Interspeech 2018
2-6 September 2018, Hyderabad

Deep Siamese Architecture Based Replay Detection for Secure Voice Biometric

Kaavya Sriskandaraja[1,2], *Vidhyasaharan Sethu*[1] , *Eliathamby Ambikairajah*[1,2]

[1]School of Electrical Engineering and Telecommunications, UNSW Australia
[2]DATA61, CSIRO, Sydney, Australia
k.sriskandaraja@unsw.edu.au, v.sethu@unsw.edu.au, e.ambikairajah@unsw.edu.au

Abstract

Replay attacks are the simplest and the most easily accessible form of spoofing attacks on voice biometric systems and can be hard to detect by systems designed to identify spoofing attacks based on synthesised speech. In this paper, we propose a novel approach to evaluate the similarities between pairs of speech samples to detect replayed speech based on a suitable embedding learned by deep Siamese architectures. Specifically, we train a deep Siamese network to identify pairs of genuine speech samples and pairs of replayed speech samples as being 'similar' and mixed pairs of genuine and replayed speech to be identified as 'dissimilar'. Siamese networks are particularly suited to this task and have been shown to be effective in problems where intra-class variability is large and the number of training samples per class is relatively small. The internal low-dimensional embedding learnt by the Siamese network to accomplish this task is then used as the basis for replay detection. The proposed approach outperforms state-of-the-art systems when evaluated on the ASVspoof 2017 challenge corpus without relying on fusion with other sub-systems.

Index Terms: voice biometrics, anti-spoofing, Siamese, deep learning, speech recognition, human-computer interaction

1. Introduction

The vulnerability of state-of-the-art voice biometric systems to spoofing attacks is well-defined [1] and development of suitable countermeasures is an active area of research [2]–[4]. Spoofing attacks on voice biometric systems fall into one of four categories, based on the practicality and methodology of the attack: impersonation [5], speech synthesis [6], voice conversion [7] and the afore-mentioned replay attacks [8]. Among these, the lack of a need for sophisticated technology makes replay attacks the simplest and easily accessible form of attack. A replay attack comprises of recording the speech of a claimant and playing it back to the voice authentication system to spoof it. Studies on assessing the vulnerability of state-of-the-art automatic speaker verification (ASV) systems to replay attacks concretely show that replay attacks are highly effective, evidenced by significant increases in both equal error rate (EER) and false acceptance rate (FAR) were observed [1].

Current countermeasures against replay attacks generally operate in one of three ways: (a) identifying exact reproduction of a previous access attempt; (b) exploiting differences in the speech transmission channel [9]; or (c) targeting artefacts in replayed speech such as pop-noise [10], and source features [11]. Commonly used spectral features include sub-band spectral centroid magnitude coefficients (SCMCs) [12], constant-Q cepstral coefficients (CQCCs) [13], single frequency filtering cepstral coefficients (SFF-CCs) [14], inverse Mel frequency cepstral coefficients (IMFCCs) [15], rectangular filter cepstral coefficients (RFCCs) [15], and scattering decomposition based features [16]. In addition, deep neural network (DNN) architectures have also been employed either as discriminative feature extractors [17] or as an end-to-end spoofing detectors [18] in a number of ways. Some systems have utilized low-level cepstral features, such as CQCCs, RFCCs and MFCCs, to learn tandem features which are proposed and aimed to maximise the channel variability [19], and build residual neural networks (ResNet) [20] and deep convolutional neural networks (CNNs) [17]. In the place of the hand-crafted features, raw spectrograms are often used as an input for CNNs, recurrent neural networks (RNNs) and ResNet [17]. The best performing spoofing detector so far utilises a reduced version of Light-CNN architecture (LCNN) [21] using the max-feature-map (MFM) activation, which is based on max-out activation function.

However, the learning of sufficient features for each class requires a large amount of training data or the neural networks may suffer from over-fitting. A key challenge in replay detection is that a replayed version of a genuine speech (replayed speech) does not contain obvious variations other than additional channel factors, which may also be present in different sessions of genuine speech.

In this work, we aim to address the afore-mentioned challenges in the field replay spoofing detection using a novel approach that uses the concept of a Siamese architecture. A Siamese architecture takes a pair of inputs and produces a similarity score, indicating whether the two inputs come from the same class. Siamese networks have been shown to be able to handle the indistinctive inter-class differences and large intra-class variations, and to be suitable for scenarios where the amount of training samples for a class is very small [22].

2. Siamese Architecture

The concept of Siamese architecture was first developed in 1993 to tackle the signature verification problem by Bromley and LeCun [23]. Siamese architectures are a class of network architectures that usually contains two identical subnetworks (twins) as shown in Figure 1. The Siamese architecture focuses on learning an embedding (with deeper layers) that places inputs of the same class close together. Hence, it can learn the similarities within each particular class, which makes the embedding more useful in a generic sense. The training process minimizes a discriminative loss function that drives the similarity metric to be small for pairs of inputs from the same class (pair of inputs are 'similar'), and large for pairs from different classes (pair of inputs are 'dissimilar'). As depicted in Figure 1, in a Siamese network, two inputs $\boldsymbol{I_p}$ and $\boldsymbol{I_q}$ are taken in parallel. These inputs are simultaneously fed

10.21437/Interspeech.2018-1819

into Siamese networks (for each sub-network) that translate each input into a latent encoding space, or 'embedding', $\boldsymbol{x_p}$ and $\boldsymbol{x_q}$ respectively. The distance between these embeddings will reflect the abstract similarity between the two inputs. The parameters of the twin networks are tied together, which means that each sub-network is trained such that they share weights. Weight tying guarantees that two extremely similar inputs could not possibly be mapped by their respective networks to very different locations in feature space because each network computes the same function [24].

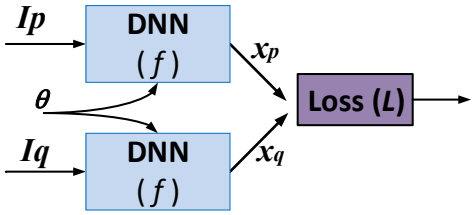

Figure 1*: A typical Siamese architecture concept involving two identical sub-networks, which share same network parameter, $\boldsymbol{\theta}$, to translate the inputs $\boldsymbol{I_p}$ and $\boldsymbol{I_q}$ to the embedding $\boldsymbol{x_p}$ and $\boldsymbol{x_q}$ with loss function, L*

The Siamese network can be represented mathematically as a function f that maps each input $\boldsymbol{I}$ into an embedding x, given parameters θ of the form

$$\boldsymbol{x} = f(\boldsymbol{I}; \boldsymbol{\theta}) \quad (1)$$

The parameter vector $\boldsymbol{\theta}$ contains all the weights and biases for the inner product layers. The aim is then to estimate the parameter vector $\boldsymbol{\theta}$ such that the embedding $\boldsymbol{x}$ produced through f has desirable properties and places 'similar' inputs nearby. If the network has learnt a good embedding, we would find that $\boldsymbol{x_p}$ and $\boldsymbol{x_q}$ are close to each other for inputs of the same class, while they would be further apart for different class inputs. These embeddings, $\boldsymbol{x}$, can be obtained by any DNN architecture. The sub-nets are joined by a loss function, *L*, at the right, which measures how well f is able to place 'similar' inputs nearby and keep 'dissimilar' inputs further apart by computing a similarity metric involving any distance measure (such as Euclidean distance [25], cosine distance [26], etc) between embeddings $\boldsymbol{x_p}$ and $\boldsymbol{x_q}$. One such loss function that is often used in Simease is contrastive loss [27] defined as:

$$L(\boldsymbol{I_p}, \boldsymbol{I_q}, l) = l * \boldsymbol{D}(\boldsymbol{x_q}, \boldsymbol{x_p}) + (\boldsymbol{1} - \boldsymbol{l}) * max(m, \boldsymbol{D}(\boldsymbol{x_q}, \boldsymbol{x_p})) \quad (2)$$

where $l \in \{0,1\}$ is a label indicating if the input classes match or not, and $\boldsymbol{D}(\boldsymbol{x_q}, \boldsymbol{x_p})$ is any distance measure (measure of similarity) between $\boldsymbol{x_p}$ and $\boldsymbol{x_q}$. m is the margin equal to 1 in most cases. There is no single standard architecture for Siamese networks. Design is largely governed by what performs well empirically for the task at hand. There are many Siamese network variants reported in the literature for different applications, such as authorship verification [28], signature verification [23], face recognition [29], person re-identification [30], image recognition [24], sentence matching[31], inertial gesture classification [26], etc. In the speech domain, variants of Siamese networks have been used for audio-visual synchrony detection [32], learning speaker and phonetic similarities [33], and to extract speaker specific information [34].

3. Spoofing Detection Based on Siamese Architecture

3.1. Proposed Architecture

To learn a suitable speech signal embedding, $\boldsymbol{x}$, for spoofing detection, based on the concept of Siamese network, we re-formulated the task of spoofing detection as follows: given a pair of input speech utterances, the network is trained to identify genuine-genuine speech or spoof-spoof speech pairs as 'similar' inputs and genuine-spoof speech pairs as 'dissimilar' inputs.

Although Siamese networks can be trained to maximize the distance between 'dissimilar' pairs and minimize the distance between 'similar' pairs of inputs, using a distance measure of some kind (e.g. Euclidean distance), this model requires searching over multiple distance functions as well as different thresholds for distances to find optimal parameters of network. Initial investigations were made into various distance functions, but they did not perform well on this dataset/task. Furthermore, even if we were able to achieve good results with such distance measures, the "real" distance measure for the input space remains uncertain. To tackle this issue, As shown Figure 2, we chose to train our Siamese network by outputting a softmax layer over the two targets: 'similar' and 'dissimilar', thus allowing the model to learn the representation and distance function that best separates the genuine and spoof classes. This is one variant of Siamese network used in [28], 'Siamese-Classification Hybrid architecture' [35].

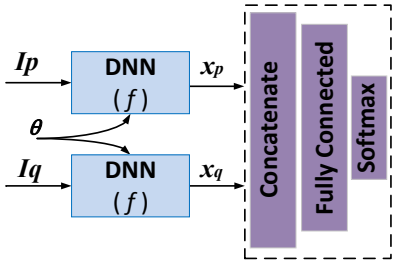

Figure 2: *Schematic diagram of Siamese-Classification Hybrid architecture, incorporating concatenation followed by fully connected layers and softmax function*

3.2. Genuine vs Replayed speech classification

The embedding learnt by the Siamese network, can be utilised as a feature by any suitable back-end to distinguish between genuine and spoofed speech. In this paper we employ a GMM based back-end for this purpose. Alternatively a fully connected back-end can also be employed if an end-to-end system is desired.

3.3. Experimental Setup

3.3.1. Corpus

The experiments reported in this paper were conducted on the ASVspoof 2017 challenge dataset [36]. All systems considered in this paper were trained using pooled training and

development sets. The development set was used for performance validation, parameter tuning and weight adjustment for the neural network systems, and the final model is then re-trained using the pooled data. With the objective of measuring the limits of replay attack detection, the ASVspoof 2017 database was designed to contain a diverse range of replay configurations ranging from conditions for which the detection of replay attacks should be relatively straightforward, to those for which detection should be considerably challenging.

3.3.2. *Front-end*

In order to compare with the LCNN based state-of-the-art spoofing detection system [17], we utilise the same features as the Light CNN system. Specifically, the mean and variance normalized log power spectrum, obtained via fast Fourier transform (FFT), were used as CNN input acoustic features for the sub-networks. Truncated normalized FFT spectrograms of size 864 × 400 × 1 were used [17] as the inputs of the first convolutional layer of each branch of the Siamese network. Spectrogram is trimmed to have 400 frames per utterance. Short utterances are extended by repeating their contents if necessary to match the required length, 400. Figure 4 shows the corresponding mean variance normalized spectrograms of a genuine and replayed version of "Birthday parties have cupcakes and ice-creams", which shows that the spectrogram with this configuration has noticeable differences between genuine and replayed speech.

3.3.3. *Data Preparation*

The training of the Siamese-Classification Hybrid network is carried out in one-go, with pairs of inputs taken from 'train' and 'dev' set of ASVspoof 2017 corpus [36]. To construct the pair-wise dataset from ASVspoof 2017 to train our model, input pairs were produced by pairing each utterance, while ensuring that they are the same phrase. For 'similar' match, we make pairs of genuine-genuine and spoof-spoof utterance from same phrase. For 'dissimilar' match, we pair each genuine with all the spoof utterances. This yield around totally 595,600 training trials, with equal numbers of 'similar' and 'dissimilar' pairs.

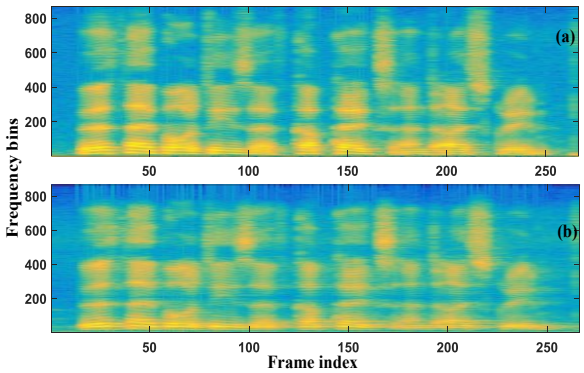

Figure 4: *Spectrograms of phrase: "Birthday parties have cupcakes and Ice-cream", showing (a) genuine speech, and (b) replayed speech of a speaker*

3.3.4. *Training the Siamese network*

The optimization objective is the average loss over all pairs in the data set. The output is mapped to [0,1] using a sigmoid function to make it a probability. Labels $l = 1$ indicates the 'similar' pairs (genuine-genuine pair or spoof-spoof pair), and $l = 0$ indicates the 'dissimilar' pairs (genuine-spoof pair). The network is trained with logistic regression, where the loss function is the binary cross entropy between the predictions and targets. The training data is randomly divided into mini-batches of 128 utterances, making sure that each mini-batch contained 50% 'similar' and 50% 'dissimilar' pairs.

Configuration and parameter details of proposed network is depicted in Figure 3 which consists of 5 convolution layers (CONV) (adjoined with dropout of 0.2, ReLu and max pooling layer (Pool)) followed by 2 fully connected (FC) layers (succeeding with dropout of 0.7, 0.5 respectively, and ReLu) for a sub-net. The output of each sub-net (embedding) is then fed into a concatenation layer (adjoined with batch-normalization followed by dropout of 0.2) and two fully connected layers (succeeding with dropout of 0.2 and ReLu) before the softmax output layer. Our concatenation layer is performing normal concatenation of two embeddings (x_p and $\boldsymbol{x_q}$) as follows:

$$concat(\boldsymbol{x_p}, \boldsymbol{x_q}) = [\boldsymbol{x_p}, \boldsymbol{x_q}] \quad (3)$$

A regularized cross-entropy objective is imposed as in

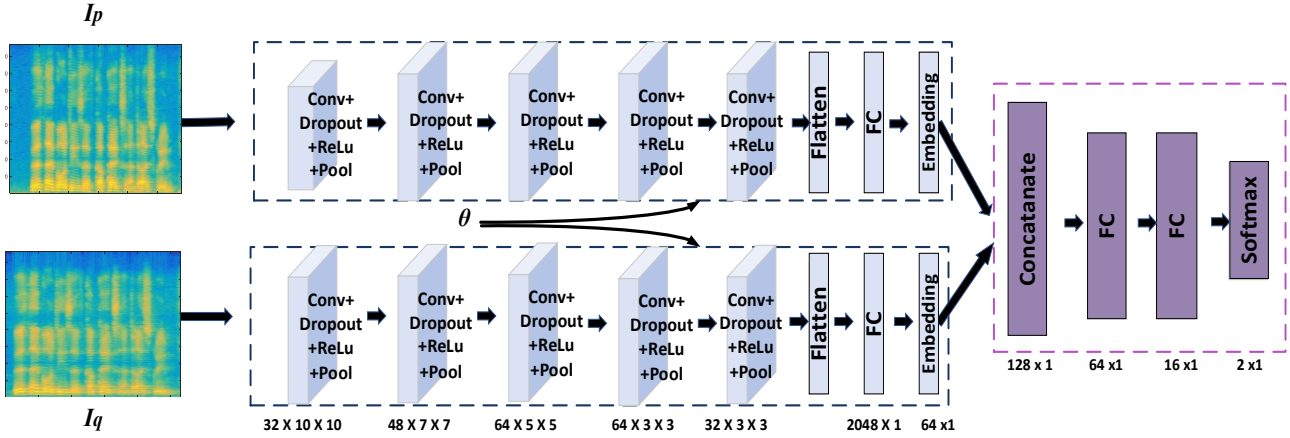

Figure 3: Overview of proposed Siamese - Classification Hybrid Architecture which includes the system parameters. The size of each convolution (Conv) layers are indicated in the order of (no of filters X filter size). $\boldsymbol{\theta}$ contains all the weights and biases for the inner product layers

[24]. Learning rates were decayed uniformly across the network by 1 percent per epoch. It was proved that by annealing the learning rate, the network was able to converge to local minima more easily without getting stuck in the error surface [37] and we also observed that. The development set is used to evaluate intermediate models and select the one that has maximum performance. All network weights are initialized in the convolutional layers from a normal distribution with zero-mean and a standard deviation of 10^{-2}, as described in [24]. Biases were also initialized from a normal distribution with mean 0.5 and standard deviation 10^{-2}. In the fully-connected layers, the biases were initialized in the same way as the convolutional layers, but the weights were drawn from a much wider normal distribution with zero-mean and standard deviation 2×10^{-1}. Layer wise L2 regularization was also performed.

3.4. Experimental Results and Discussion

At first, the n-dimensional embedding vector $\boldsymbol{x}$ for each utterance is extracted from the trained Siamese network. In our case $n = 64$. In order to compare the system directly to a baseline, a 2-class GMM back-end was used to obtain the log-likelihood ratios between genuine and replayed speech. The GMM back-end was implemented as the maximum likelihood estimates using 4 mixture components.

The primary metric for evaluation is the equal error rate (%EER). Table 1 compares the performance of the proposed Siamese architecture based spoofing detection with the baseline, and state-of-the-art spoofing detection systems on ASVspoof 2017 challenge corpus. To have the direct comparison with the existing systems we also report our results for the ASVspoof 2017 challenge corpus, version 1.0.

Table 1: *Comparison of proposed Siamese architecture based spoofing detection systems with existing systems on the evaluation set of ASVspoof 2017 corpus.*

	System	%EER
Baseline Systems	**LCNN + GMM** [17]	7.37
	LCNN + RNN + CNN + GMM + i-vector SVM [17]	6.73
	CNN + RNN + GMM [17]	10.69
	HFCC + CQCC+ DNN + SVM [38]	11.50
	SCMC + GMM [15]	11.49
	RFCC + GMM [15]	11.90
	Proposed Siamese embedding features + GMM	6.40

It should also be noted that since the Siamese network is trained on pairs of utterances, the number of training examples is in the order of the square of the number of speech samples in the database, which is beneficial given the number of parameters in the network. We examined different CNN configurations to determine which architecture provided the most informative encoding that was able to differentiate between genuine speech from replayed speech. The described CNN configuration was empirically found to be the best one. Further improvement of this Siamese architecture could incorporate the different architectures (such as ResNet, recurrent neural network (RNN)), various way of concatenating the embedding and adapt pre-trained models (such as VGGnet, GoogLeNet, etc), which will be explored in the future.

While it is straightforward to envision a replay detection scheme that directly uses the trained Siamese network by using it to compare the test utterance with known genuine and spoofed speech from the training set, the scoring would have had to be carried out against a large number of known samples making the approach expensive and wasteful. Instead, the proposed approach of using a Siamese network to learn a suitable embedding and employing that as a front-end is expected to be a more practical approach. In order to further understand that the embedding, $\boldsymbol{x}$, and to verify that it can encode the dissimilarity between genuine and spoofed speech, we have plotted a 2 dimensional t-SNE (t-distributed stochastic neighbour embedding [39]) plot of the embedding features, extracted for 'train' set (*Figure 5*). From this figure, it is obvious that the class of 'genuine' and 'spoof' are indeed separable in the embedding space.

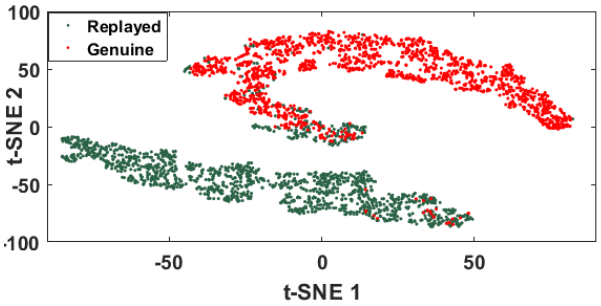

Figure 5: *2-dimensional t-SNE plot of the embedding features, extracted for 'train' set of ASVspoof 2017 corpus.*

4. Conclusions

In this paper, a novel deep leaning approach for spoofing detection based on the concept of Siamese architecture is proposed. The key idea is learn feature embeddings optimised for identifying if pairs of inputs that are 'similar' or 'dissimilar' and use this as a front-end for detecting replayed speech. In this work, the embedding is learnt by jointly tuning two identical deep convolutional neural networks (CNNs), which is trained such that they share weights, linked by concatenating the embeddings and trained using a categorical cross-entropy loss function. The proposed architecture allows a single system to outperform all the current state-of-the-art replay spoofing detection systems. This framework provides a novel avenue for developing optimised front-ends.

5. References

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

[2] X. Tian, Z. Wu, X. Xiao, E. S. Chng, and H. Li, "Spoofing detection under noisy conditions: a preliminary investigation and an initial database," 2016.

[3] K. Sriskandaraja, V. Sethu, E. Ambikairajah, and H. Li, "Front-end for antispoofing countermeasures in speaker verification: Scattering spectral decomposition," *IEEE J. Sel. Top. Signal Process.*, vol. 11, no. 4, pp. 632–643, 2017.

[4] G. Suthokumar, K. Sriskandaraja, V. Sethu, C. Wijenayake, and E. Ambikairajah, "Independent Modelling of High and Low Energy Speech Frames for Spoofing Detection," in *Interspeech*, 2017, pp. 2606–2610.

[5] R. G. Hautamäki, T. Kinnunen, V. Hautamäki, and A.-M. Laukkanen, "Automatic versus human speaker verification: The case of voice mimicry," *Speech Commun.*, vol. 72, pp. 13–31, 2015.

[6] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Commun.*, vol. 51, no. 11, pp. 1039–1064, Nov. 2009.

[7] M. Joana and R. Folgado, "Anti-spoofing : Speaker verification vs . voice conversion," *Thesis*, no. April, 2014.

[8] T. Freitas Pereira, J. Komulainen, A. Anjos, J. De Martino, A. Hadid, M. Pietikäinen, and S. Marcel, "Face liveness detection using dynamic texture," *EURASIP J. Image Video Process.*, vol. 2014, no. 1, p. 2, 2016.

[9] W. Shang and M. Stevenson, "Score normalization in playback attack detection," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pp. 1678–1681, 2010.

[10] Z. F. Wang, G. Wei, and Q. H. He, "Channel pattern noise based playback attack detection algorithm for speaker recognition," *Proceedings of International Conference on Machine Learning and Cybernetics*, vol. 4. pp. 1708–1713, 2011.

[11] A. Janicki, "Spoofing Countermeasure Based on Analysis of Linear Prediction Error," *Sixt. Annu. Conf. Int. Speech Commun. Assoc.*, pp. 2077–2081, 2015.

[12] C. Hanilçi, T. Kinnunen, M. Sahidullah, and A. Sizov, "Spoofing Detection Goes Noisy: An Analysis of Synthetic Speech Detection in the Presence of Additive Noise," *Speech Commun.*, vol. 85, pp. 83–97, 2016.

[13] M. Todisco, H. Delgado, and N. Evans, "A New Feature for Automatic Speaker Verification Anti-Spoofing: Constant Q Cepstral Coefficients," in *Odyssey*, 2016, pp. 283–290.

[14] K. N. R. K. Raju Alluri, S. Achanta, S. R. Kadiri, S. V. Gangashetty, and A. K. Vuppala, "Detection of replay attacks using single frequency filtering cepstral coefficients," in *Interspeech*, 2017, pp. 2596–2600.

[15] R. Font, J. M. Espín, and M. J. Cano, "Experimental analysis of features for replay attack detection-Results on the ASVspoof 2017 Challenge," in *Interspeech*, 2017, pp. 7–11.

[16] K. Sriskandaraja, G. Suthokumar, V. Sethu, and E. Ambikairajah, "Investigating the use of scattering coefficients for replay attack detection," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2017, pp. 1195–1198.

[17] G. Lavrentyeva, S. Novoselov, E. Malykh, A. Kozlov, O. Kudashev, and V. Shchemelinin, "Audio replay attack detection with deep learning frameworks," in *INTERSPEECH*, 2017, vol. 2017–Augus, pp. 82–86.

[18] H. Dinkel, N. Chen, Y. Qian, and K. Yu, "End-To-End Spoofing Detection with Raw Wavform CLDNNs," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2017, pp. 4860–4864.

[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*, 2015, pp. 239–243.

[20] W. Cai, D. Cai, W. Liu, G. Li, and M. Li, "Countermeasures for automatic speaker verification replay spoofing attack : on data augmentation, feature representation, classification and fusion," in *Interspeech*, 2017, pp. 17–21.

[21] X. Wu, R. He, Z. Sun, and T. Tan, "A Light CNN for Deep Face Representation with Noisy Labels," Nov. 2015.

[22] C. Zhang, W. Liu, H. Ma, and H. Fu, "Siamese neural network based gait recognition for human identification," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2016, pp. 2832–2836.

[23] J. Bromley, I. Guyon, and R. Shah, "Signature Verification Using a 'Siamese' Time Delay Neural Network," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 7, no. 4, pp. 669–688, 1993.

[24] G. Koch, "Siamese Neural Networks for One-Shot Image Recognition," University of Toronto, 2015.

[25] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015, vol. 07–12–June, pp. 815–823.

[26] S. Berlemont, G. Lefebvre, S. Duffner, and C. Garcia, "Siamese Neural Network based Similarity Metric for Inertial Gesture Classification and Rejection Siamese Neural Net- work based Similarity Metric for Inertial Gesture Classification and Rejection," in *work based Similarity Metric for Inertial Gesture Classification and Rejection. International Con- ference on Automatic Face and Gesture Recognition*, 2015, pp. 1–6.

[27] S. Appalaraju and V. Chaoji, "Image similarity using Deep CNN and Curriculum Learning," *Proc. 2017 Grace Hopper India Annu. Conf.*, pp. 1–9, 2017.

[28] W. Du and M. Shen, "Siamese Convolutional Neural Networks for Authorship Verification," 2016.

[29] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1701–1708.

[30] E. Ahmed, M. Jones, and T. K. Marks, "An improved deep learning architecture for person re-identification," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3908–3916.

[31] J. Mueller and A. Thyagarajan, "Learning Sentence Similarity with Siamese Recurrent Architectures," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*, 2016.

[32] A. Aides and H. Aronowitz, "Text-Dependent Audiovisual Synchrony Detection for Spoofing Detection in Mobile Person Recognition," in *Interspeech*, 2016, pp. 2125–2129.

[33] N. Zeghidour, G. Synnaeve, N. Usunier, and E. Dupoux, "Joint learning of speaker and phonetic similarities with siamese networks," in *Interspeech*, 2016, vol. 08–12–Sept, pp. 1295–1299.

[34] K. Chen and A. Salman, "Extracting Speaker-Specific Information with a Regularized Siamese Deep Network," in *Annual Conference on Neural Information Processing Systems*, 2011, pp. 1–9.

[35] N. N. Vo and J. Hays, "Localizing and Orienting Street Views Using Overhead Imagery," in *Computer Vision - ECCV 2016*, 2016, pp. 494–509.

[36] M. Todisco, N. Evans, T. Kinnunen, K. A. Lee, and J. Yamagishi, "ASVspoof 2017 Version 2.0: meta-data analysis and baseline enhancements," in *Odyssey (Accepted)*, 2018.

[37] V. Tolieng, B. Prasirtsak, J. Sitdhipol, N. Thongchul, and S. Tanasupawat, "Identification and lactic acid production of bacteria isolated from soils and tree barks," *Malays. J. Microbiol.*, vol. 13, no. 2, pp. 100–108, 2017.

[38] P. Nagarsheth, E. Khoury, K. Patil, and M. Garland, "Replay attack detection using DNN for channel discrimination," in *Interspeech*, 2017, pp. 97–101.

[39] L. Van Der Maaten and G. Hinton, "Visualizing Data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.