On the Use of Convolutional Neural Networks for Speech Presentation Attack Detection

P. Korshunov[1], A. R. Gonçalves[2], R. P. V. Violato[2], F. O. Simões[2], and S. Marcel[1]

[1]Idiap, Switzerland. sebastien.marcel@idiap.ch
[2]CPqD, Brazil. rviolato@cpqd.com.br

Abstract

Research in the area of automatic speaker verification (ASV) has advanced enough for the industry to start using ASV systems in practical applications. However, these systems are highly vulnerable to spoofing or presentation attacks (PAs), limiting their wide deployment. Several speech-based presentation attack detection (PAD) methods have been proposed recently but most of them are based on hand crafted frequency or phase-based features. Although convolutional neural networks (CNN) have already shown breakthrough results in face recognition, little is understood whether CNNs are as effective in detecting presentation attacks in speech. In this paper, to investigate the applicability of CNNs for PAD, we consider shallow and deep examples of CNN architectures implemented using Tensorflow and compare their performances with the state of the art MFCC with GMM-based system on two large databases with presentation attacks: publicly available voicePA and proprietary BioCPqD-PA. We study the impact of increasing the depth of CNNs on the performance, and note how they perform on unknown attacks, by using one database to train and another to evaluate. The results demonstrate that CNNs are able to learn a database significantly better (increasing depth also improves the performance), compared to hand crafted features. However, CNN-based PADs still lack the ability to generalize across databases and are unable to detect unknown attacks well.

2018 IEEE 4^{th} International Conference on Identity, Security, and Behavior Analysis (ISBA)
978-1-5386-2248-3/18/$31.00

1. Introduction

Recent years have shown an increase in both the accuracy of biometric systems and their practical use. The application of biometrics is becoming widespread from smartphones to automated border controls. The popularization of the biometric systems, however, exposed their major flaw — high vulnerability to spoofing attacks. The ease with which a biometric system can be spoofed demonstrates the importance of developing efficient anti-spoofing systems that can detect both known (conceivable now) and unknown (possible in the future) spoofing attacks.

In this paper, we focus on the spoofing attack detection or, more specifically, on presentation attack detection (PAD) systems in the context of voice biometrics. As per ISO/IEC standard [7], presentation attacks are performed by presenting a spoofed sample to the physical sensor of a biometric system, and for voice biometrics, it means a real or a synthesized speech sample is played back to a microphone of the automatic speaker verification (ASV) system. The replay attacks are easy to perform (no special skills are required) and ASV systems are shown to be vulnerable to them [13].

However, until only recently, most of the research focused on detecting synthesized speech, which is generated by voice conversion or speech synthesis algorithms. Typical detection methods use handcrafted features based on audio spectrogram, such as spectral [27, 4] and cepstral-based features with temporal derivatives [19, 26], phase spectrum based features [3], the combination of amplitude and phase features [16], recently proposed constant Q cepstral coefficients (CQCCs) [22], extraction of local binary patterns in the cepstral domain [1], and audio quality based features [9]. A survey by Wu *et al.* [24] provides a comprehensive overview of the attacks based on synthetic speech and the detection methods tailored to those types of attacks. The problems with these approaches based on handcrafted features is that they do not generalize well for different types of attacks, as several studies have shown [22, 10];

and even 'mega-fusion' based techniques, which fuse several different PAD systems, are not helpful, as is demonstrated in [11, 6].

Some recent studies have shown neural nets to be promising for detection of synthetic speech [5, 21, 17, 28] by learning features from the intermediate representations such as log-scale spectrograms or filterbanks. For instance, Godoy *et al.* [5] demonstrated that a system based on multilayer perceptron (MLP) outperformed support vector machines (SVM) and Gaussian mixture models (GMM) based classifiers in detecting synthetic speech from ASVspoof [25] database. Tian *et al.* [21] have shown on the same ASVspoof database that a temporal CNN is even better than MLP at detecting synthetic speech.

Neural nets are also promising for detection of replay or presentation attacks [14, 12]. The latest study by Muckenhirn*et al.* [15] demonstrates the high accuracy of CNNs compared to systems based on handcrafted features for attack detection. However, little is known how CNNs perform on unknown presentation attacks, and whether they can generalize across different databases with presentation attacks. The impact of the neural net's depth on the performance is also not well understood.

In this paper, we focus on CNN-based PAD systems, which learn features from raw speech data (similar to systems in [15]), and aim to answer several important questions: i) does CNN-based PAD performs better compared to state of the art systems based on handcrafted features? ii) what is the impact of depth of the architecture on the attack detection accuracy? iii) do CNNs perform better on unknown attacks or in cross-database settings? For this purpose, we implemented two open source CNNs using Tensorflow[1], one with a shallow architecture of one convolutional layer and another with a deeper architecture consisting of three convolutional and three max pooling layers. As a state of the art PAD baseline, we selected a system based on mel-scale frequency coefficients (MFCC) and Gaussian mixture model (GMM) classifier. It has been shown [18, 10, 11] that this system (besides MFCC being very common and popular handcrafted features in speech processing) is a good 'all-rounder' stand alone system (no fusion) for *unknown* attacks and in cross-database evaluations, in which it out-performed CQCC-based system [11].

Considering CNN-based PAD systems, which typically require large amount of training data, we use two databases with large number of different presentation attacks. One is a publicly available voicePA[2], an extension of the AVspoof database (with more added attacks), which was shown to be a threat to state of the art speaker verification systems [13]. Another is the new BioCPqD-PA [23] database of Portuguese speakers and many high quality *unknown* presentation attacks recorded in an acoustically isolated room. Note that, although proprietary, BioCPqD-PA database will be publicly available for machine learning experiments on a web-based BEAT platform[3].

Therefore, this paper has the following main contributions:

- Two large databases with presentation attacks: voicePA (publicly available[2]) and BioCPqD-PA (proprietary);
- A reproducible evaluation of PAD systems based on two CNNs implemented with Tensorflow and state of the art MFCC system with GMM classifier, including cross-database evaluation scenario;
- Open source implementations of tested PAD systems[4].

2. Speech presentation attack databases

We use two large speech presentation attack databases: voicePA and BioCPqD-PA. VoicePA, which has about 140 hours of audio in the training set alone, contains presentation attacks recorded using different consumer devices, such as laptops and mobile phones, in different environment conditions, when both natural and synthesized speech was played back. BioCPqD-PA database has about 250 hours of audio in the training set alone and its main difference is that it contains only replay attacks recorded in an acoustically isolated room using different combinations of speakers and microphones (including professional equipment). The databases are summarized in Table 1 and detailed descriptions are provided in the next sections.

Table 1. Number of utterances (*bona fide*, *attacks*, and *total*) in voicePA and BioCPqD-PA databases.

Database	Type of data	Train	Dev	Eval
voicePA	bona fide	4,973	4,995	5,576
	attacks	115,740	115,740	129,988
	total	120,713	120,735	135,564
BioCPqD-PA	bona fide	6,857	12,455	7,941
	attacks	98,562	179,005	114,111
	total	105,419	191,460	122,052

2.1. voicePA database

The voicePA[2] database inherits bona fide (genuine) speech samples from AVspoof database [13], which contains utterances from 44 participants (31 males and 13 females) recorded over the period of two months in four sessions, each scheduled several days apart in different setups

[1] http://tensorflow.com

[2] https://www.idiap.ch/dataset/voicepa

[3] https://www.beat-eu.org/platform/

[4] Source code: https://pypi.python.org/pypi/bob.paper.isba2018-pad-dnn

Figure 1. voicePA database recording setup.

and environmental conditions such as background noises. Speech samples were recorded using three devices: laptop using microphone AT2020USB+, Samsung Galaxy S3 phone, and iPhone 3GS (see the setup in Figure 1). For more details, please refer to [13].

Table 2. Attack types in voicePA database.

Laptop replay	**Phone replay**	**Synthetic replay**
Laptop speakers, High quality speakers	Samsung Galaxy S3, iPhone 3GS & 6S	Speech synthesis, Voice conversion

The presentation attacks were generated with assumption that a verification system, which is considered to be attacked, is installed either on a laptop (with an internal built-in microphone), on Samsung Galaxy S3, or iPhone 3GS. The attacker is trying to gain access to this system by playing back to it a pre-recorded bona fide data or an automatically generated synthetic data using some playback device.

The following devices were used to playback the attacks (see Table 2): (i) direct replay attacks using a laptop with internal speakers and a laptop with external high quality speaker, (ii) direct replay attacks using Samsung Galaxy S3, iPhone 3G, and iPhone 6S phones, and (iii) replay of synthetic speech generated with text to speech and voice conversion algorithms. Attacks targeting verification system on the laptop are the same as the attacks in AVspoof database [13], while the attacks on Samsung Galaxy S3 and iPhone 3G phones are newer and are contained only in voicePA database.

The attacks were also recorded in three different noise environments: a large conference room, an empty office with window open, and a typical lab with closed windows. In total, voicePA contains 25 different types of presentation attacks.

All utterances (see Table 1) in voicePA database are split into three non-overlapping subsets: training or *Train* (real and spoofed samples from 4 male and 10 female participants), development or *Dev* (real and spoofed samples from 4 male and 10 female participants), and evaluation or *Eval* (real and spoofed samples from 5 male and 11 female participants).

Figure 2. Example of BioCPqD-PA database recording setup. All attacks were recorded in an acoustically isolated room.

2.2. BioCPqD-PA database

BioCPqD-PA [23] is a proprietary database and it contains video (audio and image) of 222 participants (124 males and 98 females) speaking different types of content, which include free speech, read text, and read numbers (credit card, telephone, personal ID, digits sequences and other numbers set). Recordings used different devices (laptops and smartphones), were performed in different environments, and in Portuguese language.

The subset used in this paper as bona fide samples consists of only the laptop part and include all participants. The recordings used 4 different laptops, took place at 3 different environments, including a quiet garden, an office, and a noisy restaurant, and were performed during 5 recording sessions[5]. In each session, 27 utterances with variable content were recorded.

The presentation attacks were recorded in an acoustically isolated room (see Figure 2) using 3 different microphones and 8 different loudspeakers, resulting in 24 configurations (see Table 3 for details). The total number of bona fide recordings is 27,253 and presentation attacks is 391,678. This database was split in three non-overlapping subsets (see Table 1), isolating pairs of microphones and loudspeakers in each subset (each microphone and loud-

[5]Not all subjects recorded 5 sessions, due to scheduling difficulties.

Table 3. Microphone/speaker pairs forming attack types in BioCPqD-PA database. **(T)**, **(D)**, and **(E)** indicate Train, Dev, and Eval sets.

Microphone \ Speaker	**Genius SP**	**Megaware**	**Dell A225**	**Edifier**	**Logitech S-150**	**Creative SBS20**	**Dell XPS L502X**	**Mackie**
1. Genius travel	A1-1 (T)	A1-2 (T)	A1-3 (T)	A1-4 (T)	A1-5 (D)	A1-6 (D)	A1-7 (D)	A1-8 (D)
2. Dell XPS L502X	A2-1 (D)	A2-2 (D)	A2-3 (D)	A2-4 (D)	A2-5 (E)	A2-6 (E)	A2-7 (E)	A2-8 (E)
3. Logitech USB	A3-1 (D)	A3-2 (D)	A3-3 (D)	A3-4 (D)	A3-5 (E)	A3-6 (E)	A3-7 (E)	A3-8 (E)

speaker pair belongs to only one subset), thus providing a protocol to evaluate the ability of a PAD system to generalize to unseen configurations. As shown in the Table 3, *Train* set contains 4 pairs of microphone and loudspeaker, *Dev* set contains 12 pairs, and *Eval* set 8 pairs. Additionally the protocol guarantees that Train and Eval sets do not contain any repeated microphone-loudspeaker pairs. There is no split among speakers, meaning that samples from all speakers are present in all subsets.

2.3. Evaluation protocol

In a single database evaluation, the *Train* set of a given database is used for training PAD system, the *Dev* set is used for selecting hyper-parameters and *Eval* set is used for testing. In a cross-database evaluation, the *Train* and *Dev* sets are taken from one database, while the *Eval* set is taken from another database.

For evaluation of PAD systems, the following metrics are recommended [8]: attack presentation classification error rate (APCER) and bona fide presentation classification error rate (BPCER). APCER is the number of attacks misclassified as bona fide samples divided by the total number of attacks, and is defined as follows:

$$\mathrm{APCER} = \frac{1}{N_{AT}} \sum_{i=1}^{N_{AT}} (1 - Res_i), \tag{1}$$

where N_{AT} is the number of attack presentations. Res_i takes value 1 if the i-th presentation is classified as an attack, and value 0 if classified as bona fide. Thus, APCER can be considered as an equivalent to FAR for PAD systems, as it represents the ratio of falsely accepted attack samples in relation to the total number of attacks.

BPCER is the number of incorrectly classified bona fide samples divided by the total number of bona fide samples:

$$\mathrm{BPCER} = \frac{1}{N_{BF}} \sum_{i=1}^{N_{BF}} Res_i, \tag{2}$$

where N_{BF} is the number of bona fide presentations, and Res_i is defined similar to APCER. Hence, BPCER can be considered as an equivalent to FRR for PAD systems.

In this paper's evaluations, when testing PADs on each database and in cross-database scenarios, we report EER rates on *Dev* set (when BPCER and APCER are equal) and separate BPCER and APCER values on *Eval* set using the EER threshold computed on the *Dev* set.

3. PAD systems

For the baseline state of the art PAD system, we selected a system based on mel-scale frequency coefficients (MFCC) [2] and Gaussian mixture model (GMM)-based classifier, which was shown to perform well by [18, 10, 11] in single and cross-database scenarios.

To compute MFCC features, a given audio sample is first split into overlapping 20ms-long speech frames with 10ms overlap. The frames are pre-emphasized with 0.97 coefficient and pre-processed by applying Hamming window. MFCC features are obtained from a power spectrum (512-sized FFT) by applying mel-scale filter of size 20 (as per [18]). A discrete cosine transform (DCT-II) is applied to the filtered values and first 20 coefficients are taken. Then, from the resulted coefficients, only deltas and delta-deltas [20] are kept (40 in total) for the classifier, as it has been reported in [18] that the static features degraded performance of PAD systems.

The classification is done using two separately trained GMM models, one for bona fide and one for attacks from the *Train* set. The trained models are then used to compute scores for *Dev* and *Eval* sets as the difference between log-likelihoods to the two GMM models. Each model is trained using 10 expectation-maximization (EM) iterations and has 512 Gaussians components.

3.1. Convolution neural networks

Two convolutional neural networks (CNNs) are designed and trained for speech presentation attack detection: a smaller network (denoted as 'CNN-Shallow' in this paper) and a deeper model (denoted as 'CNN-Deep') with more layers stacked up. The CNNs are implemented using Tensorflow framework. The architecture of both CNNs are presented in Figure 3. The number of neurons are shown at the top of each layer. These networks are by no means the best possible architectures for PAD, as it was not our goal to find such. Instead, we aim to understand whether CNNs, even such simple ones, would be a better alternative to the systems based on handcrafted features.

Unlike the traditional MFCC-GMM model, in a CNN model the discriminative features are learned jointly with the classification model. Hence, a raw waveform is used as an input to the model and the convolutional layers are responsible to build relevant features.

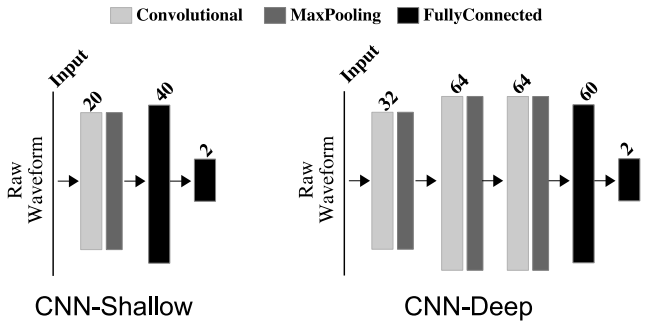


Figure 3. Architecture of the two CNNs designed for speech presentation attack detection. Two more convolutional layers and more neurons are added in CNN-Deep model.

In our CNN networks, the raw input audio is split into 20ms-long speech frames. The features vector consists of each frame plus its 20 left and right neighbors, resulting in 41 input frames.

In the CNN-Shallow network, the only convolutional layer contains 20 neurons, each with kernel=300 and stride=200, followed by a fully connected layer composed of 40 neurons. Both layers use hard tangent as an activation function. The output of convolutional layer is flattened for the fully connected layer input. The last layer has two neurons corresponding to the two output classes (bona fide and attacks). LogSoftMax function is applied to the output of the network before a negative log-likelihood loss function is computed. Gradient descent with constant learning rate 0.0001 is used to optimize the loss.

A deeper CNN (CNN-Deep) is a slightly larger network with three convolutional layers and we added it to analyze how increasing depth of CNN architecture impacts the PAD performance. The same raw data input and activation function are used as in the shallow network. The first convolutional layer has 32 neurons, each with kernel=160 and stride=20, followed by a max pooling layer (kernel=2 and stride=1). A second convolutional layer has 64 neurons (kernel=32 and stride=2) and the same max pooling layer. The third convolutional layer contains 64 neurons (kernel=1 and stride=1) followed by the same max pooling layer. The output of the last max pooling is flattened and connected to a fully connected layer of 60 neurons. The last layer is an output layer with 2 neurons-classes. Similarly to the shallow network, LogSoftMax function is applied to the outputs. For all convolutional layers, hard tangent activation function is used. Gradient descent with exponentially decay learning rate with base rate of 0.001 and decaying step 10000 is used for optimizing the negative log likelihood loss function.

4. Evaluation results

To evaluate the performance of CNN-based PAD systems, we first trained both CNN-Shallow and CNN-Deep networks, presented in the previous section, on training sets of voicePA and BioCPqD-PA databases. The two trained models (one for each database) were then used in two different capacities:

1. Use pre-trained models directly as classifiers on development and evaluation sets
2. Use models as feature extractors, by taking the output of the fully connected layer.

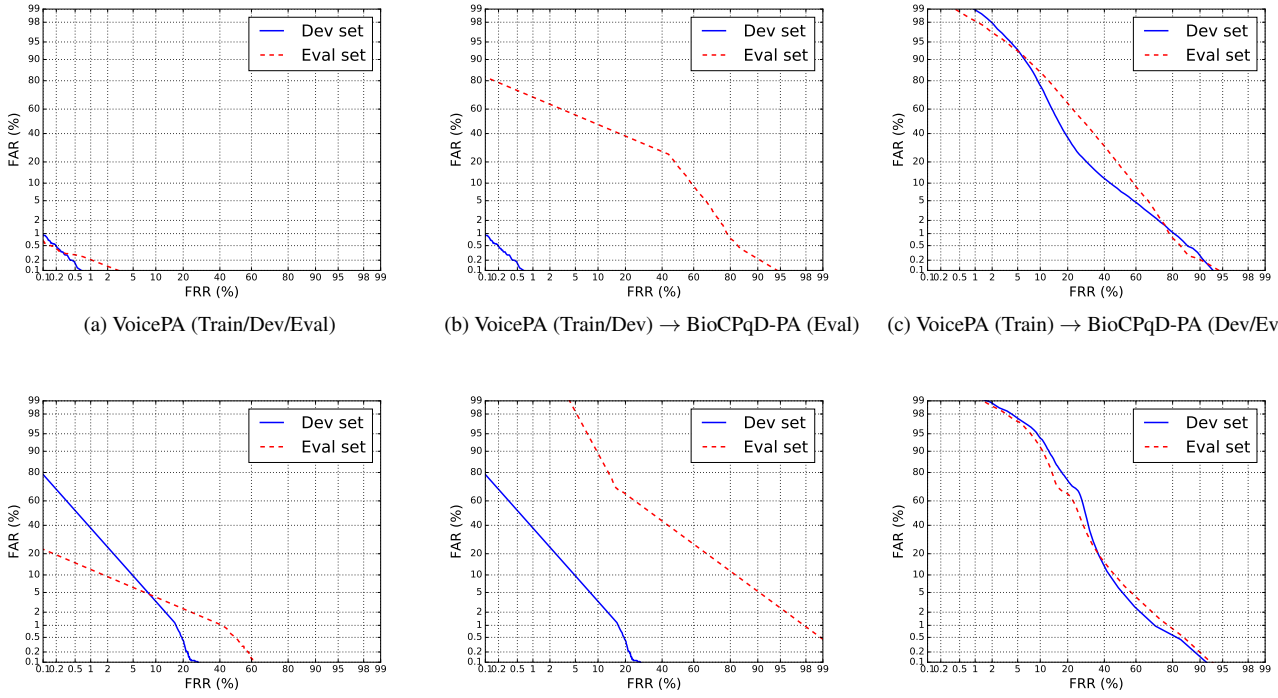
When used as a feature extractor, the feature vectors (40 values for CNN-Shallow model and 60 values for CNN-Deep model) are extracted for a training set and two GMM classifiers are trained (one for bona fide and one for attacks) in the same fashion as for MFCC-based PAD. Using the same GMM classifier allows us to understand the effectiveness of self-learned CNN-based features compared to the handcrafted MFCC features (with CNN-Shallow model, the number of features is also the same 40 as in MFCC-based PAD).

Table 4 demonstrates the evaluation results of four versions of CNN-based PAD systems and baseline MFCC-GMM based PAD using two databases voicePA and BioCPqD-PA. The first column of the table describes the combinations of the datasets used in each evaluation scenario and other columns contains the evaluation results (EER for Dev set with APCER and BPCER for Eval set) for each of the considered PAD system.

For instance, in the first row of Table 4, 'voicePA (Train/Dev/Eval)' means that the training set of voicePA was used to train the model of each evaluated PAD, the development set of voicePA was used to compute the EER value and the corresponding threshold, and this threshold was used to compute APCER and BPCER values on evaluation set from the same voicePA database. In the second row of Table 4, 'voicePA (Train/Dev) $\rightarrow$ BioCPqD-PA (Eval)' means that training and computation for development set were performed in the same way as for the system in the first row (hence, EER rate for Dev set is the same as in the first row), but the evaluation was done on the Eval set of BioCPqD-PA database instead. This cross-database evaluation simulates a practical scenario when a PAD system is built and tuned on one type of data but is deployed, as a black box, in a different setting and environment with different data. The last cross-database scenario is when only a pre-trained model is built using some existing data (a common situation in recognition), for instance from voicePA as in row 'voicePA (Train) $\rightarrow$ BioCPqD-PA (Dev/Eval)' of Table 4, but the system is tuned and evaluated on another data, e.g., from BioCPqD-PA.

Table 4. Performance of PAD systems in terms of EER (%) on Dev set, APCER (%) on Eval set, and BPCER (%) on Eval set of the scores for each voicePA and BioCPqD-PA databases and for different cross-database scenarios. **T**, **D** and **E** stand for Train, Dev, and Eval sets.

Combinations of datasets	**GMM-MFCC**			**GMM-CNN-Shallow**			**GMM-CNN-Deep**			**CNN-Shallow**			**CNN-Deep**		
	EER	APCER	BPCER	EER	APCER	BPCER	EER	APCER	BPCER	EER	APCER	BPCER	EER	APCER	BPCER
voicePA (T/D/E)	4.28	4.07	4.45	1.26	1.40	0.47	**0.26**	**0.18**	**0.39**	1.25	1.41	0.52	0.30	0.22	0.38
voicePA (T/D) → BioCPqD-PA (E)	4.28	96.18	8.89	1.26	48.65	54.40	0.26	50.45	13.76	1.25	79.59	3.49	0.30	75.69	1.73
voicePA (T) → BioCPqD-PA (D/E)	41.00	70.55	41.71	34.89	56.11	34.43	**19.98**	**46.84**	**19.93**	37.05	57.07	36.90	25.20	43.73	25.22
BioCPqD-PA (T/D/E)	41.00	70.55	41.71	11.39	22.45	11.09	7.34	24.09	7.14	11.69	23.73	11.48	**7.01**	**23.81**	**6.89**
BioCPqD-PA (T/D) → voicePA (E)	41.00	81.57	29.16	11.39	0.00	100.00	7.34	0.00	100.00	11.69	11.84	85.28	7.01	11.97	86.48
BioCPqD-PA (T) → voicePA (D/E)	50.19	37.73	47.31	**22.86**	**24.83**	**18.49**	37.04	39.59	32.08	33.29	34.02	26.54	32.23	32.52	26.15



(a) VoicePA (Train/Dev/Eval) (b) VoicePA (Train/Dev) → BioCPqD-PA (Eval) (c) VoicePA (Train) → BioCPqD-PA (Dev/Eval)

(d) BioCPqD-PA (Train/Dev/Eval) (e) BioCPqD-PA (Train/Dev) → VoicePA (Eval) (f) BioCPqD-PA (Train) → VoicePA (Dev/Eval)

Figure 4. DET curves of calibrated scores of CNN-Deep system in different evaluation scenarios (see the corresponding rows in Table 4).

The results in Table 4 demonstrate several important findings. First, it is clear that CNN-based PADs perform significantly better compared to MFCC-based PAD. This is especially evident in individual database evaluations, with 'CNN-Deep' variants showing more than 10 times lower error rates compared to MFCC-based PAD for voicePA database and a few times lower for BioCPqD-PA database. Then, deeper CNN models perform generally better compared to shallow variants. Also, using CNN models as feature extractors coupled with a GMM classifier can be beneficial and can lead to an increase in accuracy, though the increase is not as significant compared to the larger computational resources GMM-CNN based systems require.

It is important to note that CNN-based systems do not generalize well across databases, although, in the scenario when only a model is pre-trained on another database, CNNs are more stable and significantly more accurate compared to MFCC-based PAD. However, if the system is both trained and tuned (threshold is chosen) on the same database but is evaluated on another database, CNN-based systems completely fail just as MFCC-based systems.

To illustrate the performance of CNN-based systems in more details, we also plot detection error tradeoff (DET) curves for a 'CNN-Deep' system in Figure 4. You can notice the large gap between the curves for Dev and Eval sets in the Figure 4b and Figure 4e, when both training and threshold tuning is performed on one database but evaluation is done on another.

Although none of the considered PAD systems generalize well across different databases, it is also important to understand how they perform on different types of attacks, including *unknown* attacks, for which the systems were not

Table 5. Per attack APCER results for Eval sets of voicePA and BioCPqD-PA databases.

Types of attacks	GMM-MFCC	GMM-CNN-Shallow	GMM-CNN-Deep	CNN-Shallow	CNN-Deep
voicePA, laptop replay	74.19	20.19	7.19	20.12	8.94
voicePA, phone replay	51.00	2.93	0.83	2.73	0.91
voicePA, synthetic replay	0.01	1.05	0.04	1.08	0.06
BioCPqD-PA, A2-5	71.42	5.81	30.86	3.93	28.91
BioCPqD-PA, A2-6	42.65	1.86	25.10	1.04	23.22
BioCPqD-PA, A2-7	77.01	0.00	0.34	0.00	0.31
BioCPqD-PA, A2-8	76.93	53.94	14.52	60.60	13.68
BioCPqD-PA, A3-5	73.96	2.05	3.80	1.59	4.41
BioCPqD-PA, A3-6	36.67	0.02	0.10	0.02	0.10
BioCPqD-PA, A3-7	68.72	43.38	74.04	43.87	73.67
BioCPqD-PA, A3-8	70.17	0.76	0.47	0.86	0.63

trained. This analysis can help us understand which types of presentation attacks are more challenging. In this scenario, PAD systems are trained, tuned, and evaluated on the same database, only error rates are computed for specific attacks. Thus, we computed APCER value separately for each type of attacks in Eval sets of voicePA and BioCPqD-PA database. Note that EER and BPCER values do not change, since EER is computed on the whole development set and BPCER only measures the detection of bona fide utterances.

The results for different types of attacks detailed in Table 2 and Table 3 are shown in Table 5. It is important to note that in case of voicePA, the same attacks are present in all training, development, and evaluation sets (data is split by speakers), so voicePA does not contain *unknown* attacks. However, in BioCPqD-PA, different types of attacks are distributed into Train, Dev, and Eval sets differently (see Table 3), so that all attacks in Eval set are basically *unknown* to the PAD systems.

The results in Table 5 for voicePA database demonstrate that using high quality speakers as a replay device (see 'voicePA, laptop replay' row of the table) lead to significantly more challenging attacks compared to attacks replayed with mobile phone (see row 'voicePA, phone replay'). Also, synthetic speech poses considerably lesser challenge to PAD systems compared to the replay of natural speech. Also, note that we did not consider different environments and ASV systems (different microphones) for each of these types of attacks in voicePA, we only separate different speakers and natural speech from synthetic.

The attacks in BioCPqD-PA, however, are formed by combining different pairs of speakers (attack devices) and microphones of ASV systems, while influence of environment and types of speech were excluded, since acoustically isolate room was used and attacks were recorded by replaying natural speech only. Results in Table 5 for BioCPqD-PA, show the significance of the choice for both speaker, with which attacks is made, and the microphone of the attacked ASV system. For instance, the same microphone is used in attacks 'A3-6' and 'A3-7' (see attacks details in Table 3) but the difference in speakers can lead to drastically different detection results, as 'A3-6' is easily detected by all CNN-based PAD systems, while all were spoofed by 'A3-7'. Similarly, the results of the CNN-Shallow and the CNN-Deep substantially vary across different pairs of speakers and microphones, e.g., for pairs 'A2-5', 'A2-6', 'A2-8', and 'A3-7'. These differences may be due to different features learned by each neural network, as the model learns the features directly from the audio signal. Therefore, changing the neural network architecture will possibly affect the features learned and consequently the results.

5. Conclusion

In this paper, we investigated whether, for speech presentation attack detection, a convolutional neural network (CNN) is a better alternative to the systems based on handcrafted features. The evaluation results on two large voicePA and BioCPqD-PA databases demonstrate that CNNs can achieve up to the order of magnitude better performance compared to MFCC-based systems. The performance in cross-database scenario is not as impressive, although, using a network model pre-trained on an external data could be a possibility.

More work needs to be done, especially, in the direction of understanding which CNN architectures would work best for speech presentation attack detection, and how to find such architecture, including determining good loss and activation functions, cost optimizers, learning rate techniques, dropouts, etc. We also would like to explore whether transfer learning can improve the performance in the cross-database scenario. Fusion techniques are also interesting to investigate.

Acknowledgements

This work was partially funded by Norwegian SWAN project, EU H2020 project TeSLA, and Swiss Centre for Biometrics Research and Testing. Also, the authors would like to thank Tiago de Freitas Pereira for the help with Tensorflow implementation and Hannah Muckenhirn for the help with initial CNN architectures.

References

[1] F. Alegre, A. Amehraye, and N. Evans. A one-class classification approach to generalised speaker verification spoofing countermeasures using local binary patterns. In *IEEE International Conference on Biometrics: Theory, Applications and Systems*, pages 1–8, 2013.

[2] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, 1980.

[3] P. L. De Leon, M. Pucher, J. Yamagishi, I. Hernaez, and I. Saratxaga. Evaluation of speaker verification security and detection of HMM-based synthetic speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(8):2280–2290, 2012.

[4] J. Galka, M. Grzywacz, and R. Samborski. Playback attack detection for text-dependent speaker verification over telephone channels. *Speech Communication*, 67:143 – 153, 2015.

[5] A. Godoy, F. O. Simões, J. A. Stuchi, M. de Assis Angeloni, M. Uliani, and R. Violato. Using deep learning for detecting spoofing attacks on speech signals. *CoRR*, abs/1508.01746, 2015.

[6] A. R. Goncalves, P. Korshunov, R. P. V. Violato, F. O. Simões, and S. Marcel. On the generalization of fused systems in voice presentation attack detection. In A. Brömme, C. Busch, A. Dantcheva, C. Rathgeb, and A. Uhl, editors, *16th International Conference of the Biometrics Special Interest Group*, Darmstadt, Germany, Sept. 2017.

[7] ISO/IEC JTC 1/SC 37 Biometrics. DIS 30107-1, information technology – biometrics presentation attack detection. American National Standards Institute, 2016.

[8] ISO/IEC JTC 1/SC 37 Biometrics. DIS 30107-3:2016, information technology – biometrics presentation attack detection — part 3: Testing and reporting. American National Standards Institute, Oct. 2016.

[9] A. Janicki. Spoofing countermeasure based on analysis of linear prediction error. In *INTERSPEECH*, pages 2077–2081, 2015.

[10] P. Korshunov and S. Marcel. Cross-database evaluation of audio-based spoofing detection systems. In *INTERSPEECH*, pages 1705–1709, Sept. 2016.

[11] P. Korshunov and S. Marcel. Impact of score fusion on voice biometrics and presentation attack detection in cross-database evaluations. *IEEE Journal of Selected Topics in Signal Processing*, 11(4):695–705, June 2017.

[12] P. Korshunov, S. Marcel, H. Muckenhirn, A. R. Gonalves, A. G. S. Mello, R. P. V. Violato, F. O. Simoes, M. U. Neto, M. de Assis Angeloni, J. A. Stuchi, H. Dinkel, N. Chen, Y. Qian, D. Paul, G. Saha, and M. Sahidullah. Overview of BTAS 2016 speaker anti-spoofing competition. In *IEEE International Conference on Biometrics: Theory, Applications and Systems*, pages 1–6, Niagara Falls, NY, USA, 2016.

[13] S. Kucur Ergunay, E. Khoury, A. Lazaridis, and S. Marcel. On the vulnerability of speaker verification to realistic voice spoofing. In *IEEE International Conference on Biometrics: Theory, Applications and Systems*, pages 1–6, 2015.

[14] D. Luo, H. Wu, and J. Huang. Audio recapture detection using deep learning. In *IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, pages 478–482, 2015.

[15] H. Muckenhirn, M. Magimai.-Doss, and S. Marcel. End-to-end convolutional neural network-based voice presentation attack detection. In *International Joint Conference on Biometrics*, 2017.

[16] T. B. Patel and H. A. Patil. Combining Evidences from Mel Cepstral, Cochlear Filter Cepstral and Instantaneous Frequency Features for Detection of Natural vs. Spoofed Speech. In *INTERSPEECH*, pages 2062–2066, 2015.

[17] Y. Qian, N. Chen, and K. Yu. Deep features for automatic spoofing detection. *Speech Commun.*, 85(C):43–52, Dec. 2016.

[18] M. Sahidullah, T. Kinnunen, and C. Hanilçi. A comparison of features for synthetic speech detection. In *INTERSPEECH*, pages 2087–2091, 2015.

[19] S. Shiota, F. Villavicencio, J. Yamagishi, N. Ono, I. Echizen, and T. Matsui. Voice liveness detection algorithms based on pop noise caused by human breath for automatic speaker verification. In *INTERSPEECH*, pages 239–243, 2015.

[20] F. K. Soong and A. E. Rosenberg. On the use of instantaneous and transitional spectral information in speaker recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(6):871–879, 1988.

[21] X. Tian, X. Xiao, E. S. Chng, and H. Li. Spoofing speech detection using temporal convolutional neural network. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1–6, Dec 2016.

[22] M. Todisco, H. Delgado, and N. Evans. Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification. *Computer Speech & Language*, Feb. 2017.

[23] R. Violato, M. U. Neto, F. Simões, T. Pereira, and M. Angeloni. BioCPqD: uma base de dados biométricos com amostras de face e voz de indivíduos brasileiros. *Cadernos CPqD Tecnologia*, 9(2):7–18, 2013.

[24] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li. Spoofing and countermeasures for speaker verification: A survey. *Speech Communication*, 66:130–153, Feb. 2015.

[25] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah, and A. Sizov. ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge. In *INTERSPEECH*, pages 2037–2041, 2015.

[26] Z. Wu, C. E. Siong, and H. Li. Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition. In *INTERSPEECH*, 2012.

[27] Z. Wu, X. Xiao, E. S. Chng, and H. Li. Synthetic speech detection using temporal modulation feature. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7234–7238, May 2013.

[28] C. Zhang, C. Yu, and J. H. L. Hansen. An investigation of deep-learning frameworks for speaker verification antispoofing. *IEEE Journal of Selected Topics in Signal Processing*, 11(4):684–694, June 2017.