generalized face PAD can be roughly categorized into: 1) face PAD-specific feature learning to capture the intrinsic differences between real and fake faces [17], 2) data augmentation and synthesis [36], 3) auxiliary supervision [21, 33, 36, 37, 39], 4) domain adaptation [20, 23, 32] and generalization [15], and 5) continual detection and learning of novel attack types [26]. While face PAD has been traditionally treated as a "black box" binary classification problem, Jourabloo *et al.* [17] proposed a deep CNN architecture for explicitly extracting PAI dependent spoof noise, e.g., characteristic reflections, colour distortions and moiré patterns, from facial images and then use spoof noise modelling for discriminating attacks from bona fide samples. Yang *et al.* [36] introduced a data synthesis technique to simulate digital medium-based spoofing attacks and were able to significantly improve the PAD performance with their augmented training data. Liu *et al.* [21] proposed to increase the generalization of face PAD methods by exploiting spatial and temporal auxiliary supervision, where face depth can be considered as spatial information while remote photoplethysmography (rPPG) signals (pulse) are used as temporal cues. Several works have also approached the generalization issues in face PAD from domain adaptation and domain generalization point of view. Domain adaptation

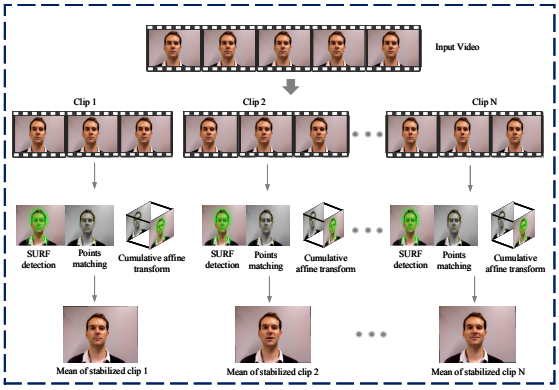

Fig. 1. In the proposed TSS method, the input video is first divided into N equal video clips and the inter-frame 2D affine motion is estimated within each video clip based on the trajectories of SURF keypoint matches. Then, the estimated inter-frame 2D affine motion is removed from the video frames and the resulting clips are accumulated into single RGB images.

Fig. 2. First frame of a sample print attack video clip (left) and the mean of the corresponding stabilized video clip (middle). The result of simple frame averaging (right) is included for comparison to demonstrate the amount of inter-frame 2D affine motion in the original print attack video clip.

[20, 23, 32] based approaches exploit some data from the target domain to match the feature distributions of source and target domains, whereas domain generalization [15] based techniques try to minimize the bias between diverse source domains without using any data from the target domain. Rostami *et al.* [26] proposed to tackle the problem of unknown attacks using continual detection and learning of novel attack types and developed a method to update a face PAD model with test samples that do not fit the training distribution in an embedding space.

Despite the generalization ability of the face PAD methods proposed in the literature has been gradually improving, the results have been still far from satisfying for real-world applications. For instance, the performance of methods using auxiliary supervision depends largely on the accuracy of the estimated auxiliary information. Monocular depth estimation from single face images or even short video sequences is rather difficult if active user interaction, e.g., challenge-response approach, is not utilized during liveness check. Also, reliable estimation of rPPG signals is hard when the subject is moving or lighting conditions are challenging. A major problem with domain adaptation based approaches is that collecting data from the target domain is expensive or even impossible in some real-world use cases.

In this work, we propose to use spatiotemporal information for face PAD because we argue that both static and dynamic information provide important visual cues for discriminating artificial faces from real ones. However, the successive frames in PAD videos are highly redundant. The videos might comprise hundreds of frames repeating similar patterns,

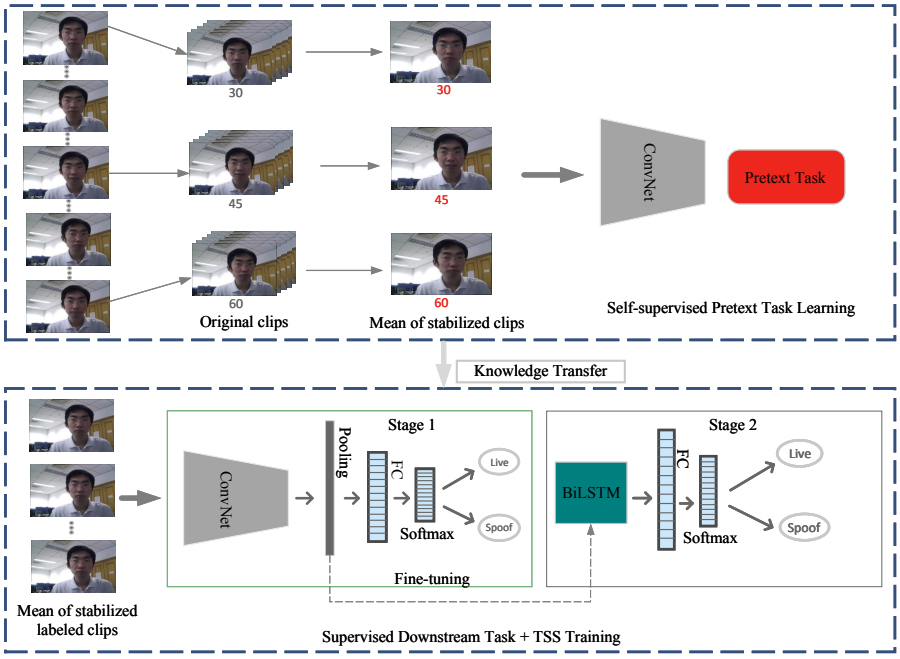

Fig. 3. An illustration of the proposed self-supervised and TSS training tasks. During the self-supervised learning phase, the CNN receives unlabelled TSS sampling outputs accumulated over different temporal lengths and the pretext task is to predict the length of the original video clip. The learnt feature representations are then fine-tuned on labelled TSS encoded video clips by performing PAD as the downstream task (stage 1). Finally, the BiLSTM subnetwork is trained using the fine-tuned features (stage 2) to make the final PAD decision.

which makes it difficult to extract meaningful liveness cues even with deep learning based approaches. Therefore, we propose a simple, yet effective pre-processing method called Temporal Sequence Sampling (TSS) to accumulate appearance and dynamic information of video sequences into single RGB images. This is achieved by splitting an input video sequence into non-overlapping segments, and then estimating the trajectories of keypoints within each video clip. We focus on the problem of print and display attacks (consisting of both digital photos and video-replays) when the PAIs can be considered as planar 2D objects. Therefore, we stabilize each video segment by removing the inter-frame 2D affine motion estimated based on the keypoint trajectories and then aggregate the resulting video frames into a single image. Fig. 1 provides an illustration of the steps described above. A comparison between straightforward frame aggregation and the output of the proposed TSS approach is shown in Fig. 2, which highlights the amount of inter-frame 2D affine motion in the original print attack video clip[1].

It is worth noting that the cumulative 2D affine transformation estimated within a video clip is not directly used for face alignment but to enrich the spatiotemporal discrepancies between real 3D faces and flat 2D face artefacts in the observed view. The proposed approach can handle also print attacks where the 2D surface is warped, as bending a photograph leads to a highly distorted cumulative 2D affine mapping that is not characteristic for real faces. The problem of video-replay attacks exhibiting also non-rigid facial motion is tackled by focusing on appearance information, which has shown promising generalization in detecting display attacks, e.g., in [6], due to evident screen bezels, video compression artefacts, display noise signatures, moiré effects, and luminance and colour distortions, for instance.

Recently, self-supervised learning [16] has been receiving increasing attention as solving pretext tasks, like patch location, order and rotation prediction, in unsupervised manner has shown to be successful in learning meaningful and more interpretable visual representations from the data itself, thus mitigating the need for human annotations for the downstream task. Inspired by the work on frame order prediction where 3D CNNs and optical flow information have been utilized [19, 35], we propose a self-supervised learning scheme where the pretext task is to predict the length of the original video clip based on the TSS encoded data. To be more specific, the stabilized frames accumulated over video segments of different temporal lengths provide the supervision for training a 2D CNN with the aim of learning more meaningful representations from the videos aggregated into single RGB images. The learnt visual features are then fine-tuned for the downstream face PAD task.

The main contributions of this work can be summarized as follows:

1. In order to reduce temporal redundancy and remove inter-frame 2D affine motion in videos, Temporal Sequence Sampling (TSS) is introduced to encode video clips into a compact representation in the form of a single RGB image.
2. The need for annotated data in face PAD is mitigated using self-supervised learning.
3. The effectiveness of the proposed approach is demonstrated using the official cross-test evaluation protocols

[1]Sample videos can be found at: `http://yty.kapsi.fi/PRL_2022/`

of the OULU-NPU database [7] and several widely used cross-database configurations, where promising generalization ability with new state-of-the-art results is achieved.

We also provide the source code[2] to the research community for reproducing, verifying and extending our results.

2. Proposed Method

The backbone of the proposed face PAD approach is the TSS method, which removes inter-frame 2D affine motion within a video segment and accumulates frames of the resulting motion-compensated video clip into a single RGB image (see, Fig. 1 and Fig. 2). The main architecture of our face PAD framework is illustrated in Fig. 3. During the self-supervised learning phase, the CNN receives unlabelled TSS encoded frames accumulated over video segments of different temporal lengths and the pretext task is to predict the length of the original video clip. The learnt feature representations are then fine-tuned on labelled TSS encoded video clips by performing PAD as the downstream task (stage 1 in Fig. 3). Finally, a Bidirectional Long Short-Term Memory (BiLSTM) [28] subnetwork is trained using the fine-tuned CNN features to make the final face PAD decision (stage 2 in Fig. 3). A more detailed description of the proposed TSS method and self-supervised learning scheme is provided in the following, while the implementation details and the training process are discussed later in Section 3.3.

2.1. Temporal Sequence Sampling (TSS)

The steps of the proposed Temporal Sequence Sampling method are illustrated in Fig 1. First, the input video is equally partitioned into S non-overlapping segments (clips), where each video clip contains the same number of frames, e.g., 45. We estimate the 2D affine motion between all adjacent frames of a video clip based on sparse point correspondences. We use first Speeded Up Robust Features (SURF) descriptor [1] to detect keypoints from both the face and background regions in each video frame and then find the corresponding points between all adjacent video frames using the Hamming distance.

The M-estimator SAmple Consensus (MSAC) algorithm [31] is utilized to mitigate the impact of incorrect point correspondences and to get robust estimates of the 2D affine transformations between the adjacent frames. MSAC is an improved version of RANdom SAmple Consensus (RANSAC) where an M-estimator is introduced to set outlier point correspondences a constant weight while inliers are weighted based on how well they fit the estimated transformation.

The resulting 2D affine transformation between adjacent frames is a 3x3 matrix:

$$\begin{bmatrix} a_1 & a_2 & t_x \\ a_3 & a_4 & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where a_n represents scale, rotation, and shearing transformations and t_x and t_y correspond to translation. However, we convert the 2D affine transformation described above into a simpler and more stable four parameter transformation to produce the final motion-compensated video clip:

$$\begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where s is a scale factor and θ rotation angle.

The 2D affine transformation between two frames F_i and F_{i+1} is denoted with A_i when the cumulative 2D affine transformation of a frame F_i with respect to the first (reference) frame F_0 of the video segment corresponds to cascaded inter-frame transformations:

$$A_{0,i} = \prod_{j=0}^{i-1} A_j \tag{3}$$

The estimated cumulative transformation $A_{0,i}$ is used to remove inter-frame 2D affine motion by warping each frame F_i relative to the first frame F_0 of the video clip. It is worth highlighting that the cumulative transformation aims at removing the infer-frame 2D motion, but the frame F_i is not necessarily aligned with the reference frame F_0 due to the cumulative errors in estimating the motion between adjacent frames and changes in the observed view within the video clip.

Finally, we take the temporal average of the motion-compensated frames to encode the whole video clip into a single RGB image. An example of the TSS output is shown in Fig. 2.

2.2. Self-supervised representation learning

The amount and nature of spatiotemporal variations within video segments, and, consequently, the TSS outputs and their corresponding CNN feature representations depend largely on the duration of the input sequence. Therefore, the key idea of our self-supervision scheme is to generate sets of TSS encoded video clips with different temporal lengths L and then learn the spatiotemporal variations and context across these different length settings. To be more specific, we first generate T classes of TSS outputs with e.g., $L = \{5, 15, 30, 45, 60\}$ and then use these labels to train a deep CNN with softmax loss to predict the length of a given TSS encoded video segment. After the self-supervised spatiotemporal context adaptation step, the learnt 2D visual features are then further fine-tuned for the actual downstream task of face PAD and finally the BiLSTM subnetwork is trained using the resulting CNN features.

3. Experimental Setup

In the following, we introduce briefly the public benchmark face PAD datasets and describe the evaluation metrics and protocols used in our experimental analysis. Finally, the implementation details of the proposed approach are also provided.

[2] https://github.com/Usman1021/Self-Supervised-2D-PAD

3.1. Experimental data

To assess the generalization of the proposed face PAD approach, we considered four widely used publicly available databases consisting of bona fide and 2D face presentation attack videos, namely Idiap Replay-Attack Database [8] (denoted as I), CASIA Face Anti-Spoofing Database [41] (denoted as C), MSU Mobile Face Spoofing Database [34] (denoted as M), and OULU-NPU Database [7] (denoted as O).

Idiap Replay-Attack Database [8] consists of bona fide and attack videos of 50 subjects captured under two different lighting conditions. Five different attacks are launched with iPhone 3GS (digital photo and video-replay), 1st generation iPad (digital photo and video-replay) and hard copies. All videos are recorded with a built-in webcam of a MacBook Air laptop. Altogether, the database contains 1,200 videos, which are divided into three subject-disjoint subsets for training, development and testing (15, 15 and 20 subjects, respectively).

CASIA Face Anti-Spoofing Database (CASIA-FASD) [41] contains bona fide and attack videos of 50 subjects recorded with three different of imaging qualities (low, normal and high) and considers three kinds of attack presentations (warped photo, cut-photo and video-replay). Consequently, each subject has three kinds of bona fide videos and nine different attack presentations. Altogether, the database contains 600 videos, which are divided into two subject-disjoint subsets for training and testing (20 and 30 subjects, respectively).

MSU Mobile Face Spoofing Database (MSU-MFSD) [34] includes bona fide and attack videos of 35 subjects recorded with two mobile devices (a Google Nexus 5 smartphone and a MacBook Air laptop). Three kinds of attack presentations are considered, including two video-replay attacks of different quality (iPhone 5S and iPad Air) and a print attack. Consequently, each subject has two kinds of real videos and six different attack presentations. Altogether, the database contains 280 videos, which are divided into two subject-disjoint subsets for the training and testing (15 and 20 subjects, respectively).

OULU-NPU Database [7] is one of the most recent commonly used face PAD datasets. It contains bona fide and attack videos of 55 subjects recorded in several acquisition conditions (six high-resolution smartphone front cameras and three sessions) and considers two kinds of print attacks and two kinds of video-replay attacks. Four cross-test protocols are used to evaluate the generalization performance of a face PAD method across different covariates. Protocols 1, 2, and 3 introduce a single previously unseen test condition, namely illumination, PAI and sensor, respectively, while the fourth and most challenging protocol evaluates the generalization performance simultaneously across unknown sensors, attacks and illumination conditions. Altogether the database contains 5,940 videos, which are divided into three subject-disjoint subsets for training, development and testing (20, 15 and 20, respectively).

3.2. Evaluation metrics and protocols

All four datasets are used in our cross-database experiments, while only the OULU-NPU database is utilized also for intra-database experiments following its official cross-test protocols that assess generalization across different covariates.

For our cross-database experiments, we follow the widely used evaluation metrics and protocols introduced in [9]. The results are reported using Half Total Error Rate (HTER), which denotes the mean of the False Acceptance Rate (FAR) and False Rejection Rate (FRR). The HTER is computed on the test set of the target domain using the threshold τ corresponding to the equal error rate (EER) operating point on the development set of the source domain. In the case of the CASIA-FASD and MSU-MFSD datasets, the threshold τ is computed on the training set because they lack pre-defined validation sets (see, Section 3.1).

The intra-database results on the official cross-test protocols of the OULU-NPU database are reported in terms of Average Classification Error Rate (ACER), which denotes the mean of Attack Presentation Classification Error Rate (APCER) and Bona Fide Presentation Classification Error Rate (BPCER). APCER and BPCER essentially correspond to FAR and FRR, respectively, but APCER is computed separately for each PAI, e.g., print and video-replay, and the final PAD performance corresponds to the attack with the highest APCER, i.e., the most successful PAI. Similarly to the HTER, the ACER is computed on the test set using the threshold τ corresponding to the EER operating point on the development set.

3.3. Implementation details

We utilize both the face and background regions for PAD, thus no face cropping is applied. The TSS method processes the frames of the input video clips at their native resolution, and the TSS generated accumulated output frames are resized to 224×224 according to the input image size of the pre-trained CNN (ResNet-101 [14]). The video segment length for TSS methods was set to 45 and the number of TSS encoded segments depends on the total length of an input video sequence. For instance, a video of 270 frames results in six TSS encoded video clips. No data augmentation is applied during training.

To evaluate the generalization performance of the proposed TSS method, the pre-trained CNN is fine-tuned using Stochastic Gradient Descent (SGD) with mini-batch size of 32 and validation frequency of 30, and by shuffling every epoch. We do not use fixed epochs because an early stopping function is utilized to automatically stop the model training when over-fitting is observed [25]. The learning rate is fixed to 0.0001 in our cross-database experiments, while we adjust the learning rate to 0.001 on all four intra-database cross-test protocols of the OULU-NPU dataset.

The fine-tuned feature vectors are extracted from the output of the last pooling layer with size of 2048. The BiLSTM subnetwork is trained using cross-entropy loss and Adam optimizer with fixed learning rate of 0.0001. The number of hidden units is fixed to 100 in the cross-database experiments, while the number of hidden units is increased to 500 on all four intra-database cross-test protocols of the OULU-NPU dataset. We set the recurrent weights with He initializer [13] that performs the best in all scenarios of our experiments.

During the self-supervised training stage, we first fine-tune the pre-trained CNN with the aforementioned settings on the unlabelled data of the pretext task, i.e., sets of TSS outputs with different temporal length combinations. Then, the model is

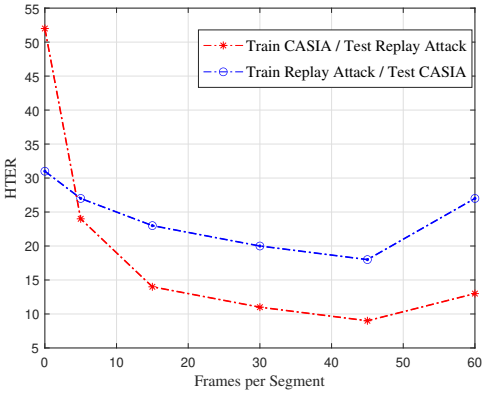

Fig. 4. Cross-database performance with different video segment lengths.

further fine-tuned using the binary labels (bona fide and attack) of the downstream task by replacing the fully connected layer with a new one with the output size of 2. Finally, the BiLSTM subnetwork takes the input of the last average pooling layer of the fine-tuned CNN and gives the final binary PAD decision.

4. Experimental Results

Our experimental analysis focuses on assessing the generalization performance of the proposed approach under two settings: 1) different cross-database configurations, and 2) the official intra-database cross-test protocols of the OULU-NPU dataset. In the following, we first investigate the effect of the input video sequence length on the TSS method and then study the effectiveness of the proposed self-supervised learning scheme in cross-database tests between the CASIA-FASD and Replay-Attack datasets. Finally, we compare the performance of the proposed approach against the state of the art in several widely used cross-database configurations and on the four official evaluation protocols of the OULU-NPU database.

4.1. The effect of video segment length

We begin our experiments by exploring how the performance of the proposed TSS method depends on the length of the video segment. We examine the generalization performance by varying the length of video segments L from 5 to 60 frames. The cross-database results on the Replay-Attack and CASIA-FASD databases shown in Fig. 4 depict that the HTER decreases as the number of frames per video segment increases. However, when we further increase the temporal length of frames to more than 45 frames, the face PAD performance on the CASIA-FASD and Replay-Attack datasets starts decreasing. Therefore, we set $L = 45$ where the best performance is achieved, i.e., HTER of 9.3% on Replay-Attack and 18.1% on CASIA-FASD database.

4.2. Effectiveness of self-supervised learning

For assessing the performance of the proposed self-supervised learning scheme, we construct three sets of TSS encoded video segments with varying number of temporal lengths $L = \{5, 15, 30, 45, 60\}$ for the pretext task. It can be observed from Table 1 that the use of pretext task with three temporal lengths ($L = \{15, 30, 45\}$) leads to best cross-database performance, improving the HTER with 3.4% and 2.9% for Replay-Attack and CASIA-FASD databases, respectively.

Table 1. Cross-database performance of the proposed self-supervised learning scheme in terms of HTER (%).

Number of classes	Segment lengths	Train C / Test I	Train I / Test C
2	30, 45	8.2	18.4
3	15, 30, 45	5.9	15.2
5	5, 15, 30, 45, 60	24.8	22.1

Table 2. Cross-database performance in terms of HTER (%) on the Replay-Attack and CASIA-FASD databases. Comparative results are obtained from [40].

Method	Train CASIA-FASD Test Replay-Attack	Train Replay-Attack Test CASIA-FASD
LBP [22]	47.0	39.6
LBP-TOP [10]	49.7	60.6
Color-LBP [4]	30.3	37.7
Motion-Mag [2]	50.1	47.0
Spectral cubes [24]	34.4	50.0
Auxiliary [21]	27.6	28.4
FaceDs [17]	28.5	41.1
STASN [36]	31.5	30.9
DSGTD [33]	17.0	22.8
CDCN [40]	6.5	29.8
TSS with ResNet	30.4	39.9
TSS with ResNet-BiLSTM	9.3	18.1
Self-supervised learning	**5.9**	**15.2**

4.3. Comparison against the state of the art

The results presented in Table 2 depict that our TSS method achieves astonishing cross-database performance between the Replay-Attack and CASIA-FASD datasets, and that the generalization ability can be further improved when the proposed self-supervised learning stage is included in training the PAD model. The TSS method combined with self-supervised learning obtains the state of the art in this widely used cross-database configuration, achieving an HTER improvement from 9.3% to 5.9% on the Replay-Attack dataset and from 18.1% to 15.2% on the CASIA-FASD dataset, respectively. Thus, the proposed self-supervised learning scheme indeed helps in fine-tuning a 2D CNN to learn more meaningful representations from the TSS encoded video segments.

Table 3. Combined cross-database evaluation using MSU-MFSD (M), Idiap Replay-Attack (I), CASIA-FASD (C) and OULU-NPU (O) databases. Comparative results are obtained from [15].

	M&I to C		M&I to O	
Method	HTER(%)	AUC(%)	HTER(%)	AUC(%)
MS-LBP [22]	51.16	52.09	43.63	58.07
LBP-TOP [10]	45.27	54.88	47.26	50.21
Color-LBP [5]	55.17	46.89	53.31	45.16
IDA [34]	45.16	58.80	54.52	42.17
MADDG [30]	41.02	64.33	39.35	65.10
SSDG-M [15]	31.89	71.29	36.01	66.88
TSS with ResNet	29.58	79.44	37.62	70.26
TSS with ResNet-BiLSTM	**28.66**	**83.73**	**30.12**	**79.06**

Table 3 presents the generalization performance of our TSS approach in another cross-database configuration, combining the MSU-MFSD and Replay-Attack databases for training and

Table 4. Intra-database evaluation on the four official protocols of the OULU-NPU database. Comparative results are obtained from [40].

Protocol	Method	APCER(%)	BPCER(%)	ACER(%)
1	DeepPixBiS [12]	0.8	0.0	0.4
	GRADIANT [3]	1.3	12.5	6.9
	Auxiliary [21]	1.6	1.6	1.6
	FaceDs [17]	1.2	1.7	1.5
	STASN [36]	1.2	2,5	1.9
	DSGTD [33]	2.0	0.0	1.0
	CDCN [40]	0.4	0.0	0.2
	BiFPN [27]	3.1	0.8	2.0
	TSS with ResNet	0.6	10.3	5.5
	TSS with ResNet-BiLSTM	0.0	0.2	**0.1**
2	DeepPixBiS [12]	11.4	0.6	6.0
	GRADIANT [3]	3.1	1.9	2.5
	Auxiliary [21]	2.7	2.7	2.7
	FaceDs [17]	4.2	4.4	4.3
	STASN [36]	4.2	0.3	2.2
	DSGTD [33]	2.5	1.3	1.9
	CDCN [40]	1.8	0.8	1.3
	BiFPN [27]	1.7	1.1	1.4
	TSS with ResNet	2.0	2.1	2.1
	TSS with ResNet-BiLSTM	0.4	0.8	**0.6**
3	DeepPixBiS [12]	11.7±19.6	10.6±14.1	11.1±9.4
	GRADIANT [3]	2.6±3.9	5.0±5.3	3.8±2.4
	Auxiliary [21]	2.7±1.3	3.1±1.7	2.9±1.5
	FaceDs [17]	4.0±1.8	3.8±1.2	3.6±1.6
	STASN [36]	4.7±3.9	0.9±1.2	2.8±1.6
	DSGTD [33]	3.2±2.0	2.2±1.0	2.7±0.6
	CDCN [40]	1.7±1.5	2.0±1.2	1.8±0.7
	BiFPN [27]	0.7±0.7	0.3±0.7	**0.5±0.6**
	TSS with ResNet	7.2±8.3	3.9±3.4	5.5±3.0
	TSS with ResNet-BiLSTM	2.5±1.8	0.5±0.6	1.5±0.8
4	DeepPixBiS [12]	36.7±29.7	13.3±14.1	25.0±12.7
	GRADIANT [3]	5.0±4.5	15.0±7.1	10.0±5.0
	Auxiliary [21]	9.3±5.6	10.4±6.0	9.5±6.0
	FaceDs [17]	1.2±6.3	6.1±5.1	5.6±5.7
	STASN [36]	6.7±10.6	8.3±8.4	7.5±4.7
	DSGTD [33]	6.7±7.5	3.3±4.1	5.0±2.2
	CDCN [40]	4.2±3.4	5.8±4.9	5.0±2.9
	BiFPN [27]	2.5±3.2	3.3±4.1	**2.9±3.4**
	TSS with ResNet	5.7±4.5	16±13	10.8±5.3
	TSS with ResNet-BiLSTM	4.7±10.5	9.2±10.4	7.1±5.3

the CASIA-FASD and OULU-NPU databases for testing. The proposed TSS method with CNN-BiLSTM framework achieves the best results and significant improvement with respect to the state of the art in HTER from 31.89% to 28.66% on the CASIA-FASD and from 36.01% to 30.12% on the OULU-NPU dataset.

The results of the intra-database experiments following the official evaluation protocols of the OULU-NPU database are presented in Table 4. From these results it can be seen that the proposed TSS method with CNN-BiLSTM framework ranks first on the protocols 1 and 2 of the OULU-NPU database obtaining ACER of 0.1% and 0.6%, respectively, and achieves very competitive performance of 1.5% and 7.1% in terms of ACER on the protocols 3 and 4, respectively.

We have also included the performance of the proposed TSS method in the cross-database experiments and the intra-database tests on the OULU-NPU without the BiLSTM subnetwork in order to demonstrate the importance of the BiLSTM component on the final face PAD performance.

4.4. Network visualization and analysis

In this section, we use Gradient-weighted Class Activation Mapping (Grad-CAM) [29] to help in explaining why the proposed method makes a particular decision. Sample Grad-CAM visualizations of real faces, video-replay attacks and print attacks are presented for further analysis in Fig. 5. The first row represents samples of real faces, from which one can see that the network gives clear focus on the actual facial region due to e.g., head motion, non-rigid facial movements, eye blinking

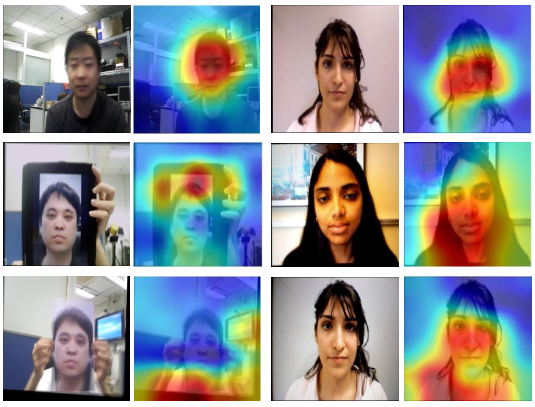

Fig. 5. Grad-CAM visualizations for TSS encoded videos corresponding to real faces (first row), video-replay attacks (second row) and print attacks (third row).

and skin texture, while the background region does not provide liveness cues. In contrast, the samples of video-replay and print attacks in the second and third rows, respectively, depict that the discriminative visual and motion cues are PAI related and attention is more dispersed and focusing also on background regions, i.e., non-face related information.

5. Conclusions

In this paper, we addressed the generalization issues in 2D face presentation attack detection. We proposed a Temporal Sequence Sampling (TSS) method that removes the estimated inter-frame 2D affine motion within short video clips and encodes the appearance and dynamics of the resulting frames into a single colour image. We also introduced a self-supervised learning scheme where the stabilized video frames accumulated over sequences of different temporal lengths provide the supervision to train a 2D Convolutional Neural Network. We conducted extensive experimental analysis using the official intra-test protocols of the OULU-NPU database and several cross-database configurations on four public face PAD databases to demonstrate the robustness of the proposed framework.

The proposed approach needs capturing and processing of relatively long input sequences, i.e., approximately two seconds of video, in order to achieve robust face PAD performance, thus it cannot be used in authentication applications requiring real-time response or biometric systems operating on single facial images. A drawback of encoding stabilized video clips into a compact representation in the form of a single colour image is that the subtle inter-frame motion (direction) information is lost due to frame aggregation. Therefore, we plan to extend our work by developing methods that explicitly model the geometrical differences in the feature or facial landmark based trajectories between motion-compensated bona fide and attack videos. In this work, we focused only on detecting attacks launched with 2D PAI, i.e., prints and displays, thus it is yet unknown whether the proposed approach generalizes well under unseen or other types of facial artefacts, including paper and 3D masks. In the future, we will explore the robustness of our method on new emerging face PAD datasets and evaluation protocols.

Acknowledgments

The financial support of the Tauno Tönning Foundation is gratefully acknowledged.

References

[1] Bay, H., Tuytelaars, T., Van Gool, L., 2006. SURF: Speeded up robust features, in: Leonardis, A., Bischof, H., Pinz, A. (Eds.), European Conference on Computer Vision (ECCV), Springer. pp. 404–417.

[2] Bharadwaj, S., Dhamecha, T.I., Vatsa, M., Singh, R., 2013. Computationally efficient face spoofing detection with motion magnification, in: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 105–110.

[3] Boulkenafet, Z., Komulainen, J., Akhtar, Z., Benlamoudi, A., Samai, D., Bekhouche, S.E., Ouafi, A., Dornaika, F., Taleb-Ahmed, A., Qin, L., et al., 2017a. A competition on generalized software-based face presentation attack detection in mobile scenarios, in: International Joint Conference on Biometrics (IJCB), pp. 688–696.

[4] Boulkenafet, Z., Komulainen, J., Hadid, A., 2015. Face anti-spoofing based on color texture analysis, in: IEEE International Conference on Image Processing (ICIP), pp. 2636–2640.

[5] Boulkenafet, Z., Komulainen, J., Hadid, A., 2016. Face spoofing detection using colour texture analysis. IEEE Transactions on Information Forensics and Security 11, 1818–1830.

[6] Boulkenafet, Z., Komulainen, J., Hadid, A., 2018. On the generalization of color texture-based face anti-spoofing. Image and Vision Computing 77, 1–9.

[7] Boulkenafet, Z., Komulainen, J., Li, L., Feng, X., Hadid, A., 2017b. OULU-NPU: A mobile face presentation attack database with real-world variations, in: IEEE International Conference on Automatic Face & Gesture Recognition (FG), pp. 612–618.

[8] Chingovska, I., Anjos, A., Marcel, S., 2012. On the effectiveness of local binary patterns in face anti-spoofing, in: International Conference of the Biometrics Special Interest Group (BIOSIG), pp. 1–7.

[9] de Freitas Pereira, T., Anjos, A., De Martino, J.M., Marcel, S., 2013. Can face anti-spoofing countermeasures work in a real world scenario?, in: International Conference on Biometrics (ICB).

[10] de Freitas Pereira, T., Komulainen, J., Anjos, A., De Martino, J.M., Hadid, A., Pietikäinen, M., Marcel, S., 2014. Face liveness detection using dynamic texture. EURASIP Journal on Image and Video Processing 2014, 1–15.

[11] Galbally, J., Marcel, S., 2014. Face anti-spoofing based on general image quality assessment, in: International Conference on Pattern Recognition (ICPR), pp. 1173–1178.

[12] George, A., Marcel, S., 2019. Deep pixel-wise binary supervision for face presentation attack detection, in: International Conference on Biometrics (ICB), pp. 1–8.

[13] He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: International Conference on Computer Vision (ICCV), pp. 1026–1034.

[14] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778.

[15] Jia, Y., Zhang, J., Shan, S., Chen, X., 2020. Single-side domain generalization for face anti-spoofing, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8484–8493.

[16] Jing, L., Tian, Y., 2020. Self-supervised visual feature learning with deep neural networks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence .

[17] Jourabloo, A., Liu, Y., Liu, X., 2018. Face de-spoofing: Anti-spoofing via noise modeling, in: European Conference on Computer Vision (ECCV), pp. 290–306.

[18] Kollreider, K., Fronthaler, H., Faraj, M.I., Bigun, J., 2007. Real-time face detection and motion analysis with application in "liveness" assessment. IEEE Transactions on Information Forensics and Security 2, 548–558.

[19] Lee, H.Y., Huang, J.B., Singh, M., Yang, M.H., 2017. Unsupervised representation learning by sorting sequences, in: International Conference on Computer Vision (ICCV), pp. 667–676.

[20] Li, H., Li, W., Cao, H., Wang, S., Huang, F., Kot, A.C., 2018. Unsupervised domain adaptation for face anti-spoofing. IEEE Transactions on Information Forensics and Security 13, 1794–1809.

[21] Liu, Y., Jourabloo, A., Liu, X., 2018. Learning deep models for face anti-spoofing: Binary or auxiliary supervision, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 389–398.

[22] Määttä, J., Hadid, A., Pietikäinen, M., 2011. Face spoofing detection from single images using micro-texture analysis, in: International Joint Conference on Biometrics (IJCB), IEEE. pp. 1–7.

[23] Mohammadi, A., Bhattacharjee, S., Marcel, S., 2020. Domain adaptation for generalization of face presentation attack detection in mobile settengs with minimal information, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

[24] Pinto, A., Pedrini, H., Schwartz, W.R., Rocha, A., 2015. Face spoofing detection through visual codebooks of spectral temporal cubes. IEEE Transactions on Image Processing 24, 4726–4740.

[25] Prechelt, L., 1998. Early stopping-but when?, in: Neural Networks: Tricks of the trade. Springer, pp. 55–69.

[26] Rostami, M., Spinoulas, L., Hussein, M., Mathai, J., Abd-Almageed, W., 2021. Detection and continual learning of novel face presentation attacks, in: International Conference on Computer Vision (ICCV).

[27] Roy, K., Hasan, M., Rupty, L., Hossain, M.S., Sengupta, S., Taus, S.N., Mohammed, N., 2021. Bi-FPNFAS: Bi-directional feature pyramid network for pixel-wise face anti-spoofing by leveraging fourier spectra. Sensors 21.

[28] Schuster, M., Paliwal, K.K., 1997. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing 45, 2673–2681.

[29] Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2017. Grad-CAM: Visual explanations from deep networks via gradient-based localization, in: International Conference on Computer Vision (ICCV), pp. 618–626.

[30] Shao, R., Lan, X., Li, J., Yuen, P.C., 2019. Multi-adversarial discriminative deep domain generalization for face presentation attack detection, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10023–10031.

[31] Torr, P., Zisserman, A., 1997. Robust parametrization and computation of the trifocal tensor. Image and Vision Computing 15, 591–605.

[32] Wang, G., Han, H., Shan, S., Chen, X., 2020a. Unsupervised adversarial domain adaptation for cross-domain face presentation attack detection. IEEE Transactions on Information Forensics and Security 16, 56–69.

[33] Wang, Z., Yu, Z., Zhao, C., Zhu, X., Qin, Y., Zhou, Q., Zhou, F., Lei, Z., 2020b. Deep spatial gradient and temporal depth learning for face anti-spoofing, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

[34] Wen, D., Han, H., Jain, A.K., 2015. Face spoof detection with image distortion analysis. IEEE Transactions on Information Forensics and Security 10, 746–761.

[35] Xu, D., Xiao, J., Zhao, Z., Shao, J., Xie, D., Zhuang, Y., 2019. Self-supervised spatiotemporal learning via video clip order prediction, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10334–10343.

[36] Yang, X., Luo, W., Bao, L., Gao, Y., Gong, D., Zheng, S., Li, Z., Liu, W., 2019. Face anti-spoofing: Model matters, so does data, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3507–3516.

[37] Yu, Z., Li, X., Shi, J., Xia, Z., Zhao, G., 2021a. Revisiting pixel-wise supervision for face anti-spoofing. IEEE Transactions on Biometrics, Behavior, and Identity Science 3, 285–295.

[38] Yu, Z., Qin, Y., Li, X., Zhao, C., Lei, Z., Zhao, G., 2021b. Deep learning for face anti-spoofing: A survey. arXiv preprint arXiv:2106.14948 .

[39] Yu, Z., Wan, J., Qin, Y., Li, X., Li, S.Z., Zhao, G., 2021c. NAS-FAS: Static-dynamic central difference network search for face anti-spoofing. IEEE Transactions on Pattern Analysis and Machine Intelligence 43, 3005–3023.

[40] Yu, Z., Zhao, C., Wang, Z., Qin, Y., Su, Z., Li, X., Zhou, F., Zhao, G., 2020. Searching central difference convolutional networks for face anti-spoofing, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

[41] Zhang, Z., Yan, J., Liu, S., Lei, Z., Yi, D., Li, S.Z., 2012. A face antispoofing database with diverse attacks, in: International Conference on Biometrics (ICB), pp. 26–31.