Pattern Recognition 88 (2019) 557–568

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Attention guided deep audio-face fusion for efficient speaker naming

Xin Liu[a,*], Jiajia Geng[a], Haibin Ling[b], Yiu-ming Cheung[c]

[a] Department of Computer Science, Huaqiao University, Xiamen 361021, China
[b] Department of Computer and Information Sciences, Temple University, PA 19122, USA
[c] Department of Computer Science and Institute of Research and Continuing Education, Hong Kong Baptist University, Hong Kong SAR, China

ARTICLE INFO

Article history:
Received 21 November 2017
Revised 24 October 2018
Accepted 15 December 2018
Available online 18 December 2018

Keywords:
Speaker naming
Deep audio-face fusion
Common attention model
Factorized bilinear model

ABSTRACT

Speaker naming has recently received considerable attention in identifying the active speaking character in a movie video, and face cue alone is generally insufficient to achieve reliable performance due to its significant appearance variations. In this paper, we treat the speaker naming task as a group of matched audio-face pair finding problems, and present an efficient attention guided deep audio-face fusion approach to detect the active speakers. First, we start with VGG-encoding of face images and extract the Mel-Frequency Cepstrum Coefficients from audio signals. Then, two efficient audio encoding modules, namely two-layer Long Short-Term Memory encoding and two-dimensional convolution encoding, are addressed to discriminate the high-level audio features. Meanwhile, we train an end-to-end audio-face common attention model to discriminate the face attention vector, featuring adaptively to accommodate various face variations. Further, an efficient factorized bilinear model is presented to deeply fuse the paired audio-face features, whereby the joint audio-face representation can be reliably obtained for speaker naming. Extensive experiments highlight the superiority of the proposed approach and show its very competitive performance with the state-of-the-arts.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

Biometric recognition greatly helps identifying, searching and organizing the human identities, and plays an important role in many attractive identification applications. In particular, speaker naming serves as a fundamental identification problem of localizing as well as labeling each visually speaking character in movies, TV series and live shows, and the reliability of such a system is now considered sufficient to support many high-level video analysis systems such as video summarization [1], semantic indexing, media retrieval [2], interaction analysis [3] and so forth. For instance, automatic labeling of the active speaking characters can be directly applied in the generation of meta-data for indexing and fine-grained retrieval of specific scenes in large-scale video datasets. However, as shown in Fig. 1, it remains a challenging task to achieve efficient speaker naming, mainly due to the severely degraded videos (e.g., low-resolution and occlusion) and unconstrained activities (e.g., facial variations, changing pose and varying view points) in real-life movies.

Intuitively, the modality of face, favored for its superiorities including easy to use and non-invasive detection, may probably be the most natural source to detect the speaking character [4]. Along this line, some researchers select to mark the active speakers by detecting their lip motions [5]. Nevertheless, the precise detections of lip movements depend on the high-quality videos, otherwise, the corresponding lip-dynamic states cannot be visually detected in a reliable way [6]. In addition, the captured speakers are not all in completely full-frontal faces, and this brings an additional challenge that we now need to identify an active speaker even when the face is not frontally observable.

As a typical multimedia data, real-life TV series or movie videos often consist of multiple data types for attractive exhibition and display. It has been shown that the selection of multi-modal data could help to alleviate the problems intrinsic to the techniques based on single modality. Since different modalities could characterize the speaker from different views, the integration of multiple sources could provide more information to name the speaker in a reliable way. In the past, some researchers have made various attempts to combine both visual and textual information (i.e., subtitles or script) to boost the character naming performance. However, neither the subtitles nor the script contain the required information to mark the identity of an active speaker in TV videos. That is because the subtitles record what is said, but not by whom, whereas the script records who says what, but not when. In addition, the textual information within the unedited movies may be

* Corresponding author.
E-mail addresses: xliu@hqu.edu.cn (X. Liu), jjgeng@hqu.edu.cn (J. Geng), hbling@temple.edu (H. Ling), ymc@comp.hkbu.edu.hk (Y.-m. Cheung).

https://doi.org/10.1016/j.patcog.2018.12.011
0031-3203/© 2018 Elsevier Ltd. All rights reserved.

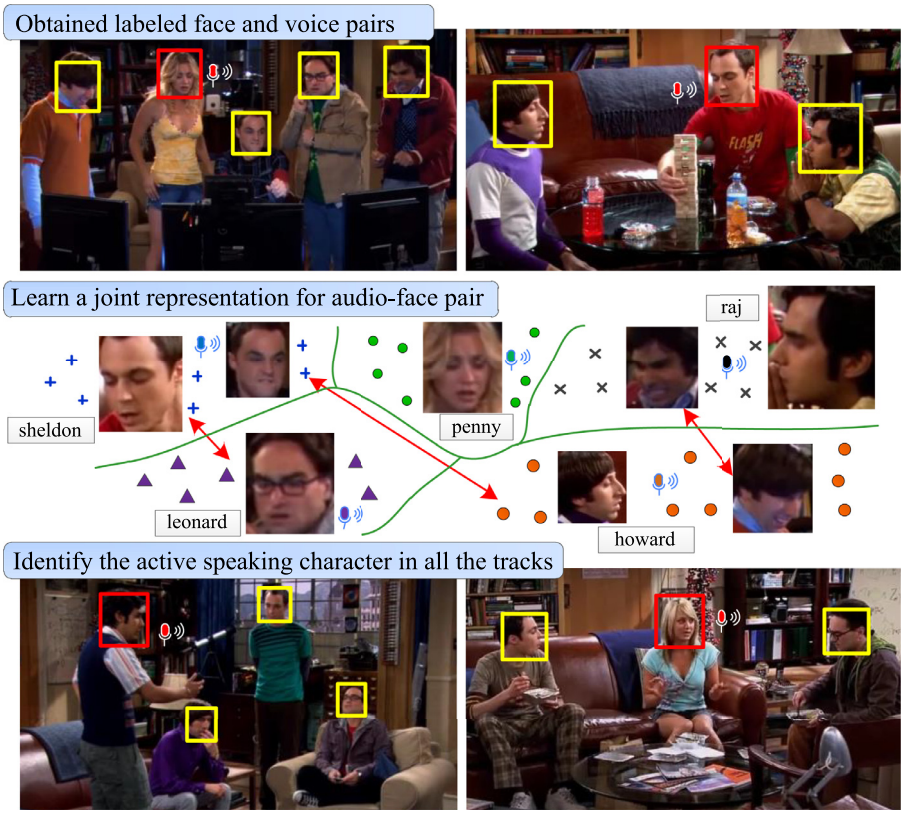

Fig. 1. The main steps of audio-face based speaker naming system. The red bounding box indicates the active speaker, while the yellow bounding box marks the non-speaking actor. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

unavailable to the subscribers, which may therefore limit their application domains in practice.

In a TV show, audio data is closely accompanied within the video during the speaking process, and it is well accepted that the audio cues also provide reliable information for speaker identification [7]. For instance, audio information can be particularly helpful to recognize the speaking character who performs far away from the camera. Importantly, face and audio information can be synchronously acquired and easily accessible. Nevertheless, the appropriate fusion between the audio and face data is still a non-trivial task, and the main difficulties are four-fold: (1) Feature heterogeneity: face and audio samples are captured by different sensors, and their feature types are totally different; (2) Data imbalance: the data sizes between audio and face modalities may be different in the same video clip, and such imbalance brings significant challenges in training process; (3) Lack of correlation mining: face and audio cues are able to characterize the identity of the same speaker, and most existing works often ignore such correlations for reliable fusion; (4) Information loss: most existing audio-face fusion approaches usually process the data of different modalities independently, and such individual mining scheme may result in information loss. In addition, as shown in Fig. 1, the existing audio-face fusion methods may fail to detect the speaker with large facial variations. Therefore, there is still a need to develop an efficient speaker naming approach for high accuracy requirements.

In recent years, there is a rising interest in processing the multimodal data with deep neural networks [8]. Inspired by these applications, we propose to treat the speaker naming task as a group of matched audio-face pair finding problems, and present an efficient attention guided deep audio-face fusion approach to detect the active speakers. The proposed approach improves the state-of-the-art works by providing the following three contributions: (1) A novel two-dimensional convolution scheme is exploited for spatiotemporal audio feature extraction, whereby the discriminative audio features can be well obtained to characterize the active speaker; (2) An end-to-end audio-face common attention architecture is proposed, through which the reliable face attention vector can be adaptively obtained to accommodate the large face variations; (3) A factorized bilinear model is exploited to deeply fuse the paired audio-face features, whereby the joint audio-face representation can be reliably obtained for speaker naming. The experiments have shown its outstanding performance.

The remainder part of this paper is structured as follows: In Section 2, we briefly overview the related works concerning to character naming and active speaker detection. Section 3 elaborates the procedures and implementation details of the proposed framework. In Section 4, we report the experimental results and extensive evaluations. Finally, we draw a conclusion in Section 5.

2. Related works

In the past, automatic naming of an active character was generally considered as the principal actor identification problem [9], and previous character identification works can be roughly categorized into two branches: labeling every character appearance with a unique identity and naming the active speaking actor in the current scene. Although these two topics are a little different, they often share the similar data processings and our work mainly falls into the latter one. This section makes an extensive survey on these two topics.

Character annotation is both an important and challenging problem in multimedia analysis, and this topic is often tackled directly based on the face information [10]. However, character naming via single face modality often suffers from the large variations in pose, illumination and facial expression [11]. Recently,

it has been shown that the utilization of multi-modal data allows better retrieval of the data and makes the automatic labeling of images possible. In photographs, the face of imaged person is likely to appear when his or her name is mentioned in the caption. Along this way, Berg et al. [12] first employed the clustering procedure to build an appearance model for each character, and then automatically labeled the faces in photographs with the names of people obtained from textual captions. Similarly, Ozkan and Duygulu [13] addressed a graph model to find the most similar subset among the set of possible faces, provided that the query name was given. Experimentally, these methods may work well for frontal faces on mug shots under controllable lighting conditions, but which are unlikely to produce meaningful results on TV videos.

In videos, the aligned scripts and subtitles are able to provide supervisory information for character naming, and a pioneer work was proposed in [14]. Within this work, a face was labeled with the name that incorporated the largest temporal overlap with a group of similar face sequences. This work was theoretically sound, but the meaningful evaluations were not reported. Later, Yang and Hauptmann [15] predicted the most likely name for each person by using the multiple text features, while Everingham et al. [16] built a group of exemplar sets for all characters by using aligned transcripts. In addition, Sivic et al. [17] exploited a weak supervision from the aligned subtitle to automatically label the characters. Later, Parkhi et al. [18] formulated the character naming as a multiple-instance learning task and attempted to mark the principal characters in TV videos by using the supervisory information provided by an aligned transcript. Although these approaches are able to annotate most face-tracks in videos, they mainly focus on processing the nearly frontal faces under controllable environments. To adapt more challenging face-tracks, Tapaswi et al. [19] characterized the actor appearance as a Markov Random Field (MRF) and integrated cues from face, speech and clothing in a common framework. Later, they further revisited the problem of matching subtitles with the face-tracks as a joint optimization problem [20]. Similarly, Bojanowski et al. [21] utilized the scripts as weak supervision to learn a joint model for actor and action in movies, whereby the name of each person can be well annotated in a clustering framework. Experimentally, these methods are able to label the character appearances in a TV video, even if the face cannot be fully detected or tracked. Nevertheless, the clothing appearance and human actions are often inconsistent with different views and times, which may degrade their identification performances in changing scenes.

In contrast to label all characters in the scene, active speaker naming aims to mark the face of a speaking character who has the same identity as the ongoing voice in a TV video, and the other characters with no speech signal available (i.e., non-speaking actors) are regarded as the distractors. Intuitively, visual speaker detection can be directly achieved by detecting the face with significant lip motions. Along this way, Everingham et al. [5] first utilized two thresholds to categorize the face sequences into “speaking” and “non-speaking” states, and then combined the subtitle/script alignments to name the speakers. Soon after, Jou et al. [22] first detected the mouth region within affine-aligned faces and then performed a hybrid multi-modal approach to mark the active speaker in broadcast videos, while Bauml et al. [23] tagged the speaking faces by thresholding the nearest neighbor distance of the mouth region to the previous frame. Although these lip-motion based methods are able to mark the speaking characters in the TV videos, the lip-dynamic states can not be reliably detected in non-frontal face sequences or low quality videos [24].

Real-life TV series or movie videos are typical multimedia data, and the audio data is closely synchronized with the video modality during the speaking process. Meanwhile, it has been demonstrated that the audio cues can provide reliable information for speaker identification [25,26]. Inspired by recent success of convolutional neural networks (CNN), Hu et al. [27] proposed a multi-modal CNN framework to fuse the face and audio data for active speaker naming. However, this method simply concatenated the audio-face features, which was found to be sensitive to the large facial variations. Meanwhile, this method did not consider the temporal property within the audio data such that the corresponding speaker naming performance was a bit poor. Later, Ren et al. [28] further fused the audio-face features in time direction by Long Short-Term Memory (LSTM). This approach has shown its outstanding performance in characterizing the temporal dependency across audio and face examples. Nevertheless, the high-level audio-face features fused by such method are not fully compatible with each other. As a result, some distractors, i.e., non-speaking actors, may be mistakenly recognized as the speaking character. Therefore, it is still imperative to develop an efficient speaker naming algorithm from a practical viewpoint.

3. The proposed methodology

In a TV video, the speaker refers to the actor who is speaking, and the goal of our proposed speaker naming approach is to identify the active speaking actor by using the audio and face information. To this end, we treat the speaker naming task as a group of matched audio-face pair finding problems, and propose an efficient attention guided deep audio-face fusion approach to detect the active speaker. As shown in Fig. 2, we start with VGG-encoding of face images and address two audio feature encoding modules, namely two-layer Long Short-Term Memory (LSTM) encoding and two-dimensional convolution encoding, to discriminate audio features. Then, we address a common attention model to discriminate the face attention vector, and further exploit a factorized bilinear model to fuse the paired audio-face features for efficient speaker naming.

3.1. Problem formulation and solution overview

Speaker naming aims to identify and mark the face of a speaking character who has the same identity as the ongoing voice. In the current scene, the other characters with no speech signal available are regarded as the distractors, and these non-speaking actors need not to be identified. For the sake of clarity, let $\mathbf{X}=\{\mathbf{x}_i\}_{i=1}^{c}$ denote the set of detected faces in the scene, and $\mathbf{y}$ represent the ongoing audio signal, the task of speaker naming is inherently a semantic matching problem, which can be achieved by predicting the most likely face by:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}_i \in \mathbf{X}} p(\mathbf{x}_i|\mathbf{y};\theta), \tag{1}$$

where θ is the learning parameter. By characterizing the same identity, one detected face and the ongoing audio component are well matched if they come from the same speaker, and such matched audio-face pair could deliver a higher likelihood value. In contrast to this, if the detected face and the ongoing audio component are captured from different actors, such non-matched audio-face pair generally produces a relatively small likelihood value. Let $l \in \{1, 0\}$ denote the label of audio-face pair (i.e., 1 for the matched pair and 0 for the non-matched pair), we treat the speaker naming task as a group of paired audio-face data finding problems, and predict the most likely face as follows:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}_i \in \mathbf{X}} p(l=1|\mathbf{x}_i, \mathbf{y};\theta) \tag{2}$$

As shown in Fig. 1, the detected actors in the current scene may vary at different time. Therefore, the number of predictive function depends on the detected face examples in the scene. For an

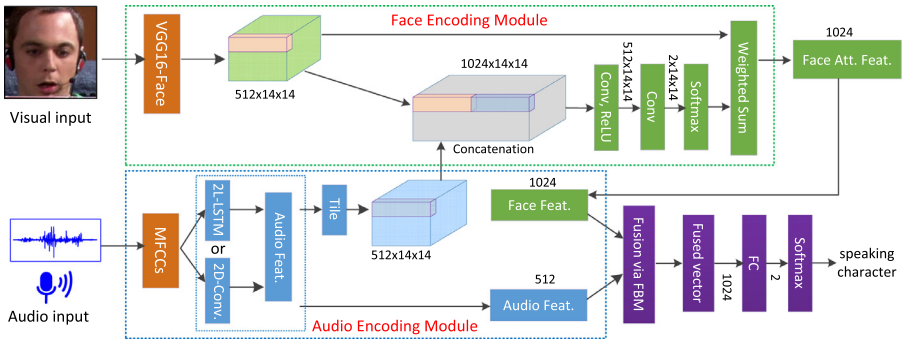

Fig. 2. Schematic pipeline of the proposed speaker naming framework.

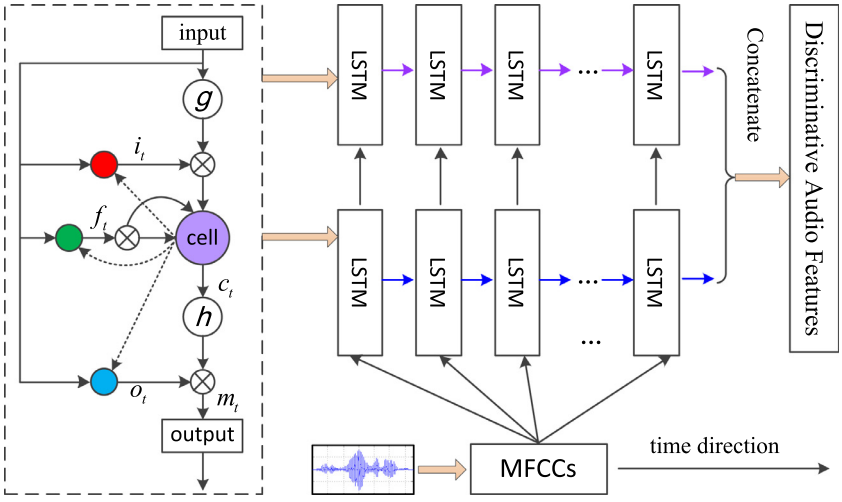

Fig. 3. An audio encoding scheme via two-layer stacked LSTMs.

input audio-face pair ($\mathbf{x}_i$, $\mathbf{y}$), it is necessary to learn a joint representation of these two heterogeneous modalities. With an efficient multi-modal fusion method that encodes the relationship between face cue $\mathbf{x}_i$ and audio example $\mathbf{y}$, it becomes easier to learn a classifier for solving Eq. (2). In the following, we present our speaker naming approach in detail. Specifically, the face features are extracted by the standard VGG-face model [29], and two alternative audio feature encoding methods are proposed in Section 3.2. Subsequently, the proposed audio-face common attention model and deep fusion scheme are carefully stated in tandem.

3.2. Audio feature encoding module

In general, the raw audio features characterized by mel-frequency cepstral coefficients (MFCCs) are not discriminative enough for speaker identification. The main reason lies that the raw MFCCs fail to characterize the temporal properties of audio signals. In recent years, deep neural networks have emerged to be a powerful tool for state-of-the-art speech recognition [30] and speaker identification [31,32]. Inspired by these works, two kinds of audio encoding modules, two-layer LSTMs encoding and two-dimensional convolution encoding, are presented in this section. As suggested in [27], we extract the standard MFCCs from raw audio signals as the input of learning networks.

3.2.1. Audio encoding via two-layer LSTMs

LSTM has been designed to address the gradient vanishing and exploding problems within the conventional Recurrent Neural Networks (RNNs). In essence, LSTM provides a solution by incorporating memory units that allow the network to learn when to forget the previous hidden states and when to update these hidden states. That is, LSTM network computes a mapping from an input to an output at time t by calculating the unit activations iteratively. A standard LSTM model is shown in the left part of Fig. 3, where i_t, f_t, o_t and c_t are respectively the input gate, forget gate, output gate and cell activation vectors, and they are of the same size as the cell output activation vector m_t. g and h are respectively the cell input and cell output activation functions.

In recent years, a stack of multiple LSTM layers and its extensions have been successfully applied to sequential data analysis [33,34]. In deep leaning networks, the low-level layers generally contain much information about the raw data features, while the high-level layers often comprise much information about the semantic features. Inspired by these findings, we utilize a two-layer stacked LSTM (abbreviated as 2L-LSTM) network to discriminate the audio features. A pictorial illustration of the 2L-LSTM network is shown in Fig. 3, where the output from the lower layer becomes the input of the upper layer. Similarly, the dropouts are utilized after each LSTM layer (i.e., dropout ratio is 0.3) [34]. Accordingly, we concatenate the last output of each stacked LSTM layer as the whole audio feature vector for discriminative representation.

3.2.2. Audio encoding via two-dimensional convolution

CNN can be generally regarded as a variant of the standard neural network, and its great success has been demonstrated on many computer vision tasks [35]. Since LSTM often requires a large number of memory cells and output units to store the temporary data information, it is computationally expensive to produce the final result. Differently, the weights of convolutional layers in CNN are shared across different channels, which can significantly reduce the

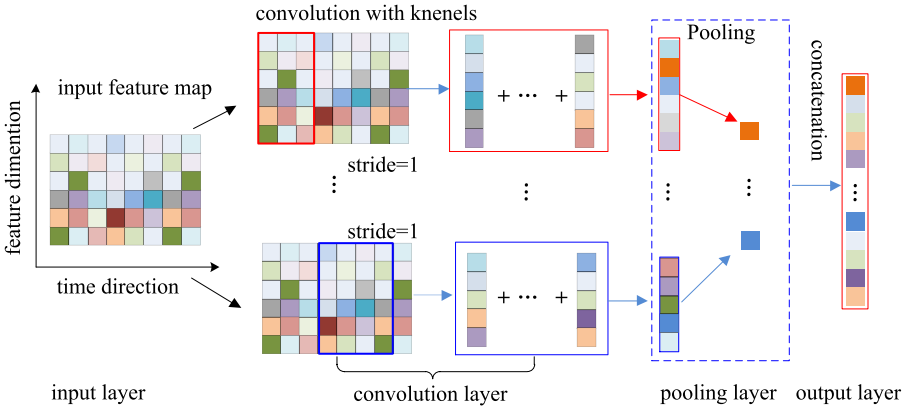

Fig. 4. An audio encoding scheme via 2D-convolutional operations.

layer parameters. It is noted that CNN explicitly characterize the structural locality in feature space and could provide the model with temporal invariance. Accordingly, we propose to encode audio feature in a convolutional way.

Within the CNN based applications, the input data are generally organized as a number of two-dimensional (2D) feature maps [36,37]. Since the audio signal is a time-series data, the raw MFCC features can be easily organized into the pattern (BatchSize, SequenceLength, FeatureDimension). As shown in Fig. 4, we utilize a 2D-convolutional operation (abbreviated as 2D-CONV) to normalize both frequency and temporal variations, and employ a pooling layer to produce the final output. Similar to the data processing in CNN [36], we reshape the raw MFCCs into the pattern: (BatchSize, 1, SequenceLength, FeatureDimension). Often, the convolutional kernels with different sizes can be selected for discriminative feature learning. For example, an input raw MFCC feature is shaped into the size of (1, 1, 49, 75) and the sizes of convolutional kernels are set at (3, 75), (4, 75), (5, 75). Meanwhile, the sizes of kernel filters and convolutional strides are fixed to (1, 1, 1), whereby the outputs of convolutional layers are (47, 1), (46, 1) and (45, 1), respectively. Consequently, we can utilize the pooling operation to get three values and further concatenate them as the final output vector. In practice, we can map the raw audio data into high-dimensional feature space by utilizing different filters, setting different kernel sizes and pooling them into one vector as the final output.

3.3. End-to-end audio-face common attention model

Face feature maps derived from the VGG extractor generally contain much information about the appearances, key points and structural information [29]. However, the face regions always exhibit the large variations caused by the expression, hair style and uncontrollable background clutters, which often make the face representation less discriminative. An attention mechanism allows the model to effectively learn which component is important for the given example [38], and its successful applications include image retrieval [39], object classification [40] and visual question answering [41]. Note that, both of the face and audio data are able to characterize the identity of the same speaker, and these two modalities should share the semantic consistency intrinsically. More specifically, predicting face attention associated with its corresponding audio counterpart allows the model to effectively discriminate the salient location, whereby the semantic consistency between the core face components and its corresponding audio part can be well preserved.

Since the encoded audio feature vector is in one-dimensional representation, it is difficult to fuse it with VGG-face feature map directly. To tackle this problem, as depicted in Fig. 2, we tile the audio feature into the same shape as the face representation in each channel, and the soft attention mechanism is employed to discriminate the common attention vector [42]. For each spatial grid location k, we first concatenate the paired audio-face features into one tensor at each channel direction, and then utilize two convolutional layers to predict the attention weight for each grid location. Given an input attention map $\mathbf{a}_t$ at channel t, we apply the softmax to generate a normalized soft attention map:

$$\tilde{\mathbf{a}}_{t,i} = \frac{\exp(\mathbf{a}_{t,i})}{\sum_{k=1}^{K}\exp(\mathbf{a}_{t,k})} \tag{3}$$

where K is the total number of the spatial grid locations. Once the attention map is obtained, we can take a weighted sum of the input VGG-face features using attention map to discriminate the face attention vector $\mathbf{z}_t$:

$$\mathbf{z}_t = \sum_{k=1}^{K}\tilde{\mathbf{a}}_{t,k}\mathbf{fea}_k \tag{4}$$

where $\mathbf{fea}_k$ represents the VGG-face feature map at spatial grid location k.

In practice, the dimension of raw audio features (i.e., MFCCs) is relative small at each time step, here we set the output size of network at 512 to balance the trade off between parameter number and discrimination power. For each spatial grid location in visual face representation (i.e., last convolutional layer of VGG-face [conv5-3]), we concatenate the slice of the visual face feature with its corresponding audio counterpart. Accordingly, a group of combined feature maps of size $1024 \times 14 \times 14$ are obtained for common attention learning. As depicted in Fig. 2, we employ two convolutional layers to capture the relationships of audio-face pair and predict the common attention weight for each grid location. As discussed in [41], the multiple attention maps are able to enhance the output of attentional image features, and the trained network with two attention maps has shown the best performance. Inspired by this finding, we first utilize one convolutional layer with kernels in size of $512 \times 1 \times 1$ to map the combined features from 1024 to 512, and then employ another convolutional layer to output two attention maps of size $2 \times 14 \times 14$. Consequently, we take a weighted sum of the VGG-face features using these two normalized attention maps and concatenate them to create the face attention vector.

Typical attention maps derived from different audio-face pairs are shown in Fig. 5. For the matched audio-face pair, it can be found that the responses within two attention maps mainly centralize on the locations of core parts in the feature map. Differently, the corresponding responses of non-matched audio-face pair have scattered for a larger part and some responses are far away

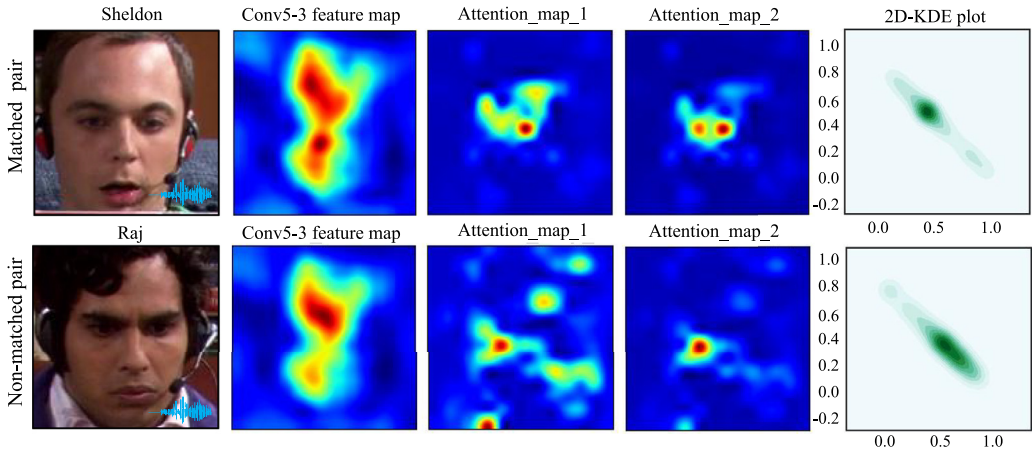

Fig. 5. Examples of attention maps derived from the matched and non-matched audio-face pairs, in which the ongoing audio signals are captured from the actor 'Sheldon'.

from the locations of core components in the feature map, especially for the derived attention map in the first channel (i.e., attention_map_1). Further, we plot 2D kernel density estimation (KDE) on the produced attention maps. As shown in Fig. 5. it can be observed that the attention densities of non-matched audio-face pair have spanned over a larger area, for reason that the selected face example and audio signals are captured from different actors. By contrast, the attention densities of matched audio-face pair (i.e., come from the same speaker) mainly aggregate on an extremely dense area. Since the face features will be weighted sum with those attended responses, the derived face attention vector can be adaptively utilized to accommodate different facial variations.

3.4. Deep audio-face fusion via factorized bilinear model

As introduced in Section 3.1, the fusion of audio-face data plays an important role in speaker naming. In general, concatenation or element-wise summations are most frequently utilized schemes for heterogeneous feature fusion. Since the distributions of audio and face features often vary significantly and their feature dimensions are generally different, the representation capacity of these simple fused schemes may be insufficient for reliable speaker naming performance.

In recent years, fusion by bilinear model is able to capture the inherent interactions between two different modalities more expressively and usually outperforms the simple fusion approaches (e.g., concatenation) [43]. Inspired by such learning architecture, we exploit a factorized bilinear model (FBM) to fuse the paired audio-face features. Without loss of generality, bilinear model considers each feature pair by a linear transformation:

$$\mathbf{z}_i = \mathbf{x}^{\mathrm{T}}\mathbf{W}_i\mathbf{y} + b_i \tag{5}$$

where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$ are the input feature vectors from two different modalities (e.g., high-level features of face and audio), $\mathbf{W}_i \in \mathbb{R}^{n\times m}$ is a weight matrix and b_i is a bias item for the output item of linear model $\mathbf{z}_i$. Although the bilinear model can capture the pairwise interactions between two modalities, it often induces a huge number of parameters that may lead to large computations. To handle this problem, an efficient way is to factorize the projection matrix $\mathbf{W}_i$ into two low-rank matrices: $\mathbf{W}_i = \mathbf{U}_i\mathbf{V}_i^{\mathrm{T}}$, where $\mathbf{U}_i \in \mathbb{R}^{n\times d}$ and $\mathbf{V}_i \in \mathbb{R}^{m\times d}$ impose a restriction on the rank d with constraint $d \leq \min(n, m)$. Accordingly, Eq. (5) can be further rewritten as follows:

$$\mathbf{z}_i = \mathbf{x}^{\mathrm{T}}\mathbf{U}_i\mathbf{V}_i^{\mathrm{T}}\mathbf{y} + b_i \tag{6}$$

In general, the first item in the right part of Eq. (6) can be further transformed with Hadamard product or element-wise multiplication to capture the inherent correlations between two heterogeneous modalities:

$$\mathbf{x}^{\mathrm{T}}\mathbf{U}_i\mathbf{V}_i^{\mathrm{T}}\mathbf{y} = \mathbb{1}^{\mathrm{T}}(\mathbf{U}_i^{\mathrm{T}}\mathbf{x} \circ \mathbf{V}_i^{\mathrm{T}}\mathbf{y}) \tag{7}$$

where $\mathbb{1} \in \mathbb{R}^d$ denotes a column vector of ones, and $\circ$ represents the Hadamard or element-wise product. In order to obtain the output feature vector $\mathbf{z} \in \mathbb{R}^o$, whose elements are $\{\mathbf{z}_i\}$, there still need to learn two three-order tensors: $\mathbf{U} = [\mathbf{U}_1, \ldots, \mathbf{U}_o] \in \mathbb{R}^{n\times d\times o}$ and $\mathbf{V} = [\mathbf{V}_1, \ldots, \mathbf{V}_o] \in \mathbb{R}^{m\times d\times o}$. To reduce the order of the tensors $\mathbf{U}$ and $\mathbf{V}$ by one, we replace $\mathbb{1}$ with linear projection $\mathbf{P} \in \mathbb{R}^{d\times o}$ and Eq. (6) can be converted into the following form:

$$\mathbf{z} = \mathbf{P}^{\mathrm{T}}(\mathbf{U}^{\mathrm{T}}\mathbf{x} \circ \mathbf{V}^{\mathrm{T}}\mathbf{y}) + \mathbf{b} \tag{8}$$

where $\mathbf{b} \in \mathbb{R}^o$ is the bias vector. In general, the application of non-linear activation functions often help to increase the representative capacity of bilinear model. Therefore, the non-linear activation functions are added after each linear mapping, and Eq. (8) can be further rewritten as follows:

$$\mathbf{f} = \sigma(\mathbf{P}^{\mathrm{T}}(\sigma(\mathbf{U}^{\mathrm{T}}\mathbf{x}) \circ \sigma(\mathbf{V}^{\mathrm{T}}\mathbf{y})) + \mathbf{b}) \tag{9}$$

where σ denotes an arbitrary non-linear activation function such as ReLU, sigmoid or tanh. Suppose $\mathbf{x}$ and $\mathbf{y}$ respectively represent the face attention vector and audio feature vector, the values of $\mathbf{x}$ are all larger than 0 while $\mathbf{y}$ is in the range of $[-1, 1]$. To avoid information loss, we utilize different non-linear activation functions to map the heterogeneous values into a finite interval. Since the element-wise multiplication is introduced to capture the correlations between two modalities, the magnitude of the output neurons may vary dramatically. To resist this attack, a non-linear activation function (i.e., ReLU) is further added to regularize the output of network:

$$\mathbf{z} = \mathrm{ReLU}(\mathbf{P}^{\mathrm{T}}(\mathrm{ReLU}(\mathbf{U}^{\mathrm{T}}\mathbf{x}) \circ \tanh(\mathbf{V}^{\mathrm{T}}\mathbf{y})) + \mathbf{b}) \tag{10}$$

During the training process, the fusion parameters of FBM can be updated and optimized through the error back-propagation. Once the paired audio-face features are fed into the fusion model, an output of a softmax function acted on a fully connected layer is selected to compute the final decision values. In general, one face and the input audio data are well matched if they come from the same actor, and such audio-face pair could deliver a higher likelihood value. On the contrary, the non-matched audio-face pair generally produces a relatively small likelihood value, and such detected face and the ongoing audio component are generally collected from different actors. Accordingly, we classify these audio-face pairs into matched or non-matched classes, and the loss function in training process is a standard classification loss. As a result, the active speaking actor can be well detected by predicting the

Table 1
The frame numbers of leading actors in BBT dataset.

	Leading roles (Actors) in BBT dataset				
Subsets	Howard	Lenoard	Penny	Raj	Sheldon
S01E01	1953	13454	6616	1762	14441
S01E02	2588	8889	4473	4294	9660
S01E04	2154	4931	4355	1953	12536
S01E05	2097	7689	4797	1631	9307
S01E06	3602	11303	6023	5785	11871
Train	12394	46266	26264	15425	57815
S01E03	3009	11418	4527	2242	7357

Table 2
The frame numbers of leading actors in Friends dataset.

	Leading roles (Actors) in Friends dataset					
Subsets	Chandler	Joey	Monica	Phoebe	Rachel	Ross
S01E03	2412	925	1857	2309	878	1608
S04E04	3881	5081	3921	4255	4098	3434
S07E07	4542	4589	3708	3531	4693	4626
S10E15	5333	4445	3776	3845	2758	6822
Train	16168	15040	13262	13940	12373	16490
S05E05	6621	3585	6556	2196	4998	5593

most likely speaker for a given face and audio input within the detected actors in the current scene.

4. Experiment

In the experiments, two public available datasets: BigBang Theory (BBT) and Friends [27,28], are selected for evaluation. These two TV series have been proved to be very challenging for multi-modal data analysis, mainly due to various image degradations, low resolutions and high variations on facial appearances [19,23]. For BBT dataset, the audio-face pairs selecting from *S01E01*, *S01E02*, *S01E04*, *S01E05* and *S01E06* series are enrolled as the training set, while those collecting from *S01E03* are taken as testing set. For Friends dataset, the audio-face pairs selecting from *S01E03* (Season 01, Episonde03), *S04E04*, *S07E07* and *S10E15* corpus are served as the training set, while those choosing from *S05E05* are taken as the evaluation set. All the experiments are implemented using Python and conducted on a computer running at an Intel®Core™ i5 3.40 GHz processor with 8 GB memory. In the training, we utilize Adam solver with similar parameter settings as in [41] and d is fixed at 4.

It is noted that the performance of speaker naming depends on the accuracy of the underlying face detector. Since the face regions are already cropped within these two public datasets, we just utilize these processed datasets for fair evaluation. Meanwhile, we refer to [27] and only report the speaker naming performance on the leading roles, including five characters in BBT dataset (i.e., Howard, Leonard, Penny, Raj and Sheldon) and six actors in Friends dataset (i.e., Chandler, Joey, Monica, Phoebe, Rachel and Ross). The statistical frames of these two datasets are shown in Table 1 and Table 2, respectively.

4.1. Data processing

The popular VGG-face model is selected as the face feature extractor [29], and we derive the face feature maps from conv5 − 3 layer as the input of our network model, whose size is $514 \times 14 \times 14$. As introduced in [27], the sequential data including both faces and audio cues over 0.5 s time window can be treated as the same resources that come from one speaking character. In general, the video often comprises of 24 frames per second, and there are 12 consecutive frames within each face sequence at 0.5 s. For audio data processing, we set 0.5 s time window over audio sequence to ensure the usability of the resulting system, because the changes of acoustic features over 0.5 s time window can be regarded as the feature unit. As shown in Fig. 6, a window size of 20 ms and a frame shift of 10 ms are employed to process the raw MFCC audio features. Accordingly, we extract the mean and standard deviation of 25D MFCCs, and also derive the standard deviation of 2 − ΔMFCCs, resulting in a total of 75 features per audio sample.

In a video clip, each face sequence has only 12 time steps, which are inconsistent with the 49 time steps in audio sequences. Meanwhile, the face and audio data may differ a bit with the increasing of age. For instance, the whole Friends TV series are taken over a large time periods across ten years, and the facial appearances of leading roles are varying to some degree. To maximize the diversity of the training set, we randomly sample the face examples of each actor in different seasons. More specifically, we first randomly sample the face examples within the periods of per audio clip (0.5 s window size) and then composite the audio-face pairs. For matched training pairs, we randomly choose the face data of current speaker at all seasons and group these samples with the current audio data for positive training pairs. For the non-matched training pairs, we randomly choose the face examples excluding the current speaker and pair these samples with the input audio data for negative training pairs. It is noted that the label sets in [28] represent the identity of leading actors, while the label sets in our training model denote the matched or non-matched audio-face pairs.

4.2. Evaluation metric

The active speaker can be detected by predicting the most likely speaker within the detected actors in the current scene. Given a speaking clip, there is only one speaker in the scene and the decision can be obtained by Eq. (2). Therefore, we can define the speaker naming accuracy (snA) as follows:

$$\text{snA} = \frac{N_{[p^{sn}==s^{tr}]}}{N_{s^{tr}}} \times 100\% \tag{11}$$

where p^{sn} and s^{tr}, respectively, denote the labels of predicted samples and ground truth, $N_{[p^{sn}==s^{tr}]}$ and $N_{s^{tr}}$, respectively, represent the numbers of correctly named samples and total testing examples.

4.3. Audio encoding performance analysis

In Section 3.2, two different audio encoding modules are addressed to discriminate the high-level audio features. Note that, the lengths of processed audio clips (0.5 s) are significantly less than the ones in traditional speaker identification [31,32]. In general, the short audio clips capturing from the same speaker can be categorized as the same class. To validate the efficiency of the proposed audio encoding schemes, we select the MFCCs as the baseline and perform unsupervised classification on different audio encoding modules. Specifically, support vector machine (SVM) and nearest neighbour (NN) are selected as the classifiers. The classification results tested on BBT (S01E03) and Friends (S05E05) datasets are shown in Table 3, it can be observed that the audio representations encoded by both of 2L-LSTM and 2D-CONV modules have delivered the better classification accuracies than that achieved by the MFCC counterparts. Although the SVM classifier is generally more powerful than NN classifier, the proposed 2L-LSTM and 2D-CONV modules can well discriminate the high-level audio features.

Further, it can be found that the audio features derived from 2D-CONV encoding module have delivered a higher classification

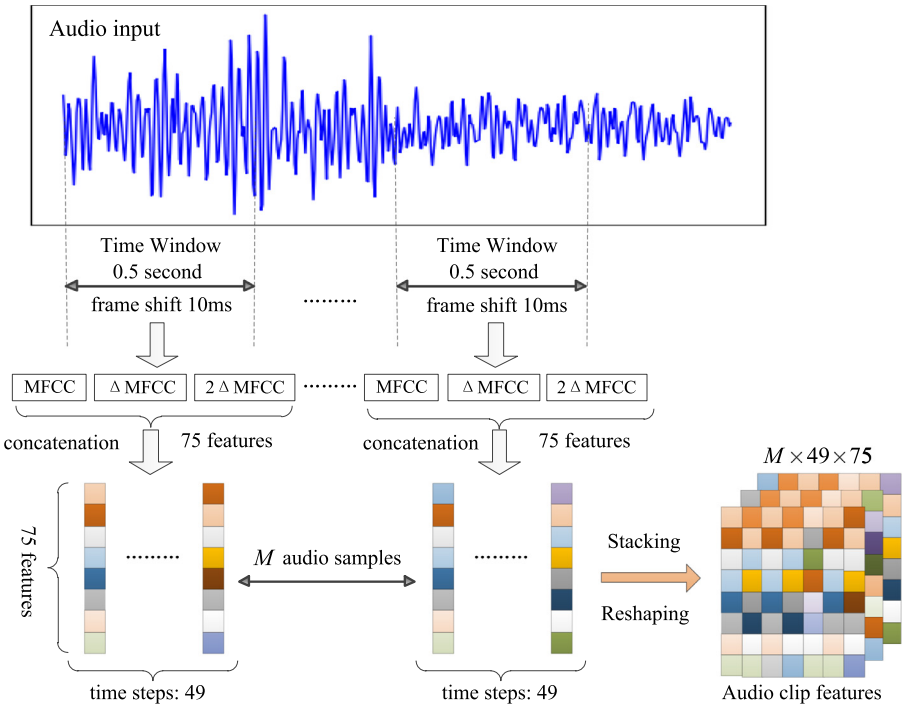

Fig. 6. Audio data precessing procedure, in which a window size of 20 ms and a frame shift of 10ms are employed to generate 75×49 MFCCs within 0.5,s time window.

Table 3
Audio classification obtained by different encoding modules.

Encoding Modules	Classification accuracy	
	BBT (S01E03)	Friends (S05E05)
MFCCs+SVM	0.621	0.634
2L-LSTM+NN	0.743	0.764
2D-CONV+NN	**0.792**	**0.813**

accuracy than the results obtained from 2L-LSTM encoding module. That is, the audio features encoded by 2D-CONV are more semantically interpretable than the features encoded by 2L-LSTM scheme. Essentially, a LSTM layer is composed of recurrently connected memory blocks, each of which contains one or more recurrently connected memory cells. That is, the outputs of each cell consist of the outputs of previous steps and current inputs, and there is no exploitation about the future input features. Accordingly, the audio information may not be well exploited by the last output of 2L-LSTM encoding module. In contrast to this, 2D-CONV encoding module employs different kernels to capture the local relationship among neighboring feature frames, which can access to both past and future features for a given time. For instance, a kernel of size 5×75 can extract the discriminative local features with two steps front and back from the current frame. Therefore, these outputs derived from different convolutional kernels can be well utilized as the inputs of the next pooling layer. As a result, the final outputs not only can preserve much information about the identity, but also contain the temporal information for discriminative analysis.

Moreover, we set the input per-batch size at M=50, and record the running time of these two audio encoding methods. In the experiments, the execution time of per-batch audio processing obtained by 2D-CONV encoding module was significantly less than the results obtained by the 2L-LSTM encoding module, i.e., 2D-CONV only spent 4.767 s to produce the encoding features, while 2L-LSTM cost over 130 s to process per-batch audio data. The main reason lies that the stacked 2L-LSTM network requires a large number of output units and memory cells to store the temporary data information, thereby the learning of related parameters are computationally expensive. In contrast to this, the 2D-convolutional operation needs not to store and process large temporary data, which can significantly reduce the processing time.

4.4. Speaker naming performance analysis

Since the speakers in TV series often exist large facial variations and non-frontal appearances during the speaking process, it is unsuitable to name the active speaker by detecting the facial key points or lip-dynamic states. In the literature, Tapaswi et al. [19] and Bauml et al. [23] have reported all character naming accuracies on BBT dataset, respectively reached up to 77.81% and 80.80% on evaluating the S01E03 subset. The former approach utilized the face and clothing appearance for speaker identification, while the latter method tagged the speaking faces by using subtitles and transcripts in the videos. Since our proposed approach selects the face and audio data to achieve speaker naming, it is very difficult to perform a relatively fair and meaningful comparison with these two approaches appropriately. With the same audio-face data, we carefully compare the proposed approach with Hu et al. [27] and Ren et al. [28] to evaluate the speaker naming performance. Specifically, Hu et al. [27] exploited a multi-modal CNN framework to automatically learn the fusion function between face and audio cues, while Ren et al. [28] presented a multi-modal LSTM model to characterize the long-term dependencies across the audio and face modalities. Since Ren et al. [28] only provided the trained models of the second season on BBT dataset, and we thus mainly focus on comparing our proposed approach with Ren et al. [28] on BBT dataset extensively.

For each detected face in a video frame, we group it with the audio cues of the current speaker, and input them into our network model to predict its probability. Note that, all the detected faces in a video frame are independent, and we choose the actor who produces the maximal probability as the active speaking character. Representative naming examples tested on two datasets

Fig. 7. Representative active speaker naming examples. The green bounding box indicates the active speaker, while the yellow bounding box annotates the distractors. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4
The snA performances of leading actors in BBT dataset.

Actors	Times	Frames	Hu [27]	Ren [28]	Our (2L-LSTM)	Our (2D-CONV)
Howard	31	775	74.97%	86.47%	87.09%	**89.41%**
Lenoard	91	2275	74.51%	86.71%	**87.12%**	86.32%
Penny	15	375	74.93%	87.35%	86.67%	**89.06%**
Raj	5	125	73.33%	84.66%	85.33%	**86.47%**
Sheldon	52	1300	75.78%	86.37%	87.85%	**88.92%**
Average	194	4850	74.93%	86.59%	87.23%	**87.73%**

are shown in Fig. 7, it can be clearly observed that our proposed approach is able to well identify the active speaker in real video examples. In particular, the matched probabilities are very high if the detected faces are of high quality, e.g., frontal appearance or high resolution. For instance, the matching probabilities of 'Sheldon' (i.e., first image in the first row) and 'Chandler' (i.e., first image in the third row) have reached up to 0.961 and 0.982, respectively. Although some matching probabilities are a bit small when the faces are not completely frontal, e.g., 0.780 of 'Monica' (i.e., third image of second row), our proposed approach is also able to detect their speaking states.

Further, the snA values tested on individual actors and different datasets are shown in Tables 4 and 5, respectively. It can be observed that Hu et al. [27] and Ren et al. [28] have delivered a bit lower snA values on BBT dataset, and the speaking actors within a certain part of video frames were mistakenly identified. Similarly, the snA values of leading actors obtained by Hu et al. [27] are all less than 83% when tested on Friends dataset. In contrast to this, our proposed speaker naming approach has yielded an improved performance than those obtained by Hu et al. [27] and Ren et al. [28]. The main reason lies that the methods [27,28] seldom consider the large face variations. It is noted that the face appearances in TV series always accompany with significant changes in scale, pose and expressions, which often make it difficult to characterize the active speaker and correlate the audio components in a reliable way.

Table 5
The snA performances of leading actors in Friends dataset.

Actors	Times	Frames	Hu [27]	Our (2L-LSTM)	Our (2D-CONV)
Chandler	35	875	80.38%	83.11%	**85.38%**
Joey	30	750	80.08%	82.82%	**84.94%**
Monica	35	875	80.91%	83.45%	**85.84%**
Phoebe	12	300	82.48%	83.02%	**85.68%**
Rachel	35	875	80.68%	82.88%	**84.70%**
Ross	42	1050	81.71%	84.31%	**86.58%**
Average	31	4725	80.92%	83.35%	**85.56%**

Compared to compress an entire face image into a static representation, our proposed approach exploits a common attention model to discriminate the face attention vector adaptively. Although the detected TV faces often suffer from the large vari-

Table 6
Ablation results of different learning combinations.

Dataset	Actors	N-Att.+Con.	Att.+Con.	N-Att.+FBM	Att.+FBM
BBT	Howard	85.69%	86.46%	86.16%	**89.41**%
	Lenoard	86.38%	**86.43**%	86.12%	86.32%
	Penny	87.21%	87.03%	87.57%	**89.06**%
	Raj	85.13%	85.48%	86.64%	**86.47**%
	Sheldon	84.04%	86.85%	86.22%	**88.92**%
	Average	85.67%	86.57%	86.28%	**87.73**%
Friends	Chandler	83.56%	**85.51**%	84.67%	85.38%
	Joey	83.34%	84.25%	84.24%	**84.94**%
	Monica	82.40%	85.46%	**86.09**%	85.84%
	Phoebe	84.28%	84.13%	84.25%	**85.68**%
	Rachel	80.36%	84.73%	**85.62**%	84.70%
	Ross	83.75%	85.10%	83.62%	**86.58**%
	Average	82.81%	84.97%	84.78%	**85.56**%

ations in pose and facial expression, the proposed common attention model is able to well discriminate the core face features, thereby the proposed deep audio-face fusion module is capable of detecting the active speaker in a reliable way. For instance, the snA value was reached up to 89.41% (2D-CONV) when the actor 'Howard' was speaking in the testing sequence. The main advantages of our proposed approach are three-fold: (1) The core face features derived from the proposed audio-face common attention model is robust against various facial variations; (2) The fusion by FBM can capture the inherent interactions of paired audio-face features more expressively; (3) The non-matched audio-face pairs are selected as the negative examples to train the network model, whereby some confused examples can be well differentiated. The experimental results have shown its outstanding performance.

4.5. Ablation studies

Differing from the multi-modal architecture in [27], we have carefully considered the attention mechanism and factorized bilinear fusion. Next, we further evaluate the effectiveness of each learning module and validate the performance of different learning combinations. For VGG-face features, we utilize the fully connected layers after the non-linear pooling to obtain 1024 dimensional feature vector. Without attention mechanism, we concatenate the VGG-face feature with 2D-CONV audio feature as the baseline (abbreviated as N-Att.+Con.), and also employ the FBM to fuse the paired audio-face features (abbreviated as N-Att.+FBM). By considering the attention mechanism, we further concatenate the face attention vector and the audio feature vector (abbreviated as Att.+Con.) to perform speaker naming task. Accordingly, we compare our proposed attention guided deep audio-face fusion approach (abbreviated as Att.+FBM) with these different learning combinations.

Table 6 shows the speaker naming performances obtained by different learning combinations. Comparing with non-attention mechanism and concatenation fusion, it can be found that the utilization of attention mechanism or FBM often improves the speaker naming accuracies in most cases. For instance, the average naming accuracies derived from 'Att.+Con.' reach up to 86.57% and 84.97%, respectively tested on BBT and Friends datasets. That is, the attention mechanism often contributes to a better performance than the model without attention. Similarly, the fusion by FMB also improves the speaker naming performance than that obtained by the concatenation fusion. Further, our proposed method (Att.+FBM) generally outperforms these different learning combinations and performs better in naming most active speakers. The average naming accuracies obtained by our method reach up to 87.73% and 85.56%, respectively evaluated on BBT and Friends datasets. That is, the proposed audio-face common attention model is adaptive to facial appearance variations, while the presented deep audio-face fusion scheme is capable of capturing the inherent interactions of audio-face pair more expressively.

Table 7
Overall snA performances with different (d, o) values in FBM module.

Settings	BBT dataset	Friends dataset
FBM (d=2, o=1024)	87.63%	85.43%
FBM (d=4, o=512)	87.71%	85.52%
FBM (d=8, o=256)	87.69%	85.48%
FBM (d=4, o=256)	87.35%	85.43%
FBM (d=4, o=1024)	**87.73**%	**85.56**%
FBM (d=4, o=2048)	87.71%	**85.56**%

It is noted that there are two variables (i.e., d and o) to determine the hyper-parameters in FBM module. Further, we evaluate the performance of FBM with different parameter settings. One the one hand, we fix the $d \times o$ as a constant, i.e., 2048. As shown in Table 7, the number of factors determined by d affects the performance. If the value of d is increased from 2 to 8, the overall performances have gained the improvements of 0.06% and 0.05%, respectively evaluated on the BBT and Friends. Meanwhile, the performance has approached almost saturation when d is equal to 8. This phenomenon can be explained by the fact that a large d involves a large window to sum pool the features, which can be treated as a compressed representation and may loss some information. On the other hand, we fix d at 4, and vary the value of o from 256 to 2048. It is worth noting that the increase of o does not produce further improvements. A possible reason is that the high-dimensional features may be easier to overfit. From the experimental results, it can be found that parameter d and o are insensitive to the overall performance, and the settings of d=4 and o=1024 are suitable for efficient fusion of the paired audio-face features.

4.6. Complexity analysis

In our learning model, we improve the multi-modal architecture in [27] by considering the discriminative audio encoding, attention mechanism and factorized bilinear fusion. Intrinsically, our learning network is not difficult, for reason that the audio-face features are learned independently and their learning parameters are not shared across different modalities. Meanwhile, the computational complexities of attention mechanism and FBM have proven to be acceptable [42,43], and these two modules can be easily trained in an end-to-end manner. More importantly, the presented audio-face common attention model is able to discriminate the core face features, while the fusion by FBM can capture the inherent interactions between heterogeneous audio-face features more expressively. The experimental results have shown that the deeply fused audio-face features often yield outstanding speaker naming performance.

5. Conclusion

This paper has presented an efficient attention guided deep audio-face fusion approach to achieve speaker naming. The proposed approach not only can discriminate the high-level features from both of the face and audio modalities, but also could automatically learn the fusion function to seamlessly fuse the audio-face features. Accordingly, the presented speaker naming framework provides an effective way to distinguish the matched or non-matched audio-face pairs such that the active speakers can be well detected. The extensive experiments have shown that the proposed approach is adaptive to different facial variations and can well name the speaking actors in various challenging TV series.

Further research is warranted along the present lines of work in order to solve several challenging problems. For example, if the au-

dio signals are corrupted by the significant background noise, the proposed common attention model may fail to jointly discriminate the face attention vector. Therefore, it would be necessary to extend the presented algorithm so that it can handle the noise problem adaptively. In addition, questions like how to efficiently handle the crowd talking in outdoor scenes and how to adaptively name the new coming speakers, deserve further attention in future studies.

Acknowledgment

This work was supported by the National Science Foundation of China (Nos. 61673185 and 61672444), National Science Foundation of Fujian Province (No. 2017J01112), Promotion Program for Young and Middle-aged Teacher in Science and Technology Research of Huaqiao University (No. ZQN-PY309), Science and Technology Project of Quanzhou (No. 2018C107R), the National Key Research and Development Plan (No. 2016YFB1001200), US National Science Foundation (Nos. 1814745, 1407156 and 1350521), SZSTI Grant (No. JCYJ20160531194006833) and the Faculty Research Grant of Hong Kong Baptist University (No. FRG2/17-18/082).

References

[1] K. Takenaka, T. Bando, S. Nagasaka, T. Taniguchi, Drive video summarization based on double articulation structure of driving behavior, in: Proceedings of ACM International Conference on Multimedia, 2012, pp. 1169–1172.
[2] H. Zhang, Z.-J. Zha, Y. Yang, S. Yan, Y. Gao, T.-S. Chua, Attribute-augmented semantic hierarchy: towards bridging semantic gap and intention gap in image retrieval, in: Proceedings of ACM International Conference on Multimedia, 2013, pp. 33–42.
[3] C. Liu, S. Jiang, Q. Huang, Naming faces in broadcast news video by image Google, in: Proceedings of ACM International Conference on Multimedia, 2008, pp. 717–720.
[4] S. Satoh, Y. Nakamura, T. Kanade, Name-it: naming and detecting faces in news videos, IEEE Multimed. 6 (1) (1999) 22–35.
[5] M. Everingham, J. Sivic, A. Zisserman, Taking the bite out of automated naming of characters in tv video, Image Vis. Comput. 27 (5) (2009) 545–559.
[6] P. Tiawongsombat, M.-H. Jeong, J.-S. Yun, B.-J. You, S.-R. Oh, Robust visual speakingness detection using bi-level HMM, Pattern Recognit. 45 (2) (2012) 783–793.
[7] A. Noulas, G. Englebienne, B.J. Krose, Multimodal speaker diarization, IEEE Trans. Pattern Anal. Mach. Intell. 34 (1) (2012) 79–93.
[8] X. Xu, Y. Li, G. Wu, J. Luo, Multi-modal deep feature learning for rgb-d object detection, Pattern Recognit. 72 (2017) 300–313.
[9] R. Poppe, Facing scalability: naming faces in an online social network, Pattern Recognit. 45 (6) (2012) 2335–2347.
[10] W. Zhao, R. Chellappa, P.J. Phillips, A. Rosenfeld, Face recognition: a literature survey, ACM Comput. Surv. 35 (4) (2003) 399–458.
[11] W. Deng, J. Hu, Z. Wu, J. Guo, Lighting-aware face frontalization for unconstrained face recognition, Pattern Recognit. 68 (2017) 260–271.
[12] T.L. Berg, A.C. Berg, J. Edwards, M. Maire, R. White, Y.-W. Teh, E. Learned-Miller, D.A. Forsyth, Names and faces in the news, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2004, pp. 848–854.
[13] D. Ozkan, P. Duygulu, A graph based approach for naming faces in news photos, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2006, pp. 1477–1482.
[14] S. Satoh, T. Kanade, Name-it: Association of face and name in video, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 1997, pp. 368–373.
[15] J. Yang, A.G. Hauptmann, Naming every individual in news video monologues, in: Proceedings of ACM International Conference on Multimedia, 2004, pp. 580–587.
[16] M. Everingham, J. Sivic, A. Zisserman, Hello! my name is... buffy–automatic naming of characters in tv video, in: Proceedings of British Machine Vision Conference, 2006, pp. 899–908.
[17] J. Sivic, M. Everingham, A. Zisserman, "who are you?"-learning person specific classifiers from video, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2009, pp. 1145–1152.
[18] O.M. Parkhi, E. Rahtu, A. Zisserman, Its in the bag: Stronger supervision for automated face labelling, in: Proceedings of ICCV Workshop: Describing and Understanding Video & The Large Scale Movie Description Challenge, 2015, pp. 1–9.
[19] M. Tapaswi, M. Bäuml, R. Stiefelhagen, "knock! knock! who is it?" probabilistic person identification in tv-series, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2012, pp. 2658–2665.
[20] M. Tapaswi, M. Bauml, R. Stiefelhagen, Improved weak labels using contextual cues for person identification in videos, in: Proceeding of IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, 2015, pp. 1–8.
[21] P. Bojanowski, F. Bach, I. Laptev, J. Ponce, C. Schmid, J. Sivic, Finding actors and actions in movies, in: Proceedings of IEEE International Conference on Computer Vision, 2013, pp. 2280–2287.
[22] B. Jou, H. Li, J.G. Ellis, D. Morozoff-Abegauz, S.-F. Chang, Structured exploration of who, what, when, and where in heterogeneous multimedia news sources, in: Proceedings of ACM International Conference on Multimedia, 2013, pp. 357–360.
[23] M. Bauml, M. Tapaswi, R. Stiefelhagen, Semi-supervised learning with constraints for person identification in multimedia data, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2013, pp. 3602–3609.
[24] S.-L. Wang, A.W.-C. Liew, Physiological and behavioral lip biometrics: a comprehensive study of their discriminative power, Pattern Recognit. 45 (9) (2012) 3328–3335.
[25] H. Vajaria, T. Islam, P. Mohanty, S. Sarkar, R. Sankar, R. Kasturi, Evaluation and analysis of a face and voice outdoor multi-biometric system, Pattern Recognit. Lett. 28 (12) (2007) 1572–1580.
[26] N. Le, J.-M. Odobez, Learning multimodal temporal representation for dubbing detection in broadcast media, in: Proceedings of ACM on Multimedia Conference, 2016, pp. 202–206.
[27] Y. Hu, J.S. Ren, J. Dai, C. Yuan, L. Xu, W. Wang, Deep multimodal speaker naming, in: Proceedings of ACM International Conference on Multimedia, 2015, pp. 1107–1110.
[28] J. Ren, Y. Hu, Y.-W. Tai, C. Wang, L. Xu, W. Sun, Q. Yan, Look, listen and Learn multimodal Lstm for speaker identification, in: Proceedings of AAAI Conference on Artificial Intelligence, 2016, pp. 3581–3587.
[29] O.M. Parkhi, A. Vedaldi, A. Zisserman, Deep face recognition, in: Proceedings of British Machine Vision Conference, 2015, pp. 41.1–41.12.
[30] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, et al., Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups, IEEE Signal Process. Mag. 29 (6) (2012) 82–97.
[31] E. Variani, X. Lei, E. McDermott, I.L. Moreno, J. Gonzalez-Dominguez, Deep neural networks for small footprint text-dependent speaker verification, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 2014, pp. 4052–4056.
[32] F. Richardson, D. Reynolds, N. Dehak, Deep neural network approaches to speaker and language recognition, IEEE Signal Process. Lett. 22 (10) (2015) 1671–1675.
[33] A. Graves, N. Jaitly, A.-r. Mohamed, Hybrid speech recognition with deep bidirectional LSTM, in: Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding, 2013, pp. 273–278.
[34] H. Sak, A. Senior, F. Beaufays, Long short-term memory recurrent neural network architectures for large scale acoustic modeling, in: Proceedings of Annual Conference of the International Speech Communication Association, 2014, pp. 338–342.
[35] A. Sharif Razavian, H. Azizpour, J. Sullivan, S. Carlsson, Cnn features off-the-shelf: an astounding baseline for recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014, pp. 806–813.
[36] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[37] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, D. Yu, Convolutional neural networks for speech recognition, IEEE/ACM Trans Audio Speech Lang Process 22 (10) (2014) 1533–1545.
[38] A. Borji, L. Itti, State-of-the-art in visual attention modeling, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 185–207.
[39] G.-H. Liu, J.-Y. Yang, Z. Li, Content-based image retrieval using computational visual attention model, Pattern Recognit. 48 (8) (2015) 2554–2566.
[40] B. Zhao, X. Wu, J. Feng, Q. Peng, S. Yan, Diversified visual attention networks for fine-grained object classification, IEEE Trans. Multimed. 19 (6) (2017) 1245–1256.
[41] Z. Yu, J. Yu, J. Fan, D. Tao, Multi-modal factorized bilinear pooling with co-attention learning for visual question answering, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2017, pp. 1821–1830.
[42] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in: Proceedings of IEEE International Conference on Machine Learning, 2015, pp. 2048–2057.
[43] Y. Gao, O. Beijbom, N. Zhang, T. Darrell, Compact bilinear pooling, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2016, pp. 317–326.

Xin Liu received the Ph.D. degree in computer science from Hong Kong Baptist University, Hong Kong, in 2013. He is currently an Associate Professor at the Department of Computer Science and Technology, Huaqiao University, Xiamen, China, and also a Visiting Scholar in Temple University, PA, USA. His research interests include computer vision, pattern recognition, and machine learning.

Jiajia Geng received the B.S. degree in applied mathematics from Qingdao University of Technology, China, in 2015, and M.S. degree in computer science from Huaqiao University, Xiamen, China, in 2018. His current research interests include pattern recognition, deep learning and computer vision.

Haibin Ling received the B.S. degree in mathematics and the M.S. degree in computer science from Peking University, China, in 1997 and 2000, respectively, and the PhD degree from the University of Maryland, College Park, in Computer Science in 2006. From 2000 to 2001, he was an assistant researcher at Microsoft Research Asia. From 2006 to 2007, he worked as a postdoctoral scientist at the University of California Los Angeles. After that, he joined Siemens Corporate Research as a research scientist. Since fall 2008, he has been with Temple University where he is now an Associate Professor. Dr. Lings research interests include computer vision, augmented reality, medical image analysis, and human computer interaction. He received the Best Student Paper Award at the ACM Symposium on User Interface Software and Technology (UIST) in 2003, and the NSF CAREER Award in 2014. He is an Associate Editor of IEEE Trans. on Pattern Analysis and Machine Intelligence and serves on the editorial board of Pattern Recognition, and served as Area Chairs for CVPR 2014 and CVPR 2016.

Yiu-ming Cheung received the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong, Hong Kong, in 2000. He is currently a Full Professor in the Department of Computer Science, Hong Kong Baptist University, Hong Kong. His research interests include machine learning, pattern recognition, and visual computing. Prof. Cheung is a Senior Member of the Association for Computing Machinery. He is the Founding Chairman of the Computational Intelligence Chapter of the IEEE Hong Kong Section. He is also an Associate Editor of the IEEE Transactions on Neural Networks and Learning Systems, Pattern Recognition, Knowledge and Information systems, and the International Journal of Pattern Recognition and Artificial Intelligence.