Biomedical Signal Processing and Control 47 (2019) 312–323

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

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

Speech emotion recognition using deep 1D & 2D CNN LSTM networks

Jianfeng Zhao[a,b], Xia Mao[a], Lijiang Chen[a,*]

[a] School of Electronics and Information Engineering, Beihang University, 100083, Beijing, China
[b] School of Information Engineering, Inner Mongolia University of Science & Technology, 014010, Baotou, China

ARTICLE INFO

Article history:
Received 12 July 2017
Received in revised form 26 July 2018
Accepted 27 August 2018
Available online 11 September 2018

Keywords:
Speech emotion recognition
CNN LSTM network
Raw audio clips
Log-mel spectrograms

ABSTRACT

We aimed at learning deep emotion features to recognize speech emotion. Two convolutional neural network and long short-term memory (CNN LSTM) networks, one 1D CNN LSTM network and one 2D CNN LSTM network, were constructed to learn local and global emotion-related features from speech and log-mel spectrogram respectively. The two networks have the similar architecture, both consisting of four local feature learning blocks (LFLBs) and one long short-term memory (LSTM) layer. LFLB, which mainly contains one convolutional layer and one max-pooling layer, is built for learning local correlations along with extracting hierarchical correlations. LSTM layer is adopted to learn long-term dependencies from the learned local features. The designed networks, combinations of the convolutional neural network (CNN) and LSTM, can take advantage of the strengths of both networks and overcome the shortcomings of them, and are evaluated on two benchmark databases. The experimental results show that the designed networks achieve excellent performance on the task of recognizing speech emotion, especially the 2D CNN LSTM network outperforms the traditional approaches, Deep Belief Network (DBN) and CNN on the selected databases. The 2D CNN LSTM network achieves recognition accuracies of 95.33% and 95.89% on Berlin EmoDB of speaker-dependent and speaker-independent experiments respectively, which compare favourably to the accuracy of 91.6% and 92.9% obtained by traditional approaches; and also yields recognition accuracies of 89.16% and 52.14% on IEMOCAP database of speaker-dependent and speaker-independent experiments, which are much higher than the accuracy of 73.78% and 40.02% obtained by DBN and CNN.

1. Introduction

Speech emotion recognition has attracted much attention in the last decades. Emotions are specific and intense mental activities, which can be signed outward by many expressive behaviors. Speech, facial expression, body gesture, and brain signals etc., are the cues of the whole-body emotional phenomena [1–3]. Speech is a fast, efficient and essential pathway of human's communication. So, recognizing speech emotion is one of the important research directions in emotion detection and recognition naturally [4,5].

In order to recognize the emotional state of the speaker, distinguishing paralinguistic features which do not depend on the speaker or the lexical content need to be extracted from the speech. In general, there are two types of information in speech: linguistic information, and paralinguistic information. The linguistic information always refers to the context or the meaning of the speech. The paralinguistic information comes to mean the implicit messages such as the emotion contained in the speech [4,6–8].

There are many distinguishing acoustic features usually used into recognizing the speech emotion: continuous features, qualitative features, and spectral features [9–13]. Many features have been investigated to recognize speech emotion. Some researchers weighted the pros and cons of each feature, but no one can identify which category is the best one until now [4,6,14,15].

In order to learn high-level features from emotion utterances and form a hierarchical representation of the speech, many deep learning architectures have been introduced in speech emotion recognition. The classification accuracy of handcrafted features extracted from certain emotion utterances is relatively high, but the extraction of handcrafted features always consumes expensive manual labor and depend on professional knowledge [6,16,17]. The handcrafted features extraction normally overlooks the high-level features, which are derived from lower level features. So, hierarchical learning, also known as deep learning, is introduced to model high-level abstractions of the data.

Speech signal processing has been revolutionized by deep learning. More and more researcher achieved excellent results

* Corresponding author at: School of Electronics and Information Engineering, Beihang University, Mailbox 206, 37 XueYuan Road, Beijing, China.
E-mail address: chenlijiang@buaa.edu.cn (L. Chen).

https://doi.org/10.1016/j.bspc.2018.08.035
1746-8094/

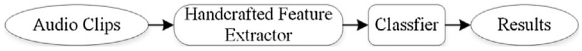


Fig. 1. A general flow chart of traditional speech emotion recognition approach. The handcrafted features are extracted from raw data.
(a) The deep features are extracted from raw data.
(b) The deep features are learned from handcrafted features.

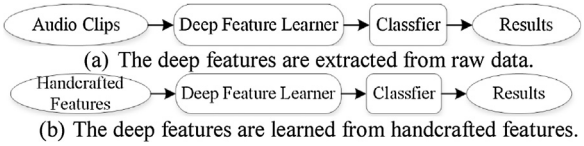


Fig. 2. Two flow charts of the speech emotion recognition approaches adopted in this paper.

in certain applications using deep belief networks (DBNs), convolutional neural networks (CNNs) and long short-term memory (LSTM) [18–20,32]. Deep neural networks are typical "black box" approaches, because it is extremely difficult to understand how the final output is arrived at. There are two models or methods have been introduced to study relevant problems or coincidences. Compared to the "data model" used largely by statisticians, deep networks focus on finding an algorithm to do prediction, so they are called "algorithmic model" [55], [56]. The interpretability of how the highly abstracted features are learned by deep neural networks (DNNs) is poor [57]. But deep neural networks perform dramatically better than traditional approaches (see Fig. 1) in some experiments [21,22].

We constructed two convolutional neural network and long short-term memory (CNN LSTM) networks by stacking four designed local feature learning blocks (LFLBs) and other building layers to extract emotional features. The speech signal is a time-varying signal which needs special processing to reflect time-varying properties. Therefore, LSTM layer is introduced to extract long-term contextual dependencies. The 1D CNN LSTM network is intended to recognize speech emotion from audio clips (see Fig. 2a); the 2D CNN LSTM network mainly focuses on learning global contextual information from the handcrafted features (see Fig. 2b). Most of the traditional features extraction algorithms can reduce data dimension dramatically. The amount of extracted low-level features, such as the spectrum features [23,24], is smaller than that of the raw data. A significant advantage of the learning from a small amount of the low-level features is the decreasing of the training time. The experimental results show that the designed CNN LSTM networks can recognize the speech emotion effectively. Moreover, the designed 2D CNN LSTM network does not only achieve high emotion recognition accuracies but also has better generalization ability. High recognition rate and good generalization ability can provide a guarantee for the application of the designed networks in disease prevention, health care, medical diagnosis, social intercourse etc.

Our original contributions of the work are as follows: 1) a local feature learning block (LFLB), which consists of one convolutional layer, one batch normalization (BN) layer, one exponential linear unit layer, and one max-pooling layer, is designed to extract local features; 2) to learn long-term dependencies from a sequence of local features, LSTM layer is introduced to build CNN LSTM networks following the LFLB; 3) it is proved experimentally that 1D CNN LSTM network can learn lots of emotional features from raw audio utterances for the first time. In our experiments, 2D CNN LSTM network achieves better results. 2D CNN LSTM network focuses on capturing both local correlations and global contextual information from log-mel spectrogram, which is a representation of how the frequency content of a signal changes with time. When log-mel spectrogram is considered as a grid or a sequence, it can be processed by LFLB or LSTM layer.

2. Related work

Distinguishing features are essential for recognizing the speech emotion. Among many paralinguistic features, spectrum features are widely used in speech emotion recognition. AB Kandali et al. presented a method based on MFCCs as features and Gaussian mixture model classifier to recognize emotion from Assamese speeches [25]. Milton, A. et al. used a 3-stage Support Vector Machine classifier to classify seven different emotions present in the Berlin Emotional Database (Berlin EmoDB) [26]. VB Waghmare et al. adopted MFCCs to analyze and recognize speech emotion from artificial emotional Marathi speech database [27]. Demircan, S. et al. used a k-NN algorithm to classify the speech emotion after extracting MFCCs from the audio clips of the Berlin EmoDB [28]. Nalini, N. J. et al. developed a speech emotion recognition system using the residual phase and MFCCs features with the autoassociative neural network (AANN) [29]. Chenchah, Farah et al. used a Hidden Markov Model (HMM) and Support Vector Machine (SVM) to classify the spectral features extracted from audio characteristics of emotional speech [30]. Nalini, N. J. et al. combined the evidence from MFCCs and residual phase (RP) features to recognize emotion in music using AANN, SVM, RBFNN, respectively [31]. Though handcrafted features are very effective to distinguish emotions in speech, most of them are low-level features.

With numerous successful applications of DNNs, more and more researchers began to focus on the learning of deep emotional features. Andre Stuhlsatz and collaborators introduced a generalized discriminant analysis (GerDA) DNNs stacked by several restricted Boltzmann machines (RBMs) to recognize the speech emotion and obtained a highly significant improvement over the previously reported baselines by SVMs [33]. Erik M. Schmidt et al. employed a regression-based deep belief network which was configured with three hidden layers to learn features directly from magnitude spectra and recognize music emotion [34]. Duc Le et al. proposed and evaluated a set of hybrid classifiers based on hidden Markov models and deep belief networks and achieved state-of-the-art results on FAU Aibo [35]. Kun Han et al. proposed to utilize deep neural networks (DNNs) to recognize utterance-level emotions, and obtained 20% relative accuracy improvement compared to the traditional state-of-the-art approaches [17]. Qirong Mao et al. introduced a semi-CNN architecture with a linear SVM to recognize speech emotion and achieved a stable and robust recognition performance in complex scenes [36]. W. Q. Zheng et al. also constructed a CNN architecture to implement emotion recognition on labelled audio data, the preliminary experimental results showed that this approach outperformed the SVM-based classification [21].

Our work differs from the work mentioned above. The designed 1D & 2D CNN LSTM networks learn hierarchical local and global features to recognize speech emotion. Whereas most of the data models can only extract low-level features to classify emotion, and most of the previous DBN-based or CNN-based algorithmic models can only learn one type of emotion-related features to recognize emotion.

3. Methods and materials

Extracting more distinguishing emotion features is one of the main tasks for researchers to recognize speech emotion. According to the difference of feature extraction methods, speech features can be classified as handcrafted features and learned features. Most of the extraction of handcrafted features are carefully designed using ingenious strategies and can be explained in more detail how it

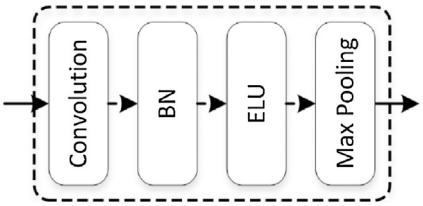


Fig. 3. Block diagram of the local feature learning block. For brevity, batch normalization and exponential linear unit are abbreviated as BN and ELU.
(a) Diagrams of 1D convolution and pooling: the first represents the 1D convolution with a kernel of size 4 and stride 1; the second represents the 1D pooling with a kernel of size 3 and stride 3.
(b) Diagrams of 2D convolution and pooling: the first represents the 2D convolution with a kernel of size 2×2 and stride 1×1; the second represents the 2D pooling with a kernel of size 2×2 and stride 2×2.

works and what it does. The learned features extracted by different deep networks, such RBM based DNN [33,53], CNN [12], perform incredibly well on prediction. Hence, learning deep features to make predictions is becoming more and more popular.

3.1. Deep feature learning

LFLB and LSTM are combined together to learn local features and global features from raw audio clips and log-mel spectrograms respectively in this paper. Convolution layer, the core layer of LFLB, is specialized for processing a grid of values X [39–42]. It can learn a sequence feature where each member of the feature is a function of a small number of neighboring members of the input. Whereas LSTM is specialized for processing a sequence of values X [43], each member of the learned feature is a function of the previous members of the output. The combination of the CNN and LSTM can learn the high-level features, which contain both the local information and the long-term contextual dependencies.

3.1.1. Local feature learning

A local feature learning block (LFLB), a substitute of CNN, is designed to extract emotional features. Each LFLB consists of one convolutional layer, one batch normalization (BN) layer [37], one exponential linear unit (ELU) layer [38], and one max-pooling layer, as illustrated in Fig. 3. Convolution layer and pooling layer are the core layers of the LFLB (see Fig. 4). The most outstanding of the convolution layer is spatially local connectivity and shared weights [39–42]. These properties allow the convolution layer to perform the function of the learning kernel. BN layer normalizes the activations of the convolutional layer at each batch, and improves the performance and stability of deep networks. The transformation applied by batch normalization maintains the mean activation close to 0 and the activation standard deviation close to 1 [37]. ELU layer defines the output of the BN layer. Different to other activation functions, ELU has negative values which push the mean of the activations closer to zero, so it can speed up learning in the designed networks and lead to higher recognition accuracies [38]. Pooling layer can make the features robust against noise and distortion. Max-pooling is the most commonly used non-linear functions. It divides the input into a set of non-overlapping regions and outputs the maximum value of each such sub-region [67].

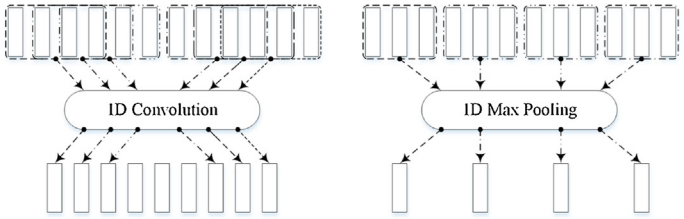
The local feature learning block can be configured differently according to different tasks. The differences in LFLB configuration are mainly reflected in various parameters of the convolution and max-pooling.

The convolution layer plays the role of a local feature extractor. When the data is passed into the convolution layer, it will be convolved with the convolution kernels across the width and height of the input volume. Then a feature map is produced by computing the dot product between the entries of the kernel and the input.

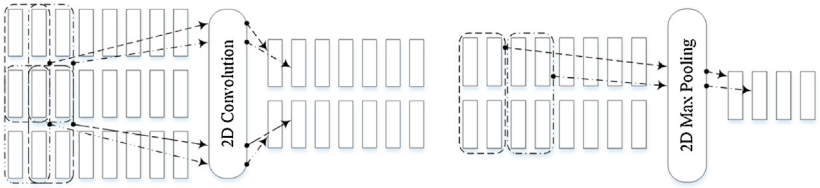
If 1D convolution layer takes as input a signal $x(n)$, the result $z(n)$ can be obtained by convolving the signal $x(n)$ with the convolution kernel $w(n)$ of size l. The 1D convolution kernel $w(n)$ is initialized randomly in our experiments.

$$z(n) = x(n) * w(n) = \sum_{m=-l}^{l} x(m) \cdot w(n-m) \tag{1}$$

While if the input of the 2D convolution layer is $x(i, j)$, the result $z(i, j)$ can be obtained by convolving the signal $x(i, j)$ with the con-



(a) Diagrams of 1D convolution and pooling: the first represents the 1D convc with a kernel of size 4 and stride 1; the second represents the 1D pooling wit kernel of size 3 and stride 3.



(b) Diagrams of 2D convolution and pooling: the first represents the 2D convolution with a kernel of size 2×2 and stride 1×1; the second represents the 2D pooling with a kernel of size 2×2 and stride 2×2.

Fig. 4. Illustration of 1D and 2D convolution and pooling.

volution kernel $w(i,j)$ of size $a \times b$. The 2D convolution kernel $w(i,j)$ is also initialized randomly in our experiments.

$$z(i,j) = x(i,j)^* w(i,j) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} x(s,t) \cdot w(i-s, j-t) \tag{2}$$

Then the convolved features are inputted into the BN layer, which normalizes the activations of the previous layer at each batch. BN layer applies a transformation that maintains the mean of the convolved features close to zero and the variance of the convolved features close to one. When the normalized features are inputted into the ELU layer, the output features can be expressed as

$$z_i^l = \sigma(BN(b_i^l + \sum_j z_j^{l-1} * w_{ij}^l)) \tag{3}$$

Where z_i^l and z_i^{l-1} represent the *i-th* output feature at the *l-th* layer and the *j-th* input feature at the *(l-1)-th* layer; w_{ij}^l denotes convolution kernel between the *i-th* and *j-th* feature.

The function $BN(\cdot)$ normalizes the features learned by the convolution layer. The function $\sigma(\cdot)$ is the ELU activation function of the network, and can be represented as

$$\sigma(x) = \begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases} \tag{4}$$

The extra alpha constant should be a positive number ($\alpha > 0$), e is the Euler's number. Then the features are passed into the max-pooling layer. The pooling layer performs the non-linear down-sampling function and reduces the resolution of the features. The features produced by max-pooling layer can be expressed as

$$z_k^l = \max_{\forall p \in \Omega_k} z_p^l \tag{5}$$

Where Ω_k represents the pooling region with index k, z_k^l and z_p^l represents the output and input feature of the *l-th* max-pooling layer with index k and p.

3.1.2. Global feature learning

LSTM, a recurrent neural network (RNN) architecture, is adopted to learn the long-term contextual dependencies [43,44]. LSTM is explicitly designed for learning long-term dependencies from sequences. So it is stacked upon the LFLB to learn contextual dependencies from the learned local feature sequences. The LSTM can remove or add information to the block state using four components: an input gate, an output gate, a forget gate and a cell with a self-recurrent connection. The updating of an LSTM unit at every timestep t is described by Eqs. (6)–(10) [43,44].

Let z_t^{l-1} and z_t^l be the input and output of an LSTM unit, the relationship between them can be expressed as

$$f_t = \sigma_g(W_f z_t^{l-1} + U_f z_{t-1}^l + b_f) \tag{6}$$

$$i_t = \sigma_g(W_i z_t^{l-1} + U_i z_{t-1}^l + b_i) \tag{7}$$

$$o_t = \sigma_g(W_o z_t^{l-1} + U_o z_{t-1}^l + b_o) \tag{8}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c z_t^{l-1} + U_c z_{t-1}^l + b_c) \tag{9}$$

$$z_t^l = o_t \circ \sigma_z c_t \tag{10}$$

Where c_t represents the LSTM unit state; W, U, and b are parameter matrices and vector; f_t, i_t and o_t are gate vectors; σ_g is a sigmoid function, σ_c and σ_z are hyperbolic tangents; the operator $\circ$ represents the Hadamard product. The superscript $l-1$ and l in Eqs. (6)–(10) are the indices of the input and output features; the subscript i, o, f, c in Eqs. (6)–(8) represent input gate, output gate, forget gate and cell respectively; the subscript g in Eqs. (6)–(8) represent gate.

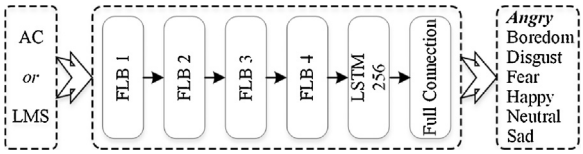


Fig. 5. Block diagram of the overall architecture of the designed 1D and 2D CNN LSTM networks. For brevity, audio clip and log-mel spectrogram are abbreviated as AC and LMS.

Table 1
The layer parameters of the 1D CNN LSTM network. The output dimension is given by length × number. L is the length of the audio clip. The kernel size K of 1F is the number of the emotions. 1C1 and 1P1 are the first convolutional layer and the max-pooling layer of 1LFLB1, and so on.

Name		Output Dim	Kernel Size	Stride
1 LFLB1	1C1	L×64	3	1
	1P1	L/4 × 64	4	4
1 LFLB2	1C2	L/4 × 64	3	1
	1P2	L/16 × 64	4	4
1 LFLB3	1C3	L/16 × 128	3	1
	1P3	L/64 × 128	4	4
1 LFLB4	1C4	L/64 × 128	3	1
	1P4	L/256 × 128	4	4
1 L		–	256	–
1 F		–	K	–

3.2. 1D CNN LSTM network

The CNN LSTM networks are constructed by stacking four LFLBs, one LSTM layer and one fully connected layer. In order to distinguish the same building block or layer, we use the following coding to designate them: 1) the digit before the name indicates that which network this building block or layer is in; 2) the digit after the name is the index of the building block or layer in the networks. The overall architecture of the CNN LSTM networks is illustrated in Fig. 5.

The 1D CNN LSTM network is built by linking four LFLBs (1LFLB1, 1LFLB2, 1LFLB3, 1LFLB4), one LSTM layer (1L), and one fully connected layer (1F). This network is designed to learn deep features from raw audio clips. Therefore, the convolution and pooling kernels in each LFLB are all one-dimensional. The convolution kernel in each LFLB has the same size 3, the same stride 1, and the SAME padding. The number of the convolution kernels in the first and second LFLBs (1LFLB1 and 1LFLB2) is 64, in the third and fourth LFLBs (1LFLB3 and 1LFLB4) is 128. The kernel size and the stride of the max-pooling in each LFLB are 4. The parameters of this architecture are shown in Table 1. The top layer of this architecture is softmax classifier, which is utilized to recognize the emotion according to the learned features.

When audio clip, represented by a one-dimensional vector, is passed into this 1D network, local features are learned by these LFLBs. After been reshaped, these features outputted from 1 LFLB4 are inputted into the LSTM layer (1L). Then the contextual dependencies are learned from these inputted local hierarchical features. The learning of local features and contextual dependencies is shown in Fig. 6. So, the features outputted from the LSTM layer contain local information and long-term contextual dependencies.

Then the learned features are passed to the fully connected layer (1F), which is connected to the 1L layer directly. The fully connected layer can be expressed as

$$z^l = b^l + z^{l-1} \cdot w^l \tag{11}$$

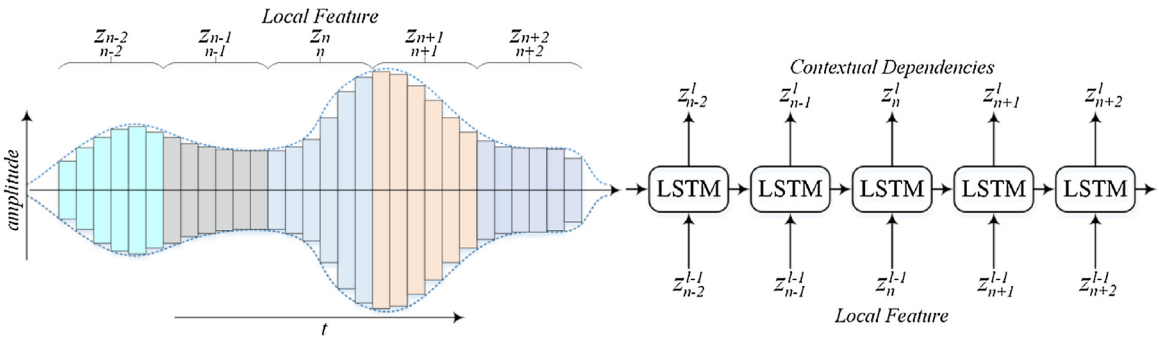


Fig. 6. Illustration of the learning of local features from audio clips by 1D LFLB (left figure, the colours stand for different receptive fields of the 1D LFLB), and the learning of contextual dependencies from local features by LSTM (right figure).

Table 2
The layer parameters of the 2D CNN LSTM network. The output dimension is represented as height × width × number. M × N is the size of the low-level features. The kernel size K of 2 F is the number of the emotions. 2C1 and 2P1 are the convolutional layer and the max-pooling layer of 2 LFLB1, and so on.

Name		Output Dim	Kernel Size	Stride
2 LFLB1	2C1	M × N×64	3 × 3	1 × 1
	2P1	M/2 × N/2 × 64	2 × 2	2 × 2
2 LFLB2	2C2	M/2 × N/2 × 64	3 × 3	1 × 1
	2P2	M/8 × N/8 × 64	4 × 4	4 × 4
2 LFLB3	2C3	M/8 × N/8 × 128	3 × 3	1 × 1
	2P3	M/32 × N/32 × 128	4 × 4	4 × 4
2 LFLB4	2C4	M/32 × N/32 × 128	3 × 3	1 × 1
	2P4	M/128 × N/128 × 128	4 × 4	4 × 4
2 L		–	256	–
2 F		–	K	–

Softmax is the classifier of this architecture, it makes the prediction using the inputted features. Softmax is a generalization of logistic regression to the problem of multi-class classification. The class label y takes on more than two values. Softmax function can be defined to be

$$z_i = \sum_j h_j W_{ji} \tag{12}$$

$$\text{softmax}(z)_i = p_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_i}} \tag{13}$$

Where z_i is the input of softmax, h_j is the activation in the penultimate layer and W_{ji} is the weight connecting between the penultimate layer and softmax layer. Therefore, the predicted class label $\hat{y}$ would be

$$\hat{y} = \underset{i}{arg\max} p_i \tag{14}$$

3.3. 2D CNN LSTM network

The 2D CNN LSTM network has the same structure as the 1D CNN LSTM network. It also has four LFLBs (2LFLB1, 2LFLB2, 2LFLB3, 2LFLB4), one LSTM layer (2L), and one fully connected layer (2F) (see Fig. 5). The convolution and pooling kernels in each LFLB are all two-dimensional. The convolution kernel has the same size 3 × 3, the same stride 1 × 1, and the SAME padding. The number of the convolution kernels in the first and second LFLBs (2LFLB1 and 2LFLB2) is 64, in the third and fourth LFLBs (2LFLB3 and 2LFLB4) is 128. The kernel size and the stride of the max-pooling in the first LFLB are 2 × 2, in other blocks is 4 × 4. The parameters of this network are shown in Table 2. Softmax classifier of this architecture is at the top layer. The implementations of the two models were done with public python deep learning library Keras [52].

This network is constructed to learn high-level emotional features from log-mel spectrograms. When a log-mel spectrogram in the form of a matrix is inputted into the network, local features with local correlations are learned by four LFLBs. The features outputted from 2LFLB4 are reshaped into a form of a temporal sequence and are inputted into the LSTM layer (2L). Then the contextual dependencies are learned from these local features. The learning of local features and contextual dependencies is shown in Fig. 7. So, the features outputted from the LSTM layer contain local correlations and global contextual information.

The fully connected layer (2F) is utilized to generalize from these features into the output-space, and softmax is adopted to make the prediction using the learned features which contain both the local correlations and the global contextual dependencies.

3.4. Hyperparameter optimization

It is very important to choose a set of hyperparameters for a deep architecture. The goal of hyperparameter optimization is to improve the performance of the deep network on an independent data set. Grid search and random search all have been applied to some deep learning frameworks successfully, and simplify the training of the deep architecture. When Bayesian optimization is proposed, it is shown to obtain better results in fewer experiments [46–49]. In our experiments, the Bayesian optimization method is adopted to select the hyperparameters for the proposed deep networks.

Bayesian optimization is a sequential design strategy and can minimize the objective function efficiently. Hyperopt, a Python library, is used to optimize the hyperparameters in our experiments [49]. Hyperopt defines an objective function which can be minimized, and treats it as a random function. A prior is also placed over the objective function. According to the gathered function evaluations, prior is updated to form the posterior distribution over the objective function. An acquisition function is created by using the posterior distribution. Then the hyperparameters are picked iteratively. In order to select a suitable optimization algorithm, the distribution over the choice (*'adagrad'*, *'adam'*, *'sgd'*, *'rmsprop'*) is chosen. After the training with the optimized hyperparameters, the best model is returned.

3.5. Databases and data pre-processing

The two designed CNN LSTM networks were evaluated on two public emotional speech datasets, Berlin EmoDB and Interactive Emotional Dyadic Motion Capture (IEMOCAP) database. The two selected databases are all acted emotional speech databases. The invited actors expressed the pre-determined sentences with required emotions.

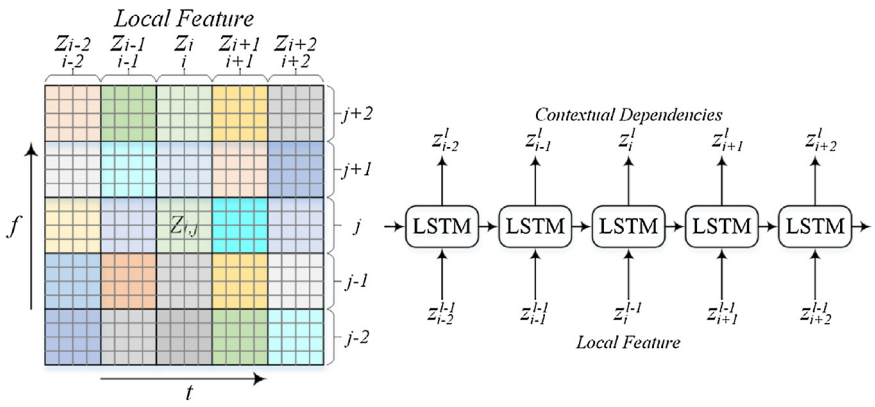


Fig. 7. Illustration of the learning of local features from log-mel spectrogram by 2D LFLB (left figure, the colours stand for different receptive fields of the 2D LFLB), and the learning of contextual dependencies from local features by LSTM (right figure).

The Berlin EmoDB recorded in 2005 contains seven emotions, and each emotion comprises nearly the same number of utterances to evaluate the classification accuracy properly. It provides labeled audio clips and some analysis results. Ten professional actors spoke these acted emotion utterances in an angry, boredom, disgust, fear, happy, neutral and sadness way. There are 535 sentences of the utterances, which come from everyday communication and can be interpretable in all applied emotions [50].

The IEMOCAP contains both motion capture markers and audio data from five pairs of actors (male-female). Two emotion elicitation methods, performances of theatrical scripts and improvisations of affective scenarios are used in the dialog recording. At least three evaluators labeled the emotion label of the data. There are 1150 utterances in the prototypical data (complete agreement on the affective state from evaluators) of improvisations of affective scenarios (Angry: 71, Excited: 178, Frustrated: 271, Happy: 31, Neutral: 328, Sad: 265, and Surprise: 6 utterances). We only use the utterances with labels from the following emotion: Angry, Excited, Frustrated, Happy, Neutral, and Sad [51].

All of the audio clips of the two selected databases are adopted to recognize the emotion and extract log-mel spectrogram. The sample rate of the audio clips used is 16 kHz. The length of the raw audio clips used is 8 s long. If the audio clip is longer than 8 s, it will be segmented to 8 s long. Otherwise, it is padded to 8 s long. At 16 kHz sampling rate, the audio clip can be represented as a 128000-bit vector. So, the input of 1D CNN LSTM network is the 128000-bit vectors in our experiments.

Log-mel spectrogram has been shown to be effective distinguishing features in emotion recognition. It is a representation of the short-term power spectrum of an audio clip. In the processing of computing log-mel spectrogram, the FFT window length is 2048 and the hop length is 512. Thus, the log-mel spectrogram with 251 frames and 128 mel frequency bins is calculated [54]. The log-mel spectrogram can be considered as a grid or a sequence (see Fig. 8). The input of the 2D CNN LSTM network is the 128×251 matrices in our experiments. So, 2D CNN LSTM network can learn high-level features from the 2D image-like patches.

4. Experimental results

Speaker-dependent and speaker-independent experiments are conducted on Berlin EmoDB and IEMOCAP database. Each experiment consists of two parts, the first one is conducted on raw audio clips, the second one is conducted on log-mel spectrograms. The 1D CNN LSTM network is utilized to learn the emotional features from raw audio clips, the 2D CNN LSTM network is adopted to learn high-level features from log-mel spectrograms.

Deep networks are generally considered to be "black box" approaches, how the networks achieve the goals is obscure. Therefore, deep networks are always used to find an algorithm to do prediction. In our experiments, the designed CNN LSTM networks are also used for the predictive power rather than the weak explanatory power.

Several techniques are introduced to lessen the chance of or amount of overfitting in experiments. Overfitting is one of the reasons for poor predictions for untrained sample data. When overfitting occurs, the overfitted model will memorize training data rather than learn to predict better. There are many reasons for the occurrence of overfitting. If a deep network is very complex, overfitting will occur. If a deep network is overtrained, overfitting will also arise. When model degrees of freedom adopted in network training is excessive, a condition of overfitting will exist [66]. So, regularization [59], batch normalization [37], cross-validation [60], early stopping [45], model selection [58] are adopted to overcome overfitting.

4.1. Speaker-dependent experiments

We conducted extensive speaker-dependent experiments on all the labeled audio clips and log-mel spectrograms of the selected databases firstly. In each experiment, the experimental data was split into two sets randomly, the training set took 80% of the data, and the testing set took the remaining 20% of the data. The similar results of the experiments show that the designed 1D and 2D CNN LSTM networks are reliable to recognize the speech emotion.

The goal of our work is to recognize speech emotion with high generalization performance and high accuracy. So only the best predictive and fitted models are recorded in experiments. The validation accuracy is a key indicator of the generalization of the trained model. When validation accuracy reaches its maximum during the training of the 1D and 2D CNN LSTM networks, the best predictive model will be there. Therefore the recorded model does not only fit the experimental data well but also have the superior predictive performance to recognize speech emotion.

The average accuracies reported in this paper are not the highest accuracies. There will be lots of models in the training of the designed networks. In order to reduce overfitting, only the best predictive and fitted models are selected. The average accuracies and the validation accuracies reported in this paper are produced by the selected models. The experimental results performed on Berlin EmoDB are illustrated in Table 3, and the experimental results conducted on IEMOCAP database are shown in Table 4.

In experiments, the best predictive and fitted models are recorded. When validation accuracy stops increasing during training, the model will have the superior predictive performance (see

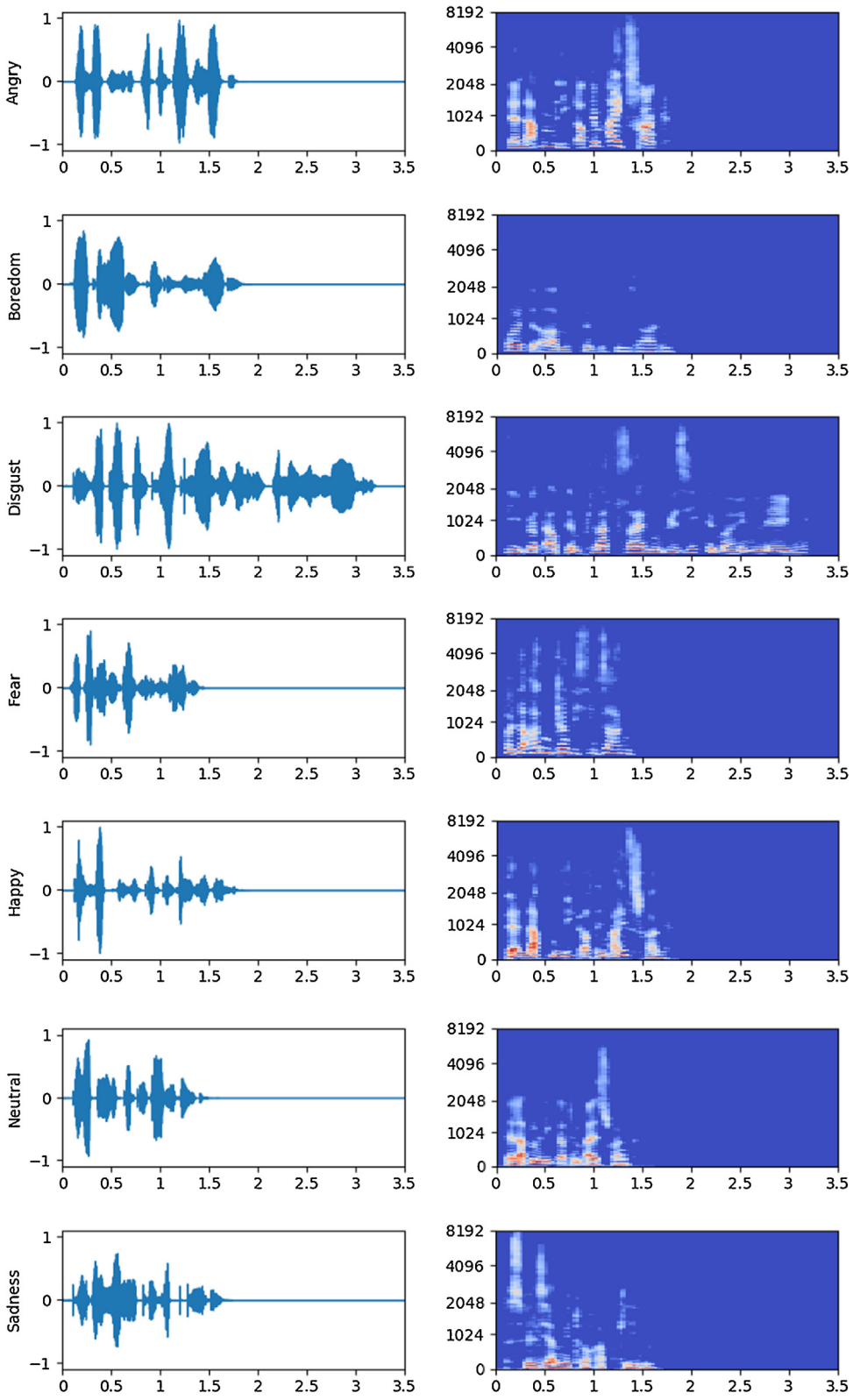


Fig. 8. Waveforms and Log-mel spectrograms of the 03a01Wa.wav (Angry), 03a04Lc.wav (Boredom), 03b10Ec.wav (Disgust), 03a04Ad.wav (Fear), 03a01 Fa.wav (Happy), 03a01Nc.wav (Neutral), and 03a02Ta.wav (Sadness) in Berlin EmoDB.

Fig. 9). From the figure, we can see that when the validation accuracy reaches its maximum, the training accuracy does not reach its maximum. When the validation accuracy decreases while the training accuracy steadily increases, a situation of overfitting has occurred. So, the training will be stopped by early stopping.

Early stopping can avoid overtraining and improve the model's predictive performance. If a monitor has stopped improving, the training of the model will stop. The monitor can be training accuracy, validation accuracy etc. The patience is the number of epochs with no improvement of the monitor. To get a better predictive

Table 3
The confusion matrixes of speaker-dependent experiments on Berlin EmoDB (Ang = Anger, Bor = Boredom, Dis = Disgust, Fea = Fear, Hap = Happiness, Neu = Neutral, Sad = Sadness).

Confusion matrix based on audio clips (accuracy %)

	Ang	Bor	Dis	Fea	Hap	Neu	Sad
Ang	**95.28**	0	0	0.79	3.15	0	0.79
Bor	0	**87.65**	0	1.23	1.23	9.88	0
Dis	4.35	4.35	**82.61**	4.35	2.17	2.17	0
Fea	1.45	0	1.45	**92.75**	2.9	1.45	0
Hap	8.45	0	4.23	0	**87.32**	0	0
Neu	0	0	0	0	0	**100**	0
Sad	0	1.61	1.61	0	0	1.61	**95.16**

Average accuracy = 92.34

Confusion matrix based on log-mel spectrograms (accuracy %)

	Ang	Bor	Dis	Fea	Hap	Neu	Sad
Ang	**99.21**	0	0	0	0.79	0	0
Bor	0	**95.06**	0	0	0	3.7	1.23
Dis	2.17	2.17	**95.65**	0	0	0	0
Fea	1.45	0	1.45	**92.75**	1.45	2.9	0
Hap	5.63	0	2.82	0	**88.73**	2.82	0
Neu	0	5.06	0	0	0	**93.67**	1.27
Sad	0	0	0	0	0	0	**100**

Average accuracy = 95.33

Note: The highest emotion predictions are indicated in boldface.

Table 4
The confusion matrixes of speaker-dependent experiments on IEMOCAP database (Ang = Angry, Exc = Excited, Fru = Frustrated, Hap = Happy, Neu = Neutral, Sad = Sadness).

Confusion matrix based on audio clips (accuracy %)

	Ang	Exc	Fru	Hap	Neu	Sad
Ang	**40.85**	40.85	4.23	1.41	12.68	0
Exc	0.56	**84.83**	2.25	0	10.11	2.25
Fru	0.74	22.88	**46.86**	0	23.99	5.54
Hap	0	**29.03**	9.68	25.81	**29.03**	6.45
Neu	0.3	12.2	0.61	0	**70.12**	16.77
Sad	0	1.51	0.38	0	10.57	**87.55**

Average accuracy = 67.92

Confusion matrix based on log-mel spectrograms (accuracy %)

	Ang	Exc	Fru	Hap	Neu	Sad
Ang	**90.14**	5.63	2.82	0	1.41	0
Exc	2.25	**89.89**	2.25	0	5.06	0.56
Fru	3.32	5.17	**83.03**	0	7.75	0.74
Hap	3.23	25.81	0	**51.61**	16.13	3.23
Neu	0	4.27	3.05	0	**90.24**	2.44
Sad	0	0	0.75	0	1.51	**97.74**

Average accuracy = 89.16

Note: The highest emotion predictions are indicated in boldface.

performance, validation accuracy is monitored in our experiments, patience is set to eight. When validation accuracy is no longer increasing in training, the network will have superior predictive performance.

4.2. Speaker-independent experiments

Extensive speaker-independent experiments are conducted using the same approach as the speaker-dependent experiments. But the division of the data sets is not the same as the speaker-dependent experiments. In experiments, the data are divided into two sets according to the subjects. Because the emotional utterances of the selected databases are all performed by ten speakers, the data of eight subjects are chosen as the training set, the data of other two subjects are chosen as the testing set.

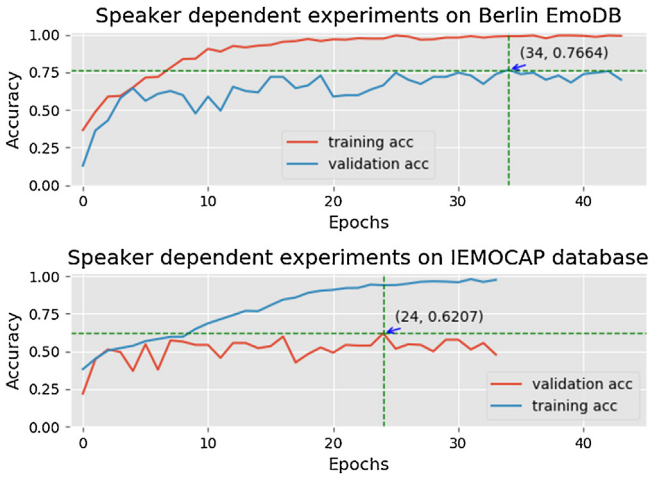


Fig. 9. The 2D CNN LSTM network's training accuracy and validation accuracy on Berlin EmoDB and IEMOCAP database in speaker-dependent experiments per epoch. The annotation shown in parenthesis is the number of epochs that yields the best results and the highest accuracy on the validation set. The annotated validation accuracies are shown in Table 7(b).

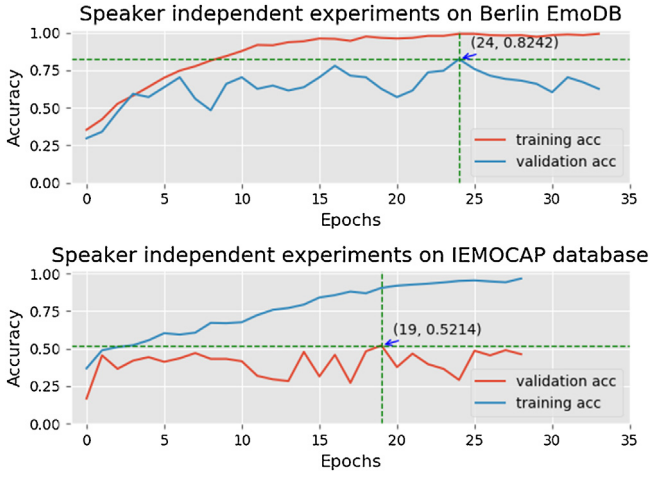


Fig. 10. The 2D CNN LSTM networks training accuracy and validation accuracy on Berlin EmoDB and IEMOCAP database in speaker-independent experiments per epoch. The annotation shown in parenthesis is the number of epochs that yields the best results and the highest accuracy on the validation set. The annotated validation accuracies are shown in Table 7(b).

The best predictive and fitted models are also recorded when validation accuracy reaches its maximum during training (see Fig. 10). The recorded model fits the experimental data well, and has better predictive performance. The confusion matrix performed on Berlin EmoDB is illustrated in Table 5, and the confusion matrix conducted on IEMOCAP database is shown in Table 6.

From the Tables 3–6, we can see that the emotions of Berlin EmoDB are recognized with high recognition accuracies, but most of the happy speeches of IEMOCAP database are misrecognized as neutral, excited etc. We listened to some of these misrecognized happy speeches. The emotion of many happy speeches could hardly be recognized without the help of linguistic information. This is consistent with our practical experience. But the happy emotion of Berlin EmoDB is recognized correctly by the designed networks. This is probably due to the difference of the culture, environment and education of speakers. So, the same speech emotion of different speakers from different cultures has not the same probability of being recognized or misrecognized.

Table 5
The confusion matrixes of speaker-independent experiments on Berlin EmoDB (Ang = Anger, Bor = Boredom, Dis = Disgust, Fea = Fear, Hap = Happiness, Neu = Neutral, Sad = Sadness).

Confusion matrix based on audio clips (accuracy %)

	Ang	Bor	Dis	Fea	Hap	Neu	Sad
Ang	**92.91**	0	0	3.94	3.15	0	0
Bor	0	**98.77**	1.23	0	0	0	0
Dis	8.7	0	**76.09**	10.87	2.17	0	2.17
Fea	1.45	0	1.45	**94.2**	0	2.9	0
Hap	19.72	0	1.41	8.45	**69.01**	1.41	0
Neu	0	15.19	0	3.8	1.27	**78.48**	1.27
Sad	0	9.68	0	0	0	1.61	**88.71**

Average accuracy = 86.73

Confusion matrix based on log-mel spectrograms (accuracy %)

	Ang	Bor	Dis	Fea	Hap	Neu	Sad
Ang	**100**	0	0	0	0	0	0
Bor	0	**97.53**	0	0	0	2.47	0
Dis	6.52	0	**86.96**	4.35	2.17	0	0
Fea	2.9	0	0	**97.1**	0	0	0
Hap	0	0	0	8.45	**91.55**	0	0
Neu	0	1.27	0	5.06	0	**93.67**	0
Sad	0	0	0	0	0	1.61	**98.39**

Average accuracy = 95.89

Note: The highest emotion predictions are indicated in boldface.

Table 6
The confusion matrixes of speaker-independent experiments on IEMOCAP database (Ang = Angry, Exc = Excited, Fru = Frustrated, Hap = Happy, Neu = Neutral, Sad = Sadness).

Confusion matrix based on audio clips (accuracy %)

	Ang	Exc	Fru	Hap	Neu	Sad
Ang	**90.14**	2.82	1.41	2.82	2.82	0
Exc	1.69	**83.71**	3.37	0	9.55	1.69
Fru	1.11	6.27	**78.6**	0.74	9.23	4.06
Hap	3.23	9.68	12.9	25.81	**41.94**	6.45
Neu	0.3	6.1	4.57	0.3	**68.29**	20.43
Sad	0	0	1.13	0	3.02	**95.85**

Average accuracy = 79.72

Confusion matrix based on log-mel spectrograms (accuracy %)

	Ang	Exc	Fru	Hap	Neu	Sad
Ang	**84.51**	15.49	0	0	0	0
Exc	1.12	**88.2**	3.93	0	6.74	0
Fru	1.85	6.27	**75.65**	0	15.5	0.74
Hap	3.23	19.35	16.13	16.13	**41.94**	3.23
Neu	0.61	5.18	3.35	0	**89.94**	0.91
Sad	0	0.38	0.75	0	1.89	**96.98**

Average accuracy = 85.58

Note: The highest emotion predictions are indicated in boldface.

4.3. Recognition accuracy comparison

The comparison of the experimental results of the two designed CNN LSTM networks is listed in Table 7. From this table, we can see that the 1D CNN LSTM network also can learn emotional features from raw audio clips to recognize speech emotion. Moreover, when compared with the 1D CNN LSTM network, the 2D CNN LSTM network shows a certain advantage in overall performance. The average recognition accuracies and the validation accuracies achieved by learning deep features from log-mel spectrograms are higher than that from raw audio clips. From Figs. 9 and 10, we can also see that the 2D CNN LSTM network achieves the highest validation accuracies with fewer epochs than the 1D CNN LSTM network. In other words, 2D CNN LSTM network converges faster compared to 1D CNN LSTM network.

Table 7
The comparison of average and validation accuracy of the designed 1D and 2D CNN LSTM networks. The best performances are indicated in boldface.

(a) average accuracy

Datasets	Audio Clip	Log-mel Spectrogram
Berlin EmoDB[a]	92.34	**95.33**
IEMOCAP database[a]	67.92	**89.16**
Berlin EmoDB[b]	86.73	**95.89**
IEMOCAP database[b]	79.72	**85.58**

(b) validation accuracy

Datasets	Audio Clip	Log-mel Spectrogram
Berlin EmoDB[a]	61.68	**76.64**
IEMOCAP database[a]	46.12	**62.07**
Berlin EmoDB[b]	57.14	**82.42**
IEMOCAP database[b]	45.52	**52.14**

[a] Speaker-dependent experiments.
[b] Speaker-independent experiments.

Table 8
The comparison of average recognition accuracy of 2D CNN LSTM network conducted on Berlin EmoDB with other well-established feature representations and methods. The best performances are indicated in boldface.

Research work	Accuracy (speaker-dep)	Accuracy (speaker-indep)
Wu et al. [10]	91.6	85.8
Zhengwei Huang et al. [12]	88.3	85.2
Huang, Yongming, et al. [13]	75.5	–
Semiye Demircan et al. [14]	/	92.9
Our Work	**95.33**	**95.89**

Table 9
The comparison of recognition accuracy of 2D CNN LSTM network conducted on IEMOCAP database with other well-established feature representations and methods. The best performances are indicated in boldface.

Research work	Accuracy (speaker-dep)	Test Accuracy (speaker-indep)
W. Q. Zheng et al. [21]	/	40.02
Yelin Kim et al. [53]	73.78	/
Our Work	**89.16**	**52.14**

When compared with other well-established feature representations and methods on average accuracy, the designed 2D CNN LSTM network also performs satisfactorily. Table 8 shows that the average accuracy of the 2D CNN LSTM network conducted on the log-mel Spectrograms of Berlin EmoDB achieves the highest accuracy. Table 9 indicates that the 2D CNN LSTM network conducted on the log-mel Spectrograms of IEMOCAP database also performs well.

5. Discussion

In this work, 1D & one 2D CNN LSTM networks, which consists of four LFLBs and one LSTM layer, are built to learn both the local and global emotion-related features. Speeches are time-varying signals, need more sophisticated analysis to reflect time-varying properties. The designed networks with the strength of CNN and LSTM are utilized to recognize the speaker's emotional state.

The experiments have accomplished the task of learning more emotional information from the experimental data, but how to deduce the causal relation between acted emotions and audio features is worth investigating deeply. The designed networks learned a lot of causal features from the data about the underlying mechanism, and recognized the emotions with high accuracies in experiments. So, the mechanism was deduced from the data to some extent, rather than an exact form of an assumed algorithm.

The similar prediction performances of the designed networks in extensive experiments show that the designed networks are effective approaches for recognizing speech emotion.

5.1. "Black Box"

Some researchers have begun to look into the "black box" problem to explain in more detail what's happening in such "black box" in recent years. In 2015, researchers at Google developed a deep-learning-based image recognition algorithm to discover which features are used by the program to recognize the objects. In the same year, a research team at the University of Wyoming found how certain images could fool a network by testing deep neural networks. An MIT research group is devoted to giving a deep learning-based breast cancer diagnosis algorithm some ability to explain its reasoning. In 2007, a computer scientist and neuroscientist from Hebrew University of Jerusalem shared his idea called the "information bottleneck" to explain how deep learning works [61]. Researchers at Columbia and Lehigh universities developed a tool, DeepXplore, to debug the neural networks by error-checking the reasoning of the thousands or millions of neurons [62]. Another tool developed at Stanford University, called ReluPlex, used the power of mathematical proofs to verify properties of deep neural networks, especially small networks [63]. These approaches have made much progress in some image-based applications, but they are not the general solutions to solve the "black box" problem [64,65].

We have also done a great deal of work to better understand the designed deep networks which are used to process speech. Many architectures which were constructed by using a different number of layers and a different number of filters at each layer were tested to learn the effect of basic parameters of the designed networks on prediction performance. When the basic parameters of the architectures were experimentally determined, experiments were also conducted on different handcrafted features to detect which were efficient to recognize emotion. These efforts have helped us to reveal more details of the designed deep networks in experiments.

5.2. Over fitting

Regularization was adopted to prevent overfitting by modifying a learning algorithm in our experiments. It always involves imposing some kind of smoothness constraint on a model, and allows to fix the number of parameters in the model or to augment the cost function. In our experiments, regularization implemented by applying penalties on layer parameters and layer activity during optimization. The loss function of the designed deep networks incorporates the penalties. The experiments show that regularization can guarantee the convergence of the designed networks and reduce overfitting.

BN layer was adopted in the designed CNN LSTM networks to combat overfitting. When applying batch normalization, an inputted feature is always in conjunction with other features in each batch. Therefore, each normalized feature produced by BN layer is no longer a deterministic value for each inputted feature. This effect facilitates the generalization of the deep networks, speeds up the training and reduces overfitting in experiments.

Different cross-validations were used to avoid an overtrained model to a particular dataset in speaker-dependent and speaker-independent experiments. The goal of cross-validation is to limit problems like overfitting, by using a dataset to "test" a model in the training phase. In speaker-dependent experiments, experimental data were randomly divided into train and test sets. In speaker-independent experiments, experimental data were divided into two sets according to the subjects. The experiments show that cross-validation is an effective way to avoid overtraining. The networks can learn to generalize better.

Early stopping was also utilized to reduce overfitting in our experiments. The designed networks were trained with an iterative method which can make the model to better fit the training data. Early stopping tends to improve the model's performance on data outside of the training set. Different monitored quantities and patiences will affect the experimental results. A small patience will lead to an undertrained model, and a big patience will produce an overtrained model. The experiments show that early stopping can enable the networks to learn more general features, and have superior predictive performance.

Model selection was used to lessen the chance of overfitting by selecting a model from a set of candidate models. Extensive experiments have been conducted on the selected databases in our experiments. Only the best predictive and fitted models are selected. These models were recorded when the validation accuracy would not increase during training. So, the recognition accuracies produced by the best predictive and fitted models are not the highest accuracies. The selected models not only fit the experimental data well but also have superior predictive performance.

Several effective methods have been used to reduce overfitting in our experiments, but overfitting has not been completely avoided. From Figs. 9 and 10, the training accuracies have always been higher than the validation accuracy. This means the networks learn some random features of the training data which have no causal relation to the target label. So the models memorize more information about the training data, rather than learning to generalize. Of course, there is another possibility that learned random features are also the emotion features which do not exist in the validation data. Hence, recording a speech emotion database with a large number of utterances is very important to promote the development of this field.

6. Conclusion

This paper presents 1D and 2D CNN LSTM networks to recognize speech emotion. The method of how to learn local correlations and global contextual information from raw audio clips and log-mel spectrograms is investigated. LFLB which consists of one convolutional layer, one BN layer, one exponential linear unit layer, and one max-pooling layer is designed to learn local features. When local features learned by LFLBs are reshaped, they are inputted into an LSTM layer. The LSTM layer can learn contextual dependencies from inputted local features. So, the features learned by the designed CNN LSTM networks contain local information and long-term contextual dependencies.

The performances of the two networks were tested on two benchmark databases. The results show that the two designed CNN LSTM networks can learn distinguishing features and model high-level abstractions of the emotional information. The comparison of the experimental results shows that the 2D CNN LSTM network has a certain advantage over 1D CNN LSTM network in overall performance. When compared with other well-established feature representations and methods, 2D CNN LSTM network also has the edge on average accuracy.

Though the deep networks presented in this paper have gotten better performance in speech emotion recognition, there are many aspects still need to be improved. Firstly, how the designed networks recognize the emotion cannot be explained in more detail, meaning the "black box" of these networks have not been uncovered. Some researchers mentioned in Section 5.1 have done a lot of works to open the "black box" of deep learning, but most of them focus on the deep networks applied in image processing.

Speech is different from the image, so how to uncover the "black box" of the deep networks designed for speech processing deserves deeply study. Secondly, it is not the end to acquire higher accuracy in speech emotion recognition. Any new network architectures or new optimization algorithms which can learn more general features or can train a superior predictive model have to be worth investigating. Finally, in order to combine the advantages of different features, developing a method to merge different deep features which are learned by different deep networks is also worth delving into.

Acknowledgements

Part of this work was done when the first author worked in Advanced Analytics Institute (AAI), University of Technology, Sydney as a visiting scholar. Jianfeng Zhao, Xia Mao, and Lijiang Chen's work in this paper was supported in part by the National Natural Science Foundation of China under Grant No. 61603013. This article recently received funding from the Fundamental Research Funds for the Central Universities (Grant No. YWF-18-BJ-Y-181).

References

[1] H. Gunes, M. Piccardi, Bi-modal emotion recognition from expressive face and body gestures, J. Netw. Comput. Appl. 30 (4) (2007) 1334–1345.
[2] K. Wan, S.Z. Bong, M. Murugappan, N.M. Ibrahim, K. Mohamad, Y. Rajamanickam, Implementation of wavelet packet transform and non linear analysis for emotion classification in stroke patient using brain signals, Biomed. Signal Process. Control 36 (2017) 102–112.
[3] R. Yuvaraj, M. Murugappan, N.M. Ibrahim, K. Sundaraj, M.I. Omar, K. Mohamad, R. Palaniappan, Detection of emotions in Parkinson's disease using higher order spectral features from brain's electrical activity, Biomed. Signal Process. Control 14 (1) (2014) 108–116.
[4] M.E. Ayadi, M.S. Kamel, F. Karray, Survey on speech emotion recognition: features, classification schemes, and databases, Pattern Recognit. 44 (3) (2011) 572–587.
[5] M. Nardelli, G. Valenza, A. Greco, A. Lanata, E. Scilingo, Recognizing emotions induced by affective sounds through heart rate variability, IEEE Trans. Affect. Comput. 6 (4) (2015) 385–394.
[6] C.N. Anagnostopoulos, T. Iliou, I. Giannoukos, Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011, Artif. Intell. Rev. 43 (2) (2012) 155–177.
[7] Z. Zhang, E. Coutinho, J. Deng, B. Schuller, Cooperative learning and its application to emotion recognition from speech, IEEE/ACM Trans. Audio Speech Lang. Process. 23 (1) (2015) 115–126.
[8] A. Guidi, N. Vanello, G. Bertschy, C. Gentili, L. Landini, E.P. Scilingo, Automatic analysis of speech F0 contour for the characterization of mood changes in bipolar patients, Biomed. Signal Process. Control 17 (2015) 29–37.
[9] L. He, M. Lech, N.C. Maddage, N.B. Allen, Study of empirical mode decomposition and spectral analysis for stress and emotion classification in natural speech, Biomed. Signal Process. Control 6 (2) (2011) 139–146.
[10] S. Wu, T.H. Falk, W.Y. Chan, Automatic speech emotion recognition using modulation spectral features, Speech Commun. 53 (5) (2011) 768–785.
[11] H. Pérez-Espinosa, C.A. Reyes-García, L. Villaseñor-Pineda, Acoustic feature selection and classification of emotions in speech using a 3D continuous emotion model, Biomed. Signal Process. Control 7 (1) (2012) 79–87.
[12] Z. Huang, M. Dong, Q. Mao, Y. Zhan, Speech Emotion Recognition Using CNN, ACM Multimedia, 2014, pp. 801–804.
[13] Y. Huang, A. Wu, G. Zhang, Y. Li, Extraction of adaptive wavelet packet filter-bank-based acoustic feature for speech emotion recognition, IET Signal Process. 9 (4) (2015) 341–348.
[14] S. Demircan, H. Kahramanli, Application of fuzzy C-means clustering algorithm to spectral features for emotion classification from speech, Neural Comput. Appl. (2016) 1–8.
[15] Y. Sun, G. Wen, J. Wang, Weighted spectral features based on local Hu moments for speech emotion recognition, Biomed. Signal Process. Control 18 (2015) 80–90.
[16] L. Chen, X. Mao, Y. Xue, L.L. Cheng, Speech emotion recognition: features and classification models, Digit. Signal Process. 22 (6) (2012) 1154–1160.
[17] L. Chen, X. Mao, H. Yan, Text-independent phoneme segmentation combining EGG and speech data, IEEE/ACM Trans. Audio Speech Lang. Process. 24 (6) (2016) 1029–1037.
[18] A. Graves, N. Jaitly, A.R. Mohamed, Hybrid speech recognition with Deep Bidirectional LSTM, Autom. Speech Recognit. Underst. (2013) 273–278.
[19] K. Han, D. Yu, I. Tashev, Speech emotion recognition using deep neural network and extreme learning machine, in: Conference of the International Speech Communication Association, 2014, pp. 223–227.
[20] H. Lee, L. Yan, P. Pham, A.Y. Ng, Unsupervised feature learning for audio classification using convolutional deep belief networks, Neural Inf. Process. Syst. (2009) 1096–1104.
[21] W.Q. Zheng, J.S. Yu, Y.X. Zou, An experimental study of speech emotion recognition based on deep convolutional neural networks, in: International Conference on Affective Computing and Intelligent Interaction IEEE, 2015, pp. 827–831.
[22] G. Antipov, S.A. Berrani, N. Ruchaud, J.L. Dugelay, Learned vs. Hand-crafted features for pedestrian gender recognition, in: ACM International Conference on Multimedia ACM, 2015, pp. 1263–1266.
[23] K. Wang, N. An, B.N. Li, Y. Zhang, L. Li, Speech emotion recognition using fourier parameters, IEEE Trans. Affect. Comput. 6 (1) (2015) 69–75.
[24] T. Ogunfunmi, R. Togneri, M. Narasimha (Eds.), Speech and Audio Processing for Coding, Enhancement and Recognition, Springer, New York, 2015.
[25] A.B. Kandali, A. Routray, T.K. Basu, Emotion recognition from Assamese speeches using MFCC features and GMM classifier, in: TENCON 2008 - 2008 IEEE Region 10 Conference IEEE, 2008, pp. 1–5.
[26] A. Milton, S.S. Roy, S.T. Selvi, SVM scheme for speech emotion recognition using MFCC feature, Int. J. Comput. Appl. 69 (9) (2013) 34–39.
[27] V.B. Waghmare, R.R. Deshmukh, P.P. Shrishrimal, G.B. Janvale, Emotion recognition system from artificial marathi speech using MFCC and LDA techniques, in: International Conference on Advances in Communication, Network, and Computing, 2014.
[28] S. Demircan, H. Kahramanl, Feature extraction from speech data for emotion recognition, J. Adv. Comput. Netw. 2 (1) (2014) 28–30.
[29] N.J. Nalini, S. Palanivel, M. Balasubramanian, Speech emotion recognition using residual phase and MFCC features, Int. J. Eng. Technol. 5 (6) (2013) 4515–4527.
[30] F. Chenchah, Z. Lachiri, Acoustic emotion recognition using linear and nonlinear cepstral coefficients, Int. J. Adv. Comput. Sci. Appl. 6 (11) (2015).
[31] N.J. Nalini, S. Palanivel, Music emotion recognition: the combined evidence of MFCC and residual phase, Egypt. Inf. J. 17 (1) (2015) 1–10.
[32] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.
[33] A. Stuhlsatz, C. Meyer, F. Eyben, T. Zielke, G. Meier, B. Schuller, Deep neural networks for acoustic emotion recognition: raising the benchmarks, in: IEEE International Conference on Acoustics, Speech, and Signal Processing IEEE, 2011, pp. 5688–5691.
[34] E.M. Schmidt, J.J. Scott, Y.E. Kim, Feature learning in dynamic environments: modeling the acoustic structure of musical emotion, in: International Symposium/Conference on Music Information Retrieval, 2012, pp. 325–330.
[35] D. Le, E.M. Provost, Emotion recognition from spontaneous speech using hidden markov models with deep belief networks, in: Automatic Speech Recognition and Understanding (ASRU) IEEE, 2013, pp. 216–221.
[36] Q. Mao, M. Dong, Z. Huang, Y. Zhan, Learning salient features for speech emotion recognition using convolutional neural networks, IEEE Trans. Multimedia 16 (8) (2014) 2203–2213.
[37] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, 2015, pp. 448–456.
[38] D. Clevert, T. Unterthiner, S. Hochreiter, Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs), arXiv preprint arXiv: 1511.07289, 2015.
[39] Y. Lecun, B.E. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (4) (1989) 541–551.
[40] B.E. Boser, E. Sackinger, J. Bromley, Y. Lecun, An analog neural network processor and its application to high-speed character recognition, Ijcnn-91-Seattle International Joint Conference on Neural Networks IEEE 1 (1991) 415–420.
[41] S. Behnke, Discovering hierarchical speech features using convolutional non-negative matrix factorization, International Joint Conference on Neural Networks IEEE 4 (2003) 2758–2763.
[42] D. Palaz, R. Collobert, M.M. Doss, Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks, in: Conference of the International Speech Communication Association, 2013, pp. 1766–1770.
[43] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[44] Felix A. Gers, J.A. Schmidhuber, F.A. Cummins, Learning to forget: continual prediction with LSTM, Neural Comput. 12 (10) (2014) 2451–2471.
[45] J. Loughrey, P. Cunningham, Using Early Stopping to Reduce Overfltting in Wrapper-Based Feature Weighting, Trinity College Dublin Department of Computer Science, 2005, TCD-CS-2005-41, pp12.
[46] J.S. Bergstra, R. Bardenet, Y. Bengio, B. Kégl, Algorithms for hyper-parameter optimization, Neural Inf. Process. Syst. (2011) 2546–2554.
[47] J. Snoek, H. Larochelle, R.P. Adams, Practical Bayesian optimization of machine learning algorithms, Neural Inf. Process. Syst. 4 (2012) 2951–2959.
[48] C. Thornton, F. Hutter, H.H. Hoos, K. Leytonbrown, Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms, Knowl. Discov. Data Min. (2013) 847–855.
[49] J. Bergstra, D. Yamins, D.D. Cox, Hyperopt: a python library for optimizing the hyperparameters of machine learning algorithms, in: Proceedings of the 12th Python in Science Conference, 2013, pp. 13–20.

[50] F. Burkhardt, A. Paeschke, M. Rolfes, W.F. Sendlmeier, B. Weiss, A database of German emotional speech, INTERSPEECH 2005 - Eurospeech, European Conference on Speech Communication and Technology (2005) 1517–1520.
[51] C. Busso, M. Bulut, C.C. Lee, A. Kazemzadeh, E. Mower, S. Kim, et al., IEMOCAP: interactive emotional dyadic motion capture database, Lang. Resour. Eval. 42 (4) (2008) 335–359.
[52] Franois Chollet, Keras, 2015 https://github.com/fchollet/keras.
[53] Y. Kim, H. Lee, E.M. Provost, Deep learning for robust feature generation in audiovisual emotion recognition, Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on 32 (2013) 3687–3691.
[54] B. McFee, C. Raffel, D. Liang, D.P. Ellis, M. McVicar, E. Battenberg, O. Nieto, Librosa: audio and music signal analysis in python, in: In Proceedings of the 14th Python in Science Conference, 2015, pp. 18–25.
[55] P. Diaconis, F. Mosteller, Methods for studying coincidences, J. Am. Stat. Assoc. 84 (408) (1989) 853–861.
[56] L. Breiman, Statistical Modeling: the two Cultures (with comments and a rejoinder by the author), Stat. Sci. 16 (3) (2001) 199–231.
[57] Z.C. Lipton, The Mythos of Model Interpretability, arXiv preprint arXiv: 1606.03490, 2016.
[58] K. Aho, D. Derryberry, T. Peterson, Model selection for ecologists: the worldviews of AIC and BIC, Ecology 95 (3) (2014) 631–636.
[59] A. Neumaier, Solving ill-conditioned and singular linear systems: a tutorial on regularization, Siam Rev. 40 (3) (1998) 636–666.
[60] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, International Joint Conference on Artificial Intelligence 14 (1995) 1137–1143.
[61] New Theory Cracks Open the Black Box of Deep Learning, Quanta Magazine, 2017, Accessed 22 December 2017 https://www.quantamagazine.org/new-theory-cracks-open-the-black-box-of-deep-learning-20170921/.
[62] K. Pei, Y. Cao, J. Yang, S. Jana, Deep Xplore: Automated Whitebox Testing of Deep Learning Systems, 2017, pp. 1–18.
[63] G. Katz, C. Barrett, D.L. Dill, K. Julian, M.J. Kochenderfer, Reluplex: an efficient SMT solver for verifying deep neural networks, International Conference on Computer Aided VerificationSpringer (2017) 97–117.
[64] D. Castelvecchi, Can we open the black box of AI? Nature 538 (7623) (2016) 20.
[65] MIT Technology Review, the Dark Secret at the Heart of AI, 2017, Accessed 22 December 2017 https://www.technologyreview.com/s/604087/the-dark-secret-at-the-heart-of-ai/.
[66] D.H. Alman, L. Ningfang, Overtraining in back-propagation neural networks: a CRT color calibration example, Color Res. Appl. 27 (2) (2002) 122–125.
[67] Dan Ciregan, Ueli Meier, Jurgen Schmidhuber, Multi-column deep neural networks for image classification, Comput. Vis. Pattern Recognit. 3 (2012) 642–3649.