Improving Multi-Scale Aggregation Using Feature Pyramid Module for Robust Speaker Verification of Variable-Duration Utterances

Youngmoon Jung, Seong Min Kye, Yeunju Choi, Myunghun Jung, Hoirin Kim
School of Electrical Engineering, KAIST, Daejeon, South Korea
{dudans, kye9165, wkadldppdy, kss2517, hoirkim}@kaist.ac.kr

Abstract
Currently, the most widely used approach for speaker verification is the deep speaker embedding learning. In this approach, we obtain a speaker embedding vector by pooling single-scale features that are extracted from the last layer of a speaker feature extractor. Multi-scale aggregation (MSA), which utilizes multi-scale features from different layers of the feature extractor, has recently been introduced and shows superior performance for variable-duration utterances. To increase the robustness dealing with utterances of arbitrary duration, this paper improves the MSA by using a feature pyramid module. The module enhances speaker-discriminative information of features from multiple layers via a top-down pathway and lateral connections. We extract speaker embeddings using the enhanced features that contain rich speaker information with different time scales. Experiments on the VoxCeleb dataset show that the proposed module improves previous MSA methods with a smaller number of parameters. It also achieves better performance than state-of-the-art approaches for both short and long utterances.

Index Terms: Speaker verification, deep speaker embedding, multi-scale aggregation, feature pyramid module, short duration

1. Introduction
Speaker verification (SV) is the task of verifying a speaker’s claimed identity based on his or her speech. Depending on the constraint of the transcripts used for enrollment and verification, SV systems fall into two categories, text-dependent SV (TD-SV) and text-independent SV (TI-SV). TD-SV requires the content of input speech to be fixed, while TI-SV operates on unconstrained speech. Before the deep learning era, the combination of i-vector [1] and probabilistic linear discriminant analysis (PLDA) [2] was the dominant approach for SV [3,4]. Although this approach performs well with long utterances, it suffers from performance degradation with short utterances.

Recently, the most widely used SV approach is the deep speaker embedding learning which extracts speaker embeddings directly from a deep-learning-based speaker-discriminative network [5–18]. This approach outperforms i-vector/PLDA approach, especially on short utterances. In deep speaker embedding learning, convolutional neural networks (CNNs) such as time-delay neural network (TDNN) [10–12] or ResNet [5, 8, 9, 13–19] are mostly used as the speaker-discriminative network. Specifically, the network is trained to classify training speakers [9–19] or to separate same-speaker and different-speaker utterance pairs [5, 6]. After training, an utterance-level speaker embedding called deep speaker embedding is obtained by aggregating speaker features extracted from the network. Most of these approaches use a pooling layer for feature aggregation, mapping variable-length speaker features to a fixed-dimensional embedding. There are several pooling methods, such as global average pooling [5, 7, 14], statistics pooling [10, 13], attentive statistics pooling [11], learnable dictionary encoding [9], and spatial pyramid encoding [16].

Meanwhile, all these pooling layers use only single-scale features from the last layer of the feature extractor. To aggregate speaker information from different time scales, multi-scale aggregation (MSA) methods have been proposed recently [12–15]. The MSA aggregates multi-scale features from different layers of a feature extractor to generate a speaker embedding. In [12–15], the authors show the effectiveness of the MSA in dealing with short or long utterances.

In this work, we propose a new MSA method using a feature pyramid module (FPM). A top-down architecture with lateral connections is used to generate feature maps with rich speaker information at all selected layers. Then, we exploit the rich multi-scale features of a ResNet-based feature extractor to extract speaker embeddings. In addition, we present a novel interpretation of the MSA using the theory of [20]. We evaluate our method using various pooling layers for TI-SV on the VoxCeleb dataset [7, 8]. Experimental results show that the performance of MSA is further improved by the FPM with a smaller number of parameters. Besides, the effectiveness of our method is verified on variable-duration test utterances.

2. Relation to prior works
Gao et al. [13] proposed multi-stage aggregation, where ResNet is used as a feature extractor. The output feature maps of stage 2, 3, and 4 (see Table 1) are concatenated along the channel axis. To make feature maps match in time-frequency resolution, the feature map from stage 2 is downsampled by convolution with stride 2, and the feature map from stage 4 is upsampled by bilinear interpolation or transposed convolution. After concatenation, speaker embeddings are generated by statistics pooling.

Seo et al. [14] also utilize feature maps from different stages of ResNet to fuse information at different resolutions. Unlike the method of Gao et al., global average pooling (GAP) is applied to the feature maps, and the pooled feature vectors are concatenated into a long vector. The concatenated vector is fed into fully-connected layers to generate the speaker embedding. Hajavi et al. [15] proposed a similar approach to the study of Seo et al. Their proposed model, UterIdNet, shows significant improvement in speaker recognition with short utterances.

To sum up, the method of Gao et al. fuses feature maps from different stages to form a single feature map, and applies pooling to the fused feature map to generate the speaker embedding. On the other hand, the method of Seo et al. applies pooling to feature maps respectively, and aggregates the resulting vectors to obtain the speaker embedding. In this paper, we take these two methods as baselines. For clarity, we denote the first method as multi-scale feature aggregation (MSEA) and the second one as multi-scale embedding aggregation (MSEA). These approaches are illustrated in Fig. 1. We will show that the pro-
The architecture of the feature extractor based on ResNet-34 [22]. Inside the brackets is the shape of a residual block and outside the brackets is the number of stacked blocks on a stage. The input size is $64 \times T$.

### Table 1

<table>
<thead>
<tr>
<th>Layer name</th>
<th>Output size</th>
<th>ResNet-34</th>
<th>Stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>conv1</td>
<td>$64 \times 3 \times 32$</td>
<td>$7 \times 7$, stride 1</td>
<td>-</td>
</tr>
<tr>
<td>conv2$_i$</td>
<td>$64 \times T \times 32$</td>
<td>$3 \times 3$, stride 2, stride 1</td>
<td>$i$</td>
</tr>
<tr>
<td>conv3$_i$</td>
<td>$32 \times T/2 \times 64$</td>
<td>$3 \times 3$, stride 2, stride 1</td>
<td>$i$</td>
</tr>
<tr>
<td>conv4$_i$</td>
<td>$16 \times T/4 \times 128$</td>
<td>$3 \times 3$, stride 2, stride 1</td>
<td>$i$</td>
</tr>
<tr>
<td>conv5$_i$</td>
<td>$8 \times T/8 \times 256$</td>
<td>$3 \times 3$, stride 2, stride 1</td>
<td>$i$</td>
</tr>
</tbody>
</table>

3. **Proposed approach**

In this section, we introduce the proposed multi-scale aggregation using the feature pyramid module motivated by [21]. In this work, we use ResNet as our feature extractor just as in the baselines [13, 14]. The architecture is described in Table 1.

3.1. **Multi-scale aggregation**

Deep CNNs such as ResNet are usually bottom-up, feed-forward architectures, which use repeated convolutional and subsampling layers to learn sophisticated feature representations. Deep CNNs compute a feature hierarchy layer by layer, and with subsampling layers, the feature hierarchy is inherently multi-scale of pyramidal shape. This in-network feature hierarchy produces feature maps of different time-frequency scales and resolutions, but introduces large semantic gaps caused by different depths. In SV, as the network is trained to discriminate speakers, features of higher layers contain more speaker-discriminative information (higher-level speaker information) but have lower resolutions due to the repeated subsampling.

Deep CNNs are robust to variance in scale and thus facilitate extraction of speaker embeddings from feature maps computed on a single input scale (Fig. 2(a)). However, even with this robustness, using multi-scale features from multiple layers (Fig. 2(b)) improves the performance as discussed in Section 2.

According to the previous works, the MSA improves speaker recognition performance by extracting speaker embeddings from multiple temporal scales [12, 13]. It is also useful for short-duration speaker recognition through an efficiently increased use of information in short utterances [15]. Besides, it passes error signals back to earlier layers, which helps alleviate the vanishing gradient problem [12, 14].

Two types of MSA are described in Fig. 1. For MSFA (Fig. 1(a)), we use the same method in [13]. For MSEA (Fig. 1(b)), $1 \times 1$ convolutions are added before pooling. The embeddings of different stages are concatenated and the output of the following fully-connected (FC) layer is used as the speaker embedding.

3.2. **Feature pyramid module**

In deep CNNs, feature maps of lower layers have less speaker-discriminative information compared to those of higher layers. Intuitively, if we can enhance the speaker discriminability of the low-layer feature maps, the performance of MSA will be improved. Motivated by this intuition, we aim to create multi-scale features that have high-level speaker information at all layers. The feature pyramid module (FPM) is used to achieve this goal. The MSA with FPM is illustrated in Fig. 2(c). The dotted box indicates the building block of FPM that consists of the lateral connection and the top-down pathway, merged by addition.

The detailed architecture is shown in Fig. 3, involving a bottom-up pathway, a top-down pathway, and lateral connections. The bottom-up pathway is the feed-forward computation of the backbone ResNet. It computes a feature hierarchy consisting of feature maps at multiple scales with a scaling step of 2. In each ResNet stage, there are many layers producing feature maps of the same scale (see Table 1). We select the last layer among the multiple layers of a stage, and then use the output of the layer as the output of the stage. It is because the deepest layer contains the strongest features. We denote the output of convi as $C_i$ for $i=2,3,4,5$.

The procedure of the proposed approach is as follows: (1) Using bilinear interpolation or transposed convolution, we upsample lower-resolution feature maps from higher stages by a factor of 2. That is, the top-down pathway hallucinates higher resolution features by upsampling low-resolution feature maps, but with more speaker-discriminative information, from higher stages. (2) The upsampled feature maps are then enhanced with features from the bottom-up pathway via lateral connections.
Table 2: Comparison of Single, MSFA, and MSEA. The softmax loss and GAP are used for all systems. In this paper, Single denotes using only single-scale features (Fig. 2(a)), FPM-B and FPM-TC are the FPMs with bilinear upsampling and transposed convolution upsampling, respectively.

<table>
<thead>
<tr>
<th>Systems</th>
<th>C_{det}^{\text{min}}</th>
<th>EER (%)</th>
<th># Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single</td>
<td>0.423</td>
<td>4.35</td>
<td>5.77M</td>
</tr>
<tr>
<td>MSFA w/o FPM</td>
<td>0.437</td>
<td>4.30</td>
<td>6.20M</td>
</tr>
<tr>
<td>w/ FPM-B</td>
<td>0.398</td>
<td>4.22</td>
<td>5.82M</td>
</tr>
<tr>
<td>w/ FPM-TC</td>
<td>0.408</td>
<td>4.01</td>
<td>5.85M</td>
</tr>
<tr>
<td>MSEA w/o FPM</td>
<td>0.416</td>
<td>4.22</td>
<td>5.98M</td>
</tr>
<tr>
<td>w/ FPM-B</td>
<td>0.403</td>
<td>4.20</td>
<td>5.83M</td>
</tr>
<tr>
<td>w/ FPM-TC</td>
<td>0.411</td>
<td>4.01</td>
<td>5.85M</td>
</tr>
</tbody>
</table>

When evaluating the performance on short utterances, our test recordings are cut into four different durations: 1 s, 2 s, 3 s, and 5 s, which is determined by the energy-based voice activity detection (VAD). If the length of the utterance is less than the given duration, the entire utterance is used.

4.2. Implementation details

The input features are 64-dimensional log Mel-filterbank features with a frame-length of 25 ms, which are mean-normalized over a sliding window of up to 3 s. Neither VAD nor data augmentation is used for training. In training, the input size of the ResNet is $d = 64 \times 300$ for 3 s of speech (i.e., $d = 4$ and $T = 300$ in Fig. 3). In testing, the entire utterance is evaluated at once. The 128-dimensional speaker embeddings are extracted from the network. We report the equal error rate (EER) in % and the minimum detection cost function ($C_{det}^{\text{min}}$) with the same settings as in [8]. Verification trials are scored using cosine distance.

The models are implemented using PyTorch [24] and optimized by stochastic gradient descent with momentum 0.9. The mini-batch size is 64, and the weight decay parameter is 0.0001. We use the same learning rate (LR) schedule as in [9], with the initial LR of 0.1. For LDE, we use the same method as in [16]. Before the LDE layer, a $1 \times 1$ convolution is applied to change the number of channels to 64. After the LDE layer, $L_2$-normalization and an FC layer are added to reduce the dimension to 128. The number of codewords is 64. When applying MSA, both the LDE and FC layers are shared by all stages. On the other hand, the parameters of the SAP layers are not shared.

5. Results

5.1. Performance in different MSA methods

In Table 2 and 3, the models are trained on the VoxCeleb1 dev set. Table 2 compares with (w/) and without (w/o) the FPM in two MSA methods. MSFA w/o FPM and MSEA w/o FPM correspond to the approaches in Gao et al. [13] and Seo et al. [14], respectively. In both MSA approaches, we aggregate feature maps from 3 different stages: $C_0$, $C_4$, and $C_5$ for w/o FPM and $P_i$, $P_{i+1}$, and $P_{i+2}$ for w/ FPM. Note that the output of stage $i$ is $C_{i+1}$ for $i = 1, 2, 3, 4$ because the stage 1 corresponds to conv2 as shown in Table 1. We can see that both MSAs yield better performance than Single, but with more parameters.

As we discussed in Section 2, the MSFA uses upsampling so that the three feature maps have the same spatial size. In this work, bilinear upsampling is applied to $C_5$ since using transposed convolution does not improve the performance of MSFA with additional parameters. On the other hand, for the FPM, both bilinear and transposed convolution upsampling are used in the top-down pathway, which correspond to FPM-B and FPM-TC, respectively. Among the three systems, w/o FPM has the worst performance with the largest number of parameters. By
Table 3: Performance comparison with and without the FPM for three pooling strategies: GAP, SAP [9], and LDE [9]. MSEA with softmax loss is applied for all the systems.

<table>
<thead>
<tr>
<th>Systems</th>
<th>Train set</th>
<th>C_mins</th>
<th>EER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Single</td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/o FPM</td>
<td>0.424</td>
<td>4.22</td>
<td>4.38</td>
</tr>
<tr>
<td>w/ FPM-B</td>
<td>0.403</td>
<td>4.01</td>
<td>4.09</td>
</tr>
<tr>
<td>w/ FPM-TC</td>
<td>0.411</td>
<td>4.08</td>
<td>4.09</td>
</tr>
</tbody>
</table>

adding the FPM, we obtain better performance with fewer parameters. The number of parameters is reduced because the channel dimension of feature maps at selected stages is reduced as we discussed in Section 3.2. FPM-TC has slightly more parameters than FPM-B because the transposed convolutional layer has learnable parameters. FPM-B achieves a relative improvement of 8.92% in C_mins over w/o FPM. FPM-TC shows a relative improvement of 6.74% in EER over w/o FPM.

For MSEA, FPM-B provides the best C_mins (0.403), and FPM-TC obtains the best EER (4.01%). The proposed FPM improves the performance of both MSFA and MSEA, achieving similar performance with a similar number of parameters. Therefore, we only use MSEA in the following experiments. Moreover, the MSEA is more flexible to use various numbers of stages unlike the MSFA which is developed to use only 3 stages.

5.2. Performance in different pooling methods

In Table 3, we compare the performance among the three pooling strategies. For the SAP layer, FPM-B provides the best C_mins (0.393), and FPM-TC obtains the best EER (4.09%). For the LDE layer, we only aggregate features from stage 3 and 4 because using features from stage 2 does not improve the performance. FPM-TC achieves the best performance on both C_mins (0.368) and EER (3.65%). For all the three pooling strategies, the FPM improves the performance of the MSA method. We can see that the FPM is most effective for the LDE layer.

5.3. Comparison with recent methods

In Table 4, we compare the two proposed systems with recently reported 5V systems. Both proposed systems apply the MSEA with FPM-TC, and the combination of A-softmax and ring loss with the same settings as in [16]. When we use the VoxCeleb2 dataset for training, we extract 256-dimensional speaker embeddings. In the first system (Proposed 1), we aggregate features from all the stages using GAP. In the second system (Proposed 2), we aggregate features from stage 2, 3, and 4 using LDE. The proposed systems outperform other baseline systems in terms of both C_mins and EER. Proposed 2 achieves the best performance among all the systems. Using the VoxCeleb1 dataset for training, Proposed 2 obtains a C_mins of 0.350 and an EER of 3.22%. Using the VoxCeleb2 dataset for training, Proposed 2 achieves a C_mins of 0.205 and an EER of 1.98%.

In Table 5 and 6, we evaluate the performance of several systems in 5 s and full-length enrollment conditions, respectively. For each condition, we evaluate the performance on five different test durations: 1 s, 2 s, 3 s, 5 s, and original full-length (full). The average duration of ‘full’ is 6.3 s. All the results are obtained by our own implementation and VoxCeleb2 dataset is used for training. For TDNN, we follow the same architecture as in [11]. For all the baseline systems, we use the same acoustic features and optimization as our proposed systems.

Among baselines, the ResNet-based system using SPE (the advanced version of LDE) outperforms TDNN-based systems except for the 1 s test condition. Similarly, Proposed 2 achieves better results than Proposed 1 for utterances longer than 1 s. From these, we can see that LDE-based pooling shows a greater performance degradation on very short utterances than the other pooling strategies. MSEA w/o FPM is the approach in the 5th row of Table 2. We find that applying the FPM improves the performance of MSA for variable-duration test utterances. When we compare proposed systems with other state-of-the-art baseline systems including TDNN (x-vector) and ResNet-based systems, we also observe that the MSA with FPM achieves higher performance for both short and long utterances.

6. Conclusions

In this study, we proposed a novel MSA method for TI-SV using a FPM. We applied the FPM to two types of MSA methods, MSFA and MSEA. It enhances speaker-discriminative information on multi-scale features at multiple layers of a speaker feature extractor. On the VoxCeleb dataset, experimental results showed that the FPM improves both MSA methods with fewer parameters, and works well with three popular pooling layers. The proposed systems obtained better results for both short and long utterances than the state-of-the-art baseline systems.

7. Acknowledgements

This material is based upon work supported by the Ministry of Trade, Industry and Energy (MOTIE, Korea) under Industrial Technology Innovation Program (No.10063424, Development of distant speech recognition and multi-task dialog processing technologies for in-door conversational robots).
8. References


