Learning Spatiotemporal Features using 3DCNN and Convolutional LSTM for Gesture Recognition

Liang Zhang, Guangming Zhu, Peiyi Shen, Juan Song
School of Software, Xidian University
{liangzhang, gmzhu, pyshen, songjuan}@xidian.edu.cn

Syed Afaq Shah, Mohammed Bennamoun
University of Western Australia
{afaq.shah, mohammed.bennamoun}@uwa.edu.au

Abstract

学习姿态识别的时空域特征

Gesture recognition aims at understanding the ongoing human gestures. In this paper, we present a deep architecture to learn spatiotemporal features for gesture recognition. The deep architecture first learns 2D spatiotemporal feature maps using 3D convolutional neural networks (3DCNN) and bidirectional convolutional long-short-term-memory networks (ConvLSTM). The learnt 2D feature maps can encode the global temporal information and local spatial information simultaneously. Then, 2DCNN is utilized further to learn the higher-level spatiotemporal features from the 2D feature maps for the final gesture recognition. The spatiotemporal correlation information is kept through the whole process of feature learning. This makes the deep architecture an effective spatiotemporal feature learner. Experiments on the ChaLearn LAP large-scale isolated gesture dataset (IsoGD) and the Sheffield Kinect Gesture (SKIG) dataset demonstrate the superiority of the proposed deep architecture.

使用3D卷积神经网络和双向卷积LSTM学习2D时空域特征map。

2D时空特征map可以同时对全局时域信息和局部空间信息编码

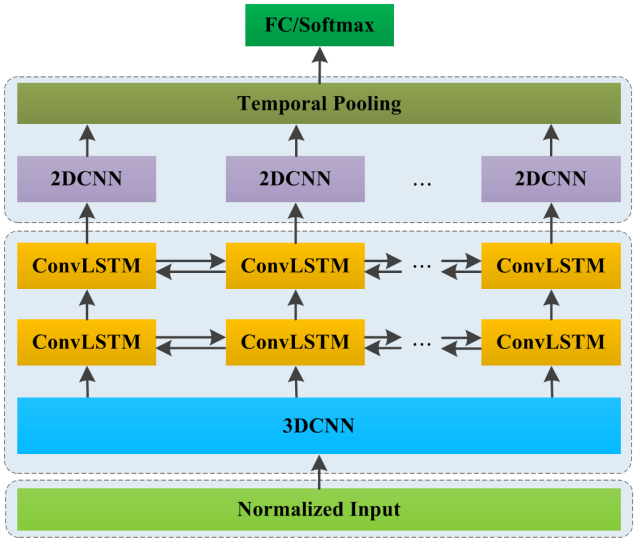

Figure 1. Overview of the proposed deep architecture. 3DCNN and bidirectional ConvLSTM are utilized to learn the short-term and long-term spatiotemporal features successively, and then 2DCNN is used to learn higher-level spatiotemporal features based on the learnt 2D long-term spatiotemporal feature maps for the final gesture recognition.

1. Introduction

Gestures, as a nonverbal body language, play a very important role in humans 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 [23].

Effective and universal gesture recognition from videos is extremely difficult; partly due to the large gesture vocabularies with cultural differences, various illumination conditions, out-of-vocabulary motions, inconsistent and non-standard behaviors among different performers, etc [12]. Moreover, 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 [32], 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 [17, 29, 31], lots of neural network based frameworks are proposed to learn spatiotemporal features

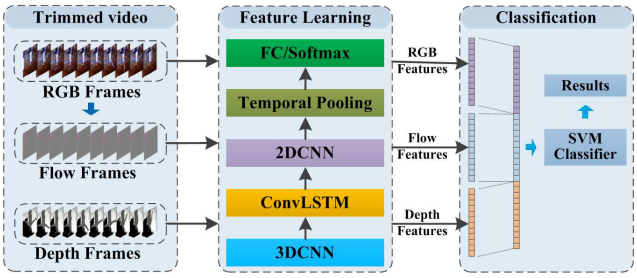

Figure 2. Pipeline of the proposed framework. Multimodal data is used to train the proposed deep architecture respectively, and RGB/Depth/Flow based spatiotemporal features are extracted and combined into large multimodal spatiotemporal feature vectors further. Linear SVM classifier is utilized for the final gesture recognition.

for human action/gesture recognition. *Two-Stream Convolutional Networks* [28] learn spatial and temporal features separately. *Long-term Recurrent Convolutional Networks* (LRCN) [6] learn spatial and temporal features using convolutional neural networks (CNN) and long-short-term-memory (LSTM) networks successively. Tran et al. [32] 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. [24] 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 from the various-length videos. Although Molchanov et al. [24] proposed to combine 3DCNN and RNN, fully-connected spatiotemporal features are transferred into RNN, this make 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 learn long-term spatiotemporal features further using bidirectional convolutional LSTM (ConvLSTM), lastly recognize gestures using 2DCNN based on the learnt 2D spatiotemporal feature maps. An overview of the proposed deep architecture is illustrated in Figure 1, and the pipeline of the proposed framework is in Figure 2.

In brief, our contributions in this paper include:

- 2D spatiotemporal feature maps are learnt using 3DCNN and bidirectional convolutional LSTM. 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.
- The proposed spatiotemporal features with a linear SVM classification model outperform or achieve performance in par with the state-of-the-art methods on two different benchmarks.
- To the best of our knowledge, this is the first time to learn 2D spatiotemporal feature maps using 3DCNN and bidirectional 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. Various deep neural networks have been proposed recently [15]. 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 [30], but 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 [28] and their derivations [36, 13] obtain the state-of-the-art performance on HMDB51 [18] and UCF101 datasets, but they fail to achieve a similar performance in the case of gesture recognition. Another obvious approach is to learn spatial and temporal features successively, such as LRCN [6]. However, Pigou et al. [26] demonstrated that LRCN-style networks are not optimal, while bidirectional recurrence and temporal convolutions can improve gesture recognition performance significantly. The huge success of 2DCNN on image recognition has encouraged researchers to transform video files into particular 2D image files, so that the state-of-the-art 2DCNN networks can be applied on gesture recognition [39]. 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. [32] constructed a deep 3D ConvNet to learn spatiotemporal features directly and achieved the best performance on different types of video analysis tasks. Inspired by [32], 3DCNN-based neural networks obtained the remarkable performances on gesture recognition [11]. In the past 2016 ChaLearn LAP Large-scale Isolated/ Continuous Gesture Recognition Challenges [35], 3DCNN demon-

strated excellent performance [3, 41, 19, 10]. However, 3DCNN use the stacked pooling layers to reduce the spatial and temporal size 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 successively [24, 42, 2].

Generally, the fully-connected features of 3DCNN or 2DCNN are transferred into RNN/LSTM networks [6, 24, 2]. The spatial correlation information is lost in the input-to-state and state-to-state transitions of RNN/LSTM. ConvLSTM [27] is originally proposed for precipitation nowcasting. The spatial correlation information is encoded explicitly in the input-to-state and state-to-state transitions of ConvLSTM. We, therefore, propose to first learn short-term spatiotemporal features using a shallow 3DCNN, and then learn long-term spatiotemporal features using bidirectional ConvLSTM. The bidirectional ConvLSTM layers do not shrink the spatial size, but learn the global temporal correlation information completely. The combination of the shallow 3DCNN and the bidirectional ConvLSTM can transform video files into 2D spatiotemporal feature maps, which encode the global temporal information and local spatial information simultaneously. This transformation makes it possible to utilize 2DCNN further for the final gesture recognition.

3. Method

As illustrated in Figures 1 and 2, the proposed deep architecture is mainly composed of two components: 2D spatiotemporal feature map learning and classification based on the 2D feature maps. The former learns 2D spatiotemporal feature maps from the normalized inputs whose length is down-sampled to 32 frames per video. The latter learns higher-level spatiotemporal features further using 2DCNN, and then uses a linear Support Vector Machine (SVM) classifier for the final gesture recognition.

3.1. 2D Spatiotemporal Feature Map Learning

Three facts are taken into consideration when constructing the proposed deep architecture: **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. 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. Bidirectional ConvLSTM is designed to learn global or long-term spatiotemporal features. The spatiotemporal correlation information is encoded during the recurrent process.

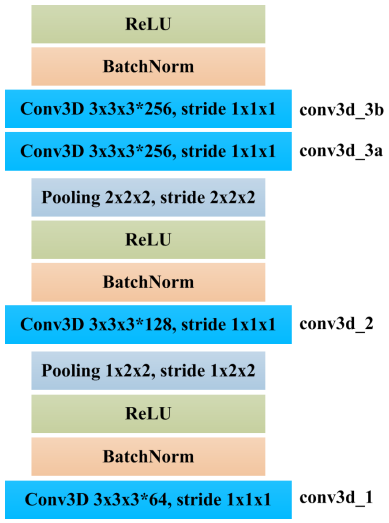

Figure 3. The 3DCNN component.

A. 3DCNN Component

The 3DCNN component of the proposed deep architecture is similar in design to the C3D model [32]. According to the aforementioned analysis, the 3DCNN does not need to be deep; only four Conv3D layers are therefore constructed, as displayed in Figure 3. 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. Based on the setting of the two pooling layers as illustrated in Figure 3, 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 only learn the short-term spatiotemporal features. Batch normalization [16] can allow using much higher learning rates and being less careful about initialization, so batch normalization is utilized to optimize our networks.

The proposed deep architecture does not need all input sequences have the same length. But, we still preprocess the input sequences to make them of the same length for simplicity during training. Uniform sampling with temporal jitter [42] is utilized for the input preprocessing, which can store the temporal information and augment the dataset. Each sequence is down-sampled to the fixed 32 frames. If one input sequence is less than 32 frames, the last frame is used with the same padding.

B. Convolutional LSTM Component

Generally, the fully-connected LSTM, which takes vectorized features as input, is used to learn temporal features. The limitation of this 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 [27] 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.

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 "*" denote the convolution operator, and let "o" denote 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. A two-layer bidirectional ConvLSTM is constructed as illustrated in Figure 1.

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

$$STFM = BICLSTM(3DCNN(I)) \quad (6)$$

where

$$STFM = \{STFM_t \in \mathbb{R}^{\frac{w}{m} \times \frac{h}{m} \times c} | t = 1, 2, ..., T_N\} \quad (7)$$

T_N is the recurrent step count of ConvLSTM ($T_N = T_I/2$ in this implementation). 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. 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 video files into 2D feature maps. This is very important, because the deep architecture can transform various-length 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.

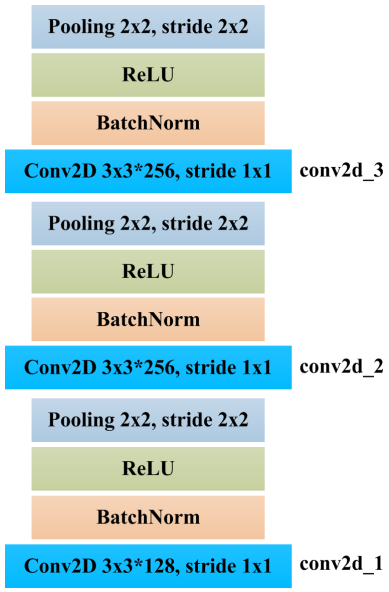

Figure 4. The 2DCNN component.

3.2. Classification based on the 2D Feature Maps

Generally, video files need to be decoded into separate image files [20] or encoded into special images [37] 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. This enables 2DCNN to be used in video-based applications in an alternative way.

A. 2DCNN Component

Since the 2D spatiotemporal feature maps still have large spatial size, dimensionality reduction is necessary for the final recognition. A simple 2DCNN is employed to reduce the dimensionality and to learn the higher-level spatiotemporal features, based on learnt the 2D spatiotemporal feature maps at each recurrent step of ConvLSTM. 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. The 2DCNN component, displayed in Figure 4, consists of three "Convolution-BatchNorm-ReLU" layers. The 2DCNN finally outputs deeper spatiotemporal features which are 4096-dimensional after vectorization. Formally, the deeper spatiotemporal feature (*DSTF*) can be represented as

$$DSTF_t = 2DCNN(\overrightarrow{W}_{fw}\overrightarrow{STFM}_t + \overleftarrow{W}_{bw}\overleftarrow{STFM}_t) \quad (8)$$

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

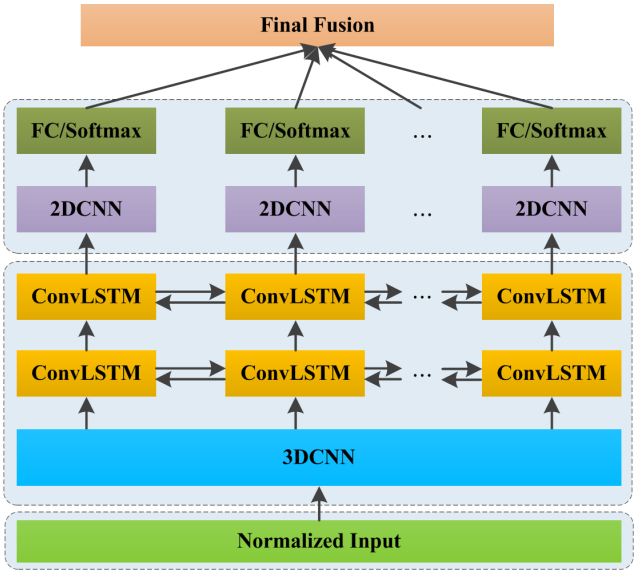

Figure 5. Variant of the proposed deep architecture.

STM 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. The *DSTF* learnt by the 2DCNN can be denoted as

$$DSTF = \{DSTF_t \in \mathbb{R}^{4096} | t = 1, 2, ..., T_N\} \quad (9)$$

B. Classification

Two fusion methods are generally used after features extraction at each recurrent step: one is to calculate the loss of each recurrent step and minimize the cumulative loss over the steps [34], the other is to accumulate the outputs over the steps and minimize the final softmax loss [8]. In this implementation, two fusion methods on *DSTF* are evaluated. The first one is illustrated in Figure 1: 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

$$A = TPooling(DSTF_t | t = 1, 2, ..., T_N) \quad (10)$$

$$p(C_k) = \frac{e^{A_k}}{\sum_{i=0}^{C-1} e^{A_i}} \quad (11)$$

where $TPooling(\cdot)$ is the pooling method of the temporal pooling layer, C is the category count of gestures. The other is illustrated in Figure 5: 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

$$p_t(C_k) = \frac{e^{DSTF_t^k}}{\sum_{i=0}^{C-1} e^{DSTF_t^i}} \quad (12)$$

$$p(C_k) = SFusion(p_t(C_k) | t = 1, 2, ..., T_N) \quad (13)$$

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.

The fusion on the prediction results of the multi-modal data can be used to improve the prediction accuracy. Besides, we also tried to fuse the higher-level spatiotemporal features of each modality directly. An integration strategy is used to combine the learnt features of each modality into larger feature vectors, and linear SVM classifier is used for the final classification, as illustrated in Figure 2.

3.3. Network Training

The proposed deep architecture is designed originally and it has to be trained from scratch. In our implementation, the end-to-end training of the network (illustrated in Figure 1) cannot give the optimal convergence speed and recognition accuracy. Therefore, the variant of the proposed deep architecture, as displayed in Figure 5, is firstly trained from scratch.

The temporal pooling layer is removed in the variant version. The softmax classifier works directly on the global spatiotemporal feature $DSTF_t$. This strategy works well, because the bidirectional recurrence makes each $DSTF_t$ encode the global spatiotemporal information. Thus, the classification on each $DSTF_t$ can also give a convincing prediction result.

The proposed deep architecture in Figure 1 is fine-tuned based on the training result of the variant. The differences on the computation and accuracy between these two deep architectures will be analyzed in the following section.

4. Experiments

We extensively evaluate the proposed framework and the learnt spatiotemporal features under various settings for the task of gesture recognition.

4.1. Datasets

Two public datasets are used to evaluate the performance of the proposed framework: the ChaLearn LAP large-scale isolated gesture dataset (IsoGD) [35] and the Sheffield Kinect Gesture dataset (SKIG) [21].

IsoGD is a large-scale isolated gesture dataset derived from the ChaLearn Gesture Dataset (CGD) [14]. IsoGD 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 subsets: the training, validation and testing subsets. The labels of the testing subset have not been released, thus the validation subset 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.

Besides the RGB and depth modalities, optical flow data are also used to improve the prediction accuracy. The Brox-OpticalFlow 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[1] on the Tensorflow-0.11 [1], the Tensorlayer-1.2.8 [7], and the implementation of ConvLSTM[2].

The deep architectures are trained from scratch on the large-scale IsoGD dataset, and then fine-tuned on the SKIG dataset. 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 45,000 iterations (10 epochs) are executed for IsoGD. The learning rate is initialized as 0.01 and dropped its 1/10 every 2,000 iterations when fine-tuning on SKIG. The weight decay is set to 0.00004 and at most 5,000 iterations are executed for SKIG. Each video is down-sampled to 32 frames using the sampling method in [42]. 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 Figures 1 and 5 are evaluated respectively. Two score fusion (i.e., $SFusion(\cdot)$ in Eq.(13)) methods are examined for the deep architecture in Figure 5: ***maximum fusion*** and ***average fusion***. Two temporal pooling (i.e., $TPooling(\cdot)$ in Eq.(10)) methods are examined for the deep architecture in Figure 1: ***maximum pooling*** and ***average pooling***. Two multimodal fusion methods are examined for the RGB/Depth/OpticalFlow 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 subset of IsoGD. The two deep architectures are trained only on the training subset of IsoGD, without using any pre-trained models on other gesture datasets.

A. How to Fuse?

[1]The code of the proposed framework has been released on the Github https://github.com/GuangmingZhu/Conv3D_BICLSTM.

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

Fusion Methods	Modality	Accuracy(%)
[a]MaxFusion	RGB	50.48
[a]MaxFusion	Depth	47.93
[a]MaxFusion	RGBD	54.55
[a]AvgFusion	RGB	50.97
[a]AvgFusion	Depth	48.89
[a]AvgFusion	Flow	45.28
[a]AvgFusion	RGBD	55.29
[a]AvgFusion	RGBD+Flow	**57.09**
[b]MaxPooling	RGB	50.38
[b]MaxPooling	Depth	49.65
[b]AvgPooling	RGB	51.31
[b]AvgPooling	Depth	49.81
[b]AvgPooling	Flow	45.30
[b]AvgPooling	RGBD+Flow	**57.50**
[b]AvgPooling+SVM	RGBD+Flow	**58.65**

Table 1. Recognition results on the validation subset of IsoGD. (The superscripts [a] and [b] denote the deep architectures in Figure 5 and Figure 1 respectively. MaxFusion and AvgFusion denote the two kinds of score fusion methods used in Eq.(13) for the deep architecture in Figure 5. MaxPooling and AvgPooling denote the two kinds of temporal pooling methods used in Eq.(10) for the deep architecture in Figure 1. If not stated explicitly, average fusion is used for multimodal fusion on the prediction scores.)

Although the prediction scores are fused in Figure 5 while the spatiotemporal features are fused (or pooled) in Figure 1, fusion methods do matter for both the two kinds of information. Both the comparison between MaxFusion and AvgFusion and the comparison between MaxPooling and AvgPooling, as illustrated in Table 1, demonstrate that average outperforms maximum. As we know, Max pooling is more frequently used in the Conv-Pooling blocks in the state-of-the-art neural networks (e.g., Alexnet, Caffenet, VGG16, VGG19, GoogLeNet, Two-Stream ConvNets, C3D, etc). 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. So, 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 [6, 34, 8].

B. What to Fuse?

What to fuse over the recurrent steps? Eqs.(10)-(13) 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. The comparison between the prediction accuracy 58.65% and 57.50% supports the conclusion further. Besides, feature fusion over the recurrent steps can significantly reduce the computational cost of the fully-connected layers, compared with the deep architecture in Figure 5. Thus, 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 Figure 1.

C. Spatiotemporal Feature Learner

The evolution of the deep architectures from our previous work [42] to the Figures 5 and 1 in this paper shows that it is effective to learn spatiotemporal features using 3DCNN and convolutional LSTM. The learnt 2D spatiotemporal feature maps encode the global temporal information and local spatial information. Thus, it is reasonable to learn deeper spatiotemporal features further using 2DCNN. Our previous work [42] uses spatial pyramid pooling to extract higher-level spatiotemporal features from the 2D feature maps. The improvement from 51.02% to 58.65% demonstrates the superiority of 2DCNN to learn higher-level features further. 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 actually, and it can also process gestures with various time durations effectively.

4.4. Comparison with the state-of-the-art

Table 2 gives the comparison results with the previous published methods, which are evaluated on the validation subset of IsoGD. The methods in [38] and [39] propose 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 [38] and [39], demonstrates the superiority of learning to transform video files into 2D spatiotemporal feature maps adaptively using 3DCNN and ConvLSTM. The methods in [41] and [19] use C3D [32] based deep architectures for gesture recognition. The proposed deep architecture outperforms the two deep architectures, and is more flexible for the recognition of various-length gestures even when the pre-processing of inputs is absent.

Table 3 gives the comparison results with the previous published methods, which are evaluated on the testing subset of IsoGD. The proposed deep architecture outperforms the methods in [39, 41, 19] on the testing subset, but the 2SCVN-3DDSN framework in [9] obtains the state-of-the-art recognition accuracy. 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, only the proposed deep architecture in Figure 1 is used to report our recognition accuracy. If we only compare our network with the 3DDSN in [9], the proposed deep architecture still demonstrates its superiority on 2D spatiotemporal feature map learning. This also proves the superiority of the proposed deep architecture, compared with the traditional 3D convolutional neural networks.

Method	Accuracy(%)
Action Map [38]	36.27
Wang et al. [39]	39.23
Pyramidal C3D [41]	45.02
Li et al. [19]	49.20
Zhu et al. [42]	51.02
Proposed	**57.50**
Proposed + SVM	**58.65**

Table 2. Recognition results on the validation subset of IsoGD.

Finally, we evaluate the proposed deep architecture on the SKIG dataset. The performance comparison on SKIG is shown in Table 4. The proposed deep architecture both achieves the state-of-the-art accuracy when using multimodal score fusion and multimodal feature fusion. Although the proposed deep architecture only obtains less than 1% improvement compared with [24] and [42], this is due to the fact that [24] and [42] have achieved extremely high recognition accuracy. The multi-stream recurrent neural network (MRNN) [25] first learns spatial features using 2DCNN, and then feeds the spatial features into MRNN for gesture recognition. The 3DCNN+RNN+CTC network [24] first learns the spatiotemporal features using 3DCNN, and then feeds the vectorized features into RNN. 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 exactly confirm the importance of the spatiotemporal correlation information when learning the spatiotemporal features for gesture recognition.

Therefore, 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.

Method	Accuracy(%)
Pyramidal C3D [41]	50.93^a
Wang et al. [39]	55.57^a
Li et al. [19]	56.90^a
3DDSN-Fusion [9]	56.37^b
2SCVN-3DDSN [9]	$\mathbf{67.26}^b$
Proposed + SVM	60.47^b
Proposed + SVM	62.14^a

Table 3. Recognition results on the test subset of IsoGD. (The superscript ^{a} indicates that both the training and validation subsets are used for training. The superscript ^{b} indicates that only the training subset is used for training.)

Method	Accuracy(%)
RGGP+RGB-D [21]	88.70
Choi et al. [4]	91.90
4DCOV [5]	93.80
Depth Context [22]	95.37
Tung et al. [33]	96.70
MRNN [25]	97.80
DLEH2(DLE+HOG2) [40]	98.43
3DCNN+RNN+CTC [24]	98.60
Zhu et al. [42]	98.89
Proposed	**99.52**
Proposed + SVM	**99.53**

Table 4. Recognition Results on the SKIG dataset.

5. Conclusion

In this paper, we proposed a deep architecture for learning novel spatiotemporal features for gesture recognition. The deep architecture learns 2D spatiotemporal feature maps using 3DCNN and bidirectional convolutional LSTM. The learnt 2D feature maps can encode the global temporal information and local spatial information. 2DCNN can be used further to learn higher-level spatiotemporal features on the learnt 2D spatiotemporal feature maps. 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 architecture. The state-of-the-art skills of 2DCNN, 3DCNN and LSTM networks can be further utilized to construct an improved version in order to obtain higher recognition accuracy.

Acknowledgements

This work is partially supported by the National Natural Science Foundation of China under Grant No.61702390, No.61401324, and No.61305109, and the China Postdoctoral Science Foundation under Grant No.2016M592763.

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. A. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zhang. Tensorflow: A system for large-scale machine learning. In *OSDI*, 2016.

[2] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential deep learning for human action recognition. In *HBU*, 2011.

[3] N. C. Camgoz, S. Hadfield, O. Koller, and R. Bowden. Using convolutional 3d neural networks for user-independent continuous gesture recognition. In *ICPR*, 2016.

[4] H. Choi and H. Park. A hierarchical structure for gesture recognition using rgb-d sensor. In *HAI*, 2014.

[5] P. Cirujeda and X. Binefa. 4dcov: A nested covariance descriptor of spatio-temporal features for gesture recognition in depth sequences. In *3DV*, 2014.

[6] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[7] H. Dong, A. Supratak, L. Mai, F. Liu, A. Oehmichen, S. Yu, and Y. Guo. Tensorlayer: A versatile library for efficient deep learning development. *arXiv preprint arXiv:1707.08551*, 2017.

[8] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, 2015.

[9] J. Duan, J. Wan, S. Zhou, X. Guo, and S. Li. A unified framework for multi-modal isolated gesture recognition. *ACM TOMM*, 2017.

[10] J. Duan, S. Zhou, J. Wan, X. Guo, and S. Z. Li. Multi-modality fusion based on consensus-voting and 3d convolution for isolated gesture recognition. *CoRR*, abs/1611.06689, 2016.

[11] H. J. Escalante, V. Ponce-Lopez, J. Wan, M. A. Riegler, . Chen, A. Claps, S. Escalera, I. Guyon, X. Bar, P. Halvorsen, H. Muller, and M. Larson. Chalearn joint contest on multimedia challenges beyond visual analysis: An overview. In *ICPR*, 2016.

[12] S. Escalera, V. Athitsos, and I. Guyon. Challenges in multimodal gesture recognition. *JMLR*, 17:1–54, 2016.

[13] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016.

[14] I. Guyon, V. Athitsos, P. Jangyodsuk, and H. J. Escalante. The chalearn gesture dataset (cgd 2011). *Machine Vision and Applications*, 25:1929–1951, 2014.

[15] S. Herath, M. Harandi, and F. Porikli. Going deeper into action recognition: A survey. *Image Vis. Comput.*, 60:4–21, 2017.

[16] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2014.

[18] H. Kuehne, H. Jhuang, E. Garrote, T. A. Poggio, and T. Serre. Hmdb: A large video database for human motion recognition. In *ICCV*, 2011.

[19] Y. Li, Q. Miao, K. Tian, Y. Fan, X. Xu, R. Li, and J. Song. Large-scale gesture recognition with a fusion of rgb-d data based on the c3d model. In *ICPR*, 2016.

[20] Z. Li, E. Gavves, M. Jain, and C. Snoek. Videolstm convolves, attends and flows for action recognition. *CoRR*, abs/1607.01794, 2016.

[21] L. Liu and L. Shao. Learning discriminative representations from rgb-d video data. In *IJCAI*, 2013.

[22] M. Liu and H. Liu. Depth context: a new descriptor for human activity recognition by using sole depth sequences. *Neurocomputing*, 175:747–758, 2016.

[23] S. Mitra and T. Acharya. Gesture recognition: A survey. *IEEE TSMCC*, 37:311–324, 2007.

[24] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In *CVPR*, 2016.

[25] N. Nishida and H. Nakayama. Multimodal gesture recognition using multi-stream recurrent neural network. In *PSIVT*, 2015.

[26] L. Pigou, A. v. d. Oord, S. Dieleman, M. V. Herreweghe, and J. Dambre. Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video. *IJCV*, pages 1–10, 2016.

[27] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. k. Wong, and W. c. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NIPS*, 2015.

[28] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR, abs/1409.1556*, 2014.

[30] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

[31] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[32] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[33] P. T. Tung and N. Q. Ly. Elliptical density shape model for hand gesture recognition. In *SoICT*, 2014.

[34] V. Veeriah, N. Zhuang, and G.-J. Qi. Differential recurrent neural networks for action recognition. In *ICCV*, 2015.

[35] J. Wan, S. Z. Li, Y. Zhao, S. Zhou, I. Guyon, and S. Escalera. Chalearn looking at people rgb-d isolated and continuous datasets for gesture recognition. In *CVPRW*, 2016.

[36] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

[37] P. Wang, W. Li, Z. Gao, J. Zhang, C. Tang, and P. Ogunbona. Deep convolutional neural networks for action recognition using depth map sequences. *CoRR*, abs/1501.04686, 2015.

[38] P. Wang, W. Li, Z. Gao, Y. Zhang, C. Tang, and P. Ogunbona. Scene flow to action map: A new representation for rgb-d based action recognition with convolutional neural networks. In *CVPR*, 2017.

[39] P. Wang, W. Li, S. Liu, Z. Gao, C. Tang, and P. Ogunbona. Large-scale isolated gesture recognition using convolutional neural networks. In *ICPR*, 2016.

[40] J. Zheng, Z. Feng, C. Xu, J. Hu, and W. Ge. Fusing shape and spatio-temporal features for depth-based dynamic hand gesture recognition. *Multimedia Tools and Applications*, 2016.

[41] G. Zhu, L. Zhang, L. Mei, J. Shao, J. Song, and P. Shen. Large-scale isolated gesture recognition using pyramidal 3d convolutional networks. In *ICPR*, 2016.

[42] G. Zhu, L. Zhang, P. Shen, and J. Song. Multimodal gesture recognition using 3d convolution and convolutional lstm. *IEEE Access*, 2017.