Odyssey 2020 The Speaker and Language Recognition Workshop
1-5 November 2020, Tokyo, Japan

PHASE SPECTRUM OF TIME-FLIPPED SPEECH SIGNALS FOR ROBUST SPOOFING DETECTION

Sung-Hyun Yoon[1], *Min-Sung Koh*[2], *and Ha-Jin Yu*[1*]

[1] School of Computer Science, University of Seoul, Seoul 02504, Republic of Korea
[2] School of Computing and Engineering Sciences, Eastern Washington University, Cheney, WA 99004 USA

ysh901108@naver.com, mkoh@ewu.edu, hjyu@uos.ac.kr

ABSTRACT

In spoofing detection, it is important to capture the attributes related to spoofing attacks from a speech signal. A speech signal has various information such as the speaker, phrase, and environment. When the time sequence of the speech signal is flipped (i.e., time reversal and an additional circular shift), phase spectrum is changed although magnitude spectrum is not changed. It has the effect of data augmentation showing additional attributes in phase spectrum which are not included in magnitude spectrum. We assume that those additional attributes in phase spectrum of time-flipped speeches are related to unseen intraclass conditions. Motivated by our assumption, we propose a method of using the phase spectrum based features from both the original and time-flipped speech signals together. If our assumption stands good, it has the effect of reducing intraclass variances because the previously unseen attributes in magnitude spectrum can be considered in phase spectrum. The additional attributes in phase spectrum are helpful to build more robust spoofing detection systems. The experimental results on ASVspoof 2019 logical and physical access scenarios exhibit significant performance improvements for both scenarios compared to that of the baseline.

1. INTRODUCTION

Automatic speaker verification (ASV) has been widely used as a biometric method because of its convenience: it requires only a voice for verification, and has shown remarkable verification performance. As the popularity of ASV has increased, the reliability of ASV has also become important. Reliability means the ability to distinguish whether a given speech is bona fide speech (i.e., spoken by a human) or spoofed speech (i.e., generated by spoofing attacks). There are some kinds of spoofing attacks that can be detected in practice: text-to-speech (TTS) synthesis, voice conversion (VC), and replay attack. ASV systems with low reliability cannot be used for verification even if they show high verification performance. However, ASV is vulnerable to various spoofing attacks [1, 2]. Therefore, spoofing detection techniques are required for the reliability of ASV.

In recent years, ASV spoofing and countermeasures (ASVspoof) challenges have been organized periodically to develop spoofing detection techniques [3, 4, 5]. In particular, ASVspoof 2019 [5], which is the latest challenge on spoofing and countermeasures for ASV, covers all the spoofing attacks mentioned above. This challenge includes two scenarios: logical access (LA) and physical access (PA). The LA scenario covers spoofing attacks generated by TTS and VC. The PA scenario covers a replay attack. Many studies have shown remarkable performance on the LA and/or PA scenarios of ASVspoof 2019.

In common to both scenarios, most studies have focused on analyzing the frequency response of speech signals in various ways. The spectra of speech signals include various attributes related to the conditions under which the signal was produced, including the speaker, phrase, background noise, and so on. If the speech was generated by a spoofing attack, it would also include the attributes generated through the spoofing attack. For example, the speeches generated by TTS or VC do not have the proper dynamic information of bona fide speeches [6], and there is no phase information of bona fide speeches [7]. The speeches generated by replay attacks have the attributes of the devices used for the replay attack, such as playback and/or recording device(s). Moreover, the attributes related to spoofing attacks are in the entire frequency domain. Therefore, for spoofing detection, it is important to adequately capture the attributes related to spoofing attacks in speeches.

When the time sequence of a speech signal is flipped, it sounds different from the original signal. It can be seen that many attributes included in the signal are distorted, resulting in the change of their identity. For example, compared to the original signal, the time-flipped speech signal sounds like an arbitrary phrase is being spoken in an unknown language. This means that the attributes of the phrase and language are distorted, resulting in a change in the identities of the original phrase and language. Similar to other attributes, the attributes related to spoofing attacks may also be distorted if the time sequence of the signal is flipped. We assume that it has the effect of generating the speeches of unseen intraclass conditions (e.g., phrase and language) for each class (i.e., bona fide and spoof). Motivated by this assumption, we propose a method of using both the original and time-flipped speech signals together for ASV spoofing detection, which is expected to build more robust spoofing detection systems against the changes in the intraclass variance. We expect that the performance of the spoofing detection system is improved if the assumption holds.

The remainder of this paper is organized as follows. Section 2 outlines preliminaries related to our research. Section 3 introduces the proposed method and the ways to utilize it.

* Corresponding author: Ha-Jin Yu (hjyu@uos.ac.kr)

10.21437/Odyssey.2020-45

Section 4 describes the experiments and their results. Finally, Section 5 concludes the paper.

2. PRELIMINARIES

2.1. Spectrum-based acoustic features

Most of the features used for spoofing detection, such as the log power spectrum, constant Q cepstral coefficients (CQCC) [6], and linear frequency cepstral coefficients (LFCC) [8], are also derived from the magnitude spectrum. In particular, the CQCC and LFCC are used in the baseline systems of ASVspoof 2019. The magnitude spectrum-based features have shown remarkable performance in various ways. However, these features do not have information contained in the phase spectrum.

The phase spectrum gives high spectral resolution not contained in the magnitude spectrum [9]. It has been found that the phase spectrum-based features are effective for spoofing detection [10, 11, 12]. For our experiments, we used group delay [13] as a phase spectrum-based feature, which has shown a significant effect on spoofing detection [14, 15, 16]. Let $x(n)$ be a sequence of speech signals corresponding to one frame. The group delay is defined as the negative gradient of the unwrapped phase spectrum $\theta(\omega)$ of $x(n)$ with respect to the frequency ω, as follows:

$$\tau(\omega) = -\frac{d\big(\theta(\omega)\big)}{d\omega}. \quad (1)$$

In many cases, the group delay function is directly computed as follows, like Eq. (5) in [13]:

$$\tau(\omega) = \frac{X_R(\omega)Y_R(\omega) + Y_I(\omega)X_I(\omega)}{|X(\omega)|^2} \quad (2)$$

where $X(\omega)$ and $Y(\omega)$ are the spectra of $x(n)$ and $nx(n)$, respectively, and the subscripts R and I denote the real and imaginary parts of the spectrum, respectively. In this paper, however, we computed the group delay function by approximating the gradient using numerical differentiation [17]. Second-order central differences were used in the interior points, and first-order forward/backward differences were used at the boundaries as follows:

$$\left.\frac{d\theta}{d\omega}\right|_{\omega=\omega_i} \approx \begin{cases} \dfrac{\theta(\omega_{i+1}) - \theta(\omega_i)}{\omega_{i+1} - \omega_i} & i = 0 \\ \dfrac{\theta(\omega_{i+1}) - \theta(\omega_{i-1})}{\omega_{i+1} - \omega_{i-1}} & 1 \le i \le D-2 \\ \dfrac{\theta(\omega_i) - \theta(\omega_{i-1})}{\omega_i - \omega_{i-1}} & i = D-1 \end{cases} \quad (3)$$

where ω_i is the i-th fast Fourier transform (FFT) bin center frequency, and D is the number of FFT bins.

2.2. Convolutional neural network based models

A convolutional neural network (CNN) effectively captures various local information in an input feature with many receptive fields that have small sizes. There are some kinds of CNN-based models such as residual network (ResNet) [18] and light CNN (LCNN) [19]. These models were initially proposed for image recognition and showed state-of-the-art performance. They have also been successfully adapted in the field of spoofing detection [16, 20, 21, 22, 23, 24, 25, 26].

In this paper, we build spoofing detection systems based on ResNet. This network has a residual learning framework to optimize very deep networks easily and has become a widely used structure in many studies. In [27], a squeeze-and-excitation (SE) block was proposed to model the relationship among the channels. This approach adaptively recalibrates channel-wise responses: the squeeze operation captures global information of each channel, and then the excitation operation recalibrates the weights for all channels. SE can be directly integrated into each residual block of the ResNet, and the result is called the squeeze-and-excitation residual network (SE-ResNet). The SE-ResNet showed promising performance in the ASVspoof 2019 challenge [21].

All features of an input utterance in each mini-batch have to be the same size to be fed into the CNN in the minibatch unit. However, speech signals have variable lengths, and we do not know the length in advance. In [21], the feature segmentation method was used to make all features have the same length. In this method, all utterances were extended by repeating some of its frames to have the longest length of all utterances over entire dataset. Then, the utterances were cut into multiple segments of length T with shift S. All features have the same number of segments. Note that all the segments obtained from the same input utterance share the same class label that corresponds to the label of the input. In the training phase, each segment is considered as individual input, regardless of which utterance the segment came from. In the evaluation phase, the score of each original input utterance is computed by averaging the scores of all the segments from the original utterance. In this paper, we used a slightly different method of feature segmentation to avoid data redundancy: we extended only the utterance shorter than the fixed length of a segment T to have the length of T. In this case, the number of segments in the utterance is one. Therefore, all utterances have different numbers of segments.

3. THE PROPOSED METHOD

3.1. Phase spectrum of the time-flipped speech signal

In this paper, we propose the method of using the original and time-flipped speech signals together. We apply our method only to the phase spectrum-based feature, because the magnitude spectrum of the original signal $x(n)$ and that of the time-flipped signal $\tilde{x}(n) = x(-n)$ are equal to each other, that is, $|X(\omega)| = \left|\tilde{X}(\omega)\right| = |X(-\omega)|$, where $|X(\omega)|$ and $\left|\tilde{X}(\omega)\right|$ are the magnitude spectra of $x(n)$ and $\tilde{x}(n)$, respectively.

The phase spectrum is changed when the time order is flipped, that is, $\theta(\omega) \neq \tilde{\theta}(\omega) = \theta(-\omega)$, where $\theta(\omega)$ and $\tilde{\theta}(\omega)$ denote the phases of $x(n)$ and $\tilde{x}(n)$ respectively. It can be seen that the attributes of the phase spectrum are changed, and their identities are also changed. We assumed that only the identities not related to spoofing attacks (i.e., those that compose the intraclass variance) are changed, and those related to spoofing attacks (i.e., those that compose the interclass variance) are not changed. Note that the **attributes** related to spoofing attacks may be changed, but it is not enough to distort the **identities** related to spoofing attacks because traditional speaker identification algorithms utilize the magnitude spectrum only. In our assumption, therefore, the class

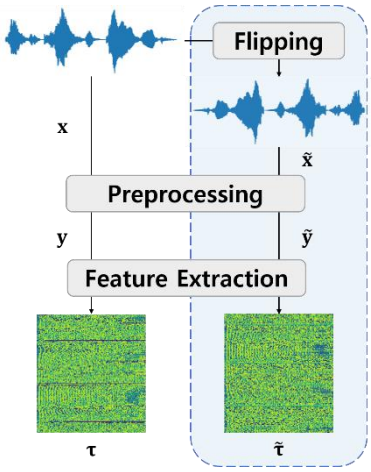

Figure 1. *The framework of (left) the conventional and (right, highlighted in blue) proposed methods.*

information (i.e., bona fide or spoof) of the signal is not changed even if the time order of the signal is flipped. This assumption has the effect of generating data that have unseen attributes while maintaining class information.

Figure 1 shows the framework of the proposed method, where $\mathbf{x} = [x_1(n), \ldots, x_T(n)]$ is a segment with T frames, $\tilde{\mathbf{x}} = [\tilde{x}_T(n), \ldots, \tilde{x}_1(n)]$ is the time-flipped segment, $\mathbf{y} = [y_1(n), \ldots, y_T(n)]$ and $\tilde{\mathbf{y}} = [\tilde{y}_T(n), \ldots, \tilde{y}_1(n)]$ are the preprocessed segments from $\mathbf{x}$ and $\tilde{\mathbf{x}}$, respectively, and $\boldsymbol{\tau} = [\tau_1(\omega), \ldots, \tau_T(\omega)]$ and $\tilde{\boldsymbol{\tau}} = [\tilde{\tau}_T(\omega), \ldots, \tilde{\tau}_1(\omega)]$ are the features from $\mathbf{y}$ and $\tilde{\mathbf{y}}$, respectively. The feature τ corresponds to the group delay, as mentioned in Section 2.1.

3.2. The methods to combine features

We can obtain two input features, $\boldsymbol{\tau}$ and $\tilde{\boldsymbol{\tau}}$, from a signal corresponding to one segment. We propose several methods that enable CNN-based models to use both features (i.e., $\boldsymbol{\tau}$ and $\tilde{\boldsymbol{\tau}}$) together at one time.

3.2.1. 2-channel input

Most spectrogram-based features from speech signals are in 2-D matrix form, that is, have the shape of $(T \times D)$, where M is the number of frames in a segment and D is the dimensionality of the feature. Such a feature can be viewed as a 3-D tensor that has the shape of $(1 \times T \times D)$, that is, the number of channels is 1. This method combines the two features at the input level (denoted as *2ch*). Figure 2b depicts the framework of the *2ch* method. It stacks the features $\boldsymbol{\tau}, \tilde{\boldsymbol{\tau}}$ to make the 2-channel input $[\boldsymbol{\tau}, \tilde{\boldsymbol{\tau}}]$ so that the input has the shape of $(2 \times T \times D)$, and then, the 2-channel input is fed into the CNN. Consequently, the number of parameters for the first layer of the CNN doubles.

3.2.2. Embedding-level combination

This method combines the two features at the embedding-level. The embedding means the output of the global average pooling (GAP) layer. Figure 2c depicts the framework of this method. Up to the GAP layer, each feature is fed into the same CNN independently. That is, the CNNs for each feature share the same parameters. After the two embeddings are computed from

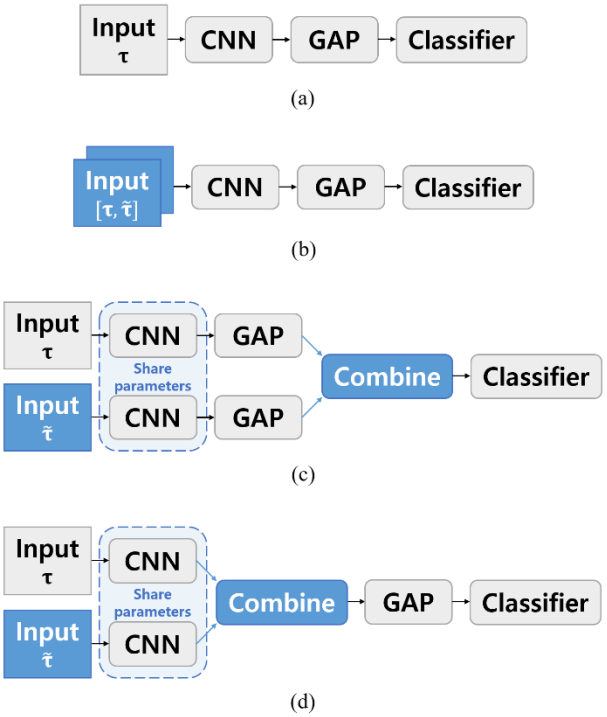

Figure 2. *The framework of the (a) baseline and proposed systems. The proposed systems consist of (b) 2-channel inputs, (c) embedding-level concatenation, and (d) feature map-level combination.*

$\boldsymbol{\tau}$ and $\tilde{\boldsymbol{\tau}}$, they are combined to form a single embedding in one of the following three ways. One way is concatenation (denoted as *concat*), in which the combined embedding vector has twice the size of each original embedding. Consequently, the number of parameters for the classifier doubles. Another way is elementwise maximum (denoted as *vmax*), and the other is elementwise averaging (denoted as *vmean*). In contrast to that in *concat*, the combined embedding vector has the same size as each original embedding in *vmax* and *vmean*. Therefore, both methods require no additional parameters.

3.2.3. Feature map-level combination

This method combines the two features at the feature map-level. The feature map means the output of the last layer of the CNN. Figure 2d depicts the framework of this method. In this method, as in Section 3.2.2, each feature is fed into the same CNN independently. After the two feature maps are computed from $\boldsymbol{\tau}$ and $\tilde{\boldsymbol{\tau}}$, they are combined to form a single feature map by taking elementwise maximum (denoted as *fmax*). In this way, the combined feature map has the same shape as each of the original feature maps. Therefore, this method also requires no additional parameters. The combined feature map is used to compute an embedding by GAP.

4. EXPERIMENTS

4.1. Database

We used the ASVspoof 2019 database for our experiments. Table 1 shows the number of utterances in the ASVspoof 2019

LA and PA databases. We used the training set to build the systems, used the development set for validation and to search hyperparameters, and used the evaluation set to evaluate the performance of the selected systems.

Table 1. *The number of utterances in the ASVspoof 2019 LA and PA databases*

		LA	PA
Training	Genuine	2,580	5,400
	Spoof	22,800	48,600
Development	Genuine	2,548	5,400
	Spoof	22,296	24,300
Evaluation	Genuine	7,355	18,090
	spoof	63,882	116,640

4.2. Experimental setup

We used 257-dimensional group delay as the acoustic feature for each frame. The 25 ms frames were extracted at 10 ms intervals. The features τ_t and $\tilde{\tau}_t$ were extracted from the original frame x_t and the flipped frame $\tilde{x}_t$, respectively. The same subsequent process was performed for both types of frames. The preprocessing was performed for all frames: remove DC offset followed by pre-emphasis filtering with a coefficient of 0.97. Hamming window was applied for all the preprocessed frames. The number of FFT points was 512. We did not apply mean and variance normalization. No silence frames were removed because silence frames can have useful information for spoofing detection [28]. As mentioned in Section 2.2, we used the method of feature segmentation with length $T = 400$ frames and overlap $S = 200$ frames, as in [21]. We set the length of a segment T to 400 because the average number of frames of the utterances of ASVspoof 2019 LA and PA databases are about 325 and 428, respectively. For the features longer than 400, we segment the features into multiple segments of length 400. For the features shorter than 400, we extended the features to have the length of 400 by repeating some of its frames.

We used SE-ResNet34 with a softmax classifier. Table 2 shows the architecture of the SE-ResNet34 model, where C_i is the number of channels of input and N_e is the number of embeddings. C_i is 2 only for the *2ch* method and is 1 for the other methods. N_e is 2 only for the *concat* method and is 1 for the other method. For the LA scenario, the number of classes N_c was 3: bona fide, TTS and VC. For the PA scenario, N_c was 2: bona fide and spoof. Cross-entropy loss was used. AMSGrad [29], a variant of the Adam [30] optimizer, was used to optimize the network with a learning rate of 10^{-3} and weight decay of 10^{-4}. All weights were initialized from the He normal distribution [31]. No bias was used. We trained the networks for 100 epochs with a minibatch size of 64, and selected the best model by validation. We implemented the networks using PyTorch [32].

Table 2. *The architecture of SE-ResNet34*

Layer	Structure	Output size
Input	-	C_i ×400×257
CNN	7×7, stride 2	16×200×129
	max pool, 3×3, stride 2 $\begin{bmatrix} 3 \times 3, 16 \\ 3 \times 3, 16 \\ SE, [1, 16] \end{bmatrix} \times 2$, stride 1	16×100×65
	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \\ SE, [2, 32] \end{bmatrix} \times 2$, stride 2	32×50×33
	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \\ SE, [4, 64] \end{bmatrix} \times 2$, stride 2	64×25×17
	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \\ SE, [8, 128] \end{bmatrix} \times 2$, stride 2	128×13×9 (feature map)
GAP	-	128× N_e (embedding)
Classifier	-	N_c

We used two evaluation metrics: equal error rate (EER) and minimum normalized tandem detection cost function (t-DCF$_n^m$) [33]. The parameters for computing t-DCF$_n^m$ were the same as those used in the ASVspoof 2019 challenge.

4.3. Results

Table 3 shows the EERs and t-DCF$_n^m$ of the systems on the development and evaluation trials of the LA scenario. Note that compared to the baseline, *2ch* has 49 ($= 7 \times 7$; the filter size of the first layer of the baseline CNN) more parameters and *concat* has 384 ($= 128 \times 3$; the number of parameters of the baseline classifier) more parameters. In the development trial, *2ch* showed the worst performance, and all other proposed systems showed better performance than the baseline. *fmax* showed the lowest EER and t-DCF$_n^m$, exhibiting relative reductions of approximately 96% and 97%, respectively. In contrast, in the evaluation trial, *vmax* showed the worst performance. Additionally, only *vmean* and *fmax* showed better performance than the baseline. In particular, *vmean* showed the lowest EER and t-DCF$_n^m$, exhibiting relative reductions of approximately 12% and 4%, respectively. If both *2ch* and *concat* had shown the performance improvement, we can say that the cause of the performance improvement may be due to the increase in the number of the parameters. In other words, we couldn't have claimed that the proposed method of combining two features itself can improve the performance. However, both *2ch* and *concat* showed worse performances than the baseline. Therefore, the increase in the number of parameters cannot be the cause of the performance improvement. Even so, we cannot say that the result is due to the overfitting with an increased number of parameters, because *concat*, which has more parameters than *2ch*, showed better performance than *2ch* except for t-DCF$_n^m$ on the evaluation trial. Additionally, the

increased number of parameters accounts for a small proportion of the total number of parameters. Rather, it is more convincing to explain that the deep embedding-level combinations are more efficient than the input feature-level combination.

Table 3. *The EERs (%) and t-DCF$_n^m$ of the systems on the development and evaluation trials of ASVspoof 2019 LA*

System	Development		Evaluation	
	EER	t-DCF$_n^m$	EER	t-DCF$_n^m$
Baseline	0.157	0.0038	8.3889	0.2206
2ch	0.5881	0.0137	9.5613	0.2221
concat	0.1104	0.0024	8.7702	0.2759
vmax	0.1149	0.0025	10.483	0.301
vmean	0.0376	0.0004	**7.4233**	**0.2112**
fmax	**0.0067**	**0.0001**	7.7469	0.219

Table 4 shows the EERs and t-DCF$_n^m$ of the systems on the development and evaluation trials of the PA scenario. Note that compared to the baseline, *2ch* has 49 ($= 7 \times 7$; the filter size of the first layer of the baseline CNN) more parameters and *concat* has 256 ($= 128 \times 2$; the number of parameters of the baseline classifier) more parameters. Unlike in the LA scenario, all proposed systems showed better performance than the baseline, except that *2ch* showed a slightly higher EER on the development trial. Among the proposed systems, *2ch* showed the worst performance, which may be because the deep embedding-level combinations are more efficient, as mentioned above. *vmax* showed the best performance on both trials, presenting a relative EER reduction of approximately 27% and 31% on the development and evaluation trials, respectively, and showed a relative t-DCF$_n^m$ reduction of approximately 31% on both trials.

Table 4. *The EERs (%) and t-DCF$_n^m$ of the systems on the development and evaluation trials of ASVspoof 2019 PA*

System	Development		Evaluation	
	EER	t-DCF$_n^m$	EER	t-DCF$_n^m$
Baseline	4.4599	0.1261	5.4174	0.154
2ch	4.536	0.1183	5.3468	0.1498
concat	3.4465	0.0947	4.3512	0.1232
vmax	**3.2572**	**0.0873**	**3.7325**	**0.107**
vmean	3.7953	0.1104	4.6167	0.1316
fmax	3.7562	0.1081	4.5994	0.1295

4.4. Results with fusion

As mentioned in Section 2, the magnitude spectrum and phase spectrum have different information. Better performance is expected when using both types of information together. Table 4 shows the EERs and t-DCF$_n^m$ values of the baseline system with log power spectra on the development and evaluation trials of the LA and PA scenarios. The only difference is that the baseline systems shown in Table 5 used the log power spectrogram as an acoustic feature, rather than the group delay. We empirically found that the log power spectrum, which is one of the magnitude spectrum-based features, showed significantly better performance than the group delay in the PA scenario. In the LA scenario, these features showed similar performance. It can be seen that the information in the magnitude spectrum is more effective for replay attack detection than that in the phase spectrum. We expected to obtain more robust spoofing detection systems when using the magnitude and phase spectrum-based features together because they have different kinds of information. We performed score-level fusion of the phase and magnitude spectrum-based systems. The weights for each score were estimated by logistic regression using the scores from the development set.

Table 5. *The EERs (%) and t-DCF$_n^m$ of the baseline system with log power spectrum on the development and evaluation trials of ASVspoof 2019 LA and PA*

Scenario	Development		Evaluation	
	EER	t-DCF$_n^m$	EER	t-DCF$_n^m$
LA	0.0045	0.0001	8.754	0.1852
PA	0.8107	0.0238	1.255	0.0382

Table 6 shows the performance of the fused systems on both trials of the LA scenario. In the development trial, all the systems showed perfect performance (i.e., both EER and t-DCF$_n^m$ are 0). Unlike the results without fusion, in which *vmean* showed the best performance, *fmax* showed the best performance in terms of EER, exhibiting a relative reduction of approximately 12%. However, the baseline system showed the lowest t-DCF$_n^m$. These results are quite different from those of the systems without fusion (Table 3).

Table 6. *The EERs (%) and t-DCF$_n^m$ of the fused systems on the development and evaluation trials of ASVspoof 2019 LA*

Fusion System	Development		Evaluation	
	EER	t-DCF$_n^m$	EER	t-DCF$_n^m$
Baseline	0	0	7.125	**0.1576**
2ch	0	0	8.3889	0.1764
concat	0	0	6.5669	0.1751
vmax	0	0	7.9822	0.1877
vmean	0	0	6.7983	0.1688
fmax	**0**	**0**	**6.3054**	0.1638

Table 7 shows the performance of the fused systems on both trials of the PA scenario. As expected, the fused systems showed higher performance for all conditions. *vmax* showed the best performance on both trials, which is equivalent to the results of the systems without fusion (Table 4). In terms of the EER, *vmax* showed relative reductions of approximately 3% and 8% on the development and evaluation trials, respectively. In terms of t-DCF$_n^m$, it showed the relative reduction of about 4% and 12% on the development and evaluation trials, respectively.

Table 7. *The EERs (%) and t-DCF$_n^m$ of the fused systems on the development and evaluation trials of ASVspoof 2019 PA*

Fusion System	Development		Evaluation	
	EER	t-DCF$_n^m$	EER	t-DCF$_n^m$
Baseline	0.6121	0.0191	1.0449	0.0325
2ch	0.6471	0.021	1.1342	0.0333
concat	0.6296	0.0196	1.0281	0.032
vmax	**0.5926**	**0.0183**	**0.9619**	**0.0287**
vmean	0.6255	0.0198	1.1271	0.0339
fmax	0.6626	0.0212	1.1173	0.0331

5. CONCLUSION

In this paper, we propose a method in which the phase spectrum-based features are extracted from the original and time-flipped speech signals and are used together for ASV spoofing detection. This approach is motivated by our assumption that the time-flipped signals have the effect of data augmentation to show the additional attributes in phase spectrum, which are not included in magnitude spectrum but related to unseen intraclass conditions. We also propose methods to combine both features for CNNs. The proposed methods show better performance than the conventional method (i.e., using the original signal only) on the tasks of ASVspoof 2019 LA and PA. In particular, we have found that the deep embedding-level combination is more efficient than the input-level combination. Based on the results, we claim that the assumption holds and more robust spoofing detection systems can be built by the proposed method.

6. ACKNOWLEDGMENT

This research was supported by Projects for Research and Development of Police Science and Technology under the Center for Research and Development of Police Science and Technology and Korean National Police Agency funded by the Ministry of Science, ICT and Future Planning (PA-J000001-2017-101).

7. REFERENCES

1. Johan Lindberg and Mats Blomberg, "Vulnerability in speaker verification – A study of technical impostor techniques," in European Conference on Speech Communication and Technology, 1999.
2. Tomi Kinnunen, Zhi-Zheng Wu, Kong Aik Lee, Filip Sedlak, Eng Siong Chng, and Haizhou Li, "Vulnerability of speaker verification systems against voice conversion spoofing attacks: The case of telephone speech," in IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP), 2012, pp. 4401-4404.
3. Zhizheng Wu, Tomi Kinnunen, Nicholas Evans, Junichi Yamagishi, Cemal Hanilci, Md Sahidullah, and Aleksandr Sizov, "ASVspoof 2015: The first automatic speaker verification spoofing and countermeasures challenge," Interspeech, 2015.
4. Tomi Kinnunen, Md Sahidullah, Hector Delgado, Massimiliano Todisco, Nicholas Evans, Junichi Yamagishi, and Kong Aik Lee, "The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection," Interspeech, 2017.
5. Massimiliao Todisco, Xin Wang, Ville Vestman, Md Sahidullah, Hector Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Tomi Kinnunen, and Kong Aik Lee, "ASVspoof 2019: Future horizons in spoofed and fake audio detection," Interspeech, 2019.
6. Massimiliano Todisco, Hector Delgado, and Nicholas Evans, "Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification," Computer, Speech and Language, vol. 45, pp. 516-535, 2017.
7. Zhizheng Wu, Eng Siong Chng, and Haizhou Li, "Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition," Interspeech, 2012.
8. Hema A Murthy and Venkata Gadde, "The modified group delay function and its application to phoneme recognition," in IEEE International Conference on

Acoustic, Speech, and Signal Processing (ICASSP), 2003, vol. 4 no. 1, pp. 68-70.

9. Meng Liu, Longbiao Wang, Jianwu Dang, Seiichi Nakagawa, Haotian Guan, and Xiangang Li, "Replay attack detection using magnitude and phase information with attention-based adaptive filters," in IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP), 2019, pp. 6201-6205.
10. Jon Sanchez, Ibon Saratxaga, Inma Hernaez, Eva Navas, Daniel Erro, and Tuomo Raitio, "Toward a universal synthetic speech spoofing detection using phase information," IEEE Transactions on Information Forensics and Security, vol. 10 no. 4, pp. 810-820, 2015.
11. Longbiao Wang, Yohei Yoshida, Yuta Kawakami, and Seiichi Nakagawa, "Relative phase information for detecting human speech and spoofed speech," IEEE Journal of Selected Topics in Signal Processing, vol. 11, no. 4, pp. 660-670, 2017.
12. Shaik Mohammad Rafi B and K Sri Rama Murty, "Importance of analytic phase of the speech signal for detecting replay attacks in automatic speaker verification systems," in IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP), 2019, pp. 6306-6310.
13. Rajesh M. Hegde, Hema A. Murthy, and Venkata Ramana Rao Gadde, "Significance of the modified group delay feature in speech recognition," IEEE Transactions on Audio, Speech, and Language Processing, vol. 15 no. 1, pp. 190-202, 2006.
14. Zhizheng Wu, Eng Siong Chng, and Haizhou Li, "Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition," Thirteenth Annual Conference of the International Speech Communication Association, 2012.
15. Francis Tom, Mohit Jain, and Prasenjit Dey, "End-to-end audio replay attack detection using deep convolutional networks with attention," Interspeech, 2018.
16. Weicheng Cai, Haiwei Wu, Danwei Cai, and Ming Li, "The DKU replay detection system for the ASVspoof 2019 challenge: On data augmentation, feature representation, classification, and fusion," Interspeech, 2019.
17. Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri, "Numerical mathematics," vol. 37, Springer Science & Business Media, 2010.
18. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," IEEE conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, 2016.
19. Xiang Wu, Ran He, Zhenan Sun, and Tieniu 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.
20. Cheng-I Lai, Alberto Abad, Korin Richmond, Junichi Yamagishi, Najim Dehak, and Simon King, "Attentive filtering networks for audio replay attack detection," in IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP), 2019, pp. 6316-6320.
21. Cheng-I Lai, Nanxin Chen, Jesus Villalba, and Najim Dehak, "ASSERT: Anti-spoofing with squeeze-excitation and residual networks," Interspeech, 2019.
22. Galina Lavrentyeva, Sergey Novoselov, Andzhukaev Tseren, Marina Volkova, Artem Gorlanov, and Alexandr Kozlov, "STC antispoofing systems for the ASVspoof2019 challenge," Interspeech, 2019.
23. Radoslaw Bialobrzeski, Michal Kosmider, Mateusz Matuszewski, Marcin Plata, and Alexander Rakowski, "Robust Bayesian and light neural networks for voice spoofing detection," Interspeech, 2019.
24. Yexin Yang, Hongji Wang, Heinrich Dinkel, Zhengyang Chen, Shuai Wang, Yanmin Qian, and Kai Yu, "The SJTU robust anti-spoofing system for the ASVspoof 2019 challenge," Interspeech, 2019.
25. Hossein Zeinali, Themos Stafylakis, Georgia Athnaasopoulou, Johan Rohdin, Inannic Gkinis, Lukas Burget, and Jan Honza Cernocky, "Detecting spoofing attacks using VGG and SinceNet: BUT-Omilia submission to ASVspoof 2019 challenge," Interspeech, 2019.
26. Moustafa Alzantot, Ziqi Wang, and Mani B. Srivastava, "Deep residual neural networks for audio spoofing detection," Interspeech, 2019.
27. Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation networks," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7132-7141, 2018.
28. Md Sahidullah, Tomi Kinnunen, and Cemal Hanilci, "A comparison of features for synthetic speech detection," Interspeech, 2015.
29. Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar, "On the convergence of adam and beyond," arXiv:1904.09237v1, 2019.
30. Diederik P. Kingma and Jimmy Lei Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980, 2014.
31. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian 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.
32. Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in PyTorch," NIPS 2017 Workshop Autodiff Submission, 2017.
33. Tomi Kinnunen, Kong Aik Lee, Hector Delgado, Nicholas Evans, Massimiliano Todisco, Md Sahidullah, Junichi Yamagishi, and Douglas A. Reynolds, "t-DCF: A detection cost function for the tandem assessment of spoofing countermeasures and automatic speaker verification," Proc. Odyssey 2018 – The Speaker and Language Recognition Workshop.