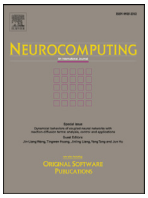
Neurocomputing 317 (2018) 50–57

Contents lists available at ScienceDirect

Neurocomputing

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



Spatio-temporal convolutional features with nested LSTM for facial expression recognition

Zhenbo Yu[a,*], Guangcan Liu[a], Qingshan Liu[a], Jiankang Deng[b]

[a] *Jiangsu Key Laboratory of Big Data Analysis Technology, School of Information and Control, Nanjing University of Information Science and Technology, Nanjing 210044, China*
[b] *Department of Computing, Imperial College London, UK Office: 351 Huxley Building, 180 Queens Gate, SW7 2AZ, UK*

ARTICLE INFO

Article history:
Received 23 January 2018
Revised 30 June 2018
Accepted 10 July 2018
Available online 18 August 2018

Communicated by Dr. M. Wang

Keywords:
Facial expression recognition
LSTM
3DCNN
Multi-level features

ABSTRACT

In this paper, we propose a novel end-to-end architecture termed Spatio-Temporal Convolutional features with Nested LSTM (STC-NLSTM), which learns the muti-level appearance features and temporal dynamics of facial expressions in a joint fashion. More precisely, 3DCNN is used to extract spatio-temporal convolutional features from the image sequences that represent facial expressions, and the dynamics of expressions are modeled by Nested LSTM, which is actually coupled by two sub-LSTMs, saying T-LSTM and C-LSTM. Namely, T-LSTM is used to model the temporal dynamics of the spatio-temporal features in each convolutional layer, and C-LSTM is adopted to integrate the outputs of all T-LSTMs together so as to encode the multi-level features encoded in the intermediate layers of the network. We conduct experiments on four benchmark databases, CK+, Oulu-CASIA, MMI and BP4D, and the results show that the proposed method achieves a performance superior to the state-of-the-art methods.

1. Introduction

Facial Expression Recognition (FER) [1], in general, is to automatically group various kinds of facial muscle motions into similar emotion categories purely based on the visual information in images or videos. Due to its potentials in a broad range of applications such as face recognition [2–5], face alignment [6–14] and human-computer interface [15], FER has received extensive attentions in the literatures, e.g., [16–18]. Essentially, facial expression is a dynamic process consisting of multiple stages, mainly including neutral, onset, apex and offset [19], so how to learn the dynamics of facial expressions is a key issue in FER [20].

Early FER methods [21,22] are often built upon some predefined features such as the Gabor filters, haar-like features and Local Binary Patterns (LBP). These methods may work well only on limited occasions, as the pre-defined features are incapable of fitting well with the data from a wide range of applications. To overcome this issue, it would be natural to consider the deep learning methods such as Convolutional Neural Network (CNN) [23], which can seamlessly integrate feature extraction and expression classification into a unified procedure. Extensive experiments demonstrate that CNN achieves substantial improvement in recognition accuracy over conventional methods [16,24–26], but most of CNN-based methods consider a video as a collection of multiple static images, and thus they may not handle well the dynamic nature of facial expressions.

In order to make better use of the features that capture the motion of facial muscles, sequence-based methods [25,27–29], which represent an expression by a sequence of images with known time stamp, have emerged as a preferable choice. To analyze sequential data, the deep learning community has also established several tools, e.g., Recurrent Neural Network (RNN) [30], Long Short-Term Memory (LSTM) [31–34] and 3D Convolutional Neural Network (3DCNN) [35]. Especially, the CNN-RNN (or CNN-LSTM) framework attracts much attention [25,27,28], in which RNN (or LSTM) takes the appearance features extracted by CNN over individual frames as inputs and encodes the temporal dynamics for later use, because it can combine the advantages of CNN and RNN to model both the appearance features and temporal dynamics simultaneously. Recently, researchers have investigated the framework of 3DCNN-RNN (or 3DCNN-LSTM) [36,37]. Unlike CNN, which only deals with 2D inputs, 3DCNN takes image sequences as inputs and can therefore extract directly the spatio-temporal features underlying the image sequences. Despite of the considerable improvement attained with the help of deep learning in recent few years, existing methods often use only the outputs of the last fully-connected layer as features for classification, discarding much useful information encoded in the intermediate layers of the network. As can be

[*] Corresponding author.
E-mail address: zbyu@nuist.edu.cn (Z. Yu).

https://doi.org/10.1016/j.neucom.2018.07.028
0925-2312/© 2018 Elsevier B.V. All rights reserved.

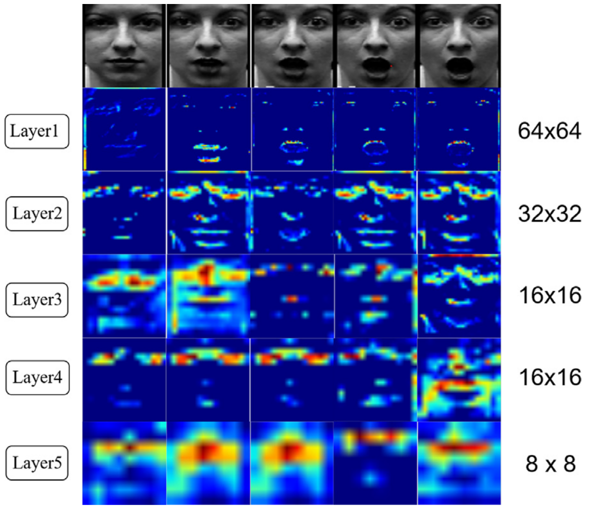


Fig. 1. Visualization of the convolutional features extracted from different layers of 3DCNN. The blue and red points correspond to the low and high response values, respectively. The emotion label for the input image sequence is surprise.

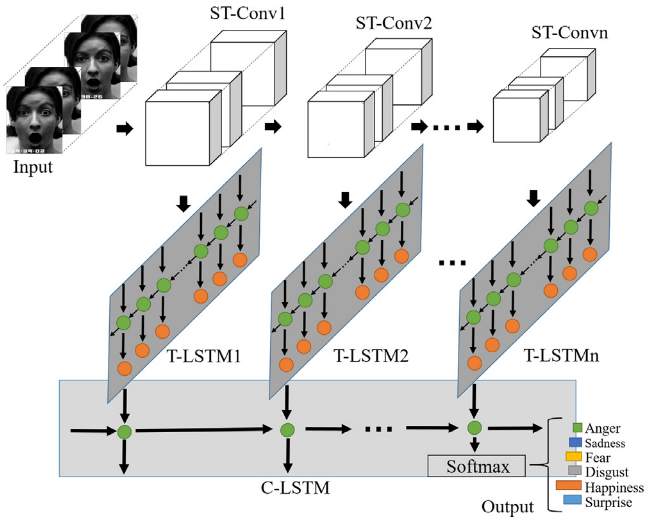


Fig. 2. Architecture of the proposed STC-NLSTM, which consists of 3DCNN and Nested LSTM, and which is coupled by temporal-LSTM (T-LSTM) and convolutional-LSTM (C-LSTM). In the figure above, the term "ST-Convs" standards for the spatio-temporal convolutional features.

seen from Fig. 1, early convolutional layers extract fine-grained details (e.g., local boundaries or illuminations) of faces, while later layers capture more detailed information, e.g., the appearance patterns of mouths and eyes. It can be seen that the features from all layers of 3DCNN indeed provide FER with a hierarchical representation of multi-level features from fine to coarse. Such a hierarchical representation, intuitively, would be more effective than the features contained in the last layer only.

In this work, we propose an end-to-end FER method that can involve various visual clues, including the multi-level appearance features and the temporal dynamics of facial expressions. To this end, we propose a novel architecture termed Spatio-Temporal Convolutional features with Nested LSTM (STC-NLSTM), which is illustrated in Fig. 2. In general, our STC-NLSTM contains three major components: (1) a 3DCNN module consisting of multiple convolutional layers, (2) multiple temporal-LSTM (T-LSTM) modules each of which corresponds to one layer of the 3DCNN, and (3) a convolutional-LSTM (C-LSTM) module that takes the outputs of T-LSTMs as inputs[1]. Given a sequence of images that represent an emotion class, first, the 3DCNN module extracts the spatio-temporal convolutional features of the expression for later use. Second, T-LSMT takes the spatio-temporal features as inputs and produces compact features that encode the appearance features as well as the temporal dynamics. Third, C-LSTM plays the role of integrating the outputs of all T-LSTMs together and encoding the multi-level features contained in each convolutional layer. Finally, the softmax classifier is used to categorize the given sequence into one of the six basic emotion classes. In contrast to the existing sequence-based methods [25,28], our STC-NLSTM can utilize not only the appearance features as well as the temporal dynamics of facial expressions, but also the multi-level semantics encoded in the individual layers of the network, so as to attain more reliable classification results. Experiments on CK+ [38], Oulu-CASIA [39], MMI [19] and BP4D [40] show that the proposed STC-NLSTM is superior to the state-of-the-art methods.

The rest of this paper is organized as follows. Section 2 provides a brief survey for FER. Section 3 introduces the proposed STC-NLSTM method. Section 4 shows some empirical results and Section 5 concludes this paper.

2. Related work

Deep learning methods have exhibited superior performance for FER, showing dramatic improvement in accuracy and robustness over the conventional methods based on pre-defined features [16,24–26,28,40–43]. According to how an expression is represented, existing methods can be roughly divided into two categories: image-based and sequence-based methods.

In general, FER is a special pattern recognition problem, and thus the techniques for generic classification can be naturally applied to FER. Yu et al. [44] proposed a FER method that combines together an ensemble of multiple CNNs by minimizing a mixture of the log likelihood loss and the hinge loss. Kim et al. [45] devised a recognition framework by combining multiple CNNs to form a hierarchical network, and they won the first place of EmotiW 2015, an international competition of FER. Bargal et al. [41] established a hybrid network that combines VGG16 [46] with Residual Network [47] to learn the appearance features of expressions, and they used SVM to produce the final classification results. Yao et al. [48] proposed a deeper and wider network consisting of three inception modules, and Zhao et al. [26] built a novel peak-piloted feature transformation network to capture the intrinsic correlations between the peak and weak expressions.

Different from the image-based methods, the sequence-based methods attempted to well capture the temporal variations of the appearance features, which are better for FER. Liu et al. [49] proposed a FER method termed 3DCNN-DAP (DAP standards for deformable action part), in which 3DCNN is used to extract the spatio-temporal features and the strong spatial structural constraints among the dynamic action parts as well. Jung et al. [16] studied an integrated network with joint fine-tuning to infer the appearance features and temporal dynamics of facial expressions. Jaiswal et al [25] utilized CNN in combination with Bi-directional LSTM (BiLSTM) for FER, achieving a performance better than the winner of FERA 2015 [40]. Fan et al. [28] established a novel hybrid network that combines 3DCNN and RNN in a late-fusion fashion and won the first place in EmotoW2016 [42]. The proposed STC-NLSTM method belongs to the sequence-based

[1] In addition to the three major components shown in Fig. 2, STC-NLSTM actually contains another component called Multi-dimensional Spatial Pyramid Pooling normalization (MSPP-norm), which is in charge of normalizing the convolutional features of different dimensions into the same size. We shall clarify this detail in Section 3.2.1.

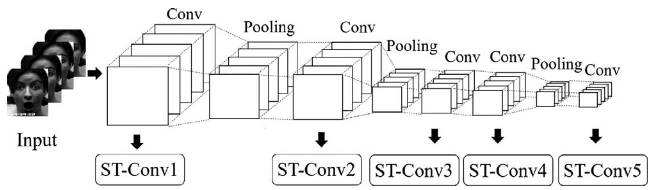


Fig. 3. Architecture of the adopted 3DCNN.

methods. Comparing to the previous works, our STC-NLSTM provides a fine grained approach for modeling multi-level features encoded in the intermediate layers of the network so as to achieve more accurate FER.

3. Spatio-Temporal convolutional features with nested LSTM

This section details the proposed STC-NLSTM. As shown in Fig. 2, our STC-NLSTM has three main components: (1) a 3DCNN module is used to extracted the spatio-temporal convolutional features (ST-Convs) of facial expressions, (2) multiple T-LSTM modules are adopted to capture the temporal dynamics of facial muscle motions, and (3) a C-LSTM module aims to seize the multi-level features encoded in the individual layers of the 3DCNN.

3.1. Spatio-temporal convolutional features

Since facial expression is essentially a dynamic process, we attempt to extract directly the spatio-temporal features of facial expressions by a more straightforward approach, that is, the well recognized 3DCNN, which has been widely used in the fields of activity recognition, lip reading recognition, gesture recognition, and so on [35]. Different from the traditional CNN that can only deal with 2D inputs, 3DCNN takes directly the image sequences as inputs and can therefore capture literally the spatio-temporal features of image sequences.

Fig. 3 illustrated the architecture of the 3DCNN. Given a sequence of images with known time stamp that represent a facial emotion class, 3DCNN processes the sequence by multiple convolutional and pooling layers, producing a collection of spatio-temporal features that characterize the expression.

3.2. Nested LSTM

To capture the multi-level features encoded in the intermediate layers of the network, we propose the so-called Nested LSTM shown in Fig. 4, which is composed of MSPP-norm, T-LSTM and C-LSTM. MSSP-norm aims to normalize the spatio-temporal features of different sizes to the same dimension, while T-LSTM and C-LSTM can capture the temporal dynamics and seize the multi-level features encoded in the individual convolutional layers of the network respectively.

3.2.1. MSPP-norm

Because the spatio-temporal features extracted by different layers of 3DCNN have different dimensions, it is impossible to input them directly to the LSTM units, which generally require the inputs to have the same dimension. To fill this gap, we design the Multi-dimensional Spatial Pyramid Pooling normalization (MSPP-norm) operation, which is inspired by the Spatial Pyramid Pooling network (SPP-net) proposed by He et al. [50]. The purpose of the MSSP-norm is to normalize the spatio-temporal features of different sizes to the same dimension. Given a 3D feature map of size $N \times a \times a$, we partition it into $N \times n \times n$ (with $n = 2, 4, 8$) sub-regions and summarize the responses within each sub-region via max pooling, resulting in a feature vector with a fixed dimension determined by the parameter n.

3.2.2. T-LSTM and C-LSTM

After the process of MSPP-norm, the spatio-temporal features in each layer of 3DCNN are transferred to feature vectors of the same dimension. Thus, it is suitable to further analyze the spatio-temporal features by LSTM, which is an advanced RNN architecture for sequential data analysis including in FER [27,51]. The commonly used LSTM can model the temporal information by transforming a sequence of inputs to a sequence of outputs; this, in general, can partially capture the correlations among the spatio-temporal features extracted by 3DCNN. However, few conventional methods based on LSTM make full use of the information encoded in all the convolutional layers, because it is hard to involve all of the appearance features, temporal dynamics and multi-level features by simply combining 3DCNN with LSTM.

To deal with the above issues, we adopt two LSTM modules, saying T-LSTM and C-LSTM to cope with the spatio-temporal features extracted by 3DCNN. For each feature vector corresponding to a certain convolution layer, a T-LSTM is constructed by stacked LSTM units, which models the temporal dynamics of facial expressions. After that, a C-LSTM is constructed to take the outputs of T-LSTMs as inputs, so the desired multi-level features can be modeled in a seamless way.

Suppose that there are in total l convolutional layers in 3DCNN. Then the procedure of our STC-NLSTM method can be summarized as follows:

$$\begin{aligned} f_j &= \text{3DCNN}(x), \, j = 1, \ldots, l, \\ f_j^{\text{mspp}} &= \text{MSPP-norm}(f_j), \, j = 1, \ldots, l, \\ h_j &= \text{T-LSTM}_{\text{j}}(f_j^{\text{mspp}}), \, j = 1, \ldots, l, \\ h &= \{h_1, \ldots, h_l\}, \\ o &= \text{C} - \text{LSTM}(h), \end{aligned}$$

where x denotes an image sequence, f_j is the 3D feature map produced by the jth convolutional layer of 3DCNN, h_j is the feature vector from the jth T-LSTM module, and o denotes the final feature vector used for classification.

4. Experiments and results

4.1. Experimental data

To verify the effectiveness of the proposed STC-NLSTM, we experiment with four benchmark datasets, CK+ [38], Oulu-CASIA [39], MMI [19] and BP4D [40]. **CK+:** This dataset has six basic emotion classes, including anger(An), disgust(Di), fear(Fe), happiness(Ha), sadness(Sa) and surprise(Su). In addition, there is another special expression called “contempt”. The dataset contains in total 593 image sequences from 123 subjects, but only 309 sequences are annotated with the six basic expression labels. We divide these 309 sequences into 10 groups, of which 9 groups are used for training and the rest for testing. In this way, we can run various FER methods multiple times and obtain an averaged accuracy for evaluation.

Oulu-CASIA: We also consider for experiments the Oulu-CASIA dataset, which is a little bit more challenging than CK+. The dataset contains 480 image sequences of six basic emotion classes (including An, Di, Fe, Ha, Sa and Su) under normal illumination conditions. Each sequence begins with a neutral expression and ends with the peak expression. The same as in CK+, a 10-fold cross validation is performed to evaluate various FER methods.

MMI: The third dataset used for experiments, MMI, is consist of 205 image sequences of the six basic emotion classes. Unlike CK+ and Oulu-CASIA, in which each sequence ends at the peak expression, the peak expressions in MMI are located in the middle of the sequences. The location of the peak frame is not provided as a prior information, which is usually the case for real-world videos.

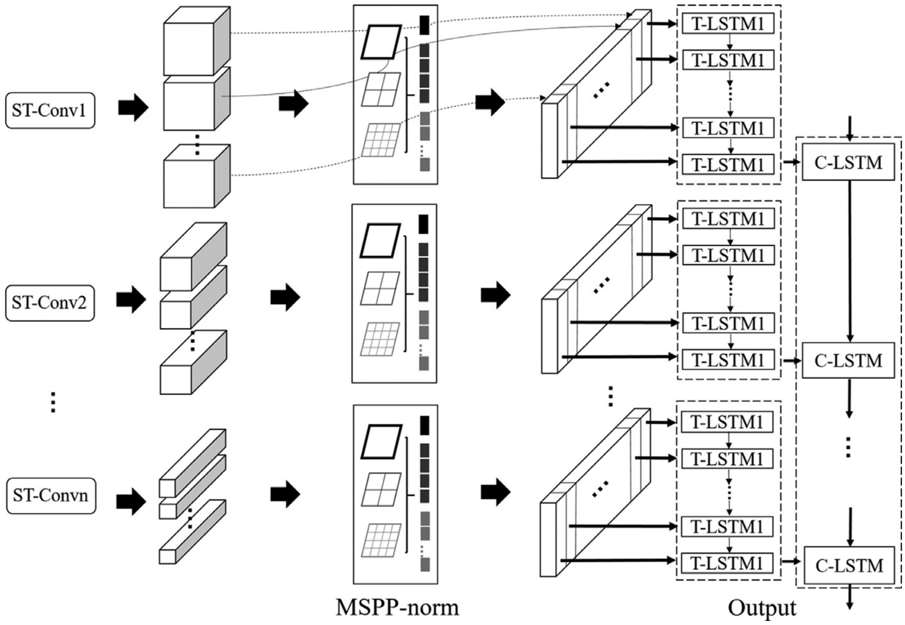


Fig. 4. Architecture of the proposed Nested LSTM.

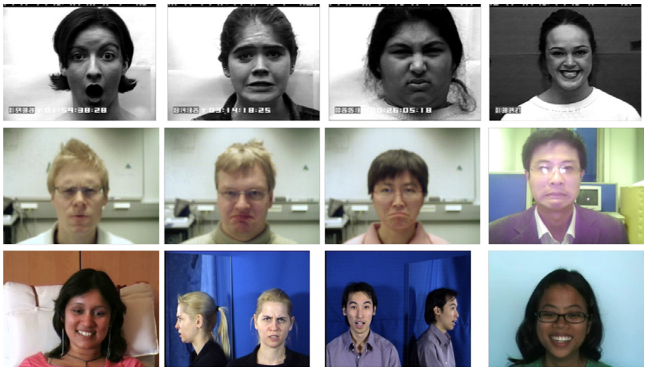


Fig. 5. Examples of the images used in our experiments. Top: CK+; Middle: Oulu-CASIA; Bottom: MMI.

Table 1
The number of image sequences in each of the six basic emotion classes: anger(An),disgust(Di), fear(Fe), happiness(Ha), sadness(Sa) and surprise(Su).

	An	Di	Fe	Ha	Sa	Su	All
CK+	45	59	25	69	28	83	309
Oulu	80	80	80	80	80	80	480
MMI	32	31	28	42	32	40	205

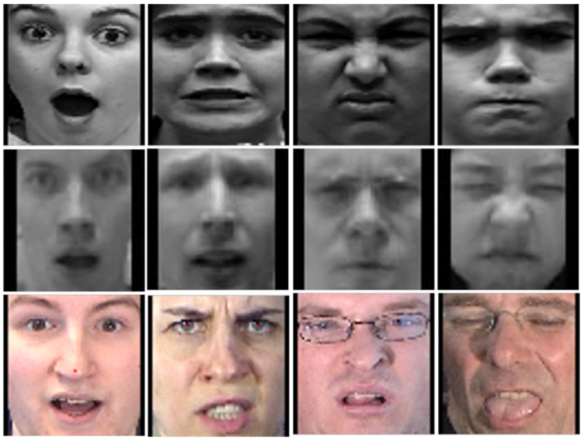


Fig. 6. Some examples of the face detection results in CK+ (top), Oulu-CASIA (middle) and MMI (bottom).

To obtain unbiased evaluation results, we perform a 10-fold cross validation in the same way as in CK+ and Oulu-CASIA.

BP4D: The last dataset used for experiments, BP4D, is divided into a fixed set of training, development and test data. In total, the training partition contains 75,586 images, the development contains 71,261 images and the test contains 75,726 images. Each of these images in BP4D are annotated with 11 Action Units. Unlike above datasets, BP4D contains large number of annotated images which benefits deep learning algorithms and provides a good platform for a fair evaluation due to a fixed training and test set.

Fig. 5 shows some examples sampled from the above four datasets, and Table 1 summarizes the number of sequences in each of the six emotion classes. To better perform FER, we need to use several pre-processing techniques, mainly including video normalization, face detection and data augmentation.

Video normalization: Since the length of the image sequences is variable, but the dimension of the inputs for a neural network is usually fixed, the normalization along the time axis is required as input for neural networks. For the sequences in CK+ and Oulu-CASIA , which have respectively averaged lengths of 18 and 22, we make each sequence into the average length via either uniform sampling (for the sequences longer than the average) or replicating the last frame (for the sequences shorter than the average). Regarding the MMI dataset, which is based on video, we convert the videos into the image sequences by uniformly selecting 10 frames per second, and normalize the sequences into a fixed length of 22 in the same way as in CK+ and Oulu-CASIA.

Face detection: We utilize the Multi-Task Cascaded Convolutional Network (MTCCN) [52] to obtain the coordinates of two eyes at first, then determine the final rectangular face by keeping the distances between two eyes invariable. Finally, we turn a rectangle into a square through zero padding and resize the square to 64×64 (see Fig. 6).

Data augmentation: Facial expression datasets, e.g., CK+, Oulu-CASIA and MMI, often contain only hundreds of image sequences. However, a typical deep neural network has many parameters, and this will make a deep network prone to overfitting. To handle this issue, we first flip each image sequence horizontally so as to double the number of sequences. Then we rotate each image by an angle in $\{-7.5^\circ, -5^\circ, -2.5^\circ, 2.5^\circ, 5^\circ, 7.5^\circ\}$, resulting in a new dataset which is 14 times as big as the original one. Such a data augmentation process can not only make the learnt model robust against the slight rotational changes of the input images, but also broaden the number of training samples so as to avoid overfitting.

4.2. Experimental data

To evaluate the performance of the proposed STC-NLSTM, we compare it with 12 prevalent FER methods, including 3DCNN-DAP [49], 3DSIFT [53], ARDfee [54], CSPL [20], DTAGN [16], FN2EN [55], IDT+FV [56], LOmo [43], PPDN [26], DCPN [57], STM-ExpLet [17] and ST-RBM [58].

In addition, to further investigate the effectiveness of the proposed Nested LSTM, we design four baselines by amending the architecture of STC-NLSTM:

- *STC (i.e., 3DCNN)*: This baseline is created by simply removing the T-LSTM and C-LSTM modules from the architecture of STC-NLSTM, and the outputs of the last convolutional layer of 3DCNN are taken as inputs to the softmax classifier so as to obtain the final classification results.
- *STC-LSTM*: This baseline is constructed by replacing the T-LSTM and C-LSTM modules with a traditional LSTM. Namely, the outputs of the last convolutional layer are taken as inputs to LSTM, which produces the final feature vectors for classification.
- *STC-SLSTM*: This baseline is similar to STC-LSTM. The only difference is that the outputs of all convolutional layer are taken as inputs to LSTM by sum fusion, which computes the sum of two feature maps at the same spatial locations and channels.
- *DenseNet*: This baseline [59] is a simpler and more efficient network compared to Inception networks, which also utilizes the middle latent representation. We compared the DenseNet to the other FER methods under the standard setting [26], which uses the strong expressions in each sequence(e.g., the last one to three frames) for training and testing. Because the DenseNet method is only based on the static image, we train this baseline following to [57].

For the 12 previously proposed baselines, their results are directly quoted from the original reports. For STC, STC-LSTM and STC-SLSTM, we obtain their classification results using the same parametric configuration as STC-NLSTM.

As usual, we denote a deep network by a sequence of letters and numbers, e.g., I(64,64,22)-C(3,64)-BN-P2-FC18-S6, where I(64,64,22) means the $64 \times 64 \times 22$ input image sequences, C(3,64) is a convolutional layer with 64 filters of 3×3, BN standards for the operation of batch normalization, P2 is a 2×2 max pooling layer, FC refers to a fully connected layer, and S6 denotes a softmax layer with six outputs. For simplicity, the architecture of the 3DCNN used in our STC-NLSTM is configured as I(64,64,18)-C(3,64)-BN-P2-C(3,64)-BN-P2-C(3,64)-C(3,64)-P2-C(3,64)-FC18, and a two-level LSTM architecture is used to construct the T-LSTM and C-LSTM modules. The stride of each layer is 1 with the exception of the pooling layer, which has a stride value of 2. Table 2 details the configurations of the network architecture.

Our model is implemented based on the TensorFlow library [60] and trained on four GeForce Titan X (pascal) GPU with 12GB memory. The weights of the network are initialized randomly using the “xaiver” procedure [61]. We first set the learning rate as 0.0025 and train the network until 300 iterations, and then fine-tune the network by setting the learning rate to 0.000025 and running 200 iterations. In all the experiments, the weight decay parameter is consistently set as 0.0015.

Table 2
Detailed configurations of the network for CK+.

Type	Patch size/stride	Input size
conv1	$3 \times 3 \times 3/1 \times 1 \times 1$	$18 \times 64 \times 64 \times 3$
mspp1	[8,4,2]	18×5376
pool1	$1 \times 2 \times 2/1 \times 2 \times 2$	$18 \times 32 \times 32 \times 64$
conv2	$3 \times 3 \times 3/1 \times 1 \times 1$	$18 \times 32 \times 32 \times 64$
mspp2	[8,4,2]	18×5376
pool2	$1 \times 2 \times 2/1 \times 2 \times 2$	$18 \times 16 \times 16 \times 64$
conv3	$3 \times 3 \times 3/1 \times 1 \times 1$	$18 \times 16 \times 16 \times 64$
mspp3	[8,4,2]	18×5376
conv4	$3 \times 3 \times 3/1 \times 1 \times 1$	$18 \times 16 \times 16 \times 64$
mspp4	[8,4,2]	18×5376
pool4	$1 \times 2 \times 2/1 \times 2 \times 2$	$18 \times 8 \times 8 \times 64$
conv5	$3 \times 3 \times 3/1 \times 1 \times 1$	$18 \times 8 \times 8 \times 64$
mspp5	[8,4,2]	18×5376

Table 3
Classification accuracies on CK+. The numbers for STC, STC-LSTM, STC-SLSTM and STC-NLSTM are averaged from 40 trails.

Method	Average accuracy
3DCNN-DAP [49]	92.4
STM-ExpLet [17]	94.2
LOmo [43]	95.1
IDT+FV [56]	95.8
FN2EN [55]	96.9
DTAGN [16]	97.3
ARDfee [54]	98.7
PPDN [26]	99.3
DCPN [57]	99.6
STC	98.9
STC-LSTM	99.3
STC-SLSTM	99.4
DenseNet	97.6
STC-NLSTM	**99.8(±0.2)**

Table 4
Confusion matrix of STC-NLSTM for CK+. The labels in the leftmost and topmost columns denote the ground truth and prediction results, respectively.

	An	Di	Fe	Ha	Sa	Su
An	100	0	0	0	0	0
Di	0.15	99.68	0	0	0.17	0
Fe	0	0	99.71	0	0.29	0
Ha	0	0	0.11	99.89	0	0
Sa	0	0.29	0.57	0	99.14	0
Su	0	0	0	0	0	100

4.3. Experimental results

4.3.1. Results on CK+

For fair comparison, we follow [20] to use 10-fold cross-validation and repeat the procedure 4 times, resulting in 40 trials in total. Table 3 shows the comparison results of various FER methods. Since the expressions in this dataset are easy to classify, several methods obtain superior classification results. In particular, as we can see from Table 4, our STC-NLSTM can achieve an accuracy near 100%. It performs well in anger and surprise, but for sadness , it is easy to be confused with disgust and fear. Fig. 7 compares STC-NLSTM with STC, STC-LSTM and STC-SLSTM on each of the six basic emotion classes. It can be seen that STC-NLSTM

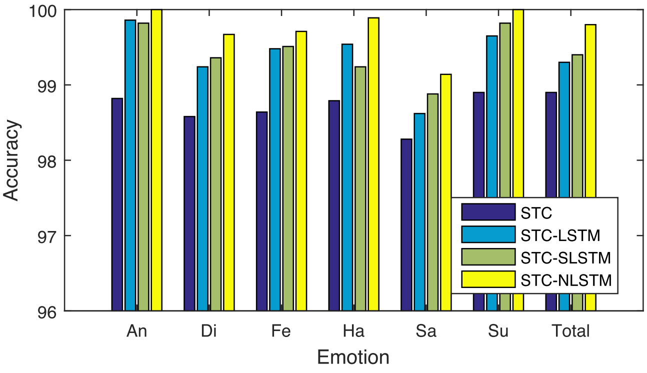


Fig. 7. Comparing STC-NLSTM with STC, STC-LSTM and STC-SLSTM on each of the six emotion classes in CK+.

Table 5
Classification accuracies on Oulu-CASIA. The numbers for STC, STC-LSTM, STC-SLSTM and STC-NLSTM are obtained by averaging the accuracies from 40 trails.

Method	Average accuracy
STM-ExpLet [17]	74.59
DTAGN [16]	81.64
LOmo [43]	82.10
PPDN [26]	84.59
DCPN [57]	86.23
FN2EN [55]	87.71
ARDfee [54]	89.60
STC	84.72
STC-LSTM	88.98
STC-SLSTM	90.12
DenseNet	87.28
STC-NLSTM	**93.45(± 0.43)**

Table 6
Confusion matrix of STC-NLSTM for Oulu-CASIA. The labels in the leftmost and topmost columns denote the ground truth and prediction results, respectively.

	An	Di	Fe	Ha	Sa	Su
An	89.82	6.20	0.75	0	3.23	0
Di	1.38	95.20	0.30	0.95	2.17	0
Fe	0	0	96.14	0.50	0.65	2.71
Ha	0	0.90	3.83	94.78	0.49	0
Sa	4.4	2.38	0.56	0	92.66	0
Su	0	0	3.95	0	0	96.05

performs consistently better than STC, STC-LSTM and STC-SLSTM, which confirms the effectiveness of our Nested LSTM.

4.3.2. Results on Oulu-CASIA

Table 5 shows the comparison results, and Table 6 gives the confusion matrix produced by our STC-NLSTM method on the Oulu-CASIA dataset. This dataset is more difficult to classify than CK+. In the cases of disgust, fear, happiness and surprise, the performance is good, but the performance for anger and sadness is slightly poor. As we can see, STC-NLSTM achieves an averaged accuracy of 93.45%, which is 4.2% higher than the 89.6% accuracy produced by the most close baseline, ARDfee. This illustrates the superiorities of STC-NLSTM over the state-of-the-art FER methods. Fig. 8 shows that STC-NLSTM is distinctly better than STC, STC-LSTM and STC-SLSTM on all expression classes except the class “happy”. It is shown that that FER can benefit a lot from the modeling of the multi-level features encoded in each convolutional layer.

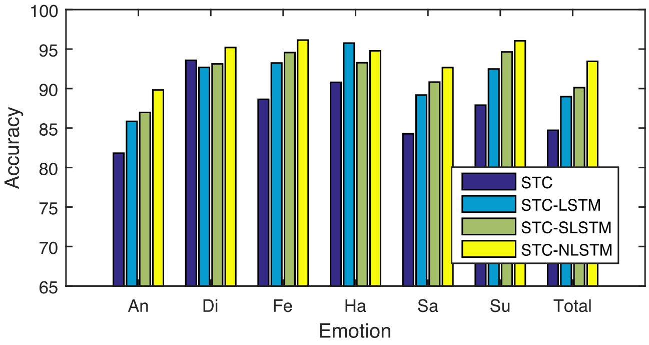


Fig. 8. Comparing STC-NLSTM with STC, STC-LSTM and STC-SLSTM on each of the six emotion classes in Oulu-CASIA.

Table 7
Classification accuracies on MMI. The numbers for STC, STC-LSTM, STC-SLSTM and STC-NLSTM are averaged from 40 trails.

Method	Average accuracy
3DCNN-DAP [49]	63.4
3DSIFT [53]	64.39
DTAGN [16]	70.24
CSPL [20]	73.53
STM-ExpLet [17]	75.12
ST-RBM [58]	81.63
STC	74.84
STC-LSTM	80.39
STC-SLSTM	81.92
DenseNet	77.68
STC-NLSTM	**84.53(± 0.67)**

Table 8
Confusion matrix of STC-NLSTM for MMI. The labels in the leftmost and topmost columns denote the ground truth and prediction results, respectively.

	An	Di	Fe	Ha	Sa	Su
An	83.24	9.04	0	5.36	1.24	1.12
Di	6.72	88.21	0	2.74	2.33	0
Fe	4.34	0	81.24	1.23	1.56	11.63
Ha	3.62	0	3.16	93.22	0	0
Sa	1.55	1.12	9.18	1.18	85.77	1.20
Su	2.64	0	8.66	3.41	0	85.29

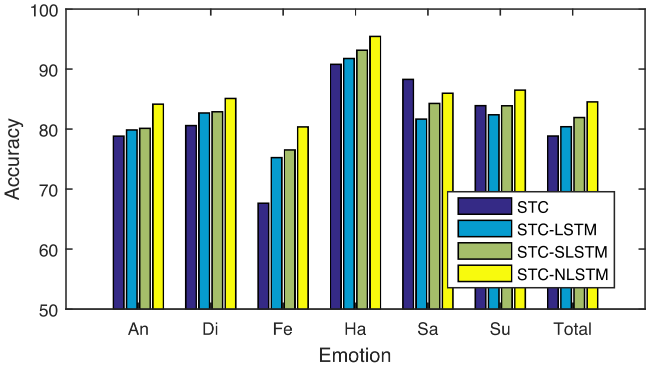


Fig. 9. Comparing STC-NLSTM with STC, STC-LSTM and STC-SLSTM on each of the six emotion classes in MMI.

4.3.3. Results on MMI

As shown in Table 7, the STC-NLSTM method can distinctly outperform previous state-of-the-art methods on the MMI dataset. Table 8 shows the confusion matrix produced by STC-NLSTM. Actually, the averaged accuracy by STC-NLSTM reaches 84.53% (see Table 7), which is 2.9% better than ST-RBM, and which archives the best performance among the previous reports. Fig. 9 shows that

Table 9
Performance (F1 scores) comparison on BP4D Test set.

Method	F1 Scores
LGBP [40]	0.44
GDNN [25]	0.48
DLE [62]	0.51
CNN+BLSTM [25]	0.52
STC-NLSTM	**0.58**

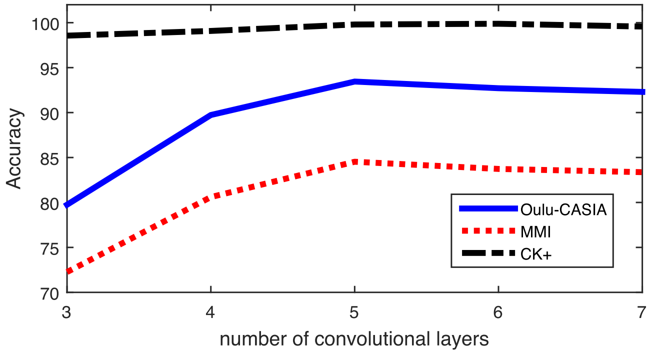


Fig. 10. Plotting the classification accuracy as a function of the number of convolutional layers in STC-NLSTM.

STC-NLSTM is distinctly better than STC-SLSTM on all the emotion classes, which is same as the results on the other three datasets.

4.3.4. Results on BP4D

Since the BP4D dataset has the training set and testing set, we do not need to use 10-fold cross-validation. Different from the above three datasets, the BP4D dataset is bigger than them. Table 9 shows the experimental results. It can be seen that the STC-NLSTM obviously outperforms previous the state-of-the-art methods. This result supports the conclusion that our STC-NLSTM can also achieve the good performance on a large scale datasets.

4.3.5. Influences of the number of layers

The above results illustrate that Nested LSTM plays a crucial role in our proposed method. To be more clear, we shall investigate the influences of the number of the convolutional layers contained in the architecture of STC-NLSTM. Fig. 10 shows the results. It can be seen that the classification accuracy gradually increases as the enlargement of the layer number, reaching the maximum at 5 convolutional layers. Since the datasets are not large, the performance drops while the number of layers exceeds 5. Regarding why the CK+ dataset is not so sensitive to the number of layers, the reason is that the dataset is easy to classify (see Table 3).

5. Conclusion

In this paper, we proposed a novel method termed STC-NLSTM for FER. Unlike most of the existing deep learning based methods, which obtain the classification results based on the outputs of the last fully-connected layer, STC-NLSTM aims to taken into account the multi-level features encoded in the intermediate layers of the network. To achieve this, the architecture of STC-NLSTM is designed to involve three major components: 3DCNN, T-LSTMs and C-LSTM. Each component is devised carefully to own a specific ability. The 3DCNN module plays the role of extracting the spatio-temporal convolutional features of facial expressions. The T-LSTM modules take charge of capturing the temporal dynamics that depict the facial appearance variations in temporal domain, and the C-LSTM is responsible for seizing the multi-level features encoded in the individual convolutional layers of the network. All the three components are integrated into an end-to-end network so as to cooperate seamlessly with each other. Experiments on four public datasets demonstrated that STC-NLSTM is superior to the state-of-the-art methods.

Acknowledgements

The work of Qingshan Liu is supported by National Natural Science Foundation of China (NSFC) under Grant61532009. The work of Guangcan Liu is supported in part by NSFC under grants 61622305 and 61502238, and in part by the Natural Science Foundation of Jiangsu Province of China (NSFJPC) under Grant BK20160040.

References

[1] P. Ekman, W.V. Friesen, Constants across cultures in the face and emotion, J. Pers. Soc. Psychol. 17 (2) (1971) 124–129.
[2] X. Li, G. Mori, H. Zhang, Expression-invariant face recognition with expression classification, in: The Canadian Conference on Computer and Robot Vision, 2006, pp. 77–84.
[3] J. Deng, Y. Zhou, S. Zafeiriou, Marginal loss for deep face recognition, in: Proceedings of the CVPRW, 4, 2017.
[4] J. Deng, J. Guo, S. Zafeiriou, Additive angular margin loss for deep face recognition, CoRR (2018) arXiv:1801.07698.
[5] J. Deng, S. Cheng, N. Xue, Y. Zhou, S. Zafeiriou, UV-GAN: Adversarial Facial UV Map Completion for Pose-Invariant Face Recognition, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[6] J. Deng, Q. Liu, J. Yang, D. Tao, M3 csr: Multi-view, multi-scale and multi-component cascade shape regression, Image Vision Comput. 47 (2016) 19–26.
[7] J. Yang, J. Deng, K. Zhang, Q. Liu, Facial shape tracking via spatio-temporal cascade shape regression, in: Proceedings of the ICCV Workshops, 2015, pp. 41–49.
[8] Q. Liu, J. Yang, J. Deng, K. Zhang, Robust facial landmark tracking via cascade regression, Pattern Recognit. 66 (2017) 53–62.
[9] J. Deng, Y. Sun, Q. Liu, H. Lu, Low rank driven robust facial landmark regression, Neurocomputing 151 (2015) 196–206.
[10] Q. Liu, J. Deng, D. Tao, Dual sparse constrained cascade regression for robust face alignment, IEEE Trans. Image Process. 25 (2) (2016) 700–712.
[11] Q. Liu, J. Deng, J. Yang, G. Liu, D. Tao, Adaptive cascade regression model for robust face alignment, IEEE Trans. Image Process. 26 (2) (2017) 797–807.
[12] S. Zafeiriou, G. Trigeorgis, G. Chrysos, J. Deng, J. Shen, The menpo facial landmark localisation challenge: a step towards the solution, in: Proceedings of the CVPR Workshops, 2017.
[13] J. Deng, Y. Zhou, S. Cheng, S. Zaferiou, Cascade multi-view hourglass model for robust 3d face alignment, in: Proceedings of the FG, IEEE, 2018, pp. 399–403.
[14] J. Deng, G. Trigeorgis, Y. Zhou, S. Zafeiriou, Joint multi-view face alignment in the wild, CoRR (2017) arXiv:1708.06023.
[15] P.V. Saudagare, D.S. Chaudhari, Facial expression recognition using neural network can overview, Int. J. Soft Comput. Eng. 2 (1) (2012) 238–241.
[16] H. Jung, S. Lee, J. Yim, S. Park, Joint fine-tuning in deep neural networks for facial expression recognition, in: Proceedings of the International Conference on Computer Vision, 2015, pp. 2983–2991.
[17] M. Liu, S. Shan, R. Wang, X. Chen, Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014a, pp. 1749–1756.
[18] P. Liu, S. Han, Z. Meng, Y. Tong, Facial expression recognition via a boosted deep belief network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014b, pp. 1805–1812.
[19] M. Valstar, M. Pantic, Induced disgust, happiness and surprise: An addition to the mmi facial expression database, Workshop on Emotion Corpora for Research on Emotion & Affect (2010) 65–70.
[20] D.N. Metaxas, J. Huang, B. Liu, P. Yang, Q. Liu, L. Zhong, Learning active facial patches for expression analysis, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2562–2569.
[21] K. Sikka, T. Wu, J. Susskind, M. Bartlett, Exploring bag of words architectures in the facial expression domain, in: Proceedings of the European Conference on Computer Vision, 2012, pp. 250–259.
[22] L. Zhong, Q. Liu, P. Yang, J. Huang, D.N. Metaxas, Learning multiscale active facial patches for expression analysis., in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2562–2569.
[23] L. Yann, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, Handwritten digit recognition with a back-propagation network, in: Proceedings of the Advances in Neural Information Processing Systems, 1990, pp. 396–404.
[24] S. Han, Z. Meng, A.S. Khan, Y. Tong, Incremental boosting convolutional neural network for facial action unit recognition, in: Proceedings of the Advances in Neural Information Processing Systems, 2016, pp. 109–117.

[25] S. Jaiswal, M. Valstar, Deep learning the dynamic appearance and shape of facial action units, in: Proceedings of the Applications of Computer Vision, 2016, pp. 1–8.

[26] X. Zhao, X. Liang, L. Liu, T. Li, Y. Han, N. Vasconcelos, S. Yan, Peak-piloted deep network for facial expression recognition, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 425–442.

[27] S. Ebrahimi Kahou, V. Michalski, K. Konda, R. Memisevic, C. Pal, Recurrent neural networks for emotion recognition in video, in: Proceedings of the ACM International Conference on Multimodal Interaction, 2015, pp. 467–474.

[28] Y. Liu, Y. Liu, Y. Liu, Y. Liu, Video-based emotion recognition using CNN-RNN and c3d hybrid networks, in: Proceedings of the ACM International Conference on Multimodal Interaction, 2016, pp. 445–450.

[29] P. Yang, Q. Liu, D.N. Metaxas, Exploring facial expressions with compositional features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2638–2644.

[30] A. Graves, M. Liwicki, S. Fernàndez, R. Bertolami, H. Bunke, J. Schmidhuber, A novel connectionist system for unconstrained handwriting recognition, IEEE Trans. Pattern Anal. Mach. Intell. 31 (5) (2009) 855–868.

[31] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[32] M. Wang, X. Liu, X. Wu, Visual classification by ℓ_1 -hypergraph modeling, IEEE Trans. Knowl. Data Eng. 27 (9) (2015) 2564–2574.

[33] M. Wang, C. Luo, R. Hong, J. Tang, J. Feng, Beyond object proposals: Random crop pooling for multi-label image recognition, IEEE Trans. Image Process. 25 (12) (2016) 5678–5688.

[34] H.L.M.W. Dan Guo, W. Zhou, Hierarchical lstm for sign language translation, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2018.

[35] T. Du, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: Proceedings of the International Conference on Computer Vision, 2016, pp. 4489–4497.

[36] L. Zhang, G. Zhu, P. Shen, J. Song, S.A. Shah, M. Bennamoun, in: Learning spatiotemporal features using 3dcnn and convolutional lstm for gesture recognition, 2017, pp. 3120–3128.

[37] G. Zhu, L. Zhang, P. Shen, J. Song, Multimodal gesture recognition using 3-d convolution and convolutional lstm, IEEE Access 5 (2017) 4517–4524.

[38] P. Lucey, J.F. Cohn, T. Kanade, J. Saragih, The extended Cohn-Kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression, in: Proceedings of the Computer Vision and Pattern Recognition Workshops, 2010, pp. 94–101.

[39] G. Zhao, X. Huang, M. Taini, S.Z. Li, M. PietikäInen, Facial expression recognition from near-infrared videos, Image Vision Comput. 29 (9) (2011) 607–619.

[40] M.F. Valstar, T. Almaev, J.M. Girard, G. Mckeown, Fera 2015 - second facial expression recognition and analysis challenge, in: Proceedings of the IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, 2016, pp. 1–8.

[41] S.A. Bargal, E. Barsoum, C.C. Ferrer, C. Zhang, Emotion recognition in the wild from videos using images, in: Proceedings of the ACM International Conference on Multimodal Interaction, 2016, pp. 433–436.

[42] A. Dhall, R. Goecke, J. Joshi, J. Hoey, T. Gedeon, Emotiw 2016: video and group-level emotion recognition challenges, in: Proceedings of the ACM International Conference on Multimodal Interaction, 2016, pp. 427–432.

[43] K. Sikka, G. Sharma, M. Bartlett, Lomo: latent ordinal model for facial analysis in videos, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5580–5589.

[44] Z. Yu, C. Zhang, Image based static facial expression recognition with multiple deep network learning, in: Proceedings of the ACM International Conference on Multimodal Interaction, 2015, pp. 435–442.

[45] B.K. Kim, H. Lee, J. Roh, S.Y. Lee, Hierarchical committee of deep CNNs with exponentially-weighted decision fusion for static facial expression recognition, in: Proceedings of the ACM International Conference on Multimodal Interaction, 2015, pp. 427–434.

[46] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceedings of the CoRR, 2014.

[47] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[48] A. Yao, D. Cai, P. Hu, S. Wang, L. Sha, Y. Chen, Holonet: towards robust emotion recognition in the wild, in: Proceedings of the ACM International Conference on Multimodal Interaction, 2016, pp. 472–478.

[49] M. Liu, S. Li, S. Shan, R. Wang, X. Chen, in: Deeply learning deformable facial action parts model for dynamic expression analysis, Asian Conf. Comput. Vis., 2014, pp. 143–157.

[50] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, in: Proceedings of the European Conference on Computer Vision, 2014, pp. 346–361.

[51] M. Liu, R. Wang, S. Li, S. Shan, Z. Huang, X. Chen, Combining multiple kernel methods on riemannian manifold for emotion recognition in the wild, in: Proceedings of the ACM International Conference on Multimodal Interaction, 2014, pp. 494–501.

[52] K. Zhang, Z. Zhang, Z. Li, Y. Qiao, Joint face detection and alignment using multitask cascaded convolutional networks, IEEE Signal Process. Lett. 23 (10) (2016) 1499–1503.

[53] P. Scovanner, S. Ali, M. Shah, A 3-dimensional sift descriptor and its application to action recognition, in: Proceedings of the 15th International Conference on Multimedia 2007, Augsburg, Germany, September 24–29, 2007, pp. 357–360.

[54] I. Ofodile, K. Kulkarni, C.A. Corneanu, S. Escalera, X. Baro, S. Hyniewska, J. Allik, G. Anbarjafari, Automatic recognition of deceptive facial expressions of emotion, in: Proceedings of the CoRR, 2017.

[55] H. Ding, S.K. Zhou, R. Chellappa, Facenet2expnet: regularizing a deep face recognition net for expression recognition (2017) 118–126.

[56] S. Afshar, A.A. Salah, Facial expression recognition in the wild using improved dense trajectories and fisher vector encoding, in: Proceedings of the Computer Vision and Pattern Recognition Workshops, 2016, pp. 1517–1525.

[57] Z. Yu, Q. Liu, G. Liu, Deeper cascaded peak-piloted network for weak expression recognition, Visual Comput. 6 (6–8) (2017) 1–9.

[58] S. Elaiwat, M. Bennamoun, F. Boussaid, A spatio-temporal RBM-based model for facial expression recognition, Pattern Recognit. 49 (C) (2015) 152–161.

[59] G. Huang, Z. Liu, L.V.D. Maaten, K.Q. Weinberger, Densely connected convolutional networks 4 (2017) 2261–2269.

[60] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, Tensorflow: large-scale machine learning on heterogeneous distributed systems, in: Proceedings of the CoRR, 2016.

[61] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, J. Mach. Learn. Res. 9 (2010) 249–256.

[62] A. Yce, H. Gao, J.P. Thiran, Discriminant multi-label manifold embedding for facial action unit detection, in: Proceedings of the IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, 2015, pp. 1–6.

Zhenbo Yu received his bachelor degree from the school of Information and Control, Nanjing University of Information Science and Technology, Nanjing, China, in 2016, where he is pursing the master degree. He took second place in 2015 and first place in 2016 in one major category of the ImageNet challenge, and got National Scholarship in 2017. His research interest is facial expression analysis.

Guangcan Liu received the bachelor's degree in mathematics and the Ph.D. degree in computer science and engineering from Shanghai Jiao Tong University, Shanghai, China, in 2004 and 2010, respectively. He was a Post-Doctoral Researcher with the National University of Singapore, Singapore, from 2011 to 2012, the University of Illinois at Urbana-Champaign, Champaign, IL, USA, from 2012 to 2013, Cornell University, Ithaca, NY, USA, in 2014. Since 2014, he has been a Professor with the School of Information and Control, Nanjing University of Information Science and Technology, Nanjing, China. His research interests touch on the areas of machine learning, computer vision, and image processing.

Qinshan Liu is a Professor with the School of Information and Control Engineering, Nanjing University of Information Science and Technology, Nanjing, China. He received the Ph.D. degree from the National Laboratory of Pattern Recognition, Chinese Academy of Sciences, Beijing, China, in 2003 and the M.S. degree from Southeast University, Nanjing, China, in 2000. He was an Assistant Research Professor with the department of Computer Science, Computational Biomedicine Imaging and Modeling Center (CBIM), Rutgers University of New Jersey, Piscataway, NJ, USA. Before joining Rutgers University, from 2010 to 2011. Before he joined Rutgers University, he was an Associate Professor with the National Laboratory of Pattern Recognition. He was a recipient of the President Scholarship of the Chinese Academy of Sciences in 2003. His research interests include image and vision analysis, including face image analysis, graphand hypergraphbased image and video understanding, medical image analysis, and event-based video analysis.

Jiankang Deng is a Ph.D. candidate in the Intelligent Behaviour Understanding Group (IBUG), Department of Computing, Imperial College London. He is funded by the Imperial President's PhD Scholarships and his research interest is face image analysis.