Learning spectro-temporal features with 3D CNNs for speech emotion recognition

Jaebok Kim, Khiet P. Truong, Gwenn Englebienne, and Vanessa Evers

Human Media Interaction, University of Twente, Enschede, The Netherlands
Email: {j.kim, k.p.truong, g.englebienne, v.evers}@utwente.nl

***Abstract*—In this paper, we propose to use deep 3-dimensional convolutional networks (3D CNNs) in order to address the challenge of modelling spectro-temporal dynamics for speech emotion recognition (SER). Compared to a hybrid of Convolutional Neural Network and Long-Short-Term-Memory (CNN-LSTM), our proposed 3D CNNs simultaneously extract short-term and long-term spectral features with a moderate number of parameters. We evaluated our proposed and other state-of-the-art methods in a speaker-independent manner using aggregated corpora that give a large and diverse set of speakers. We found that 1) shallow temporal and moderately deep spectral kernels of a homogeneous architecture are optimal for the task; and 2) our 3D CNNs are more effective for spectro-temporal feature learning compared to other methods. Finally, we visualised the feature space obtained with our proposed method using t-distributed stochastic neighbour embedding (T-SNE) and could observe distinct clusters of emotions.**

1. Introduction

Recently, deep learning methods such as Fully-connected Neural Networks (FCN) [1], Convolutional Neural Networks (CNN) [2], [3], and Long Short-Term Memory (LSTM) [4], [5] have shown considerable improvements of performance in speech emotion recognition (SER). As a potential way to improve performance, representation learning has been used to build high-level features from low-level features through several layers [2], [3], [6]. However, learning sequential structures of spectrogram representations appeared to be still challenging [7]. CNN-based methods have been investigated to this end [2], [3], [4], [8]. Although a hybrid of CNN and LSTM can be a promising method to deal with spectral variations and temporal dependencies, LSTM has the limitation of increasing depth caused by the great number of parameters. Hence, learning temporal dynamics of spectral properties for SER remains a challenge.

In this paper, we propose to learn spectro-temporal features using deep 3D CNNs. 3D CNNs are able to extract spatio-temporal features in a seamless way and have shown promising performances in computer vision tasks [9], [10]. For the task of SER, our proposed method composes a temporal series of 2D spectral feature maps and models both short and long-term dependencies simultaneously. Using large-scale datasets (7 representative, aggregated corpora) and speaker-independent classification experiments, we evaluated our proposed 3D CNNs for SER in a way that is representative of challenges for SER "in the wild". We found that 1) homogeneous layers with shallow temporal but deep spectral kernels work best among the limited set of explored architectures, and that 2) our proposed 3D CNNs are more effective and efficient for spectro-temporal feature learning in SER compared to other CNN-based methods.

This paper is structured as follows. We first introduce related studies in Section 2. Next, we present corpora in Section 3, and describe our proposed learning method in Section 4. The results will be reported in Section 5 and concluded in Section 6.

2. Related Work

The performance of machine learning relies on "feature engineering" that puts large effort in finding an optimal set of features. For the same reason, previous works in SER has focused on finding optimal feature sets and resulted in the wide usage of off-the-shelf features such as F0, Mel-Frequency Cepstrum Coefficients, and energy. However, the performance greatly varies between corpora that have distinct tasks. Representation learning offers a partial and potential solution by extracting high-level features from low-level features through a composition of multiple non-linear transformations [6]. With deep architectures, representation learning offers two advantages: 1) re-use of features which yields benefits in both computational and statistical efficiency, and 2) abstraction of features. For example, CNN can extract abstract features in a more explicit way via a pooling mechanism [11].

The performance of SER using deep architectures can still be much improved, and an optimal feature set has not been found yet for SER. For example, in [1], [5], [12], high-level features obtained from off-the-shelf features outperformed conventional methods. However, representation learning using log-spectrogram features did not outperform that of using off-the-shelf features - learning such a complex sequential structure of emotional speech appeared to be hard for representation learning [7].

More recently, the field of SER started employing CNN using low-level features. CNN is able to recognise patterns

978-1-5386-0563-9/17/$31.00

with distortions and variations [11]. It operates two functions and generates the integral of the point-wise multiplication of the two functions. For one dimensional (D) discrete data, it is defined as:

$$(f * g)(t) = \sum_{k=-T}^{T} f(t-k) \cdot g(k) \quad (1)$$

CNN-based methods using low-level features were proposed and outperformed off-the-shelf feature-based methods [2], [3], [4], [8], [13]. In [2], [3], [8] 2D feature maps were composed of spectrogram features with a fine resolution. However, these 2D CNNs cannot model temporal dependency directly. Instead, LSTM should be followed to model temporal dependencies [4], [8]. Moreover, temporal convolutions can extract spectral features from raw wave signals and capture long-term dependencies [4]. Lastly, CNN-LSTM-DNN was proposed to address frequency variations in spectral domain, long-term dependencies, separation in utterance-level feature space for the task of speech recognition [14]. While these methods augment CNNs and LSTM to handle spectral variations and temporal dynamics, a large number of parameters are required, and it is hard to learn complex dynamics with limited depths. Without these complex memory mechanisms, 3D CNNs could learn temporal features [9], [10]. In [9], [10], a series of human's motion was modelled by 3D CNNs, it empirically turned out that 3D CNNs are not only effective but also efficient to capture spatio-temporal features.

3. Data

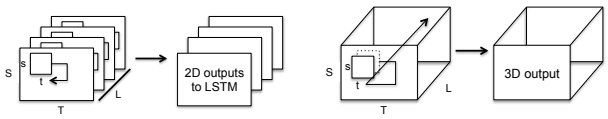
We select seven representative corpora: LDC Emotional Prosody [15], eNTERFACE [16], EMODB [17] FAU-aibo emotion corpus [18], IEMOCAP [19], SEMAINE [20], and RECOLA [21]. Since there are more corpora that have discrete labels than those that have continuous labels (e.g. arousal and valence), we focus on four discrete categories, neutral, happy, sad, and angry, which are commonly accessible as summarised in Table 1. However, the SEMAINE and RECOLA corpora provide only continuous labels such as arousal and valence, not discrete categories. To map the continuous labels into the four discrete categories, we use the landmarks of the valence and arousal dimensions as provided in FEELTRACE [22]. We extract segments by using voice activity detection or given time-alignment labels. Then, we calculate the Euclidean Distance between the landmarks and the values of the valence and arousal dimensions of each segment. Since each segment has a sequence of values of valance and arousal, we calculate the average distance for each discrete category. Next, we assign the emotional category with the smallest (average) distance to the valence and arousal values. Table 2 shows the Feeltrace landmarks for the four categories and the corresponding valence and arousal values. We use only speech utterances mapped into the four emotional categories and remove those with other categories. To the best of our knowledge, our dataset obtained from aggregating the 7 corpora has the largest number of speakers and samples in the deep-learning based experiments for SER.

TABLE 1. OVERVIEW OF THE SELECTED CORPORA (THE NUMBER OF SPEAKERS AND UTTERANCES)

Corpus ID	Speakers	Emotion			
		neutral	happy	sad	angry
AIBO	51	10967	889	0	1492
EMODB	10	77	61	58	97
ENTERFACE	43	0	208	422	211
LDC	7	80	180	161	139
IEMOCAP	10	1708	595	2168	2206
SEMAINE	20	2694	766	82	392
RECOLA	23	159	14	109	121
Total	164	15685	2713	3000	4658

TABLE 2. LANDMARKS FOR THE SEMAINE AND RECOLRA CORPORA IN FEELTRACE.

discrete emotional categories	valence	arousal
neutral	0.00	0.00
happy	0.74	0.52
angry	-0.77	0.75
sad	-0.7	-0.48



(a) 2D-CNN-LSTM (b) 3D-CNN

Figure 1. The comparison of 2D-CNN-LSTM and 3D-CNN: a) Applying 2D convolution on each spectral feature map (T x S) in a time series (L). b) Applying 3D convolution on a time series of maps (L x T x S) results in another volume, preserving temporal dynamics of the input.

4. Method

While the previous methods [8], [14] learn spectral features and the temporal dependencies via the augmentation of CNN and LSTM, our proposed method is designed to learn spectro-temporal features simultaneously as depicted in Figure 1. Particularly, spectral features should have a sufficiently fine resolution, and both short-term ($\sim$ 200ms) and long-term ($\sim$ 2s) should be considered. To this end, the following network topology is configured.

Input feature maps. First, we segment utterances into 2s long sequences after min-max normalisation of gains per speaker. Zero-padding is applied for utterances shorter than 2s while we trim those longer than 2s. Then, we extract 256 points log-spectrogram every 20ms. Since it potentially causes over-fitting, we do not use a sliding contextual window on it. Therefore, we obtain 100 frames. Lastly, we compose a temporal series of 2D feature maps that have a resolution of 10 x 256. Each feature map represents spectral features in shot-term (200ms) windows. As a result, each utterance segment has a resolution of 10 x 10 x 256. Let us denote elements of the resolution as long-term (L), short-term (T), and spectral (S).

3D CNNs. Input feature maps are directly fed into 3D CNNs. Based on the previous finding [10], we adopt a

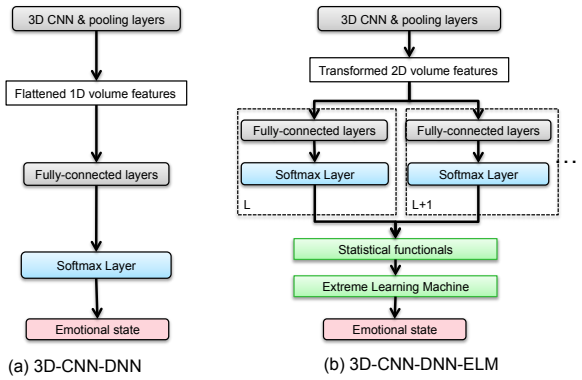


Figure 2. Block diagram of the proposed methods; L (state in long-term temporal resolution)

homogeneous architecture, i.e., all convolutional layers have the same resolutions (L x T x S). In subsequent experiments, we empirically find the optimal resolutions and the number of convolutional layers for our task. All convolutional layers have four kernels.

3D Max-pooling. A 3D max-pooling layer follows each 3D CNN. However, to preserve the spectro-temporal features at early phases [10], we do not pool outputs of the first convolutional layers. Hence, except for the first pooling layer, the rest of the pooling layers have the resolution of 2 x 2 x 2.

Lastly, we examine two methods to learn utterance-level spectro-temporal features: (a) we simply flatten 3D volume output features into 1D volume output vectors that are fed into fully-connected layers with a softmax layer; and (b) we transform 3D volume output features into 2D volume output features that are fed into a temporal series of fully-connected layers. Both methods use two fully-connected layers with 512 nodes. Let us denote method (a) and (b) as **3D-CNN-DNN** and **3D-CNN-DNN-ELM**.

3D-CNN-DNN-ELM does not use pooling for the long-term depth to train a sequence of the fully-connected layers. Moreover, it requires statistical functionals at the softmax layer and a proceeding classifier such as Extreme Learning Machine (ELM) as proposed in [1], [5]. We follow the same set-up of the functionals and ELM proposed in [1], [5]. Training a linear classifier on features from the top fully-connected layer of 3D-CNN-DNN can be an effective approach [10] but it may bring similar effects of 3D-CNN-DNN-ELM. Hence, we do not examine the variant of 3D-CNN-DNN. Figure 2 illustrates the difference between 3D-CNN-DNN and 3D-CNN-DNN-ELM.

5. Experiments and Results

First, we investigate how resolutions of kernels affect performance of the 3D CNNs. Next, based on the optimal resolution, we compare our proposed method to other state-of-the-art methods using various features and architectures. As a common set-up, we use Adam method [23] with a mini-batch of 128 samples, and a fixed learning rate of $3 \cdot 10^{-3}$.

TABLE 3. BEST, WORST, AND MEAN PERFORMANCE (UA) BY VARYING RESOLUTIONS (L X S X T) OF 3D KERNELS

Shapes	Resolutions	Best	Worst	Mean
(a) shallow temporal & deep spectral kernels	2 x 2 x 2	.518	.451	$.480 \pm .03$
	2 x 2 x 32	.541	.425	$.484 \pm .05$
	2 x 2 x 128	.550	.471	$\mathbf{.496 \pm .03}$
	2 x 2 x 256	.516	.250	$.416 \pm .10$
(b) deep temporal & shallow spectral kernels	4 x 2 x 2	.481	.393	$.450 \pm .04$
	8 x 2 x 2	.507	.394	$.454 \pm .04$
	2 x 4 x 2	.503	.419	$.450 \pm .03$
	2 x 8 x 2	.487	.441	$\mathbf{.463 \pm .02}$
(c) deep temporal & deep spectral kernels	4 x 2 x 128	.469	.437	$.451 \pm .01$
	8 x 2 x 128	.528	.421	$.462 \pm .04$
	2 x 4 x 128	.492	.446	$\mathbf{.467 \pm .02}$
	2 x 8 x 128	.488	.423	$.467 \pm .03$

We use categorical cross-entropy as the cost function. To prevent over-fitting, we use early-stopping [24] (the maximum number of epoch: 20) and dropout [25] ($p = .5$) for fully-connected layers. As an evaluation metric, Unweighted Accuracy (UA) is used to consider the imbalanced distribution of the classes. We aim to conduct cross-validation while keeping speaker-independence. Hence, we compose 5-fold cross-validation. First, we independently shuffle speakers in each corpus. Then, we divide each corpus into testing (20%), validation (20%), and training (60% of the total number of the speakers) sets. Next, seven testing sets (from the seven corpora) are merged as testing data for one fold. Validation and training data for the fold are constructed in the same way. We repeat this process five times to build 5 folds. Lastly, we use Wilcoxon signed-rank paired test (0.99 of confidence level and two-sided tests) [26] to check significance of gains.

5.1. Exploring resolutions of kernels

Table 3 summarises results of variations in temporal and spectral depth (L x T x S) of kernels. Commonly, we use three convolutional layers. In (a), we use shallow temporal kernels but vary spectral depth to find the optimal spectral depth. Shallow or moderately deep spectral kernels (2, 32, 128) significantly outperform very deep kernels (256). We assume that too deep kernels may cause over-fitting. In (b), we use shallow spectral depth but vary temporal depth. No matter which temporal depth changes, they perform worse than many cases with deep spectral depth in (a). Moreover, based on the result of (a), we keep deep spectral depth (128) but vary temporal depth. A deep long-term kernel (8 x 2 x 128) results in the best performance among them but it does not outperform the architecture with shallow temporal and deep spectral kernels (a). From these results, we empirically conclude that the kernels with shallow temporal depth and moderately deep spectral depth (2 x 2 x 128) work the best for 3D CNNs in our task. However, except for the very deep spectral kernel (256), the gap is not significant. Hence, we will examine spectral depth of 2, 32, and 128 but drop that of 256 for other methods in subsequent experiments.

TABLE 4. UA OF SPEAKER INDEPENDENT EXPERIMENTS, CNN: CONVOLUTIONAL NETWORK (RESOLUTION), FCN: FULLY-CONNECTED NETWORK (#NODE), LSTM: LONG-SHORT-TERM-MEMORY (#CELL), ELM: EXTREME-LEARNING-MACHINE

Features	Methods	Configuration of layers	Parameters (K)	UA
off-the-shelf	DNN-ELM [1]	3 x FCN (256) + ELM	141	$.384 \pm .05$
	LSTM-ELM [5]	2 x LSTM (128) + ELM	214	$.421 \pm .11$
raw waveform	1D-CNN-LSTM [4]	CNN (80) + CNN (800) + 2 x LSTM (128)	1583	$.391 \pm .06$
		CNN (80) + CNN (800) + 2 x LSTM (256)	2211	$.380 \pm .03$
log-spectrogram	2D-CNN-LSTM [8]	2 x CNN (2 x 2) + 2 x LSTM (128)	264	$.318 \pm .01$
		2 x CNN (2 x 32) + 2 x LSTM (128)	268	$.323 \pm 01$
		2 x CNN (2 x 128) + 2 x LSTM (128)	282	$.306 \pm .03$
	2D-CNN-LSTM-DNN [14]	2 x CNN (2 x 2) + 2 x LSTM (128) + 2 x FCN (512)	595	$.329 \pm .02$
		2 x CNN (2 x 32) + 2 x LSTM (128) + 2 x FCN (512)	607	$.350 \pm .03$
		2 x CNN (2 x 128) + 2 x LSTM (128) + 2 x FCN (512)	611	$.313 \pm .01$
	3D-CNN-DNN (proposed)	2 x CNN (2 x 2 x [2, 32, 128]) + 2 x FCN (512)	$792 - 810$	.250
		3 x CNN (2 x 2 x 2) + 2 x FCN (512)	798	$.480 \pm .03$
		3 x CNN (2 x 2 x 32) + 2 x FCN (512)	793	$.484 \pm .05$
		3 x CNN (2 x 2 x 128) + 2 x FCN (512)	807	$\mathbf{.496 \pm .03}$
	3D-CNN-DNN-ELM (proposed)	2 x CNN (2 x 2 x [2, 32, 128]) + 2 x FCN (512)	$527 - 548$	.250
		3 x CNN (2 x 2 x 2) + 2 x FCN (512)	528	$.495 \pm .05$
		3 x CNN (2 x 2 x 32) + 2 x FCN (512)	531	$\mathbf{.516 \pm .02}$
		3 x CNN (2 x 2 x 128) + 2 x FCN (512)	546	$.512 \pm .03$

5.2. Compared to state-of-the-art

We compare our proposed methods to state-of-the-art methods: DNN-ELM [1], LSTM-ELM [5], 1D-CNN-LSTM [4], 2D-CNN-LSTM [8], and 2D-CNN-LSTM-DNN [14]. The following descriptions explain architectures and features for the state-of-the-art methods.

DNN-ELM [1] and LSTM-ELM [5]. They use off-the-shelf features: F0, voice probability, zero-crossing-rate, 12-dimensional (D) MFCC with Root Mean Squared energy and those first time derivatives (totaling 32 features). Only DNN-ELM uses a contextual windows of five frames to model temporal dynamics. DNN-ELM has three hidden layers of 256 nodes, and LSTM-ELM has two hidden layers with 128 cells. Statistical functionals to extract utterance-level features and a proceeding Extreme Learning Machine (ELM) have the same setting in [1], [5].

1D-CNN-LSTM [4]. We extract a 32000D vector at every 2s long sequence and segment each vector to 20 sub-sequences using a contextual window with 40ms (1600D). A first temporal convolutional layer has a length of 80, followed by a max-pooling layer with a size of 2. Next, a second temporal convolutional layer with a length of 800 followed by a max-pooling with a size of 40. LSTM blocks with two hidden layers are stacked on the top conversational layer, and the cell size varies (128 and 256).

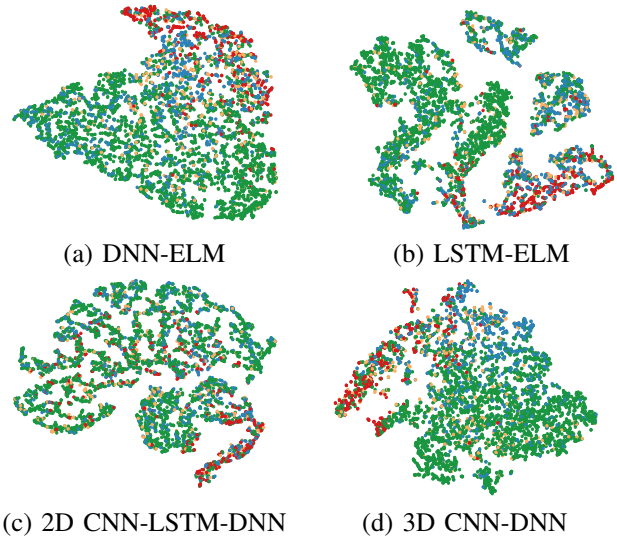
2D-CNN-LSTM [8] and 2D-CNN-LSTM-DNN [14]. The shape of feature vectors is equal to that of the proposed method (10 x 10 x 256). As the same way of our proposed method, the homogeneous kernels are adopted, and we only vary the spectral depth (2, 32, and 128). A max-pooling layer follows each convolutional layer. The first pooling layer has a resolution of 2 x 2 and the second one has that of 4 x 4. Two LSTM layers that have a cell size of 128 are stacked on the top conversational layer. Lastly, two fully-connected layers with 512 nodes are stacked too.

Table 4 summarises results. For comparison, it includes the performance of 3D-CNN-DNNs using the kernels of 2 x 2 x 2, 2 x 2 x 32, and 2 x 2 x 128 (previously presented in Table 3), too. 2 x 2 x [2, 32, 128] is short for these three shapes of kernels.

Any configuration of 2D-CNN-LSTM-DNN and 2D-CNN-LSTM does not outperform DNN-ELM and LSTM-ELM. While emotional classes can be directly learned from spectrogram features [7], outperforming off-the-shelf features is still challenging. Moreover, later experiments show that 2D-CNN-LSTM(-DNN) could not avoid critical overfitting problems as increasing the depth from two to three. Because of the complexity of LSTM, the depth is limited to two, and it might not be sufficiently deep to learn the complicated sequential structure of emotional speech [27]. 1D-CNN-LSTM using raw waveform does not outperform LSTM-ELM, too. We could not improve the performance by increasing the depth from two to three. We assume that it is mainly due to the huge number of parameters. We also differentiate other configuration (e.g. resolutions of pooling layers) later but any further gain is not observed.

On the other hand, 3D-CNN-DNN and 3D-CNN-DNN-ELM, outperform DNN-ELM and LSTM-ELM with significant gains ($\mathbf{11} \sim \mathbf{13}$ and $\mathbf{7} \sim \mathbf{9}\%$, respectively). Moreover, they outperform 1D-CNN-LSTM using raw waveform by $\mathbf{10} \sim \mathbf{12}\%$. When we use the depth of two, the trained models classify all test samples as "neutral" regardless of its kernel resolutions, resulting in UA of .25. However, they show the best performance at the depth of three. With the given data set, the shallow temporal but moderately deep spectral kernels are optimal for 3D-CNN-DNN (2 x 2 x 128) and 3D-CNN-DNN-ELM (2 x 2 x 32). We could not observe any gains as increasing the depth from three. Although 3D-CNN-DNN-ELM outperform 3D-CNN-DNN, the difference is not significant ($p = .31$).

Next, we investigate how each class becomes discriminative in the feature space. To this end, we visualise utterance-level representations of one test set of our cross-validation by using t-distributed stochastic neighbour embedding (T-SNE) [28]. T-SNE is a non-linear transform technique to



(a) DNN-ELM (b) LSTM-ELM

(c) 2D CNN-LSTM-DNN (d) 3D CNN-DNN

Figure 3. The result of T-SNE for the learned features of the test data; coloured by emotional categories (green: neutral, orange: happy, blue: sad, and red: angry)

TABLE 5. CONFUSION MATRIX (%). N: NEUTRAL, H: HAPPY, S: SAD, A: ANGRY.

Method									
DNN-ELM					LSTM-ELM				
	N	H	S	A		N	H	S	A
N	93	0	4	3	N	83	4	4	9
H	82	0	7	11	H	59	10	12	19
S	26	0	71	3	S	23	4	69	4
A	80	0	11	9	A	51	4	14	32
2D-CNN-LSTM-DNN					**3D-CNN-DNN**				
	N	H	S	A		N	H	S	A
N	80	2	10	8	N	**88**	0	7	4
H	72	1	12	15	H	64	**8**	8	20
S	38	0	58	4	S	18	1	**75**	6
A	65	0	13	22	A	41	1	8	**50**

embed high-dimensional data into a space of two or three dimensions. We obtain utterance-level features from the top fully-connected layers (before the softmax layer). Figure 3 shows results of DNN-ELM, LSTM-ELM, 2D-CNN-CNN-LSTM and 3D-CNN-DNN, and Table 5 presents their confusion matrices. Compared to 2D-CNN-LSTM-DNN, 3D-CNN-DNN shows a more discriminative separation between "neutral" and "anger". In Table 5, confusion between "neutral" and "anger" significantly decreases in 3D-CNN-DNN. The gains are $\mathbf{8}$ and $\mathbf{28}\%$, respectively. Classification of "neutral" and "sadness" shows the similar result too. The gain for "sadness" is $\mathbf{17}\%$.

5.3. Discussion

Our data has a large set of speakers from multiple corpora that have different conditions of recordings, tasks, and *etc.*. Moreover, the data has an imbalanced distribution of the categories and our evaluation is carried out in a speaker-independent manner. These settings are close to realistic challenges and pose a great potential of over-fitting (caused by the huge variance). In such a harsh condition, LSTM modelling temporal dynamics of emotional speech seems inefficient. Indeed, 1D-CNN-LSTM with a large number of parameters is vulnerable to over-fitting. Emotional speech corpora in the research community are inevitably limited compared to other tasks (speech and image recognition). Compared to the complex augmented architectures, 3D CNNs have relatively simpler architectures, and those with a moderate number of parameters could learn spectro-temporal features in a seamless way. Lastly, the variance of the performance shows the importance of large-scale cross-validation and statistical tests. While they should not be neglected, it is arduous to optimise deep architectures with the great variance of the number of samples and that of class distribution from aggregated corpora. We believe that our evaluation is able to present the realistic performance in the wild.

6. Conclusions and future work

In this paper, we proposed deep 3-dimensional convolutional networks (3D CNNs) based methods to learn spectro-temporal features for the task of speech emotion recognition (SER). We designed 3D CNNs to learn short and long-term spectro-temporal features with a moderate number of parameters. We evaluated the proposed and other state-of-the-art methods using large-scale speaker independent experiments. First, we found that shallow temporal and moderately deep spectral kernels are optimal to the explored homogeneous architectures. Next, we found that 3D CNNs are more suitable for spectro-temporal feature learning compared to other CNN based methods (e.g. CNN-LSTM). CNN-LSTMs using low-level representations do not outperform methods using off-the-shelf features. However, 3D CNNs learn the features with a moderate number of parameters and significantly outperform the other methods. In addition, we visualised the spectro-temporal features learned by the method via T-distributed stochastic neighbor embedding technique. More discriminative clusters of emotional classes can be observed in the feature space, and our proposed method significantly decreases the confusion rates. As future work, we plan to investigate identity skip-connections that are recently popular for optimising deep architectures [29].

Acknowledgments

The research leading to the results was supported by the European Community's 7th Framework Programme under Grant agreement 610532 (SQUIRREL - Clearing Clutter Bit by Bit) and 611153 (TERESA - Telepresence Reinforcement-learning Social Agent).

References

[1] I. T. Kun Han, Dong Yu, "Speech emotion recognition using deep neural network and extreme learning machine," in *Proceedings of INTERSPEECH*, 2011.

[2] W. Zheng, J. Yu, and Y. Zou, "An experimental study of speech emotion recognition based on deep convolutional neural networks," in *Proceedings of International Conference on Affective Computing and Intelligent Interaction*. IEEE, 2015, pp. 827–831.

[3] Q. Mao, M. Dong, Z. Huang, and Y. Zhan, "Learning salient features for speech emotion recognition using convolutional neural networks," *IEEE Transactions on Multimedia*, vol. 16, no. 8, pp. 2203–2213, 2014.

[4] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. A. Nicolaou, S. Zafeiriou *et al.*, "Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5200–5204.

[5] J. Lee and I. Tashev, "High-level feature representation using recurrent neural network for speech emotion recognition," in *Proceedings of INTERSPEECH*, 2015.

[6] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1798–1828, 2013.

[7] S. Ghosh, E. Laksana, L.-P. Morency, and S. Scherer, "Representation learning for speech emotion recognition," in *Proceedings of INTERSPEECH*, 2016, pp. 3603–3607.

[8] N. Anand and P. Verma, "Convoluted feelings convolutional and recurrent nets for detecting emotion from audio data."

[9] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[10] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4489–4497.

[11] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[12] J. Kim, G. Englebienne, K. P. Truong, and V. Evers, "Towards speech emotion recognition "in the wild" using aggregated corpora and deep multi-task learning," in *Proceedings of INTERSPEECH*, 2017, p. To be appeared.

[13] ——, "Deep temporal models using identity skip-connections for speech emotion recognition," in *Proceedings of ACM Multimedia*, 2017, p. To be appeared.

[14] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4580–4584.

[15] M. Liberman, K. Davis, M. Grossman, N. Martey, and J. Bell, "Emotional prosody speech and transcripts," *Linguistic Data Consortium, Philadelphia*, 2002.

[16] O. Martin, I. Kotsia, B. Macq, and I. Pitas, "The enterface'05 audio-visual emotion database," in *22nd International Conference on Data Engineering Workshops (ICDEW'06)*. IEEE, 2006, pp. 8–8.

[17] F. Burkhardt, A. Paeschke, M. Rolfes, W. F. Sendlmeier, and B. Weiss, "A database of german emotional speech." in *Proceedings of INTERSPEECH*, vol. 5, 2005, pp. 1517–1520.

[18] A. Batliner, C. Hacker, S. Steidl, E. Nöth, S. D'Arcy, M. J. Russell, and M. Wong, "You stupid tin box-children interacting with the aibo robot: A cross-linguistic emotional speech corpus." in *Proceedings of LREC*, 2004.

[19] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, no. 4, pp. 335–359, 2008.

[20] G. McKeown, M. F. Valstar, R. Cowie, and M. Pantic, "The semaine corpus of emotionally coloured character interactions," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2010, pp. 1079–1084.

[21] F. Ringeval, A. Sonderegger, J. Sauer, and D. Lalanne, "Introducing the recola multimodal corpus of remote collaborative and affective interactions," in *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*. IEEE, 2013, pp. 1–8.

[22] R. Cowie, E. Douglas-Cowie, S. Savvidou*, E. McMahon, M. Sawey, and M. Schröder, "'feeltrace': An instrument for recording perceived emotion in real time," in *ISCA Tutorial and Research Workshop (ITRW) on Speech and Emotion*, 2000, pp. 19–24.

[23] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] L. Prechelt, "Automatic early stopping using cross validation: quantifying the criteria," *Neural Networks*, vol. 11, no. 4, pp. 761–767, 1998.

[25] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[26] J. D. Gibbons and S. Chakraborti, *Nonparametric statistical inference*. Springer, 2011.

[27] Y. Kim and E. M. Provost, "Emotion classification via utterance-level dynamics: A pattern-based approach to characterizing affective expressions," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 3677–3681.

[28] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[29] A. Veit, M. J. Wilber, and S. Belongie, "Residual networks behave like ensembles of relatively shallow networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 550–558.