Detection of Speaker De-identification Disguise Based on Dense Convolutional Network

Yong Wang
Guangdong Polytechnic Normal University
Zhuoyi Su
Guangdong Polytechnic Normal University https://orcid.org/0000-0002-7123-3720
Zhengyu Zhu (zhuzhengyu0701@163.com)
Guangdong Polytechnic Normal University

Research

Keywords: dense convolutional network, speaker disguise, de-identification

Posted Date: May 27th, 2020

DOI: https://doi.org/10.21203/rs.3.rs-26755/v1

RESEARCH

Detection of Speaker De-identification Disguise Based on Dense Convolutional Network

Yong Wang[1], Zhuoyi Su[1] and Zhengyu Zhu[1,2*]

*Correspondence:
zhuzhengyu0701@163.com
[1]School of Electronic and Information Engineering, Guangdong Polytechnic Normal University, 510006 GuangZhou, China
Full list of author information is available at the end of the article

Abstract

Nowadays, speaker disguise is a common operation that presents a great challenge to social security. Therefore, it is important to recognize the authenticity of speech. Most of the current researches focus on speech spoofing, which simulates a target speaker to break through the state-of-art ASV systems by increasing false acceptance rate. Meanwhile, there is another type of disguise, i.e. de-identification, which transforms a speech signal without a target to increase the false rejection rate in order not to be recognized. It has received far less attention. Therefore, in this paper, we investigate the de-identification model and propose a method to detect de-identification speeches from genuine speeches by using a very deep dense convolutional network with 135 layers. The experimental results show that the average accuracy of the proposed method outperforms the reported state-of-the-art methods.

Keywords: dense convolutional network; speaker disguise; de-identificatoin

1 Introduction

Speaker disguise can be divided into two categories [1]: 1) speech spoofing, including voice conversion (VC), synthesis (SS) and replay, which changes or captures a person's speech, or creates an artificial speech in order to to be recognized as a target person; 2) de-identification disguise that changes a person's speech signal without a target in order not to be recognized. Early researches [2, 3, 4, 5, 6, 7, 8, 9, 10] have revealed that either kind presents threat to security by breaking through the state-of-art automatic speaker verification (ASV) systems. Recent researches focus on speech spoofing detection. For VC and SS detection, Hanilci et al. proposed a method through linear prediction residuals to extract phase features of speech signals [11]. Janick et al. proposed an algorithm to extract audio quality features based on residual signal of linear prediction [12]. Kamble et al. proposed a detection algorithm using instantaneous frequency cosine coefficients [13]. Muckenhirn et al. employed the first-order and second-order spectrum statistics [14]. Alam et al. proposed an algorithm using the transformation feature representation of infinite impulse response constant q [15]. In addition, other artificially designed features such as long-term spectral statistics, mel-frequency cepstral coefficients (MFCC) and modified group delay were used in [16, 17, 18, 19, 20, 21, 22, 23]. Support Vector Machine (SVM), Hidden Markov Model (HMM) and Generalized Method of Moments (GMM) are the most commonly used classifiers in the papers above. In some other efforts, DNN framework is used [24, 25, 26, 27, 28]. For replay detection, algorithms were proposed using Electronic Network Frequency (ENF), MFCC

and fundamental frequency, linear predictive residual signal, time envelope, stratified scattering decomposition coefficient and Inverse MFCC (IMFCC) respectively [29, 30, 31, 32, 33, 34, 35].

The researches on de-identification detection are relatively fewer. Paper [36, 37, 38] proposed detection algorithms using MFCC features and SVM. The cross-database recognition rates were lower than 90%, and the computation load was heavy. In addition, Liang et al. proposed an approach based on convolutional neural network (CNN) [39] . The accuracy rates of the above methods are all less than 95%, indicating that an improvement is needed for practical applications.

It should be noted that it is difficult and costly to implement speech spoofing to some extent as VC and SS usually require a large amount of target person's information, and the situation of replay is uncertain. By contrast, de-identification disguise requires no additional information. It has been integrated into many popular audio/voice editing tools and been used in many criminal cases. However, compared with spoofing detection, researches on de-identification detection are relatively few and insufficient. Therefore, in this paper, we examine the model and detection of de-identification disguise. Considering the fact that a DNN can automatically extract deep features that are not artificially designed, we propose a de-identification detection method that employs a very deep dense convolutional network with 135 layers. Experimental results show that it outperforms the state-of-the-art methods.

2 Method

2.1 Feratures for De-identification Detection

The object of de-identificatoin disguise is to alter the frequency content of a signal without affecting its time evolution [40]. Therefore, coming up with a rigorous definition is not easy because time and frequency characteristics of a signal, being related by the Fourier transform, are not independent. However, we can refer to a parametric model of audio signals to facilitate the task. The most efficient model in our context is the quasi-stationary sinusoidal model. In this model, the signal is represented as a sum of sinusoids whose instantaneous frequency $w_i(t)$ and amplitude $A_i(t)$ vary slowly with time. This can be written as:

$$x(t) = \sum_{i=1}^{I(t)} A_i(t) exp(j\phi_i(t)) \tag{1}$$

and

$$\phi_i(t) = \int_{-\infty}^{t} \omega_i(\tau) d\tau \tag{2}$$

where $A_i(t)$, $\omega_i(t)$ and $\phi_i(t)$ are instantaneous amplitude, instantaneous frequency and instantaneous phase of the i^{th} sinusoid respectively. Defining an arbitrary disguise modification amounts to specifying a scaling factor $\alpha(t) > 0$, which is implicitly assumed to be a regular and 'slowly' varying function of time. Then the ideal disguise corresponding to the signal described by Eq.1 and Eq.2 would be:

$$x'(t') = \sum_{i=1}^{I(t')} A_i(t') exp(j\phi_i'(t')) \tag{3}$$

and

$$\phi_i'(t') = \int_{-\infty}^{t'} \alpha(\tau)\omega_i(\tau)d\tau \tag{4}$$

Eq.3 indicates that the sinusoids in the modified signal at time t' have the same amplitude as in the original signal at time $t = t'$, but their instantaneous frequency are multiplied by a factor $\alpha(t')$, as can be seen by differentiating $\phi_i'(t')$ with respect to t'. As a result, the time-evolution of the original signal is not modified but its frequency content is scaled by the factor. In practice, $\alpha(t)$ is usually a constant. In the following context, we use α to denote the constant factor.

In implementation, STFT can be used to represent the instantaneous frequency of each sinusoid in a short time period, and multiplied by the scaling factor to modify the frequency content. The steps are given below, and the details can be found in reference [40] :

Suppose $x_t(n)$ is a frame of length N from the input speech signal at time t. Firstly, it is windowed by $w(n)$, and then an FFT is performed on the windowed signal, using Eq.5,

$$F(k) = \sum_{n=0}^{N-1} x_t(n) \cdot w(n) e^{-i\frac{2\pi kn}{N}} \quad 0 \leqslant k < N \tag{5}$$

where $w(n)$ is a Hamming or Hanning window and k is the bin frequency index.

Then, instantaneous magnitude $|F(k)|$ and instantaneous frequency $\omega(k)$ are respectively calculated by Eq.6 and Eq.7,

$$|F(k)| = |\sum_{n=0}^{N-1} x_t(n) \cdot w(n) e^{-i\frac{2\pi kn}{N}}| \quad 0 \leqslant k < N \tag{6}$$

$$\omega(k) = (k + \Delta) \cdot F_s/N \quad 0 \leqslant k < N \tag{7}$$

where F_s is the sampling frequency and Δ is the deviation of the k^{th} bin frequency. And the computation of Δ can be referred to in [40].

For de-identification, transient frequency $\omega(k)$ is modified by Eq.8, where α is the scale factor, i.e. the disguising factor.

$$\omega'(\lfloor k \cdot \alpha \rfloor) = \omega(k) \cdot \alpha \quad 0 \leqslant k, k \cdot \alpha < N/2 \tag{8}$$

There are several ways to modify the instantaneous magnitude. The commonest method is linear interpolation, as seen in Eq.9 [40], where $0 \leqslant k$, $k' < N$ / 2, $k = [$ k' / α $]$, and $\mu = k'$ / $\alpha - k$.

$$|F'(\lfloor k \cdot \alpha \rfloor)| = \mu|F(k)| + (1 - \mu)|F(k + 1)| \tag{9}$$

Another commonly used method is energy-preserving modification by Eq.10.

$$|F'(\lfloor k \cdot \alpha \rfloor)| = \sum_{\lfloor k\cdot\alpha \rfloor \leqslant k\cdot\alpha < \lfloor k\cdot\alpha \rfloor + 1} |F(k)| \tag{10}$$

For simplicity, we still use k as the index of the modified instantaneous frequency ω' and the instantaneous magnitude F'.

The instantaneous phase $\phi'(k)$ is then calculated via the instantaneous frequency $\omega'(k)$ and the transformed FFT coefficients is obtained by Eq.11.

$$F'(k) = |F'(k)|e^{i\phi'(k)} \tag{11}$$

An inverse FFT is performed on $F'(k)$ and the disguised signal can thus be obtained.

According to Eq.8 and Eq.9, the spectrum magnitude of the speech is modified by the de-identification disguise, so that some implicit features can be introduced into the disguised signal. Therefore, in our proposed algorithm, speech spectrum is used as the input into a deep neural network to extract deep features for classification. Using STFT, we get the spectrum diagram of the input speech signal, where the window size is 175 and the overlap is 50%.

With respect to phonetics, de-identification disguise is measured by a 12-semitones division [41] leading the disguising factor α to the following form in Eq.12.

$$\alpha(s) = 2^{s/12} \tag{12}$$

s can take any integer value in the range of [–12, +12]. However, a modification too weak or too strong can lead to deception failure or auditory unnaturalness. Here in the following experiments, we consider the median interval between [–8, –4] and between [+4, +8], which have the strongest deception ability.

2.2 Deeplearning Framework

2.2.1 The Dense Convolutional Network

In a traditional CNN, the output of the previous layer X_{l-1} is transmitted to the next layer as input by a non-linear operation H_l to get the output X_l. The non-linear operation consists of convolution, ReLU, and pooling.

$$X_l = H_l(X_{l-1}) \tag{13}$$

It is difficult to train a traditional CNN as degradation occurs with the increament of network layers. In order to effectively inhibit degradation, Residual Networks (ResNets) [42], FractalNets [43] and Highway Networks [44] create short paths X_{l-n} from early layers to late layers, as shown in Eq.14.

$$X_l = H_l(X_{l-1}) + X_{l-n} \tag{14}$$

However, recent research have shown that this type of connection leads to the fact that many layers contribute very little but occupy a large amount of computation [45]. Thus, an improved structure of ResNet named Dense Convolutional Network (DenseNet) was proposed to avoid this problem. In a DenseNet, any layer is directly connected to all subsequent layers, as shown in Eq.15,

where X_0, X_1, X_{l-1} represent the output of the previous layer of layer l, and [...] on behalf of the concatenation operation. In addition, the output dimension of each layer has k feature maps, where k is usually set to a small value.

$$X_l = H_l([X_0, X_1, ..., X_{l-1}]) \tag{15}$$

Compared with the aforementioned networks, this kind of dense connection mode has obvious advantages :1) It guarantees the maximum information flow between layers and enhances feature propagation. 2) Dense connection has regularization effect, which can reduces the overfitting of tasks with small training sets. 3) It allows the DenseNet layer to become narrower, e.g. $k = 12$, thus significantly reducing the number of parameters, alleviating degradation problems, and supporting limited neuron reuse. 4) There is no need to relearn the redundant feature maps, which is convenient for training.

2.2.2 Structure of the Proposed DenseNet

The proposed DenseNet structure is shown in Fig.1. The inputs are single channel spectrum diagrams obtained by STFT, and the sizes are set as 90×88. The network consists of an initial layer, two transition layers, three dense blocks, a global pooling layer and a linear layer. The three dense blocks are composed of 6, 12 and 48 bottleneck layers respectively. The linear layer is a full connection layer, followed by softmax, which has two outputs representing the probability of "genuine" and "disguised" respectively. The internal structure of each kind of layers are shown in Fig.2. Each bottleneck layer contains 2 convolutional layers, so the DenseNet as a whole contains $(6 + 12 + 48) \times 2 + 1 + 1 + 1 = 135$ convolutional layers.

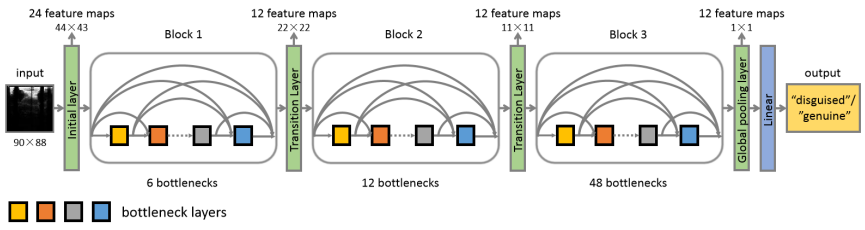

Figure 1 The architecture of the proposed network.

The bottleneck layer contains a 1×1 convolution layer, followed by a 3×3 convolutional layer, rather than two 3×3 convolution layers, to reduce the computation, as shown in Fig.2. And the transition layer connects two adjacent dense blocks, further reducing the size of the feature map.

3 Results and discussion

3.1 Experimental Corpora and Setup

Three corpora are used in the experiments, namely, UME(4040 clips, 202 speakers), NIST(3560 clips, 356 speakers) and Timit(6,300 clips, 630 speakers). They are of WAV format, 8 kHz sampling rate, 16-bit quantization and mono. We divided each corpus into 2 sets, as shown below.

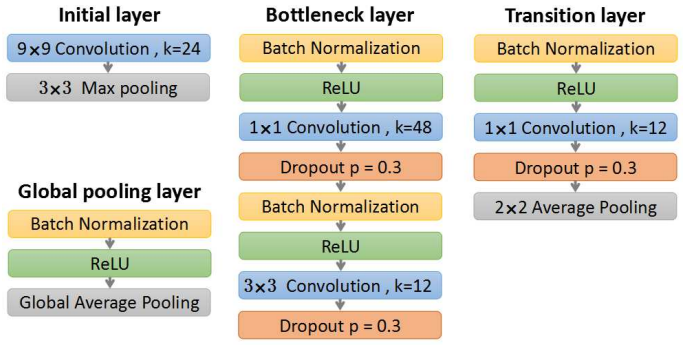

Figure 2 The internal structures of each kind of the layers. Batch Normalization [46], ReLU, Dropout [47] and Pooling are of the operations before and after the convolution layer. k is the number of convolution kernels.

Training set: UME–1(2040 clips), NIST–1(2000 clips), Timit–1 (3000 clips);

Testing set: UME–2(2000 clips), NIST–2(1560 clips), Timit–2(3300 clips).

Each clip is further cut into several 1s clips. And the number of 1s clips in each set is shown in Table 1.

Table 1 The number of 1s clips in each dataset.

Dataset	Clip number	Dataset	Clip number
NIST–1	18601	NIST–2	14589
TIMIT–1	7996	TIMIT–2	8967
UME–1	7482	UME–2	6952

Four prevailing disguising methods, i.e. Cool Edit [48], RTISI [49], PRAAT [50] and Audacity [51], with disguising factors between [–8, –4] and between [+4, +8] are taken into consideration. As a result, there are 40 times as many disguised (negative) clips as genuine (positive) clips. To achieve the balance between positive and negative data, we expand the number of positive clips by shifting every 200 samples to make it equal to the number of negative clips.

We use ADAM optimizer [52] to train the proposed DenseNet with L_2 loss function. β_1 and β_2, that is, the exponential decay rates of the first-order moment estimation value and the second-order moment estimation value are 0.9 and 0.999 respectively. Set epsilon hat ϵ as 10^{-8}, the learning rate as 10^{-4}, and the dropout rate as 0.3. The training batches are 100,000 and the batch size is 64.

The detection accuracy in Eq.16 is used to measure performance,

$$d = (G_d + D_d)/(G + D) \tag{16}$$

where G and D are the numbers of genuine clips and disguised clips in the testing sets, respectively. G_d and D_d are the number of genuine clips correctly detected from G and the number of disguised clips correctly detected from S respectively.

3.2 Intra-database evaluation

In the case of intra-database, the testing set and training set come from the same corpus. The detection results of our proposed method and the comparison with

other reported detection results are shown in Table.2. It can be seen that the average detection accuracy of our method is 2.58% higher than that of the traditional CNN model[39] and 3.66% higher than that of the SVM[38].

The method proposed by us is superior to the other two methods. The reason is that a DenseNet model has more layers than a traditional CNN, so it can extract more in-depth features to facilitate classification. In addition, in a traditional CNN, classification decision only uses deep features solely. However, in a DenseNet, due to the dense connection mode, both the deep features and the shallow edge features are utilized in the decision-making, so that the accuracy can be further improved.

Table 2 The detection accuracy of intra-database evaluation.

Case	Training set	Test set	The proposed method	Liang's method	Wu's method
Case 1	NIST–1	NIST–2	98.04%	95.93%	94.56%
Case 2	TIMIT–1	TIMIT–2	99.45%	96.52%	95.87%
Case 3	UME–1	UME–2	97.56%	94.85%	93.63%
Average			98.35%	95.77%	94.69%

3.3 Cross-database evaluation

However, in the reality scenario, testing data and training data may come from different sources, and they may have different intrinsic features. Therefore, cross-database evaluation is conducted to test the diversity of the proposed method. Here, one of the three corpora is selected as the testing dataset and the other two as the training datasets. The experimental results are shown in Table 3. We can see that the results of the first two cases are quite good, but the third case is not ideal. One possible reason is that the data volume of NIST is larger than the other two data sets, TIMIT and UME, shown in Table 1, so that the model trained by NIST has better generalization capabilities. In [39], the accuracy of case 1 is 94.37%, while our accuracy is 96.45%, indicating that our method is superior to the method in [39]. The results of the case 2 and case 3 are not given in [39].

Table 3 The detection accuracy of cross-database evaluation.

Case	Training set	Test set	The proposed method	Liang's method
Case 1	TIMIT–1&NIST–1	UME–2	96.45%	94.37%
Case 2	NIST–1&UME–1	TIMIT–2	95.26%	—
Case 3	TIMIT–1&UME–1	NIST–2	80.20%	—
Average			90.63%	—

3.4 Robustness to noise

In practical applications, noises may be introduced during recording and transmission, which may affect the detection accuracy. Here, we add Gaussian white noises into the data sets to evaluate the robustness to noise. Specifically, we add Gaussian noises of 10db, 20db and 30db, respectively, to each clean training set, and train the network by the noised sets and the clean sets. In testing, we add Gaussian noises of 10db, 15db, 20, and 30db respectively, to each clean testing set, and test the accuracy rates over each noised testing set and each clean testing set, as shown in Table 4. From the 4th column we can see that even if 10db SNR noise is added, the

accuracy rates are basically maintained at around 90%, indicating that our method has good effect on the attack from noise. It should be noted that noises of 15db are not added into the training set, but they are added into the testing sets to test the models versatility degree in the 5th column. We can see that the accuracy rates are reasonably good compared by the other conditions in column 4, 6, 7, 8.

Table 4 Detection accuracy in noised conditions.

Case	Training set	Test set	Add 10db noise	Add 15db noise	Add 20db noise	Add 30db noise	The clean voise
Case 1	NIST–1	NIST–2	91.97%	92.12%	93.83%	96.07%	96.86%
Case 2	TIMIT–1	TIMIT–2	97.65%	98.56%	99.09%	95.15%	99.18%
Case 3	UME–1	UME-2	91.82%	93.44%	94.12%	94.94%	96.31%
Case 4	TIMIT–1&NIST–1	UME–2	87.62%	90.77%	93.58%	95.84%	96.16%
Case 5	NIST–1&UME–1	TIMIT–2	90.21%	92.83%	95.06%	95.91%	96.22%
Case 6	TIMIT–1&UME–1	NIST–2	75.28%	78.12%	80.07%	80.91%	81.37%
Average			89.09%	90.97%	92.62%	93.80%	94.35%

3.5 Robustness to compression

Since compression is a necessity in audio/speech community, we test the robustness to MP3 compression in this section. We conduct MP3 compression (16:1) on the training sets and testing sets respectively. The detection accuracy is shown in Table 5. In compression cases, the average accuracy rate is 92% with a relatively slight decrease from the one of non-compression cases, indicating that the proposed method is robust to compression.

Table 5 Detection accuracy in compression and non-compression cases.

Case	Training set	Test set	Compressed wav	No compressed wav
Case 1	NIST–1	NIST–2	93.18%	98.04%
Case 2	TIMIT–1	TIMIT–2	97.57%	99.45%
Case 3	UME–1	UME–2	94.88%	97.56%
Case 4	TIMIT–1&NIST–1	UME–2	92.61%	96.45%
Case 5	NIST–1&UME–1	TIMIT–2	93.65%	95.26%
Case 6	TIMIT–1&UME–1	NIST–2	82.08%	80.20%
Average			92.33%	94.50%

4 Conclusion

This paper presents a disguised speech detection method based on dense convolutional network. Deep features can be extracted automatically by a DenseNet with 135 layers. It achieves computing efficiency by careful optimization of kernel reduction and by the employment of bottleneck layers. The experimental results show that this method is superior to the state-of-the-art methods and the future work will focus on extracting deeper features to further improve accuracy.

Abbreviations

Not applicable.

Acknowledgements

The authors would like to thank the National Natural Science Foundation of China and the the Natural Science Foundation of Guangdong Province for their support and to thank anyone who supports this paper to be published.

Author's contributions

Wang and Su conceived of the algorithm and designed the experiments. Wang and Zhu revised the manuscript. All authors read and approved the final manuscript.

Funding

This work was supported by the National Natural Science Foundation of China (61672173), the Characteristic Innovation Project of Guangdong Province Ordinary University (2015KTSCX083), the Natural Science Foundation of Guangdong Province(2014A030313623), the Guangzhou science and technology project(201803010081), the Guangzhou Science and Technology Project (201803010081), the Foundation for Innovative Young Talents by Educational Commission of Guangdong Province (2018KQNCX140) and the Qingyuan Science and Technology Plan Project (170809111721249, 170802171710591).

Availability of data and materials

The dataset used in the example are not publicly available as they are part of a larger study outside the scope of this paper but are available from the corresponding author on reasonable request.

Consent for publication

Written informed consent for publication was obtained from all participants.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]School of Electronic and Information Engineering, Guangdong Polytechnic Normal University, 510006 GuangZhou, China. [2]Audio, Speech and Vision Processing Laboratory, South China University of Technology, 510665 GuangZhou, China.

References

1. Patrick Perrot, Guido Aversano, and Gerard Chollet, "Voice disguise and automatic detection: Review and perspectives," in *The Workshop on Progress in Nonlinear Speech Processing*, 2007, pp. 101–117.
2. Z. Wu, T. Kinnunen, E. S. Chng, H. Li, and E. Ambikairajah, "A study on spoofing attack in state-of-the-art speaker verification: the telephone speech case," in *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, Dec 2012, pp. 1–5.
3. T. Kinnunen, Z. Wu, K. A. Lee, F. Sedlak, E. S. Chng, and H. Li, "Vulnerability of speaker verification systems against voice conversion spoofing attacks: The case of telephone speech," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2012, pp. 4401–4404.
4. Q. Jin, A. R. Toth, T. Schultz, and A. W. Black, "Speaker de-identification via voice transformation," in *2009 IEEE Workshop on Automatic Speech Recognition Understanding*, Nov 2009, pp. 529–533.
5. Q. Jin, A. R. Toth, T. Schultz, and A. W. Black, "Voice convergin: Speaker de-identification by voice transformation," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, April 2009, pp. 3909–3912.
6. D. Matrouf, J. . Bonastre, and C. Fredouille, "Effect of speech transformation on impostor acceptance," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, May 2006, vol. 1, pp. I–I.
7. Jean Franois Bonastre, Driss Matrouf, and Corinne Fredouille, "Artificial impostor voice transformation effects on false acceptance rates," in *Interspeech, Conference of the International Speech Communication Association, Antwerp, Belgium, August*, 2007.
8. D. Matrouf, J. . Bonastre, and C. Fredouille, "Effect of speech transformation on impostor acceptance," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, May 2006, vol. 1, pp. I–I.
9. 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*, vol. 20, no. 8, pp. 2280–2290, Oct 2012.
10. Takashi Masuko, Takafumi Hitotsumatsu, Keiichi Tokuda, and Takao Kobayashi, "On the security of hmm-based speaker verification systems against imposture using synthetic speech.," *Proc Eurospeech99 Sept*, vol. 1, pp. 1223–1226, 1999.
11. C. Hanilci, "Speaker verification anti-spoofing using linear prediction residual phase features," in *2017 25th European Signal Processing Conference (EUSIPCO)*, Aug 2017, pp. 96–100.
12. Artur Janicki, "Spoofing countermeasure based on analysis of linear prediction error," in *Interspeech*, 2015.
13. M. R. Kamble and H. A. Patil, "Novel energy separation based instantaneous frequency features for spoof speech detection," in *2017 25th European Signal Processing Conference (EUSIPCO)*, Aug 2017, pp. 106–110.
14. H. Muckenhirn, P. Korshunov, M. Magimai-Doss, and S. Marcel, "Long-term spectral statistics for voice presentation attack detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 11, pp. 2098–2111, Nov 2017.
15. J. Alam and P. Kenny, "Spoofing detection employing infinite impulse response - constant q transform-based feature representations," in *2017 25th European Signal Processing Conference (EUSIPCO)*, Aug 2017, pp. 101–105.
16. Hannah Muckenhirn, Mathew Magimai-Doss, and Sebastien Marcel, "Presentation attack detection using long-term spectral statistics for trustworthy speaker verification," in *2016 International Conference of the Biometrics Special Interest Group (BIOSIG)*, 2016, pp. 1–6.
17. Hannah Muckenhirn, Pavel Korshunov, Mathew Magimai-Doss, and Sebastien Marcel, "Long-term spectral statistics for voice presentation attack detection," *IEEE/ACM Transactions on Audio Speech & Language Processing*, vol. 25, no. 11, pp. 2098–2111, 2017.
18. Jesus Villalba and Eduardo Lleida, "Preventing replay attacks on speaker verification systems," in *IEEE International Carnahan Conference on Security Technology*, 2011, pp. 1–8.

19. Aleksandr Sizov, Elie Khoury, Tomi Kinnunen, Zhizheng Wu, and Sebastien Marcel, "Joint speaker verification and antispoofing in the i -vector space," *IEEE Transactions on Information Forensics & Security*, vol. 10, no. 4, pp. 821–832, 2015.
20. Dipjyoti Paul, Monisankha Pal, and Goutam Saha, "Spectral features for synthetic speech detection," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 4, pp. 605–617, 2017.
21. Zhizheng Wu, Phillip De Leon, Cenk Demiroglu, Ali Khodabakhsh, Simon King, Zhen Hua Ling, Daisuke Saito, Bryan Stewart, Tomoki Toda, and Mirjam Wester, "Anti-spoofing for text-independent speaker verification: An initial database, comparison of countermeasures, and human performance," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 768–783, 2016.
22. Cenk Demiroglu, Osman Buyuk, Ali Khodabakhsh, and Ranniery Maia, "Post-processing synthetic speech with a complex cepstrum vocoder for spoofing phase-based synthetic speech detectors," *IEEE Journal of Selected Topics in Signal Processing*, vol. PP, no. 99, pp. 1–1, 2017.
23. M. Sahidullah, D. A. L. Thomsen, R. G. Hautam?ki, T. Kinnunen, Z. Tan, R. Parts, and M. Pitk?nen, "Robust voice liveness detection and speaker verification using throat microphones," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 1, pp. 44–56, Jan 2018.
24. Chunlei Zhang, Chengzhu Yu, and John H. L Hansen, "An investigation of deep learning frameworks for speaker verification anti-spoofing," *IEEE Journal of Selected Topics in Signal Processing*, vol. PP, no. 99, pp. 1–1, 2017.
25. Yanmin Qian, Nanxin Chen, Heinrich Dinkel, and Zhizheng Wu, "Deep feature engineering for noise robust spoofing detection," *IEEE/ACM Transactions on Audio Speech & Language Processing*, vol. 25, no. 10, pp. 1942–1955, 2017.
26. H. Dinkel, Y. Qian, and K. Yu, "Small-footprint convolutional neural network for spoofing detection," in *2017 International Joint Conference on Neural Networks (IJCNN)*, May 2017, pp. 3086–3091.
27. Y. Qian, N. Chen, H. Dinkel, and Z. Wu, "Deep feature engineering for noise robust spoofing detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1942–1955, Oct 2017.
28. K. Lee, C. Park, N. Kim, and J. Lee, "Accelerating recurrent neural network language model based online speech recognition system," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 5904–5908.
29. A. R. Goncalves, R. P. V. Violato, P. Korshunov, S. Marcel, and F. O. Simoes, "On the generalization of fused systems in voice presentation attack detection," in *2017 International Conference of the Biometrics Special Interest Group (BIOSIG)*, Sep. 2017, pp. 1–5.
30. H. Su, R. Garg, A. Hajj-Ahmad, and M. Wu, "Enf analysis on recaptured audio recordings," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 3018–3022.
31. D. Luo, H. Wu, and J. Huang, "Audio recapture detection using deep learning," in *2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, July 2015, pp. 478–482.
32. J. Villalba and E. Lleida, "Preventing replay attacks on speaker verification systems," in *2011 Carnahan Conference on Security Technology*, Oct 2011, pp. 1–8.
33. Hardik Sailor, Madhu Kamble, and Hemant Patil, "Auditory filterbank learning for temporal modulation features in replay spoof speech detection," 06 2018.
34. M. Singh, J. Mishra, and D. Pati, "Usefulness of linear prediction residual signal for development of replay attacks detection system," in *2017 Twenty-third National Conference on Communications (NCC)*, March 2017, pp. 1–4.
35. K. Sriskandaraja, G. Suthokumar, V. Sethu, and E. Ambikairajah, "Investigating the use of scattering coefficients for replay attack detection," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Dec 2017, pp. 1195–1198.
36. Yong Wang, Yanhong Deng, Haojun Wu, and Jiwu Huang, "Blind detection of electronic voice transformation with natural disguise," in *The International Workshop on Digital Forensics and Watermarking 2012*, Yun Q. Shi, Hyoung-Joong Kim, and Fernando Pérez-González, Eds., Berlin, Heidelberg, 2013, pp. 336–343, Springer Berlin Heidelberg.
37. H. Wu, Y. Wang, and J. Huang, "Identification of electronic disguised voices," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 3, pp. 489–500, March 2014.
38. Haojun Wu, Yong Wang, and Jiwu Huang, "Blind detection of electronic disguised voice," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 3013–3017.
39. Huixin Liang, Xiaodan Lin, Qiong Zhang, and Xiangui Kang, "Recognition of spoofed voice using convolutional neural networks," in *IEEE Global Conference on Signal and Information Processing*, 2017, pp. 293–297.
40. Jean Laroche, "Time and pitch scale modification of audio signals(chapter 7)," in *International Series in Engineering and Computer Science*. 2006, vol. 437, pp. 279–309, Springer US.
41. Sandra E Trehub, Annabel J Cohen, Leigh A Thorpe, and Barbara A Morrongiello, "Development of the perception of musical relations: Semitone and diatonic structure.," *Journal of Experimental Psychology Human Perception & Performance*, vol. 12, no. 3, pp. 295, 1986.
42. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.
43. Gustav Larsson, Michael Maire, and Gregory Shakhnarovich, "Fractalnet: Ultra-deep neural networks without residuals," *CoRR*, vol. abs/1605.07648, 2016.
44. Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber, "Training very deep networks," *CoRR*, vol. abs/1507.06228, 2015.
45. Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q. Weinberger, "Deep networks with stochastic depth," *CoRR*, vol. abs/1603.09382, 2016.
46. Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," pp. 448–456, 2015.
47. Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a

simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
48. "Cool edit pro is now adobe audition," [Online]. Available: http://www.adobe.com/products/audition.html, 2012.
49. "Time-scale / pitch modification tools," [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/25880-time-scalepi, 2012.
50. "Praat: Doing phonetics by computer," [Online]. Available: http://www.fon.hum.uva.nl/praat, 2012.
51. "Audacity:free audio editor and recorder," [Online]. Available: http://audacity.sourceforge.net, 2012.
52. Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

Figures

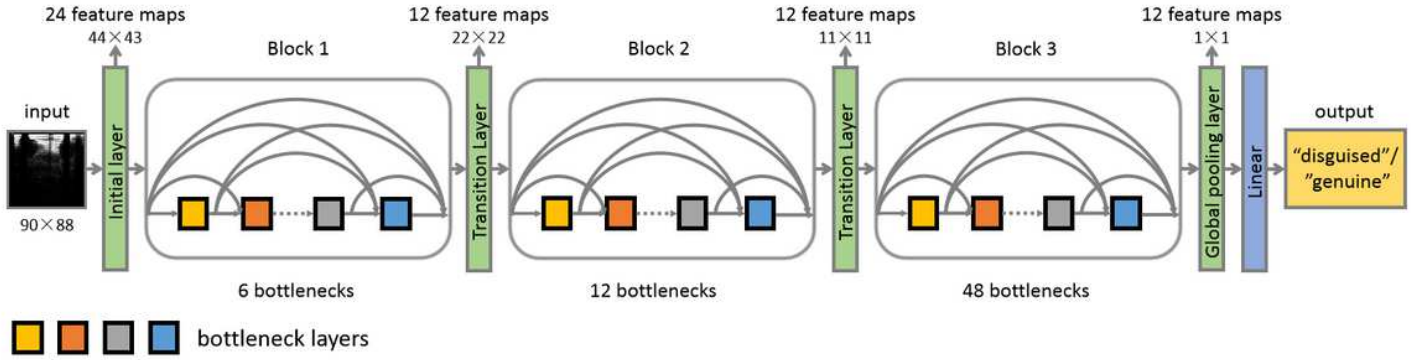

Figure 1

The architecture of the proposed network.

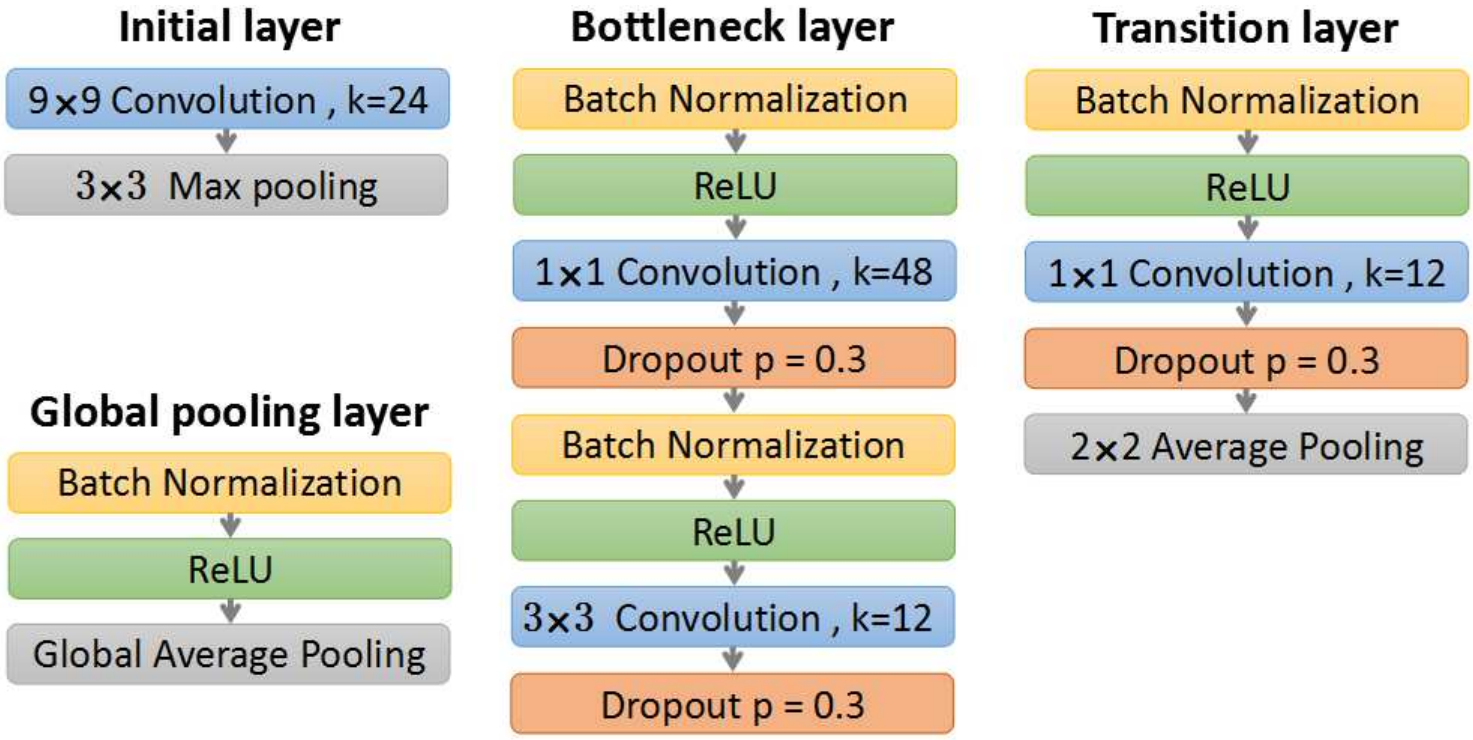

Figure 2

The internal structures of each kind of the layers. Batch Normalization [46], ReLU, Dropout [47] and Pooling are of the operations before and after the convolution layer. k is the number of convolution kernels.