Noname manuscript No.
(will be inserted by the editor)

Spatiotemporal Feature Learning Framework with 3DCNN and Convolutional LSTM

Guangming Zhu · Liang Zhang

Received: date / Accepted: date

Abstract Recent studies have demonstrated the power of neural networks for image-based applications. However, for video-based applications, such as action recognition, localization and gesture recognition, there still remain numerous open research questions. The key challenge in the video-based applications is to learn spatiotemporal features efficiently and effectively. In this paper, we present end-to-end deep architectures to learn spatiotemporal features for gesture recognition. Firstly, the proposed architectures learn short-term spatiotemporal features using a shallow 3D convolutional neural network (3DCNN), and then learn long-term spatiotemporal features using convolutional long-short-term-memory recurrent neural network (ConvLSTM). Based on these two steps, the proposed deep architectures can transform sequential video files into compact 2D spatiotemporal feature maps which could encode global temporal information and local spatial information simultaneously. We explore four such kinds of neural networks to learn the compact 2D spatiotemporal feature maps first, and then to learn higher-level spatiotemporal features further using 2DCNN or spatial pyramid pooling (SPP) for the final recognition. The objective of the proposed method is to keep the useful spatiotemporal correlation information of the video as much as possible through the whole process of feature learning. We evaluate the neural networks on the ChaLearn large-scale isolated gesture dataset (IsoGD), the Sheffield Kinect gesture (SKIG) dataset and the Montalbano gesture recognition dataset, and report the state-of-the-art performance.

Guangming Zhu
E-mail: gmzhu@xidian.edu.cn

Liang Zhang
E-mail: liangzhang@xidian.edu.cn

School of Computer Science and Technology, Xidian University

1 Introduction

Gestures, as a nonverbal body language, play a very important role in human's daily life. Gesture recognition aims at understanding the ongoing human gestures and is of great significance for human-robot/computer interaction, sign language recognition and virtual reality (Mitra and Acharya, 2007).

Effective and universal gesture recognition from videos is extremely difficult. This is partly due to the large gesture vocabularies with cultural diversity, various illumination conditions, out-of-vocabulary motions, inconsistent and non-standard behaviors among different performers, etc (Escalera et al, 2016). In addition, gestures have various time durations and involve different body parts. A small handful of gestures can be represented by a single posture of hands and arms, but most of the gestures are composed of a sequence of hand and arm postures. Therefore, learning effective spatiotemporal features is crucially important for robust gesture recognition. According to Tran et al (2015), there are four typical properties for effective spatiotemporal features of gestures: (i) *generic*, (ii) *compact*, (iii) *efficient* to compute, and (iv) *simple* to implement.

Inspired by the deep learning breakthroughs in image recognition (Krizhevsky et al, 2012; Simonyan and Zisserman, 2014b; Szegedy et al, 2015), lots of neural network based frameworks have been proposed to learn spatiotemporal features for human action/gesture recognition. *Two-Stream Convolutional Networks* (Simonyan and Zisserman, 2014a) learn spatial and temporal features separately. *Long-term Recurrent Convo-*

lutional Networks (LRCN) (Donahue et al, 2015) learn spatial and temporal features using Convolutional Neural Network (CNN) and Long-Short-Term-Memory network (LSTM) successively. Tran et al (2015) constructed a deep 3D ConvNet to learn spatiotemporal features directly and achieved the best performance on different types of video analysis tasks. Molchanov et al (2016) proposed to first learn spatiotemporal features on each clip using 3DCNN, and then to fuse the spatiotemporal features over the whole video using recurrent neural networks (RNN). Obviously, 3DCNN is superior to learn spatiotemporal features for gesture recognition. However, RNN/LSTM based networks are more suitable to encode long-term temporal information, especially for various-length videos. Although Molchanov et al (2016) proposed to combine 3DCNN and RNN, in which the key step is that fully connected spatiotemporal features are transferred into RNN, this makes the spatial correlation information lost in the RNN stage.

In this paper, we propose to first learn short-term spatiotemporal features using a shallow 3DCNN, and then to learn long-term spatiotemporal features further with convolutional LSTM (ConvLSTM). Such a deep architecture can transform sequential video files into 2D spatiotemporal feature maps. At the same time, spatiotemporal correlation information is kept through the whole feature map extraction, especially in the process of 3DCNN to ConvLSTM. Furthermore, we explore different neural networks to learn higher-level spatiotemporal features further from the learnt 2D spatiotemporal feature maps. 2DCNN and the Spatial Pyramid Pooling (SPP) are evaluated respectively on the learnt 2D feature maps for gesture recognition.

In brief, our contributions in this paper can be summarised as follows:

- Compact 2D spatiotemporal feature maps of sequential gesture video are learnt using 3DCNN and convolutional LSTM conjunctively. The 2D feature maps can encode the global temporal information and local spatial information. Spatiotemporal correlation information is kept through the whole feature map learning process.
- The proposed deep architecture can transform video files into 2D spatiotemporal feature maps. This transformation makes the deep architecture more extensible to utilize the state-of-the-art 2DCNN to learn the higher-level spatiotemporal features for gesture recognition.
- To the best of our knowledge, this is the first end-to-end framework to learn 2D spatiotemporal feature maps using 3DCNN and ConvLSTM, and then to learn higher-level spatiotemporal features using 2DCNN for the final gesture recognition.

2 Related Work

Learning spatiotemporal features is crucial for effective human action/gesture recognition. Several deep neural networks have been proposed recently (Herath et al, 2017). However, gesture recognition has significant differences from action recognition. One obvious difference is that backgrounds may be an effective clue for action recognition, but in contrast can be a challenging factor for gesture recognition. For example, scene backgrounds can help recognize human actions, especially the sports in UCF101 (Soomro et al, 2012), however they may bring negative impact on gesture recognition performance. In fact, gestures focus more on the movement of hands and arms. Thus, two-stream ConvNets (Simonyan and Zisserman, 2014a) and their variants (Wang et al, 2016a; Feichtenhofer et al, 2016) may obtain the state-of-the-art performance on the HMDB51 (Kuehne et al, 2011) and UCF101 datasets, but they fail to achieve a similar performance in the case of gesture recognition. Another approach is to learn spatial and temporal features successively, such as LRCN (Donahue et al, 2015). However, Pigou et al (2015) demonstrated that LRCN-style networks are not optimal, while bidirectional recurrence and temporal convolutions can improve gesture recognition performance significantly. Recently, the huge success of 2DCNN on image recognition has encouraged researchers to transform video files into particular 2D image files, with the aim to use the state-of-the-art 2DCNN networks for gesture recognition (Wang et al, 2016b). But, handcrafted transformation methods have inherent deficiency on adaptive learning. In this paper, a deep architecture will be described, which can learn adaptively to transform gesture video files into 2D spatiotemporal feature maps.

Tran et al (2015) constructed a deep 3D ConvNet to learn spatiotemporal features directly and achieved the best performance on different types of video analysis tasks. Inspired by Tran et al (2015), 3DCNN-based neural networks obtained the remarkable performances on gesture recognition (Escalante et al, 2016). In the past 2016 ChaLearn LAP Large-scale Isolated/ Continuous Gesture Recognition Challenges (Wan et al, 2016), 3DCNN demonstrated excellent performance (Camgoz et al, 2016; Zhu et al, 2016; Li et al, 2016a; Duan et al, 2016). However, 3DCNN uses the stacked pooling layers to reduce the spatial and temporal sizes of feature maps, which requires more layers or larger kernel and stride sizes when the networks have long inputs. This weakness drives researchers to take full use of the advantages of 3DCNN and RNN/LSTM, and combine them to learn local and global spatiotemporal features suc-

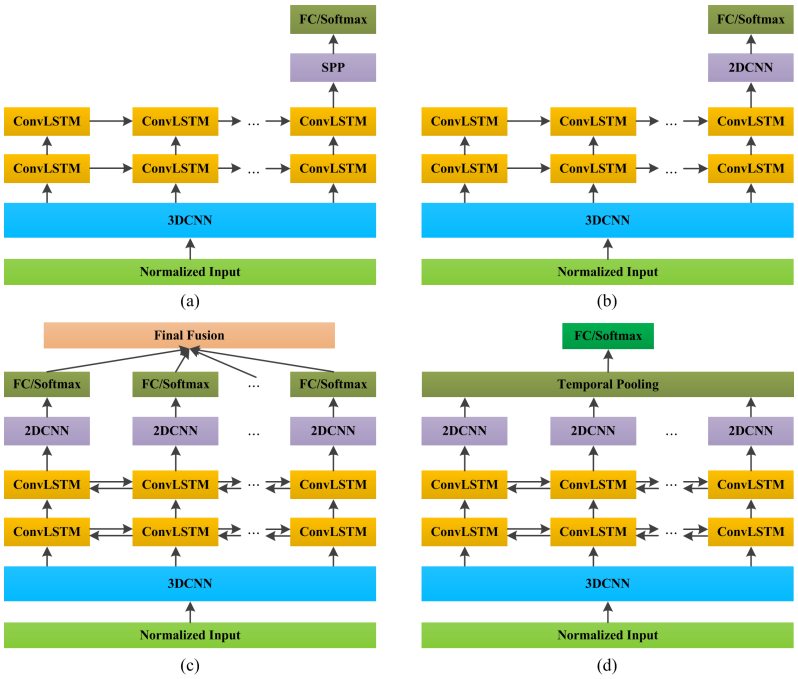

Fig. 1 Overview of the proposed deep architectures. 3DCNN and ConvLSTM are utilized to learn the short-term and long-term spatiotemporal features successively, and then 2DCNN and SPP are used to learn higher-level spatiotemporal features from the learnt 2D long-term spatiotemporal feature maps for the final gesture recognition.

cessively (Molchanov et al, 2016; Zhu et al, 2017; Baccouche et al, 2011).

Generally, the fully-connected features of 3DCNN or 2DCNN are transferred into RNN/LSTM networks (Donahue et al, 2015; Molchanov et al, 2016; Baccouche et al, 2011). The spatial correlation information is lost in the input-to-state and state-to-state transitions of RNN/LSTM due to the fully-connected features. ConvLSTM (Xingjian et al, 2015) is originally proposed for precipitation nowcasting, in which the spatial correlation information is encoded explicitly in the input-to-state and state-to-state transitions. We, therefore, propose to first learn short-term spatiotemporal features using a shallow 3DCNN, and then learn long-term spatiotemporal features using ConvLSTM. The prominent character of ConvLSTM layers is that they do not shrink the spatial size, but learn the global temporal correlation information completely. The combination of the shallow 3DCNN and the ConvLSTM can transform video files into compact 2D spatiotemporal feature maps, which can encode the global temporal and local spatial information effectively, efficiently and simultaneously. This transformation makes it possible to utilize 2DCNN or SPP for the final gesture recognition.

3 Deep Architectures

In this section, four types of proposed deep architectures are described briefly. An overview of the deep architectures[1] is illustrated in Fig. 1. Note that all the architectures use 3DCNN and ConvLSTM to learn 2D spatiotemporal feature maps first, and then four kinds of different neural networks are evaluated to verify the effectiveness of the learnt 2D feature maps. These proposed deep architectures can be divided into two components: **a)** *2D spatiotemporal feature map learning* and **b)** *classification based on the 2D feature maps*. The former learns 2D spatiotemporal feature maps from the normalized inputs using 3DCNN and ConvLSTM, and thus can transform sequential video files into 2D feature maps. The latter learns higher-level spatiotemporal features further using 2DCNN or SPP for the final gesture recognition.

The proposed deep architectures do not require all input sequences of the same length. The input sequences are preprocessed to make them of the same length for simplicity during training. The uniform sampling with temporal jitter method proposed in our previous work (Zhu et al, 2017) is utilized for the input preprocessing.

[1] The deep architecture in Fig. 1a has been evaluated in our previous work (Zhu et al, 2017).

3.1 2D Spatiotemporal Feature Map Learning

Three facts are taken into consideration when we aim to construct an effective deep architecture to learn the compact 2D spatiotemporal feature maps from videos: **a)** 3DCNN is a representative and outstanding deep architecture for spatiotemporal feature learning; **b)** RNN/LSTM networks are more suitable for long-term temporal information learning; **c)** Spatiotemporal correlation information plays an important role for gesture recognition. Our original idea is dividing learning process of the spatiotemporal feature maps into two phases, first is local or namely short-term spatiotemporal feature learning, and the second is long-term. Therefore, we propose to use 3DCNN and ConvLSTM for spatiotemporal feature learning. 3DCNN is designed to learn local or short-term spatiotemporal features, so it does not need to be deep. ConvLSTM is designed to learn global or long-term spatiotemporal features. The spatiotemporal correlation information is encoded during the recurrent process.

3.1.1 3DCNN Component

The 3DCNN component of the proposed deep architecture is similar in design to the C3D model (Tran et al, 2015). Based on our intution, the 3DCNN does not need to be deep; only four Conv3D layers are therefore constructed, as displayed in Fig. 2. The kernel size of each Conv3D layer is $3 \times 3 \times 3$ with stride $1 \times 1 \times 1$. The 3DCNN component is designed to learn local spatiotemporal features, thus only two pooling layers are used as illustrated in Fig. 2. Based on the setting of the two pooling layers, the spatial size and the temporal length are only shrunk by a ratio of 4 and a ratio of 2 respectively. This makes the 3DCNN learn the short-term spatiotemporal features effectively and efficiently. Batch normalization (Ioffe and Szegedy, 2015) can allow using much higher learning rates and being less careful about initialization, so it is utilized to optimize our networks.

3.1.2 Convolutional LSTM Component

Generally, the fully-connected LSTM, which takes vectorized features as input, is used to learn temporal features (Donahue et al, 2015; Pigou et al, 2015). The limitation of the vectorization is that it results in the loss of spatial correlation information during the recurrence. Nevertheless, position transformation of hands and arms in the spatial domain plays an important role for gesture recognition. Therefore, the ConvLSTM (Xingjian et al, 2015) is used in our proposed neural network to learn the long-term spatiotemporal features. The convolution and recurrence operations in the input-to-state and state-to-state transitions can take full use of the spatiotemporal correlation information.

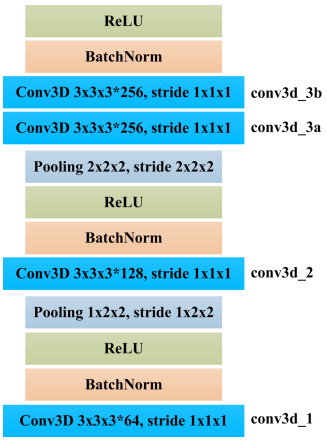

Fig. 2 The shallow 3DCNN component to learn short-term spatiotemporal features

Formally, the inputs $X_1, ..., X_t$, the cell states $C_1, ..., C_t$, the hidden states $H_1, ..., H_t$ and the gates i_t, f_t, o_t of ConvLSTM are all 3D tensors. Let "*" denotes the convolution operator, and let "o" denotes the Hadamard product. The ConvLSTM can be formulated as:

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + b_i), \quad (1)$$

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + b_f), \quad (2)$$

$$o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + b_o), \quad (3)$$

$$C_t = f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c), \quad (4)$$

$$H_t = o_t \circ \tanh(C_t), \quad (5)$$

where σ is the sigmoid function, $W_{x\sim}$ and $W_{h\sim}$ are 2-d convolution kernels. The convolutions in the ConvLSTM have kernel size 3×3 with stride 1×1 . "Same-Padding" is used to ensure that the spatiotemporal feature maps in each ConvLSTM layer have the same spatial size.

Two different types of two-layer ConvLSTM networks are constructed as illustrated in Fig 1: a) **unidirectional ConvLSTM** and b) **bidirectional ConvLSTM**. In our architecture, two-layer ConvLSTM networks are designed to learn global sptiotemporal features. Therefore, the learnt 2D spatiotemporal feature maps are only transferred into the 2DCNN or SPP networks at the last recurrent step of the unidirectional ConvLSTM (as illustrated in Fig. 1a and Fig. 1b), but at each recurrent step of the bidirectional ConvLSTM (as illustrated in Fig. 1c and Fig. 1d).

Formally, given the input $I = \{I_t \in \mathbb{R}^{w \times h \times 3} | t = 1, 2, ..., T_I\}$ where w and h are the spatial size of the inputted video, and T_I is the frame count of normalized

input, the 2D spatiotemporal feature maps ($STFM$) can be denoted as

$$STFM = ConvLSTM(3DCNN(I)), \tag{6}$$

where

$$STFM = \{STFM_t \in \mathbb{R}^{\frac{w}{m} \times \frac{h}{m} \times c} | t = T_N\}, \tag{7}$$

for the unidirectional ConvLSTM, and

$$STFM = \{STFM_t \in \mathbb{R}^{\frac{w}{m} \times \frac{h}{m} \times c} | t = 1, 2, ..., T_N\}, \tag{8}$$

for the bidirectional ConvLSTM. T_N is the recurrent step count of ConvLSTM ($T_N = T_I/2$ in this implementation), and m is the shrink coefficient on the spatial domain ($m = 4$ in this implementation).

Actually, each $STFM_t$ has encoded the global temporal information and local spatial information of the input video I simultaneously. Each $STFM_t$ keeps the same spatial size as the outputs of the 3DCNN component and just shrinks the temporal length to 1. This means that the 3DCNN and ConvLSTM components transform the input sequential video files into 2D feature maps. This is very significant, because the deep architecture can transform various-length sequential video files into 2D spatiotemporal feature maps with large spatial size. Based on this fact, the state-of-the-art 2DCNN structures can be used further for higher-level spatiotemporal feature learning. This is a novel idea for dealing with video sequences.

3.2 Classification based on the 2D Feature Maps

Generally, video files need to be decoded into separate image files (Li et al, 2016b) or encoded into special images (Wang et al, 2015) when 2DCNN is employed in video-based applications. In this paper, we propose a new deep architecture to encode video files into 2D feature maps, which enables 2DCNN to be used in video-based applications in an alternative way.

3.2.1 Higher-level Spatiotemporal Feature Learning

Since the 2D spatiotemporal feature maps still have large spatial size, dimensionality reduction is necessary for final recognition. Two dimensionality reduction networks are therefore evaluated in this paper: a) **SPP** (as displayed in Fig. 1a) and b) **2DCNN** (as displayed in Fig. 1b-Fig. 1d).

A. SPP Component

As illustrated in Fig. 1a, SPP is inserted between the ConvLSTM layer and the final fully-connected layer

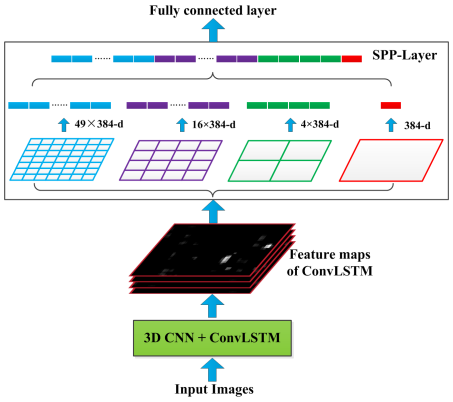

Fig. 3 The SPP component to reduce the dimensionality of features

to learn the high level feature and to reduce the feature dimensionality. Fig. 3 shows that the 4-level SPP is performed on each 2D spatiotemporal feature map. Besides the dimensionality reduction, SPP also extracts multi-scale features from the 2D feature maps, which can improve the recognition accuracy in some degree.

B. 2DCNN Component

A simple 2DCNN is also employed to reduce the dimensionality and to learn the higher-level spatiotemporal features, based on the learnt 2D spatiotemporal feature maps. Since the spatial size of inputs in our implementation is 112×112, the 2D spatiotemporal feature maps have a spatial size of 28×28. Therefore, only a shallow 2DCNN is constructed in this implementation. Nevertheless, deeper 2DCNN can also be used for different configurations or applications according to the resolution of the learnt 2D spatiotemporal feature maps. The 2DCNN component, displayed in Fig. 4, consists of three "Convolution-BatchNorm-ReLU" layers. The 2DCNN finally outputs the deeper spatiotemporal features which are 4096 dimensional after vectorization. Formally, the deeper spatiotemporal feature ($DSTF$) for the deep architecture in Fig. 1b can be represented as

$$DSTF_t = 2DCNN(STFM_t), \tag{9}$$

$$DSTF = \{DSTF_t \in \mathbb{R}^{4096} | t = T_N\}, \tag{10}$$

and the $DSTF$ for the deep architectures in Fig. 1c and Fig. 1d can be represented as

$$DSTF_t = 2DCNN(\overrightarrow{W}_{fw}\overrightarrow{STFM}_t + \overleftarrow{W}_{bw}\overleftarrow{STFM}_t), \tag{11}$$

$$DSTF = \{DSTF_t \in \mathbb{R}^{4096} | t = 1, 2, ..., T_N\}, \tag{12}$$

where $\overrightarrow{W}_{fw}$ and $\overleftarrow{W}_{bw}$ are the connection weights from the forward and backward layers of the bidirectional

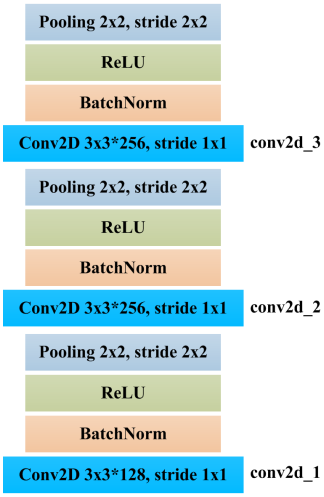

Fig. 4 The 2DCNN component to learn the higher-level spatiotemporal features

ConvLSTM to the conv2d_1 layer, and $\overrightarrow{STFM}_t$ and $\overleftarrow{STFM}_t$ are the forward and backward spatiotemporal feature maps learnt by the ConvLSTM respectively.

3.2.2 Classification

Obviously, the higher-level spatiotemporal features are learnt at each recurrent step of the deep architectures in Fig. 1c and Fig. 1d, thus the fusion over the recurrent steps is necessary for these two deep architectures. Generally, two fusion methods can be used after features extraction at each recurrent step: one is to calculate the loss of each recurrent step and minimize the cumulative loss (Veeriah et al, 2015), the other is to accumulate the outputs over the steps and minimize the final softmax loss (Du et al, 2015).

In this implementation, two fusion methods on $DSTF$ are evaluated for the deep architectures in Fig. 1c and Fig. 1d respectively. The first one is illustrated in Fig. 1d: a temporal pooling layer is used to fuse the $DSTF$ first, and then softmax classifier is used for classification. The classification functions can be denoted as

$$A = TPooling(DSTF_t|t = 1, 2, ..., T_N), \tag{13}$$

$$p(C_k) = \frac{e^{A_k}}{\sum_{i=0}^{C-1} e^{A_i}}, \tag{14}$$

where $TPooling(\cdot)$ is the pooling method of the temporal pooling layer, C is the category count of gestures. The other is illustrated in Fig. 1c: softmax classifier is used at each recurrent step first, and then the prediction results are fused over the steps. The classification functions can be denoted as

$$p_t(C_k) = \frac{e^{DSTF_t^k}}{\sum_{i=0}^{C-1} e^{DSTF_t^i}}, \tag{15}$$

$$p(C_k) = SFusion(p_t(C_k)|t = 1, 2, ..., T_N), \tag{16}$$

where $SFusion(\cdot)$ is the score fusion function, and $p_t(C_k)$ is the prediction probabilities at the recurrent step t. The cross-entropy loss function is used to learn the model parameters.

Multi-modal fusion on the prediction results of the RGB/Depth/Flow data can further be used to improve the prediction accuracy. The frequently used fusion method is **averaging**. Besides, we also evaluate fusing the higher-level spatiotemporal features of each modality, in which an **integration** strategy is used to combine the learnt features of each modality into larger feature vectors first, and then feed the feature vectors into linear SVM classifier for the final gesture recognition.

4 Experiments

We extensively evaluate the proposed deep architectures and the learnt compact 2D spatiotemporal features under various settings for the task of gesture recognition.

4.1 Datasets

Three public datasets are used to evaluate the performance of the proposed framework: the ChaLearn large-scale isolated gesture dataset (IsoGD) (Wan et al, 2016), the Sheffield Kinect Gesture dataset (SKIG) (Liu and Shao, 2013) and the Montalbano gesture dataset (Escalera et al, 2014).

IsoGD is a large-scale isolated gesture dataset which contains 47,933 RGB+D gesture videos divided into 249 kinds of gestures performed by 21 individuals. All the videos are divided into three mutually exclusive sets: the training, validation and test. The labels of the test set have not been released, thus the validation set is used to examine the various settings of the proposed framework.

SKIG contains 1,080 RGB+D videos of 10 kinds of gestures. All gestures are performed by 6 individuals with 3 kinds of hand postures under 2 illumination conditions and 3 backgrounds. Three-fold cross-validation is used to evaluate the proposed framework.

Montalbano contains 13,858 RGB+D video fragments of 20 Italian sign language gestures. All gestures are performed by 27 different individuals under diverse conditions. Each gesture is accompanied by a ground truth label as well as information about its start- and end-points. The user mask and skeleton data in Montalbano are not used in this implementation.

Besides the RGB and depth modalities, optical flow is also used to improve the prediction accuracy. The

BroxOpticalFlow method in OpenCV 2.4.23 is employed to extract the optical flow data from RGB videos.

4.2 Implementation

The proposed deep architectures are implemented[2] based on the platform Tensorflow-0.11 (Abadi et al, 2016), Tensorlayer-1.2.8 (Dong et al, 2017), and the implementation of ConvLSTM[3].

The deep architectures are trained from scratch on the large-scale IsoGD dataset, and then fine-tuned on the SKIG and Montalbano datasets respectively. Batch normalization layers make the training easier and faster. The learning rate is initialized as 0.01 and dropped to its 1/10 every 10,000 (7,500) iterations for the RGB (Depth and Optical Flow) modality when training on IsoGD. The weight decay is set to 0.00004 and at most 60,000 iterations are executed for IsoGD.

The learning rate is initialized as 0.01 and dropped its 1/10 every 2,000 (6,000) iterations when fine-tuning on SKIG (Montalbano). The weight decay is set to 0.00004 and at most 5,000 (20,000) iterations are executed for SKIG (Montalbano). Each video is downsampled to 32 frames using the sampling method in (Zhu et al, 2017). The spatial size of the inputs is restricted to 112×112. One NVIDIA TITAN X GPU is used to train the networks.

The deep architectures in Fig. 1c and Fig. 1d involve the fusion over the recurrent steps. Two score fusion (i.e., $SFusion(\cdot)$ in Eq.(16)) methods are examined for the deep architecture in Fig. 1c: ***maximum fusion*** and ***average fusion***. Two temporal pooling (i.e., $TPooling(\cdot)$ in Eq.(13)) methods are examined for the deep architecture in Fig. 1d: ***maximum pooling*** and ***average pooling***. Two multimodal fusion methods are examined for the RGB/Depth/Flow modalities: one is ***average fusion on the prediction scores of each modality***; the other is ***integrating spatiotemporal features for Linear SVM classification***.

4.3 Architecture Analysis

We begin by evaluating the proposed deep architectures using the aforementioned fusion and pooling methods. Table 1 shows the recognition results on the validation set of IsoGD. The four deep architectures are trained on the training set of IsoGD, without using any pre-trained models on other gesture datasets.

A. Spatiotemporal Feature Learning

Table 1 illustrates the recognition results of the four deep architectures in Fig. 1 under various settings. The comparison results between the architectures in Fig. 1a and Fig. 1b show that it is superior to learn higher-level spatiotemporal features using 2DCNN based on the learnt 2D spatiotemporal feature maps. In our architecture, the 3DCNN and ConvLSTM components only have four 3D convolutional layers and two convolutional LSTM layers. Although the global temporal information can be learnt by the ConvLSTM component, only local spatial information can be encoded using such a network. Thus, it is reasonable to learn higher-level spatiotemporal features further. Table 1 also shows that the bidirectional ConvLSTM can learn more effective spatiotemporal features than the unidirectional ConvLSTM, partly due to the fact that the bidirectional ConvLSTM networks can encode the temporal information from different perspectives.

The evolution of the deep architectures from Fig. 1a to Fig. 1d shows that it is effective to learn spatiotemporal features using 3DCNN and convolutional LSTM. The learnt 2D spatiotemporal feature maps can not only encode the global temporal information, but also encode the local spatial information. Thus, it is reasonable to learn deeper spatiotemporal features further using 2DCNN. Furthermore, we can regard that the deep architecture (3DCNN + ConvLSTM + 2DCNN) is an effective spatiotemporal feature learner. It is robust to various scene backgrounds and illumination conditions theoretically and pratically, and it can also process gestures with various time durations effectively.

B. How to Fuse?

It can be seen that the prediction scores are fused in Fig. 1c while the spatiotemporal features are fused (or pooled) in Fig. 1d. Fusion methods do matter for both types of information. Comparison between MaxFusion and AvgFusion, MaxPooling and AvgPooling, as illustrated in Table 1, demonstrate that averaging based methods outperform maximum ones. It can be noted that max pooling is more frequently used in the Conv-Pooling blocks in the state-of-the-art neural networks, such as Alexnet, Caffenet, VGG16, VGG19, GoogLeNet, Two-Stream ConvNets and C3D networks. This is because max pooling is more conducive to learn the significant and discriminatory features from homogeneous convolutional feature maps. On the contrary, the global spatiotemporal features at each recurrent step represent the gestures with not the same perspectives. Therefore,

[2] The code of the proposed deep architectures has been released publicly on the Github website: https://github.com/GuangmingZhu/Gestureness.

[3] The code is at https://github.com/iwyoo/ConvLSTMCell-tensorflow.

Table 1 Recognition results on the validation set of IsoGD. (MaxFusion and AvgFusion denote the two kinds of score fusion methods used in Eq.(16). MaxPooling and AvgPooling denotes the two kinds of temporal pooling methods used in Eq.(13). If not stated explicitly, average fusion is used for multimodal fusion on the prediction scores. Wider2DCNN has the same architecture as the 2DCNN in Fig.4, but has double the kernel count in each convolution layer.)

Fusion Methods	Modality	Accuracy(%)
3DCNN+UniCLSTM+SPP(Fig. 1a)	RGB	43.88
3DCNN+UniCLSTM+SPP(Fig. 1a)	Depth	44.66
3DCNN+UniCLSTM+SPP(Fig. 1a)	RGBD	51.02
3DCNN+UniCLSTM+2DCNN(Fig. 1b)	RGB	48.86
3DCNN+UniCLSTM+2DCNN(Fig. 1b)	Depth	47.51
3DCNN+UniCLSTM+2DCNN(Fig. 1b)	RGBD	53.70
3DCNN+BiCLSTM+2DCNN+MaxFusion(Fig. 1c)	RGB	50.48
3DCNN+BiCLSTM+2DCNN+MaxFusion(Fig. 1c)	Depth	47.93
3DCNN+BiCLSTM+2DCNN+MaxFusion(Fig. 1c)	RGBD	54.55
3DCNN+BiCLSTM+2DCNN+AvgFusion(Fig. 1c)	RGB	50.97
3DCNN+BiCLSTM+2DCNN+AvgFusion(Fig. 1c)	Depth	48.89
3DCNN+BiCLSTM+2DCNN+AvgFusion(Fig. 1c)	Flow	45.28
3DCNN+BiCLSTM+2DCNN+AvgFusion(Fig. 1c)	RGBD	55.29
3DCNN+BiCLSTM+2DCNN+AvgFusion(Fig. 1c)	RGBD+Flow	57.09
3DCNN+BiCLSTM+2DCNN+MaxPooling(Fig. 1d)	RGB	50.38
3DCNN+BiCLSTM+2DCNN+MaxPooling(Fig. 1d)	Depth	49.65
3DCNN+BiCLSTM+2DCNN+AvgPooling(Fig. 1d)	RGB	51.31
3DCNN+BiCLSTM+2DCNN+AvgPooling(Fig. 1d)	Depth	49.81
3DCNN+BiCLSTM+2DCNN+AvgPooling(Fig. 1d)	Flow	45.30
3DCNN+BiCLSTM+2DCNN+AvgPooling(Fig. 1d)	RGBD+Flow	57.50
3DCNN+BiCLSTM+2DCNN+AvgPooling(Fig. 1d)+SVM	RGBD+Flow	58.65
3DCNN+BiCLSTM+Wider2DCNN+AvgPooling(Fig. 1d)	RGBD+Flow	**60.93**

taking all perspectives into account is superior to selective fusion. This is why average is more frequently used to fuse such kinds of high-level information (Donahue et al, 2015; Veeriah et al, 2015; Du et al, 2015).

C. What to Fuse?

What to fuse over the recurrent steps? Eqs.(13)-(16) describe two different fusion strategies: one is to fuse the spatiotemporal features, the other is to fuse the prediction scores. What to fuse among multimodalities? Two different fusion strategies are also examined: one is to integrate the spatiotemporal features for SVM, the other is to fuse the prediction scores using average. The comparison between the prediction accuracy 57.50% and 57.09% demonstrates the superiority of the spatiotemporal feature fusion. Furthermore, the comparison of 58.65% and 57.50% accuracies further support this conclusion more strongly. Besides, feature fusion over the recurrent steps in Fig. 1d can significantly reduce the computational cost of the fully-connected layers, compared with the deep architecture in Fig. 1c. Thus, we can even conclude that early feature fusion is superior to late score fusion. The comparison and analysis exactly demonstrate the advantages of the fusion strategies of the proposed deep architecture in Fig. 1d.

D. How to Use 2D Spatiotemporal Feature Maps?

The recognition results on the IsoGD, SKIG and Montalbano datasets are illustrated in Tables 2-4 respectively. The lower accuracy on the larger-scale dataset

indicates that the network capacity may be not enough for IsoGD. Since we consider the 3DCNN+ConvLSTM architectures as 2D spatiotemporal feature map learners, we try to increase the network capacity by doubling the kernel count in each convolution layer of 2DCNN in Fig. 4 simply. We name the augmented 2DCNN as Wider2DCNN. Table 1 shows that Wider2DCNN can improve the recognition accuracy significantly. This encourages us to employ the state-of-the-art skills of 2DCNN to optimize the deep architectures further in our future works. This also demonstrates the superiority of the proposed deep architecture to transform video files into 2D feature maps.

4.4 Comparison with the state-of-the-art

Table 2 provides the comparison results with the previous published methods on the validation set of IsoGD. The methods proposed by Wang et al (2017) and Wang et al (2016b) construct handcrafted ways to transform video files into 2D feature maps, and employ AlexNet and VGG-16 networks for the final recognition respectively. The better performance of the proposed deep architecture, compared with Wang et al (2017) and Wang et al (2016b), demonstrates the superiority of our proposed learning to transform sequential video files into 2D spatiotemporal feature maps. The methods of Zhu et al (2016) and Li et al (2016a) use C3D (Tran et al, 2015) based deep architectures for gesture recognition. The proposed deep architecture outperforms these two deep architectures significantly, and is more flexible for the recognition of various-length gestures even when the pre-processing of inputs is absent.

Table 2 gives the comparison results with the previous published methods, which are evaluated on the test set of IsoGD[4]. The proposed deep architecture outperforms the methods (Wang et al, 2016b; Zhu et al, 2016; Li et al, 2016a), but the 2SCVN-3DDSN framework of Duan et al (2017) obtains the state-of-the-art recognition accuracy. Notethat, 2SCVN-3DDSN employs ensemble learning which integrates *Two Stream Consensus Voting Network* (2SCVN) and *3D Depth-Saliency Network* (3DDSN). Three kinds of neural networks are trained on the data of four modalities to get the final optimal recognition accuracy. However, in Table 2, only the proposed deep architecture in Fig. 1d is used to generate our recognition accuracy. If we only compare our network with the 3DDSN of Duan et al (2017), the proposed deep architecture still demonstrates its superiority on 2D spatiotemporal feature map learning (62.14 vs 56.37). This also proves the superiority of the proposed deep architecture, compared with the traditional 3D convolutional neural networks.

Finally, we evaluate the proposed deep architecture on the SKIG and Montalbano datasets. The performance comparison on SKIG is shown in Table 3. It is easily to see that the proposed deep architecture both achieves the state-of-the-art accuracy when using multimodal score fusion and multimodal feature fusion. The multi-stream recurrent neural network (MRNN) (Nishida and Nakayama, 2015) first learns spatial features using 2DCNN, and then feeds the spatial features into MRNN for gesture recognition. The 3DCNN+RNN +CTC network proposed by Molchanov et al (2016) first learns the spatiotemporal features using 3DCNN, and then feeds the vectorized features into RNN. It is worth noting that the spatial correlation information plays an important role for gesture recognition, but is not encoded in the recurrent process of both the two networks. On the contrary, the proposed deep architecture encodes the spatiotemporal correlation information of gestures through the whole process of feature learning. The comparison results illustrate the significance of the spatiotemporal correlation information when learning the spatiotemporal features for gesture recognition.

Table 4 shows the state-of-the-art recognition accuracy on the Montalbano dataset. Pigou et al (2015) explored five kinds of neural networks for gesture recognition and showed the advantages of recurrence and temporal convolutions for gesture recognition in video. We explore four kinds of deep architectures in this paper. We combine the recurrence and temporal convolutions, and propose to learn 2D spatiotemporal feature maps using 3DCNN and ConvLSTM. The comparison results demonstrate that 3DCNN is superior to stacking spatial and temporal convolutions, and convolutional LSTM is more suitable to learn spatiotemporal features than the traditional fully-connected LSTM.

In conclusion, it is superior to learn the 2D spatiotemporal feature maps using 3DCNN and ConvLSTM for gesture recognition. Neural network based self-learning also shows its strengths compared with the handcrafted methods.

4.5 2D Feature Map Visualization

We qualitatively evaluate our learnt spatiotemporal features by visualizing the feature maps and the feature embedding on the Montalbano dataset. Fig. 5 displays the 2D spatiotemporal feature maps of the 20 kinds of gestures extracted by the ConvLSTM layer. Unlike the feature map visualization of image classification

[4] The test labels have not been released publicly, so we can't obtain all the recognition results of each deep architecture on the test set.

Table 2 Recognition results on the IsoGD dataset.

Method	Validating Accuracy(%)	Test Accuracy(%)
Wang et al (2017) (Action Map)	36.27	-
Wang et al (2016b) (Depth Map)	39.23	55.57
Zhu et al (2016) (Pyramidal C3D)	45.02	50.93
Li et al (2016a) (C3D+SVM)	49.20	56.90
Duan et al (2017) (3DDSN-Fusion)	-	56.37
Duan et al (2017) (2SCVN-3DDSN)	-	**67.26**
Proposed(Fig. 1d) + SVM	58.65	62.14
Proposed(Fig. 1d) + Wider2DCNN	**60.93**	-

Table 3 Recognition Results on the SKIG dataset.

Method	Accuracy(%)
Cirujeda and Binefa (2014) (4DCOV)	93.80
Liu and Liu (2016) (Depth Context)	95.37
Tung and Ngoc (2014)	96.70
Nishida and Nakayama (2015) (MRNN)	97.80
Zheng et al (2016) (DLE+HOG^2)	98.43
Molchanov et al (2016) (3DCNN+RNN+CTC)	98.60
Zhu et al (2017)	98.89
Proposed(Fig. 1d) + SVM	**99.53**

Table 4 Recognition Results on the Montalbano dataset.

Method	Accuracy(%)
Pigou et al (2015) (TempPooling)	91.34
Pigou et al (2015) (TempConv)	95.33
Pigou et al (2015) (CNN+LSTM)	96.45
Neverova et al (2016) (ModDrop+Audio)	96.81
Pigou et al (2015) (TempConv+LSTM)	97.23
Molchanov et al (2016) (3DCNN+RNN)	97.40
Proposed(Fig. 1d) + SVM	**97.63**

Table 5 The parameter count and runtime analysis of the four neural networks in Fig. 1.

	Fig. 1a	Fig. 1b	Fig. 1c	Fig. 1d
Parameter ($\times 10^6$)	16.45	17.78	31.79	31.79
Runtime (fps)	1123	864	410	410

where contours of objects can be recognized distinctly, the learnt 2D spatiotemporal feature maps encode the global temporal information, thus the local spatial features cannot be displayed visually. However, it can be seen from Fig. 5 that the regions with the highest activation in the feature maps are coincident with the movements of hands of the gestures.

Furthermore, the features extracted at the temporal pooling layer of the deep architecture in Fig. 1d are projected to 2-dimensional space using t-SNE. Fig. 6 visualizes the feature embedding. It can be observed visually from Fig. 5 and Fig. 6, that the learnt features are effective and discriminative.

4.6 Parameter and Runtime Analysis

Table 5 reports the comparison of the parameter count and runtime among the proposed four neural networks in Fig. 1. The parameter count excludes the parameters of the final fully-connected layer. The runtime is evaluated on a NVIDIA TITAN X GPU. Table 5 shows that all the neural networks can run in real-time. The network in Fig. 1d only have about 32M parameters, far less than the 78M parameters of C3D. However, as we known, less parameters may cause less network capacity. The recognition accuracy comparison among the three experimental datasets and the relative high training error on the IsoGD dataset in our implementation may indicate that the proposed networks are not complex enough for large-scale gesture datasets which have more gesture categories. In our proposed method, we just give a deep architecture of how to encode spatiotemporal feature simultaneously for sequential video, and the 3DCNN, ConvLSTM and 2DCNN components are all simple in some degree, so far. Therefore, it will be natural to employ the state-of-the-art skills of deep learning to improve the proposed deep architectures for different applications.

5 Conclusion and Future Work

In this paper, we explore four deep architectures to learn novel spatiotemporal features for gesture recognition. The deep architecture learns 2D spatiotemporal feature maps using 3DCNN and convolutional LSTM jointly. The learnt 2D feature maps can encode the global temporal information and local spatial information simultaneously. Thus, 2DCNN can be used further to learn higher-level spatiotemporal features. The proposed deep architecture provides an alternative method to transform video files into 2D feature maps (or we can say 2D images).

The paper only presents the preliminary version of the deep architectures. The state-of-the-art skills of the 2DCNN, 3DCNN and LSTM networks, such as the idea of DensNet, DRN, and ResNext, can be further utilized to construct an improved version in order to obtain higher recognition accuracy. Bidirectional ConvLSTM shows its advantages on isolated gesture recognition, but unidirectional ConvLSTM will be more reasonable for continuous gesture recognition. Therefore, we will improve the deep architectures and apply them to the continuous gesture recognition in our future works.

Acknowledgements This work is partially supported by the National Natural Science Foundation of China under Grant No.61702390, the Fundamental Research Funds for the Central Universities under Grant JB181001, and the Key Research and Development Program of Shaanxi Province under Grant No.2018ZDXM-GY-036.

References

Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M, Kudlur M, Levenberg J, Monga R, Moore S, Murray DG, Steiner B, Tucker PA, Vasudevan V, Warden P, Wicke M, Yu Y, Zhang X (2016) Tensorflow: A system for large-scale machine learning. In: OSDI, pp 265–283

Baccouche M, Mamalet F, Wolf C, Garcia C, Baskurt A (2011) Sequential deep learning for human action recognition. In: International Workshop on Human Behavior Understanding, pp 29–39

Camgoz NC, Hadfield S, Koller O, Bowden R (2016) Using convolutional 3d neural networks for user-independent continuous gesture recognition. In: 23rd International Conference on Pattern Recognition (ICPR), pp 49–54

Cirujeda P, Binefa X (2014) 4dcov: A nested covariance descriptor of spatio-temporal features for gesture recognition in depth sequences. In: 2014 2nd International Conference on 3D Vision (3DV), pp 657–664

Donahue J, Anne Hendricks L, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K, Darrell T (2015) Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2625–2634

Dong H, Supratak A, Mai L, Liu F, Oehmichen A, Yu S, Guo Y (2017) Tensorlayer: A versatile library for efficient deep learning development. arXiv preprint arXiv:170708551

Du Y, Wang W, Wang L (2015) Hierarchical recurrent neural network for skeleton based action recognition.

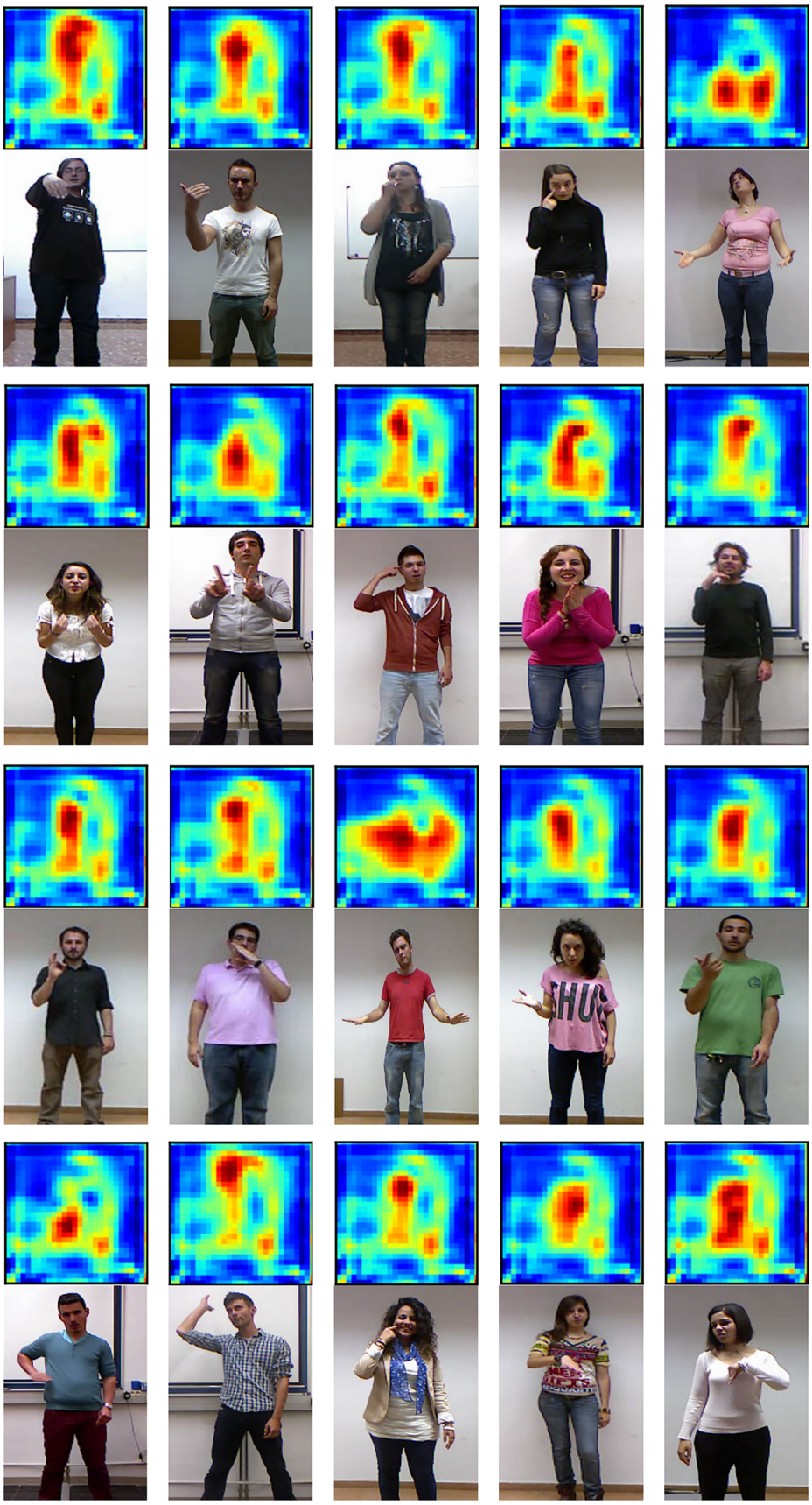

Fig. 5 The 2D spatiotemporal feature maps of each gesture category of the Montalbano dataset. Each subfigure is a 28×28 feature map with the highest average activation among the 2D spatiotemporal feature maps extracted by the ConvLSTM layer in Fig .1d. Best viewed in color.

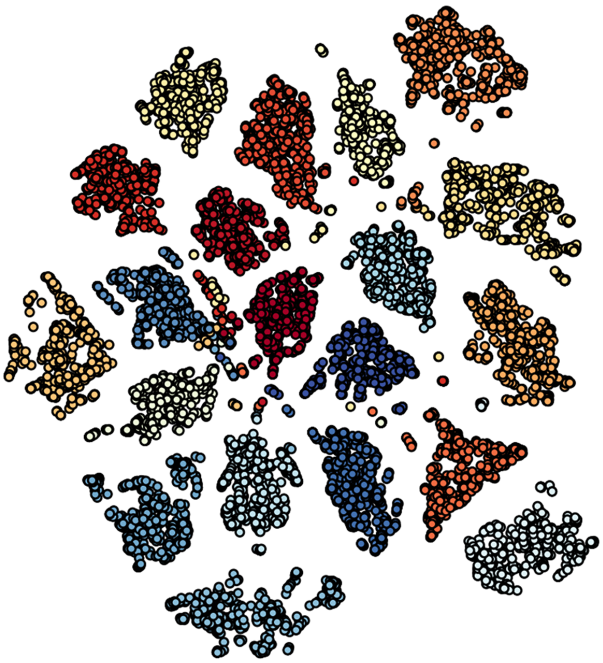

Fig. 6 Feature embedding on the Montalbano dataset. Each video is visualized as a point and videos belonging to the same gesture have the same color. Best viewed in color.

In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1110–1118

Duan J, Zhou S, Wan J, Guo X, Li SZ (2016) Multi-modality fusion based on consensus-voting and 3d convolution for isolated gesture recognition. arXiv preprint arXiv:161106689

Duan J, Wan J, Zhou S, Guo X, Li SZ (2017) A unified framework for multi-modal isolated gesture recognition. ACM Transactions on Multimedia Computing, Communications, and Applications

Escalante HJ, Ponce-López V, Wan J, Riegler MA, Chen B, Clapés A, Escalera S, Guyon I, Baró X, Halvorsen P, et al (2016) Chalearn joint contest on multimedia challenges beyond visual analysis: An overview. In: 23rd International Conference on Pattern Recognition (ICPR), pp 67–73

Escalera S, Baro X, Gonzalez J, Bautista MA, Madadi M, Reyes M, Ponce-Lopez V, Escalante HJ, Shotton J, Guyon I (2014) Chalearn looking at people challenge 2014: Dataset and results. In: ECCV Workshops, pp 459–473

Escalera S, Athitsos V, Guyon I (2016) Challenges in multimodal gesture recognition. Journal of Machine Learning Research 17(72):1–54

Feichtenhofer C, Pinz A, Zisserman A (2016) Convolutional two-stream network fusion for video action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1933–1941

Herath S, Harandi M, Porikli F (2017) Going deeper into action recognition: A survey. Image and Vision Computing 60:4–21

Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp 448–456

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105

Kuehne H, Jhuang H, Garrote E, Poggio T, Serre T (2011) Hmdb: a large video database for human motion recognition. In: IEEE International Conference on Computer Vision (ICCV), pp 2556–2563

Li Y, Miao Q, Tian K, Fan Y, Xu X, Li R, Song J (2016a) Large-scale gesture recognition with a fusion of rgb-d data based on the c3d model. In: 23rd International Conference on Pattern Recognition (ICPR), pp 25–30

Li Z, Gavves E, Jain M, Snoek CG (2016b) Videolstm convolves, attends and flows for action recognition. arXiv preprint arXiv:160701794

Liu L, Shao L (2013) Learning discriminative representations from rgb-d video data. In: Proceedings of the Twenty-Third international joint conference on Artificial Intelligence, pp 1493–1500

Liu M, Liu H (2016) Depth context: a new descriptor for human activity recognition by using sole depth sequences. Neurocomputing 175:747–758

Mitra S, Acharya T (2007) Gesture recognition: A survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 37(3):311–324

Molchanov P, Yang X, Gupta S, Kim K, Tyree S, Kautz J (2016) Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 4207–4215

Neverova N, Wolf C, Taylor G, Nebout F (2016) Moddrop: adaptive multi-modal gesture recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 38(8):1692–1706

Nishida N, Nakayama H (2015) Multimodal gesture recognition using multi-stream recurrent neural network. In: Pacific-Rim Symposium on Image and Video Technology, pp 682–694

Pigou L, van den Oord A, Dieleman S, Van Herreweghe M, Dambre J (2015) Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video. International Journal of Computer

Vision pp 1–10
Simonyan K, Zisserman A (2014a) Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems, pp 568–576
Simonyan K, Zisserman A (2014b) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556
Soomro K, Zamir AR, Shah M (2012) Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:12120402
Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
Tran D, Bourdev LD, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3d convolutional networks. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp 4489–4497
Tung PT, Ngoc LQ (2014) Elliptical density shape model for hand gesture recognition. In: Proceedings of the Fifth Symposium on Information and Communication Technology, pp 186–191
Veeriah V, Zhuang N, Qi GJ (2015) Differential recurrent neural networks for action recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp 4041–4049
Wan J, Zhao Y, Zhou S, Guyon I, Escalera S, Li SZ (2016) Chalearn looking at people rgb-d isolated and continuous datasets for gesture recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp 56–64
Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, Van Gool L (2016a) Temporal segment networks: Towards good practices for deep action recognition. In: European Conference on Computer Vision, pp 20–36
Wang P, Li W, Gao Z, Zhang J, Tang C, Ogunbona P (2015) Deep convolutional neural networks for action recognition using depth map sequences. arXiv preprint arXiv:150104686
Wang P, Li W, Liu S, Gao Z, Tang C, Ogunbona P (2016b) Large-scale isolated gesture recognition using convolutional neural networks. In: 23rd International Conference on Pattern Recognition (ICPR), pp 7–12
Wang P, Li W, Gao Z, Zhang Y, Tang C, Ogunbona P (2017) Scene flow to action map: A new representation for rgb-d based action recognition with convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition
Xingjian S, Chen Z, Wang H, Yeung DY, Wong WK, Woo Wc (2015) Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: Advances in neural information processing systems, pp 802–810
Zheng J, Feng Z, Xu C, Hu J, Ge W (2016) Fusing shape and spatio-temporal features for depth-based dynamic hand gesture recognition. Multimedia Tools and Applications pp 1–20
Zhu G, Zhang L, Mei L, Shao J, Song J, Shen P (2016) Large-scale isolated gesture recognition using pyramidal 3d convolutional networks. In: 23rd International Conference on Pattern Recognition (ICPR), pp 19–24
Zhu G, Zhang L, Shen P, Song J (2017) Multimodal gesture recognition using 3d convolution and convolutional lstm. IEEE Access 5:4517–4524