Densely Connected Convolutional Network for Audio Spoofing Detection

Zheng Wang*, Sanshuai Cui*, Xiangui Kang*, Wei Sun† and Zhonghua Li‡
* Guangdong Key Laboratory of Information Security Technology, Sun Yat-sen University, Guangzhou, China
† Information Technology Key Laboratory of the Ministry of Education, Sun Yat-sen University, Guangzhou, China
‡ School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou, China

***Abstract*—Anti-spoofing has attracted increasing attention since the inauguration of the ASVspoof Challenges, due to the fact that automatic speaker verification (ASV) systems are vulnerable to spoofing attacks. The latest ASVspoof 2019 Challenge was dedicated to addressing attacks in three major classes: speech synthesis, voice conversion, and replay audio. In this paper, we propose a novel method that includes feature extraction, a densely connected convolutional network, and fusion strategies to answer the ASVspoof 2019 Challenge and to defend against spoofing attacks. Features are extracted using different algorithms and then fed separately into variants of our model, which differ only in terms of the kernel size of the global average pooling layer. A dense connectivity pattern with better parameter efficiency is introduced to the proposed network to strengthen the propagation of the audio features. The experimental results show that the proposed method improves the tandem decision cost function and equal error rate scores by 75% and 78%, respectively, in the logical access challenge. In the physical access challenge, the proposed method improves the t-DCF and EER scores by 73% and 72%, respectively, compared with state-of-the-art methods.**

***Index Terms*—ASVspoof, Automatic Speaker Verification, Audio Spoofing Detection, Dense Connectivity**

I. INTRODUCTION

Automatic speaker verification (ASV) is deployed in an increasing number of diverse applications and services, e.g., mobile telephones, smart speakers, and call centers, in order to offer a low-cost and flexible biometric solution for personal authentication [1]. ASV systems are vulnerable to spoofing attacks, although their performance has gradually improved in recent years.

There are three major classes of spoofing attacks: replay audio (RA), speech synthesis (SS), and voice conversion (VC) [2]. These three attacks are significant threats to ASV systems. RA attacks are the most straightforward to implement and can be performed using recordings of bona fide speech [3]. RA attacks do not need any additional knowledge of audio signal processing and are more likely to be used by a non-professional attacker. However, the implementation of SS and VC attacks usually requires specific knowledge or familiarity with complex speech technology. SS systems can generate completely artificial speech signals, whereas VC systems operate on natural speech [3]. Both SS and VC technologies can produce high-quality speech signals that mimic the speech of a specific target individual.

Recent efforts in the field of anti-spoofing have been encouraged by the ASVspoof Challenges [4-6]. Previous ASVspoof Challenges have focused on raising awareness and developing solutions to spoofing attacks via SS, VC, and RA [7]. However, the ASVspoof 2019 Challenge aims to address all previous types of attack and is composed of two sub-challenges: the logical access (LA) and physical access (PA) challenges. LA considers spoofing attacks generated using SS and VC, whereas PA refers to spoofing attacks using RA. Besides using an equal error rate (EER) metric [6], a new tandem decision cost function (t-DCF) metric is adopted to reflect the impact of spoofing and countermeasures on ASV performance.

Research in the area of anti-spoofing can be divided into three categories: feature learning [8-11], statistical modeling [12-14], and deep neural networks (DNNs) [15-21]. Some DNN-based methods perform well in ASVspoof 2019. For example, the authors of [19] proposed a light convolutional gated recurrent neural network by fusing Light CNN (convolutional network) [22] and RNN (recurrent neural network) based on gated recurrent units (GRU). The network was used as a deep feature extractor to assist in the training of the classifier. Among them, LC-GRNN not only had the ability of Light CNN to extract discriminative features at the frame level but also included the ability of RNN to learn deep features. To solve the problem of poor generalization in speech detection algorithms based on a single feature, the authors of [20] proposed a speech detection framework based on multiple features integration and multi-task learning (MFMT). The authors of [18] built five DNN models and adopted different forms of feature engineering to detect spoofing attacks. Features included acoustic features, a unified feature map, and whole utterances, were fed into five DNN models based on variants of squeeze-excitation networks (SENets) [23] and ResNets [24]. Rather than using models with different architectures, the authors of [21] proposed three residual convolutional networks based on a unified residual block for anti-spoofing. Using three acoustic features as input, these Three models differed in terms of the number of blocks, the

Corresponding authors: Xiangui Kang (E-mail: isskxg@mail.sysu.edu.cn), Wei Sun (E-mail: sunwei@mail.sysu.edu.cn).

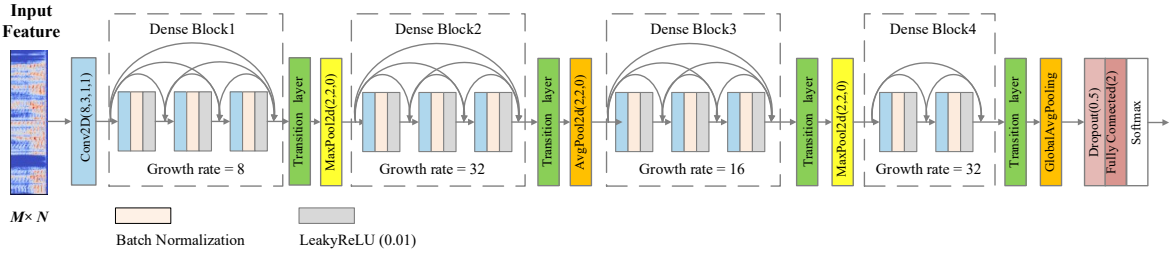

Fig.1. Illustration of the proposed densely connected convolutional network, where the input feature includes MFCC, CQCC, Spec, and LFCC with size of $M \times N$. Note that different feature input has different M and different N.

number of units in the fully connected layer, and the use of pooling layer.

In this paper, taking the advantages of multiple features integration and the success of DNN-based methods, we propose a new method that includes feature extraction, a densely connected convolutional network, and fusion strategies for audio spoofing detection. The 2D feature representation shaped by different feature extraction algorithms is then fed as input into our model. Considering that dense connectivity alleviates the vanishing gradient problem, strengthens the feature propagation, especially for audio features learned by shallow layers, the dense connectivity pattern with high parameter efficiency is introduced into our model, in which all layers with matched feature maps are directly connected. The performance of our proposed densely connected network with different types of input features is evaluated separately for the two sub-challenges in ASVspoof 2019 (LA and PA). To increase the accuracy of spoofing detection, different fusion strategies are adopted for different features for the two sub-challenge.

The major contributions of this work are as follows: We design a novel convolutional network for audio spoofing detection that includes dense connectivity. To the best of our knowledge, it is the first work to leverage dense connectivity for the task of audio spoofing detection. This dense connectivity strengthens the propagation of audio features and ensures the maximum flow of information between layers in the network through feature reuse. The developed network model with single feature-map input (single model in short form) achieves better results in the two sub-challenges (LA and PA) of ASVspoof 2019 than state-of-the-art single methods. For example, in the LA challenge, the proposed single model improves the t-DCF and EER scores by 68% and 67%, respectively. In the PA challenge, the proposed single model improves these scores by 47% and 45%, respectively. The fusion model, which is fused from several single models, improves the t-DCF and EER scores by 75% and 78%, respectively, in the LA challenge, and the t-DCF and EER scores by 73% and 72%, in the PA challenge, compared with state-of-the-art fusion models.

The rest of this paper is organized as follows. Section II describes the proposed method, including the feature extraction algorithms and the structure of the proposed network. Experimental results, comparisons, and analyses are presented in Section III. Finally, concluding remarks are made in Section IV.

II. Proposed Method

We propose a novel method with feature extraction algorithms, a densely connected convolutional network, and fusion strategies for audio spoofing detection.

A. Feature Extraction

The extracted features include Mel-frequency cepstral coefficients (MFCC), constant Q cepstral coefficients (CQCC), the logarithmic magnitude spectrum of STFT (Spec), and linear frequency cepstral coefficients (LFCC). Different input feature has different size, that is, different M and different N as shown in Fig. 1.

MFCC is one of the most popular magnitude-based features in speech processing. It uses cepstral analysis on the log magnitude spectrum in the Mel scale. The MFCC contains vocal tract dynamics, and its corresponding pulse train is related to glottal motor control, which makes the feature suitable for distinguishing spoofed speech from human speech. The MFCC is defined as:

$$C_i = \sqrt{\frac{2}{N}} \sum_{j=1}^{N} \cos\left(\frac{\pi i}{N}(j-0.5)\right), \quad (1)$$

where N is the number of Mel-frequency bins of log spectrum, and i is the number of cepstral coefficients. The first 24 coefficients are selected in this study, and the sampling rate is set to 16000. The MFCC is concatenated with its first derivative Δ MFCC and second derivative Δ^2 MFCC to obtain the MFCC feature input with size of $M \times N$ as shown in Fig.1. $M \times N$ is 72×126 here in our work.

CQCC is reported to be sensitive to the general form of spoofing attack, and yields superior performance among various kinds of features [25]. The CQCC is an amplitude-based feature that uses the constant Q transform (CQT) in combination with traditional cepstral analysis. The frequency bins of $X^{cq}(k)$ are obtained by the CQT of an input speech signal sequence $x(n)$. Uniform sampling is applied to the constant Q power spectrum $\log(|X^{cq}(k)|^2)$, and the resulting

Table I. Architecture of the proposed network, where the input feature includes MFCC, CQCC, Spec, and LFCC. Note that each "conv" layer shown in the table corresponds the sequence Conv-BN- leaky-ReLU.

Layers	Architecture	MFCC	CQCC	Spec	LFCC
Convolution	3 × 3 conv, stride 1	72 × 126	90 × 469	1025 × 126	60 × 399
Dense Block1	(3 × 3 conv) × 3	72 × 126	90 × 469	1025 × 126	60 × 399
Transition layer	1 × 1 conv	72 × 126	90 × 469	1025 × 126	60 × 399
Max pooling	2 × 2 max pool, stride 2	36 × 63	45 × 234	512 × 63	30 × 199
Dense Block2	(3 × 3 conv) × 3	36 × 63	45 × 234	512 × 63	30 × 199
Transition layer	1 × 1 conv	36 × 63	45 × 234	512 × 63	30 × 199
Average pooling	2 × 2 average pool, stride 2	18 × 31	22 × 117	256 × 31	15 × 99
Dense Block3	(3 × 3 conv) × 3	18 × 31	22 × 117	256 × 31	15 × 99
Transition layer	1 × 1 conv	18 × 31	22 × 117	256 × 31	15 × 99
Max pooling	2 × 2 max pool, stride 2	9 × 15	11 × 58	128 × 15	7 × 49
Dense Block4	(3 × 3 conv) × 2	9 × 15	11 × 58	128 × 15	7 × 49
Transition layer	1 × 1 conv	9 × 15	11 × 58	128 × 15	7 × 49
Classification layer	global average pool	1 × 1	1 × 1	1 × 1	1 × 1
	Dropout, 128 FC, softmax	-	-	-	-

$\log(|X^{cq}(k)|^2)$ can then be applied with DCT to obtain the feature representation with size 90 × 469. More details of CQCC can be found in [25].

Spec is captured by computing the STFT on hamming windows firstly, then calculating the magnitude of each component. Let $x(n)$ be a given speech sequence and $X_n(w)$ is STFT after applying a window $w(n)$ on the speech signal $x(n)$. The length of the window is set to 2048. $X_n(w)$ is defined as:

$$X_n(w) = |X_n(w)| e^{j\theta_n(w)}, \quad (2)$$

where $|X_n(w)|$ corresponds to the short-time magnitude spectrum and $\theta_n(w)$ corresponds to the phase spectrum. The square of the magnitude spectrum is called the STFT power spectrum. The logarithm of the power spectrum is adopted as the Spec feature with size of 1025 × 126.

LFCC is a kind of cepstral feature based on a triangle filter-bank similar to the MFCC. It is extracted in the similar way as MFCC, but the filters are in the triangular shape rather than on the Mel scale. Therefore, The LFCC has better resolution in the higher frequency region [26]. The first 20 coefficients are selected. The LFCC is concatenated with its first derivative △ LFCC and second derivative △² LFCC to produce the feature representation with size of 60 × 399.

B. Proposed Network Model

1. Overall Architecture

The overall architecture of the proposed network model is shown in Fig. 1. The network contains 11 regular convolutional layer groups, each of which consists of three steps: convolution, batch normalization [27], and leaky-ReLU [28]. In addition, it has one standard convolutional layer, four transitional layers, two max-pooling layers, one average-pooling layer, one global average-pooling layer, and one fully connected layer.

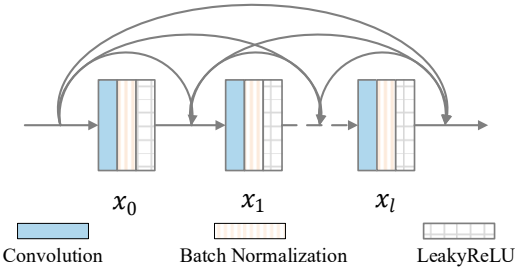

Fig.2. Illustration of the dense connectivity in dense block.

The first standard convolutional layer has a filter size of 3 × 3, with stride and padding one, and outputs eight feature maps. There are four dense blocks in total, as shown in Fig. 1. The dense connection is introduced in each dense block. For each convolutional layer in the same block, the feature maps of all preceding group layers are used as input. We adopt three convolutional layer groups in the first three dense blocks, and two convolutional layer groups in the last dense block. The receptive field of the four dense blocks is 3 × 3. For the first dense block, the growth rate is eight, which means that each convolutional layer outputs eight feature maps in this block. The growth rate is 32 for the second block, 16 for the third block and 32 for the last block. The transition layers applied after each block are designed to reduce the number of input feature maps by using 1 × 1 convolutions. The pooling layers are adopted in order to facilitate down-sampling and to change the size of the feature maps. The output from the global average pooling layer is fed into a dropout layer [29] (dropout rate = 50%) followed by a two-way Softmax layer that produces a

Table II. Summary of the ASVspoof 2019 logical access (LA) spoofing systems. Note that A04 and A16 use same waveform concatenation SS algorithm, and A06 and A19 use same VC algorithm.

Training and development set		Evaluation set	
SS	VC	SS	VC
A01		A07 A11	A13 A18
A02	A05	A08 A12	A14 A19
A03	A06	A09 A16	A15
A04		A10	A17

distribution of two class labels. Table I shows the detailed architecture of the proposed network. And when accepting different input features, the output size of each layer is presented.

The goal of the ASVspoof challenge is to calculate a countermeasure (CM) score for each input audio file. A high CM score represents bona fide speech, whereas a low CM score represents a spoofing attack. The final CM score is computed from the Softmax outputs using the log-likelihood ratio:

$$CM(s) = \log(p(bonafide|s;\theta)) - \log(p(spoof|s;\theta)), \quad (3)$$

where s is the audio signal under test and θ represents the model parameters.

2. Dense Connectivity

Dense connectivity is introduced to our model inspired by [30]. In this pattern, direct connections are applied from each layer to all subsequent layers with the same feature map size. As shown in Fig.2, the l^{th} layer receives the feature maps from all preceding layers, $x_0, x_1, \cdots, x_{l-1}$, as the input:

$$x_l = H_l([x_0, x_1, \cdots, x_{l-1}]), \quad (4)$$

where $[x_0, x_1, \cdots, x_{l-1}]$ refers to the concatenation of the feature maps produced in layer $0, \cdots, l-1$. $H_l(\cdot)$ is a composite function of operations including batch normalization, leaky-ReLU and convolution.

Concatenating the feature maps learned by different layers can increase the variation in the input of subsequent layers and ensure maximum information flow between layers in the network. Depending on the dense connectivity pattern, some general audio features that are only extracted in the preceding layers can be shared in the deeper layers in our architecture. In this way, the propagation of audio features is strengthened. In addition, the gradients can flow directly through the identity function, from the later layers to the former layers, leading to implicit deep supervision [30] that alleviates the vanishing-gradient problem and makes the network easy to train. Compared with L-layer traditional convolutional neural networks with L connections, the dense connectivity introduces $\frac{L+1}{2}$ connections without relearning redundant feature maps.

Table III. Replay attack is defined as duple (D, Q), each element of which takes one value in set (A, B, C) as a categorical value.

Attack definition	labels		
	A	B	C
D : Recording distance	10-50	50-100	>100
Q : Replay device quality	perfect	high	low

The dense pattern has better parameter efficiency than the traditional pattern in convolutional networks.

C. Model Symbol Definition and Fusion Strategies

The single CNN model variant is built by accepting MFCC, CQCC, Spec and LFCC input features. Define the single model as D_f, where f represents MFCC, CQCC, Spec, and LFCC. D_M (MFCC), D_C (CQCC), D_S (Spec), and D_L (LFCC) differ only in the kernel size of the global average pooling layer. Suppose that the size of the input feature is $M \times N$, the kernel size of the global average pooling layer is $\left\lfloor \frac{M}{8} \right\rfloor \times \left\lfloor \frac{N}{8} \right\rfloor$, where $\lfloor \cdot \rfloor$ represents the function of rounding down.

The outputs of several single-model D_f are fused together to get the fusion model. The fusion result is obtained by taking average of the outputs (CM scores) of the individual single-models. The scores of the single-models are obtained by using the formula (3) with the parameters of best performance on the development dataset. This fusion model is defined as $D_{f_1, f_2, \cdots, f_n}$. For the two sub-challenges (LA and PA) of ASVspoof 2019, different fusion strategies are adopted. For example, $D_{S,L}$ for the LA challenge and $D_{C,S,L}$ for the PA challenge.

III. Experimental Results

In this section, experiments are carried out to demonstrate the effectiveness of the proposed method. In addition to comparing with two baseline models provided by ASVspoof 2019, we also compare it with state-of-the-art DNN-based methods [21].

A. Experimental Setup

The dataset used in this study containing non-overlapping short audio files is provided by the organizers of ASVspoof 2019. The dataset consists of both bona fide and spoofed audio recordings, and is divided into three parts: training, development, and evaluation. For the LA sub-challenge, spoofed audio is generated using 19 different SS and VC algorithms (A01 to A19). Six of these attack algorithms (A01 to A06) are considered to be known attacks, and are used to generate the training and development datasets. The other 13 algorithms are used to generate the evaluation dataset. A07 to A18 (except A16) represent eleven unknown attacks, while A16 and A19 are known attacks using the same algorithms as A04 and A06. Summary of the logical access spoofing systems is shown in Table II. For the PA sub-challenge, replay attacks

Table IV. t-DCF and EER scores of different models were measured using the development and evaluation sets in logical access (LA) scenarios. Baseline models are denoted as G_L and G_C. The residual model variants provided by [21] are denoted as R_M, R_C and R_S. Our single models are denoted as D_M, D_C, D_S and D_L. The residual fusion model is denoted as $R_{M,C,S}$ and our fusion models are denoted as $D_{M,C,S}$ and $D_{S,L}$. Subscripts represent different features.

Model	Development		Evaluation	
	t-DCF	EER%	t-DCF	EER%
G_L	0.0663	2.71	0.2116	8.09
G_C	0.0123	0.43	0.2366	9.57
R_M	0.2319	7.18	0.2780	12.07
R_C	0.0899	2.98	0.2626	11.34
R_S	0.0197	0.68	0.2094	9.82
D_M	0.0580	2.00	0.2209	9.64
D_C	0.0319	1.02	0.2616	10.87
D_S	0.0029	0.11	0.1979	7.14
D_L	**0.0007**	**0.04**	**0.0676**	**3.27**
$R_{M,C,S}$	0.0231	0.82	0.1853	8.99
$D_{M,C,S}$	0.0024	0.08	0.1387	5.76
$\boldsymbol{D_{S,L}}$	**0.0001**	**0.01**	**0.0469**	**1.98**

are recorded and replayed in 27 different acoustic configurations with nine different settings (i.e. combinations of three categories of recording distance and three levels of replay device quality). Evaluation data for PA are generated from different impulse responses and therefore represent unknown attacks. The definition of nine different settings is shown in Table III.

All of the experiments using the CNN reported in this study are performed using PyTorch on Nvidia Tesla K80 GPUs. The models are trained using a batch size of 32 and a learning rate of 5×10^{-5} for 200 epochs. In order to mitigate the imbalance in the distribution of training data, we train our models by minimizing a weighted cross-entropy loss function where the ratio between the weights assigned to genuine and spoofed examples is 9:1. The cost function is minimized using the Adam optimizer [31]. After each epoch, we save the model parameters, and finally, we use the parameters with the best performance on the development dataset.

We denote the variants of single model as D_M, D_C, D_S and D_L. Several state-of-the-art methods are used for comparison with these variants. The residual model variants provided by [21] are denoted as R_M, R_C and R_S. R_M consists of nine residual blocks and inserts pooling layers between every three blocks, while R_C adopts six residual blocks and inserts pooling layers between blocks, and R_S consists of six residual blocks. The baseline models, namely Gaussian mixture models (GMMs), provided by [6] are denoted as G_L and G_C. The

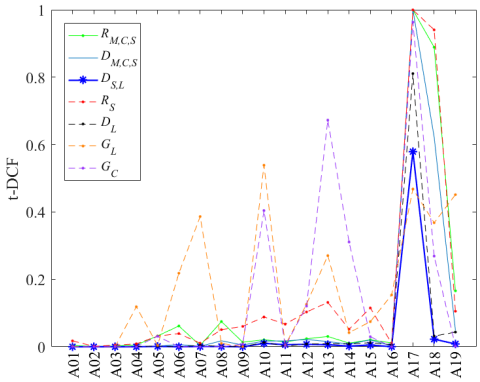

Fig.3. t-DCF scores of different models against different types of attacks (A01 to A09) in logical access (LA) scenarios, showing detailed comparisons between the two baseline models, the two single models, and the three fusion models. The meaning of model symbol can refer to the title of Table IV.

scores of these model variants are compared on both the development and evaluation datasets.

For the LA challenge, the scheme in [21] fuses MFCC, CQCC, and Spec features to get a fusion model which is denoted as $R_{M,C,S}$. For the PA challenge, the MFCC feature performs poorly on the development set, so it is not considered in the fusion strategies. The scheme in [21] fuses CQCC and Spec features to get a fusion model denoted as $R_{C,S}$ for the PA challenge. In contrast, we select two new fusion strategies and denote $D_{S,L}$ for the LA challenge and $D_{C,S,L}$ for the PA challenge. Both strategies are shown to be more suitable for audio spoofing detection than the state-of-the-art alternative.

Table VI. t-DCF and EER scores of different models are measured on the development and evaluation sets in physical access (PA) scenarios. The meaning of model symbol can refer to the title of Table IV.

Model	Development		Evaluation	
	t-DCF	EER%	t-DCF	EER%
G_L	0.2554	11.96	0.3017	13.54
G_C	0.1953	9.87	0.2454	11.04
R_M	0.3945	16.80	-	-
R_C	0.2076	8.82	0.2982	12.06
R_S	0.1256	4.74	0.1416	5.52
D_M	0.3749	15.67	-	-
D_C	0.1068	4.87	0.1518	6.40
D_S	**0.0716**	**2.80**	**0.0754**	**3.01**
D_L	0.1110	5.45	0.1871	7.48
$R_{C,S}$	0.0925	3.80	0.1274	5.02
$D_{C,S}$	0.0374	1.52	0.0445	1.76
$\boldsymbol{D_{C,S,L}}$	**0.0265**	**1.19**	**0.0341**	**1.40**

Table V. Detailed comparison of t-DCF and EER scores for the three fusion models under different replay attacks for logical access (LA) scenarios. The residual fusion model is denoted as $R_{M,C,S}$ and our fusion models are denoted as $D_{M,C,S}$ and $D_{S,L}$. Subscripts represent different features.

Attack Type	$R_{M,C,S}$		$D_{M,C,S}$		$D_{S,L}$	
	t-DCF	EER%	t-DCF	EER%	t-DCF	EER%
A01	0.0036	0.19	0.0008	0.03	**0.0000**	**0.00**
A02	0.0013	0.03	0.0016	0.03	**0.0000**	**0.00**
A03	0.0023	0.11	0.0015	0.08	**0.0000**	**0.00**
A04	0.0052	0.27	0.0015	0.08	**0.0000**	**0.00**
A05	0.0321	1.10	0.0017	0.08	**0.0011**	**0.03**
A06	0.0616	1.37	0.0065	0.16	**0.0000**	**0.00**
A07	0.0038	0.15	0.0012	0.06	**0.0006**	**0.04**
A08	0.0748	2.72	0.0177	0.63	**0.0008**	**0.04**
A09	0.0135	0.18	0.0052	0.07	**0.0000**	**0.00**
A10	0.0209	0.72	0.0168	0.61	**0.0101**	**0.38**
A11	0.0145	0.53	0.0174	0.67	**0.0061**	**0.24**
A12	0.0238	0.87	0.0218	0.80	**0.0072**	**0.26**
A13	0.0306	1.08	0.0154	0.55	**0.0060**	**0.24**
A14	0.0108	0.42	0.0081	0.31	**0.0019**	**0.08**
A15	0.0208	0.77	0.0203	0.79	**0.0052**	**0.19**
A16	0.0114	0.41	0.0026	0.11	**0.0002**	**0.02**
A17	0.9998	31.44	1.0000	18.84	**0.5791**	**9.65**
A18	0.8889	19.90	0.6221	10.13	**0.0225**	**0.34**
A19	0.1658	4.11	0.0482	1.22	**0.0084**	**0.24**

Table VII. Detailed comparison of t-DCF and EER scores for the three fusion models under different replay attacks for physical access (PA) scenarios. The residual fusion model is denoted as $R_{C,S}$ and our fusion models are denoted as $D_{C,S}$ and $D_{C,S,L}$. Subscripts represent different features.

Attack Type	$R_{C,S}$		$D_{C,S}$		$D_{C,S,L}$	
	t-DCF	EER%	t-DCF	EER%	t-DCF	EER%
AA	0.3225	11.84	0.1218	4.74	**0.1038**	**3.96**
AB	0.0675	2.32	0.0272	1.06	**0.0135**	**0.54**
AC	0.0418	1.40	0.0065	0.29	**0.0052**	**0.24**
BA	0.2090	8.02	0.0469	1.78	**0.0360**	**1.45**
BB	0.0396	1.33	0.0096	0.43	**0.0050**	**0.21**
BC	0.0226	0.84	0.0030	0.13	**0.0014**	**0.06**
CA	0.1610	5.81	0.0468	1.74	**0.0336**	**1.29**
CB	**0.0029**	0.98	0.0077	0.31	0.0034	**0.12**
CC	0.0183	0.59	0.0037	1.17	**0.0018**	**0.09**

The t-DCF [32] and EER metrics are adopted to assess the performance of the anti-spoofing methods. The t-DCF metric takes into consideration both the ASV system and spoofing countermeasure errors, and more details can be found in [32]. The EER metric is determined by the point at which the miss (false negative) rate and false alarm (false positive) rate are equal.

B. Results for ASVspoof 2019 LA

Table IV shows the scores for both the development and evaluation dataset for the LA sub-challenge. For the development set, R_M, R_C, and R_S perform worse than G_C in terms of the t-DCF and EER metrics. For the evaluation dataset, R_S performs better than G_L on the t-DCF metric, but worse than G_L on the EER metric. The single top model D_L has a significantly lowest t-DCF and EER scores for both the development and evaluation datasets among single models. Our fusion model $D_{M,C,S}$ achieves a t-DCF score of 0.1387 and an EER score of 5.76, representing improvements of 25% and 36%, respectively, over the fusion model $R_{M,C,S}$. Our fusion model $D_{S,L}$ achieves a t-DCF score of 0.0469 and an EER score of 1.98, representing improvements of 75% and 78%, respectively, over the fusion model $R_{M,C,S}$. This obvious improvement indicates that the proposed method is more

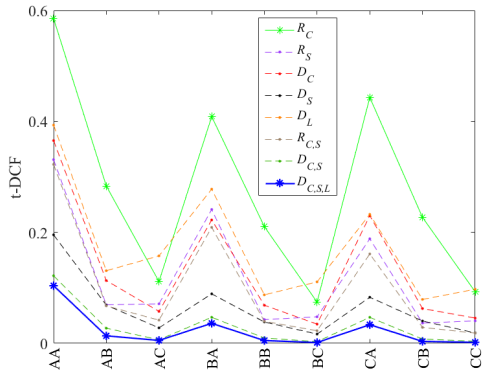

Fig.4. t-DCF scores of different models for different replay attack settings in physical access (PA) scenarios, showing detailed comparisons between the five single models, and the three fusion models. The meaning of model symbol can refer to the title of Table IV.

suitable for audio spoofing detection in the LA sub-challenge than the state-of-the-art alternative.

Fig.3. provides a detailed illustration of the performance of our model against both known and unknown attacks (A01 to A19). We show the t-DCF scores for G_L, G_C, the single residual model R_S, the single top model D_L, the fusion model $R_{M,C,S}$, the fusion model $D_{M,C,S}$, and the fusion model $D_{S,L}$. A01 to A06 are known attacks (from the development set), while A07 to A19 are unknown (from the evaluation set). It can be observed that our fusion model $D_{S,L}$ works well against almost all these attacks except for A17, and that most of the other models perform poorly on A17 and A18. Both A17 and A18 are VC algorithms, where A17 is based on waveform filtering, and A18 is based on vocoders. G_L performs best against A17, indicating that a DNN-based method is more vulnerable to vocoder based video conversion attacks. We compare the effects of the fusion models ($R_{M,C,S}$, $D_{M,C,S}$, $D_{S,L}$) on various attacks (A01 to A09) in detail in Table V. Our fusion model $D_{M,C,S}$ performs better than $R_{M,C,S}$ in most cases. Our proposed new fusion model $D_{S,L}$ works best when resisting all attacks. In particular, $R_{M,C,S}$ and $D_{M,C,S}$ perform poorly against A17. Our proposed new fusion model $D_{S,L}$ reduces the metric of t-DCF from 0.9998 to 0.5791, and the metric of EER from 31.44 to 9.65, compared to the fusion model $R_{M,C,S}$ against A17. $D_{S,L}$ reduces the metric of t-DCF from 1.0000 to 0.5791, the metric of EER from 18.84 to 9.65, compared to the model $D_{M,C,S}$ against A17.

C. *Results for ASVspoof 2019 PA*

Table VI presents the results for the PA task, for both the development and evaluation dataset. In general, R_S, D_C, D_S and D_L improve the performance in terms of the t-DCT and EER metrics. As shown in Table VI, for the development dataset, the single top model D_S is 43% and 41% better than the single residual model R_S in terms of the t-DCF and EER metrics, respectively. The single top model D_S reduces the t-DCF and EER of R_S by 47% and 45% for the evaluation dataset. The fusion model $D_{C,S}$ represents a 65% improvement compared to the fusion model $R_{C,S}$. The proposed new fusion model $D_{C,S,L}$ achieves a t-DCF score of 0.0341 and an EER score of 1.40, representing improvements of 73% and 72%, respectively.

Table VIII. Comparisons between the single proposed models and the single residual models in terms of model parameters.

Model	R_S	R_M	R_C
Param	176130	255650	167552
Model	D_S	D_M	D_C
Param	97098	97098	97098

Fig.4. provides a detailed illustration of the performance of the models for different replay attack settings. Each type of attack is represented with two letters, the first of which stands for the distance between the recording device and the bona fide speaker (where 'A' means 10–50 cm, 'B' means 50–100 cm, and 'C' means >100 cm), while the second represents the quality of the replay device (where 'A' means perfect, 'B' means high, and 'C' means low). It can be observed that the t-DCF metric of our proposed new fusion model $D_{C,S,L}$ is lowest for different replay attack settings, which implies the effectiveness of our model $D_{C,S,L}$.

To further illustrate the effects of the three fusion models, Table VII provides detailed performance results for each model for different replay attack settings. A comparison can be made between the fusion model $R_{C,S}$, the fusion model $D_{C,S}$, and the proposed new fusion model $D_{C,S,L}$. It is easy to see that anti-spoofing becomes more difficult as the distance decreases and the quality of the recording device improves. In general, the fusion model $D_{C,S,L}$ performs best under most replay attack settings.

D. *Comparison with Single Residual Networks in Model Parameters*

We show comparisons between the single proposed networks and the single residual networks [21] in Table VIII. Accepting different input features, R_f has different model parameters but D_f has the same parameters. Besides, D_f has less parameters than R_f when accepting the same input features, indicating that the DNN-based method with dense

connectivity has better parameter efficiency than the DNN-based method without this pattern.

IV. Conclusion

In this paper, we propose a novel method for anti-spoofing that includes feature extraction, a densely connected network, and fusion strategies. A dense connectivity pattern is introduced to strengthen the propagation of the extracted audio features and to give better parameter efficiency, which helps boost the detection accuracy. We compare the performance of our model, using four different feature extraction algorithms. In addition to comparing two baseline models provided by ASVspoof 2019, we also compare our proposed method with state-of-the-art DNN-based methods, and our method achieves better performance in terms of the t-DCF and EER metrics. The proposed CNN architecture performs well on the LA scenarios, which consists of VC and SS attacks, and our fusion model improves the t-DCF and EER metrics by 75% and 78%, respectively. The proposed method also performs well against the RA attack within the PA scenarios, and our fusion model improves the t-DCF and EER metrics by 73% and 72%, respectively.

Acknowledgment

This work was supported by NSFC (Grant nos. 61772571, 62072484) and Chinese national key research and development project 2019QY2203.

References

[1] K. M. Malik, H. Malik, and R. Baumann, "Towards vulnerability analysis of voice-driven interfaces and countermeasures for replay attacks," in *Proceedings of the IEEE Conference on Multimedia Information Processing and Retrieval*, pp. 523–528, 2019.

[2] J. H. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[3] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, and et al., "The ASVspoof 2019 database," *arXiv preprint arXiv:1911.01601*, 2019.

[4] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilc¸i, and et al., "ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," in *Proceedings of the Annual Conference of the International Speech Communication Association,* pp. 2037-2041, 2015.

[5] T. Kinnunen, N. Evans, J. Yamagishi, K. A. Lee, M. Sahidullah, and et al., "ASVspoof 2017: Automatic speaker verification spoofing and countermeasures challenge evaluation plan," 2016. [Online]. Available: http://www.spoofingchallenge. org/

[6] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, and et al., "ASVspoof 2019: Future horizons in spoofed and fake audio detection," *arXiv preprint arXiv:1904.05441*, 2019.

[7] N. W. Evans, T. Kinnunen, and J. Yamagishi, "Spoofing and countermeasures for automatic speaker verification." in *Proceedings of the Annual Conference of the International Speech Communication Association,* pp. 925-929, 2013.

[8] G. Suthokumar, V. Sethu, C. Wijenayake, and E. Ambikairajah, "Modulation dynamic features for the detection of replay attacks," in *Proceedings of the Annual Conference of the International Speech Communication Association,* pp. 691–695, 2018.

[9] M. J. Alam, G. Bhattacharya, and P. Kenny, "Boosting the performance of spoofing detection systems on replay attacks using q-logarithm domain feature normalization," in *Proceedings of the Odyssey of the Speaker and Language Recognition Workshop*, pp. 393–398, 2018.

[10] H. B. Sailor, M. R. Kamble, and H. A. Patil, "Auditory filterbank learning for temporal modulation features in replay spoof speech detection." in *Proceedings of the Annual Conference of the International Speech Communication Association,* pp. 666–670, 2018.

[11] Z. Wu, X. Xiao, E. S. Chng, and H. Li, "Synthetic speech detection using temporal modulation feature," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 7234–7238, 2013.

[12] H. Delgado, M. Todisco, M. Sahidullah, N. Evans, T. Kinnunen, and et al., "ASVspoof 2017 Version 2.0: meta-data analysis and baseline enhancements," in *Proceedings of the Odyssey of the Speaker and Language Recognition Workshop*, pp.296-303, 2018.

[13] Adiban, H. Sameti, N. Maghsoodi, and S. Shahsavari, "SUT system description for anti-spoofing 2017 challenge," in *Proceedings of the 29th Conference on Computational Linguistics and Speech Processing,* pp. 264–275, 2017.

[14] E. Khoury, T. Kinnunen, A. Sizov, Z. Wu, and S. Marcel, "Introducing i-vectors for joint anti-spoofing and speaker verification," in *Proceedings of the Annual Conference of the International Speech Communication Association*, pp. 61-65, 2014.

[15] Z. Chen, Z. Xie, W. Zhang, and X. Xu, "Resnet and model fusion for automatic spoofing detection," in *Proceedings of the Annual Conference of the International Speech Communication Association,* pp. 102–106, 2017.

[16] C.-I. Lai, A. Abad, K. Richmond, J. Yamagishi, N. Dehak, and et al., "Attentive filtering networks for audio replay attack detection," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6316–6320, 2019.

[17] G. Valenti, H. Delgado, M. Todisco, N. W. Evans, and L. Pilati, "An end-to-end spoofing countermeasure for automatic speaker verification using evolving recurrent neural networks." in *Proceedings of the Odyssey of the Speaker and Language Recognition Workshop,* pp. 288-295, 2018.

[18] C.-I. Lai, N. Chen, J. Villalba, and N. Dehak, "Assert: Anti-spoofing with squeeze-excitation and residual networks," in *Proceedings of the Annual Conference of the International Speech Communication Association*, pp. 1013-1017, 2019.

[19] J. Monteiro, J. Alam, and T. H. Falk, "End-to-end detection of attacks to automatic speaker recognizers with time-attentive light convolutional neural networks," in *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing*, pp. 1–6, 2019.

[20] H. Zeinali, T. Stafylakis, G. Athanasopoulou, J. Rohdin, I. Gkinis, and et al., "Detecting Spoofing Attacks Using VGG and SincNet: BUT-Omilia Submission to ASVspoof 2019 Challenge," in *Proceedings of the Annual Conference of the International Speech Communication Association*, pp. 1073-1077, 2019.

[21] M. Alzantot, Z. Wang, and M. B. Srivastava, "Deep residual neural networks for audio spoofing detection." in *Proceedings of the Annual Conference of the International Speech Communication Association*, pp. 1078–1082, 2019.

[22] X. Wu, R. He, Z. Sun, and T. Tan, "A light CNN for deep face representation with noisy labels," *IEEE Transactions on*

Information Forensics and Security, vol. 13, no. 11, pp. 2884–2896, 2018.

[23] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[25] M. Todisco, H. Delgado, and N. W. Evans, "A new feature for automatic speaker verification anti-spoofing: Constant q cepstral coefficients." in *Proceedings of the Odyssey of the Speaker and Language Recognition Workshop*, pp. 283–290, 2016.

[26] M. Sahidullah, T. Kinnunen, and C. Hanilc¸i, "A comparison of features for synthetic speech detection," in *Proceedings of the Annual Conference of the International Speech Communication Association*, pp. 2087-2091, 2015.

[27] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International Conference on Machine Learning*, pp. 448–456, 2015.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhut-dinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp.1929–1958, 2014.

[30] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980,* 2014.

[32] T. Kinnunen, K. Lee, H. Delgado, N. Evans, M. Todisco, and et al., "t-DCF: a detection cost function for the tandem assessment of spoofing countermeasures and automatic speaker verification," *arXiv preprint arXiv:1804.09618,* 2018.