# Multimodal Gesture Recognition Using 3D Convolution and Convolutional LSTM

Guangming Zhu, Liang Zhang, Peiyi Shen and Juan Song

***Abstract*—Gesture recognition aims to recognize meaningful movements of human bodies, and is of utmost importance in intelligent human computer/robot interactions. In this paper, we present a multimodal gesture recognition method based on 3D convolution and convolutional Long-Short-Term-Memory (LSTM) networks. The proposed method firstly learns short-term spatiotemporal features of gestures through 3D convolutional neural network, and then learns long-term spatiotemporal features by convolutional LSTM networks based on the extracted short-term spatiotemporal features. In addition, fine-tuning among multimodal data is evaluated, and we find that it can be considered as an optional skill to prevent overfitting when no pre-trained models exist. The proposed method is verified on the ChaLearn LAP large-scale isolated gesture dataset (IsoGD) and the Sheffield Kinect Gesture (SKIG) dataset. The results show that our proposed method can obtain the state-of-the-art recognition accuracy (51.02% on the validation set of IsoGD and 98.89% on SKIG).**

***Index Terms*—3D convolution, convolutional LSTM, gesture recognition, multimodal**

## I. INTRODUCTION

GESTURES, as a nonverbal body language, play a very important role in human daily life. It undoubtedly will be of great importance in computer vision applications, such as human robot interaction, human computer interaction [44], sign language recognition and virtual reality. Gesture recognition is aimed to recognize and understand meaningful movements of human bodies [5]. Effective gesture recognition is still a very challenging problem [6], partly due to the cultural differences, various observation conditions, noises, relative small size of fingers in images, out-of-vocabulary motions, etc.

In the traditional gesture recognition, handcrafted features and conventional machine learning methods are utilized mostly, e.g. hidden Markov models, particle filtering, finite-state machines and connectionist models [5]. Handcrafted features cannot fulfill the requirements of the practical gesture recognition systems completely, because of the aforementioned challenging factors. With the rapid development of the deep learning theory [10], data-driven methods have demonstrated amazing performances in image classification [13], image segmentation [15], object detection [16], scene recognition [18], face recognition [19], human action recognition [20], and human gesture recognition [21].

Different from the image-based applications, e.g. image classification and scene labeling, gesture recognition is generally based on video or skeleton sequences. Only a small handful of gestures could be recognized from one single static image. Therefore, the temporal information plays a key role in the gesture recognition process. Backgrounds may be an effective hint for scene recognition or action recognition. Unfortunately, complex backgrounds would bring more challenges to gesture recognition, because gestures focus more on the movements of hands and arms. Hands and arms have the relative small size compared with the whole scene, so the effective spatial features of gestures may be overwhelmed in backgrounds. Therefore, the temporal information becomes more discriminating for gesture recognition than video classification [20]. Learning spatiotemporal features simultaneously will be more informative for gesture recognition.

Two-Stream Convolutional Networks in [22] extract spatial and temporal features from RGB and stacked optical flow images separately. Long-term Recurrent Convolutional Networks (LRCN) in [23] first learn spatial features from each frame, and then learn temporal features based on the spatial feature sequences using Recurrent Neural Networks (RNN). VideoLSTM [24] uses convolutional LSTM networks to learn spatiotemporal features from the previously extracted 2D spatial features. These three representative methods learn spatiotemporal features separately or in different stages. Learning spatiotemporal features simultaneously from videos will be more effective for gesture recognition when various backgrounds are taken into consideration. For example, 3D ConvNets [36] utilize 3D Convolutional Neural Networks (3D CNN) to learn spatiotemporal features straightforwardly. Nevertheless, LSTM/RNN is more suitable to learn the long-term temporal information. Therefore, it will be more reasonable to learn short-term spatiotemporal features by 3D CNN, and to learn long-term spatiotemporal features by LSTM/RNN for the long-term dependent applications. Fully-connected features are generally used as the input of LSTM [25], but keeping the spatial correlation information in LSTM processes can learn more informative spatiotemporal features. So, the convolutional LSTM [26] is utilized in our proposed method.

This work is partially supported by the China Postdoctoral Science Foundation (Grant No. 2016M592763), the Fundamental Research Funds for the Central Universities (Grant NO. JB161006, JB161001), and the National Natural Science Foundation of China (Grant NO. 61401324, 61305109).

All authors are with school of software, Xidian University, Xi'an, P.R. China (e-mail: gmzhu@xidian.edu.cn; liangzhang@xidian.edu.cn; pyshen@xidian.edu.cn; songjuan@mail.xidian.edu.cn).

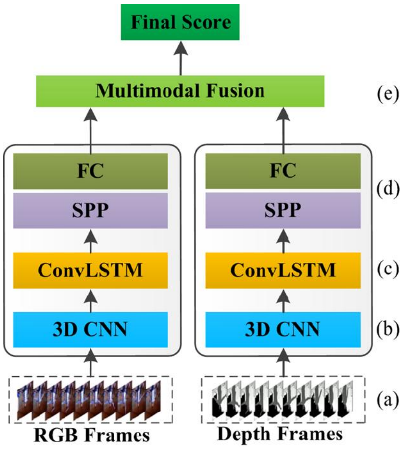

Fig. 1. An overview of the proposed method. The proposed deep architecture is composed of five components: (a) Input Preprocessing, (b) 3D CNN, (c) Convolutional LSTM, (d) Spatial Pyramid Pooling and Fully Connected Layers, (e) Multimodal Fusion.

In this paper, a multimodal gesture recognition method based on 3D convolution and convolutional LSTM is proposed for gesture recognition, as illustrated in Fig. 1. First, 3D CNN is utilized to extract short-term spatiotemporal features from the input video. And then convolutional LSTM is employed to learn long-term spatiotemporal features further. Lastly, Spatial Pyramid Pooling (SPP) [40] is adopted to normalize the spatiotemporal features for the final classification. RGB and depth modalities based networks are trained respectively and their predictions are fused to get the final prediction results.

The main contribution of the paper can be summarized as follows:

a) Multimodal deep architectures based on 3D convolution and convolutional LSTM are proposed originally for isolated gesture recognition.

b) Fine-tuning among multimodal data is evaluated and considered as an optional skill to prevent overfitting when no pre-trained models exist.

c) The state-of-the-art performances on the IsoGD and SKIG datasets are reported.

The remaining of the paper is organized as follows: Section II reviews the related work of gesture recognition. Section III gives the details of the proposed method. Section IV presents the experiments and discussions. Finally, Section V gives the conclusions and future work.

## II. Related Work

In this section, the related work of human gesture recognition will be reviewed from two aspects: handcrafted feature based methods and neural network based methods.

### A. *Handcrafted feature based methods*

Various handcrafted features have been proposed for gesture recognition. Prival et al. [43] separated hands from forearm regions, normalized the hand rotation using the geometry of gestures, and classified gestures based on the Krawtchouk moment features of normalized binary silhouettes. Konecnỳ and Hagara [27] used the dynamic time warping (DTW) method to recognize gestures based on the histograms of oriented gradients (HOG) and histograms of optical flow (HOF). Wu et al. [28] extracted the extended motion history image (Extended-MHI) from RGB and depth sequences, and used maximum correlation coefficient to recognize gestures. Lui [29] characterized gesture videos as points on a Grassmann manifold and adopted the least square regression method for gesture recognition. Wan et al. [30] first proposed the 3D enhanced motion scale invariant feature transform (3D EMoSIFT) and 3D Sparse Motion SIFT (3D SMoSIFT) to extract spatiotemporal features from RGB-D images, and then the 3D EMoSIFT and 3D SMoSIFT features are evaluated under the bag of visual words (BoVW) model. Recently, mixed features around sparse keypoints (MFSK) [31] are proposed to extract spatiotemporal features from RGB-D data. Based on these handcrafted features, linear discriminant analysis (LDA), linear support vector machine (linear SVM), principal component analysis (PCA), nearest neighbors (NN) classifier, DTW, naïve Bayes model, etc. are utilized for gesture recognition [31].

However, handcrafted features cannot take all factors into consideration at the same time. The methods based on the state-of-the-art handcrafted features failed on the *2016 ChaLearn LAP Large-scale Isolated Gesture Recognition Challenge* [1]. At the same time, neural network based methods have demonstrated remarkable performances on the large-scale and challenging gesture dataset [1, 32].

### B. *Neural network based methods*

Recently, deep neural networks (DNN) have been introduced to the field of computer vision applications. Convolutional neural networks (CNN) and RNN based methods have demonstrated the state-of-the-art performances on human gesture/action recognition [6, 32, 33]. The key of neural network based gesture recognition methods is to learn the spatiotemporal features. The most obvious approach is to learn spatial and temporal features consecutively. Pigou et al. [34] explored five kinds of deep architectures for gesture recognition in video, and showed that LRCN-style networks are not optimal for gesture recognition. They also demonstrated that bidirectional recurrence and temporal convolutions can improve frame-wise gesture recognition significantly. Another obvious approach is to extend 2D CNN to 3D CNN [35]. C3D [36] based networks demonstrated the state-of-the-art performance on the *2016 ChaLearn LAP Large-scale Isolated Gesture Recognition Challenge* [7, 12, 32]. Li et al. [12] applied the convolutional 3D (C3D) model on the RGB and depth data respectively. Zhu et al. [7] embedded pyramidal input and pyramidal fusion strategies into the C3D model for gesture recognition. Molchanov et al. [25] proposed recurrent 3D convolutional neural networks which integrate 3D CNN and RNN for gesture recognition. Moreover, two-stream based networks have obtained remarkable performances for human action recognition [20]. Duan et al. combined a convolutional two-stream consensus voting network and a 3D depth-saliency ConvNet for gesture recognition in [33]. Their method obtained the state-of-the-art performances on the Chalearn IsoGD and

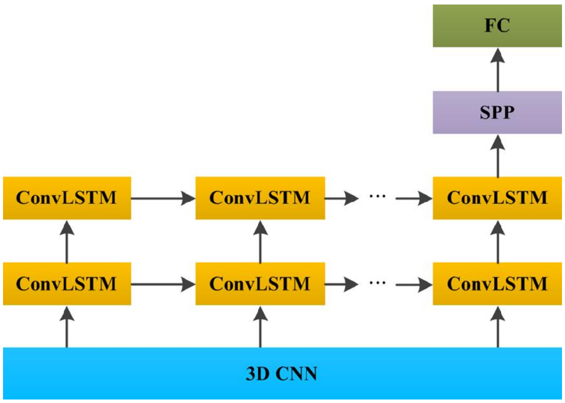

Fig. 2. Overview of the proposed network.

RGBD-HuDaAct datasets. In addition, converting videos into 2D images is also a popular way to apply the state-of-the-art image-based CNN architectures to the video-based applications. Wang et al. [4] constructed dynamic depth images, dynamic depth normal images, and dynamic depth motion normal images, and then fine-tuned VGG-16 networks [37] on these images for gesture recognition.

Generally, CNN+LSTM [23] or 3D CNN+RNN [25] networks utilize the fully *connected LSTM (FC-LSTM).* FC-LSTM uses full connections in input-to-state and state-to-state transitions, so the spatial correlation information is not encoded. Li et al. [38] proposed VideoLSTM which performs LSTM on the two-dimensional spatial features directly, but the 2D convolutional features of VGG-16 do not take the short-term temporal information into consideration. Because of the relative small size of hands in the complex backgrounds, the absence of temporal information may cause that the effective features of hands are not well extracted. So, learning spatiotemporal features simultaneously is a better choice for gesture recognition. C3D based networks [7, 12, 36] can learn spatiotemporal features simultaneously, but LSTM/RNN is more suitable for the long-term dependent applications. Therefore, convolutional LSTM is better to learn long-term spatiotemporal features.

In order to take full use of the advantages of 3D CNN and convolutional LSTM, these two networks are utilized in the proposed deep architecture to learn short-term and long-term spatiotemporal features respectively.

## III. Proposed Method

As illustrated in Figs. 1 and 2, the proposed deep architecture is composed of five components: input preprocessing, 3D CNN, convolutional LSTM, spatial pyramid pooling and multimodal fusion.

### *A. Input Preprocessing*

Generally, gestures contain three temporally overlapping phases: preparation, nucleus, and retraction [38]. Different individuals may perform gestures with various speeds. These two factors cause that gesture sequences may have various lengths. However, almost all the gesture recognition neural networks require that the input has the same size. Therefore, length normalization of inputs is necessary.

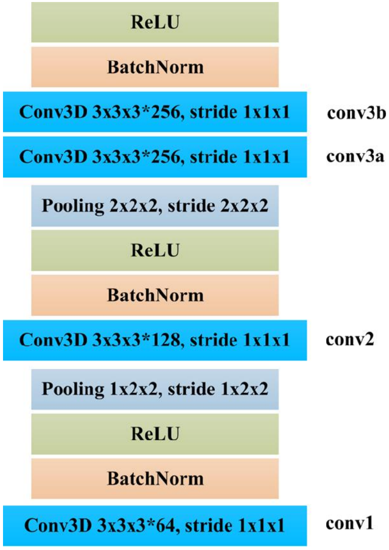
ReLU
BatchNorm
Conv3D 3x3x3*256, stride 1x1x1
conv3b
Conv3D 3x3x3*256, stride 1x1x1
conv3a
Pooling 2x2x2, stride 2x2x2
ReLU
BatchNorm
Conv3D 3x3x3*128, stride 1x1x1
conv2
Pooling 1x2x2, stride 1x2x2
ReLU
BatchNorm
Conv3D 3x3x3*64, stride 1x1x1
conv1

Fig. 3. The 3D CNN component

One way is to split each gesture sequence into clips which have the fixed length, but one clip cannot represent the whole gesture. Another way is to down-sample each gesture sequence into a fixed length $L$. The second way is utilized in the proposed method. At the same time, the uniform sampling with temporal jitter strategy is used to augment the dataset. Specifically, given one gesture sequence which has $S$ frames, the sampling process can be described as:

$$Idx_i = \frac{S}{L} * (i + jit / 2) \tag{1}$$

where $Idx_i$ is the index of the $i$ th sampled frame, and $jit$ is a random value sampled from the uniform distribution between -1 and 1. The temporal jitter can augment the dataset without disturbing the timing sequence of the sampled frames of each gesture. The sampling result can be represented as

$$US = \{Idx_1, Idx_2, ..., Idx_L\} \tag{2}$$

### *B. 3D CNN*

C3D [36] is a representative 3D convolutional neural network for human action recognition. The 3D CNN component in the proposed deep architecture is designed by referring to the C3D model, as displayed in Fig. 3. Batch normalization [39] is also utilized to accelerate deep network training. Batch Normalization allows us to use much higher learning rates and be less careful about initialization. Experiment in this study also demonstrates that there will be a substantial speedup in training when using batch normalization. The kernel size[1] of each Conv3D layer is $3\times3\times3$, the stride and padding of each Conv3D layer are all size of $1\times1\times1$. The filter counts of the four Conv3D layers are 64, 128, 256, 256, respectively. Each Conv3D layer, except the conv3a, is followed by a batch normalization layer and a ReLU layer. The first pooling layer before conv2 has kernel size $1\times2\times2$ and

[1] The kernel size of all 3D Convolutional layers and 3D Pooling layers is in length*height*width format.

stride $1\times2\times2$. This means that only the spatial pooling is performed on the first Conv3D layer. The second pooling layer has kernel size $2\times2\times2$ and stride $2\times2\times2$. This means that spatiotemporal pooling is performed on the second Conv3D layer. These two pooling layers make the output size of the 3D CNN component shrank by the ratios of 4 and 2 on the spatial size and the temporal length respectively. It means that the 3D CNN component only learns the short-term spatiotemporal features, as we stated ahead.

### C. Convolutional LSTM

The traditional *fully connected LSTM* does not take spatial correlation into consideration. However, the *convolutional LSTM* (ConvLSTM) has convolutional structures in both the input-to-state and state-to-state transitions, which can model the spatiotemporal relationships quite well [26].

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 whose last two dimensions are spatial dimensions (rows and columns). Let '*' denote the convolution operator, and let 'o' denote the Hadamard product. The ConvLSTM can be formulated as:

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + b_i) \tag{3}$$

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + b_f) \tag{4}$$

$$o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + b_o) \tag{5}$$

$$C_t = f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \tag{6}$$

$$H_t = o_t \circ \tanh(C_t) \tag{7}$$

where $\sigma$ is the sigmoid function, $W_{x\sim}$ and $W_{h\sim}$ are 2-d convolutional kernels.

As illustrated in Fig. 2, two-level ConvLSTM is deployed in the proposed algorithm. The final output of the high level ConvLSTM layer is considered as the final long-term spatiotemporal features for each gesture. So, the temporal length of the final spatiotemporal features will be 1. The convolutional kernel size is $3\times3$ with stride $1\times1$. The convolutional filter counts of the two-level ConvLSTM layers are 256 and 384, respectively. "Same-Padding" is performed during the convolution processes of ConvLSTM in our implementation, so the spatiotemporal features at different stages of ConvLSTM have the same spatial size. Specifically, the output of ConvLSTM has the same spatial size as the output of 3D CNN in the proposed deep architecture.

### D. Spatial Pyramid Pooling

Because the 3D CNN component shrinks images only with a small ratio of 4 on the spatial domain and the ConvLSTM component does not change the spatial size of feature maps, the final long-term spatiotemporal feature maps have relative high spatial size (e.g., $28\times28$ in our implementation as the input size of 3D CNN is $112\times112$). The spatial pyramid pooling [40] is inserted between the ConvLSTM and fully connected (FC) layers to reduce the dimensionality; so that the final FC layers can have less parameter. Spatial pyramid pooling, as an extension of the Bag-of-Words model, is one of the most successful methods in computer vision. It can pool features at multi-level and make it possible to generate representations from arbitrarily sized images.

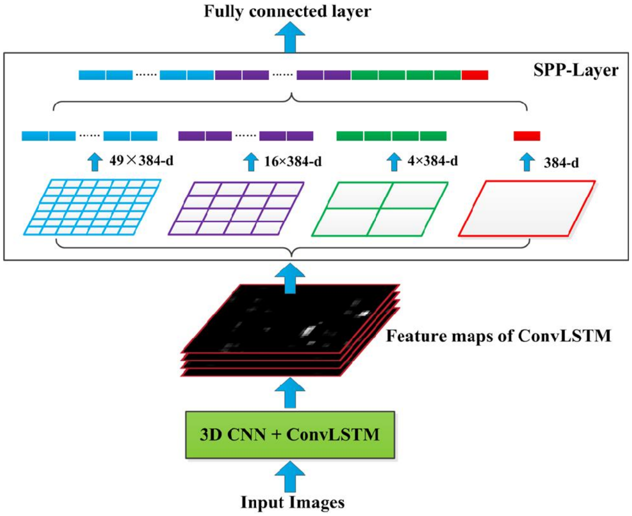

Fig. 4. Spatial Pyramid Pooling layer

As displayed in Fig. 4, four-level spatial pyramid pooling is performed on each feature map in the proposed deep architecture. The spatial size of the final long-term spatiotemporal feature maps of ConvLSTM is $28\times28=784$, and the dimension of each feature map after the SPP layer is $49+16+4+1=70$. Besides the dimensionality reduction, the multi-scale features extracted by SPP can also improve the recognition accuracy in some degree.

### E. Multimodal Fusion

Multimodal fusion can be generally divided into two categories: early multimodal fusion and late multimodal fusion [14]. Early multimodal fusion integrates multimodal data before inputs of networks. This approach may be problematic, because some multimodal data cannot be fused directly as they may be unadjusted and do not have consistent characteristics. Late multimodal fusion integrates multimodal data at the late stage of networks. This approach makes it possible that different networks can be trained according to characteristics of the data, respectively. In the proposed method, late multimodal fusion is adopted, and the predictions of different networks are fused by averaging to obtain the final prediction scores.

## IV. Experiments

In this section, the proposed method will be evaluated systematically on two public datasets: the ChaLearn LAP large-scale isolated gesture dataset (IsoGD) [1] and the Sheffield Kinect Gesture (SKIG) dataset [2]. First, the two datasets will be described briefly. And then, the training processes will be described in detail. Finally, the evaluation results will be reported respectively.

### A. Dataset Description

**IsoGD** [1] is a large-scale isolated gesture dataset which is derived from the ChaLearn Gesture Dataset (CGD) [41]. The dataset includes 47,933 RGB+D gesture videos, and each RGB

TABLE I
INFORMATION OF THE ISOGD DATASET

| Subsets | #of Gestures | #of RGB Videos | #of Depth Videos | #of Performers |
|---|---|---|---|---|
| Training | 35878 | 35878 | 35878 | 17 |
| Validation | 5784 | 5784 | 5784 | 2 |
| Testing | 6271 | 6271 | 6271 | 2 |

or depth video file represents one gesture instance. The dataset includes 249 kinds of gestures performed by 21 different individuals. The 249 kinds of gestures include: 1) body language gestures, 2) gesticulations performed to accompany speech, 3) illustrators like Italian gestures, 4) emblems like Indian Mudras, 5) sign languages for the deaf, 6) signals, 7) actions, 8) pantomimes, and 9) dance postures [1, 6]. All the videos are divided into three mutually exclusive subsets, as illustrated in TABLE I. No videos performed by the same person appear both in the training and validation/testing subsets. The labels of the testing subset have not been publicly available yet, so the validation subset will be used to evaluate the proposed method.

**SKIG** [2] contains 1080 RGB+D videos which belong to 10 gesture categories. All gestures are performed by 6 subjects with 3 kinds of hand postures (i.e., fist, flat and index) under 2 illumination conditions (i.e., strong and poor light) and 3 backgrounds (i.e., white plain paper, wooden board and paper with characters). The dataset is divided into three subsets according to the subjects: subject1+subject2, subject3+subject4, subject5+subject6. The 3-fold cross-validation as [2] is used to evaluate the proposed method.

### B. Network Training

The proposed networks[2] are implemented based on the Tensorflow and Tensorlayer platforms. No pre-trained models are compatible with the proposed deep architecture, so the networks are trained from scratch. Batch normalization makes training processes easier and faster. Therefore, higher learning rates are used and fewer epochs are needed. We train the networks first on the IsoGD dataset from scratch. The initial learning rate is set to 0.1 and dropped to its 1/10 every 15,000 iterations. The weight decay is initialized as 0.004 and decreases to 0.00004 after 40,000 iterations. At most 60,000 iterations are needed for the training on IsoGD. Then, the networks are fine-tuned for SKIG based on the pre-trained models of IsoGD. The initial learning rate is 0.01 and dropped to its 1/10 every 5,000 iterations for SKIG. The weight decay is set to 0.00004 during the whole fine-tuning process. At most 10,000 iterations are needed for the fine-tuning on SKIG.

For both IsoGD and SKIG, the batch size is 13, the temporal length of each clip is 32 frames, and the crop size for each image is 112. One NVIDIA TITAN X GPU is used to train each network. Uniform sampling with temporal jitter as described in Section III-A is used for training. Only uniform sampling is used for testing to keep the testing accuracy consistent. RGB and depth modalities based networks are trained separately.

TABLE II
COMPARISON OF RECOGNITION ACCURACY OF DIFFERENT TRAINING STRATEGIES ON THE VALIDATION SUBSET OF ISOGD

| Training Strategy | Modality | Accuracy(%) |
|---|---|---|
| Strategy 1 | RGB Only | 34.00 |
| Strategy 2 | RGB Only | 37.63 |
| | Depth Only | 35.72 |
| | RGB+Depth | 43.07 |
| Strategy 3 | RGB Only | 43.88 |
| | Depth Only | 44.66 |
| | RGB+Depth | **51.02** |

As no pre-trained models on other datasets are used in our training, a cross-modality fine-tuning strategy for IsoGD is evaluated in the experiments. We fine-tune the RGB based neural network based on the pre-trained model of the depth modality, vice versa. Several different training strategies are utilized to evaluate the proposed method when training on IsoGD:

**Strategy 1**: Adding an extra 3D pooling layer (with $2\times2\times2$ kernel and $2\times2\times2$ stride) on the top of the 3D CNN component (as displayed in Fig. 3) to evaluate the influence of the spatial size on ConvLSTM. In such case, the spatial size of the final spatiotemporal feature maps is $14\times14$, so only 3-level spatial pyramid pooling (i.e., the number of bins is 1, 4, 16, respectively) is utilized.

**Strategy 2**: Training the RGB and depth based networks from scratch on IsoGD, respectively.

**Strategy 3**: Fine-tuning the RGB based neural network based on the pre-trained model of the depth modality for IsoGD, vice versa.

### C. Evaluation on IsoGD

Three different training strategies are evaluated on IsoGD, as illustrated in TABLE II. The results show that larger spatial size of inputs for ConvLSTM results in better recognition accuracy. This also demonstrates the importance of the spatial correlation information for gesture recognition.

The proposed deep architecture is designed originally, so no pre-trained models exist. Fine-tuning on pre-trained models is an important skill to prevent overfitting when no enough data exists for training from scratch. The RGB and depth based networks are trained from scratch on the IsoGD dataset firstly, and the testing results are listed as "Strategy 2" in TABLE II. Then, we fine-tune the RGB based network on the pre-trained model of the depth modality, vice versa. In particular, the parameters of the fully connected layer are learned from scratch during fine-tuning; although the RGB and depth based networks have the same gesture categories. The results in TABLE II show that the fine-tuning processes improve the performances of the two neural networks significantly. RGB and depth datasets have different kinds of information, although they all belong to the same IsoGD dataset. So, fine-tuning among multimodal data can also be considered as

[2] The code of the proposed network has been released on the Github website: https://github.com/GuangmingZhu/Conv3D_CLSTM

TABLE III
COMPARISON OF PROPOSED METHOD AND OTHER METHODS ON THE VALIDATION SUBSET OF ISOGD

| Methods | Accuracy(%) |
|---|---|
| MFSK [1] | 18.65 |
| MFSK+Deep ID [1] | 18.23 |
| Wang et al. [4] | 39.23 |
| Pyramidal C3D (RGB Only) [7] | 36.58 |
| Pyramidal C3D (Depth Only) [7] | 38.00 |
| Pyramidal C3D (RGB+Depth) [7] | 45.02 |
| Li et al. (RGB Only) [12] | 37.3 |
| Li et al. (Depth Only) [12] | 40.5 |
| Li et al. (RGB+Depth) [12] | 49.2 |
| Proposed (RGB Only) | 43.88 |
| Proposed (Depth Only) | 44.66 |
| **Proposed (RGB+Depth)** | **51.02** |

TABLE IV
COMPARISON OF PROPOSED METHOD AND OTHER METHODS ON SKIG DATASET

| Methods | Accuracy(%) |
|---|---|
| RGGP + RGB-D [2] | 88.7 |
| Choi et al. [3] | 91.9 |
| 4DCOV [8] | 93.8 |
| Depth Context [9] | 95.37 |
| Tung et al. [11] | 96.7 |
| MRNN (depth only) [14] | 95.9 |
| MRNN [14] | 97.8 |
| DLEH$^2$ (DLE+HOG$^2$ ) [17] | 98.43 |
| Proposed (RGB Only) | 95.93 |
| Proposed (Depth Only) | 98.70 |
| **Proposed (RGB+Depth)** | **98.89** |

an optional skill to prevent overfitting when no pre-trained models exist.

TABLE III displays the comparison results with the previously published methods on the validation subset of IsoGD, because the labels of the testing subset have not been released publicly. TABLE III shows that the proposed method outperforms other methods. Both Zhu et al. [7] and Li et al. [12] trained their models based on the pre-trained model on the large-scale Sports-1M dataset [36], but the proposed networks are only trained on the IsoGD dataset. This superiority also partly demonstrates that LSTM is more suitable to learn long-term temporal dependency. Besides, at most 60,000 iterations are needed when we train from scratch or fine-tune the proposed networks on the IsoGD dataset. The training is extremely fast for the neural networks which have 22M parameters.

The total validating time on the RGB and depth datasets of IsoGD is about 32 minutes for 5784 video files. This means that the proposed networks can recognize gestures at 95.8 fps on a NVIDIA TITAN X GPU.

## D. Evaluation on SKIG

The performance of the proposed method on the SKIG dataset is shown in TABLE IV. It can be seen that the proposed method obtains the state-of-the-art accuracy of 98.70%, even when only the depth data is utilized. The proposed multimodal gesture recognition method demonstrates the state-of-the-art accuracy of 98.89%, which outperforms all the previously published methods. The multi-stream recurrent neural network (MRNN) [14] learns spatial features using 2D CNN, and then feeds the spatial features into MRNN for gesture recognition. Learning spatiotemporal simultaneously is more suitable than learning spatial and temporal features consecutively for gesture recognition. The fact that the proposed method outperforms MRNN which has deeper architectures, also demonstrates the conclusion. TABLE IV shows that the multimodal fusion strategy only contributes tiny improvement on the recognition accuracy of SKIG. This is because the depth-based network has already obtained extremely high recognition accuracy on SKIG, and the rest of the room to improve the recognition accuracy is very little.

## E. Discussion

Generally, backgrounds are less informative for gesture recognition when object affordances [42] are not involved in gestures. In such case, complex backgrounds bring negative influences to effective gesture recognition. Therefore, learning spatiotemporal features simultaneously becomes the key of effective gesture recognition methods. 3D convolutional neural networks are well designed for the spatiotemporal feature extraction, and LSTM networks are more suitable for variable length temporal information fusion. Therefore, the integration of 3D CNN and convolutional LSTM may be an excellent framework for robust gesture recognition.

Fine-tuning on pre-trained models is an important skill to prevent overfitting for relative small datasets, and the essence of fine-tuning on pre-trained models is to involve more data for training. Multimodal data of gestures are captured by different ways and represent different characteristics of gestures from different aspects. So, multimodalities can also be viewed as a special data-augment method. Cross-modality fine-tuning can also be considered as another practical skill to prevent overfitting.

According to the recognition results of IsoGD, some kinds of gestures are very difficult to recognize for the proposed method. 1) The proposed method does not separate region of hands from the whole scene, and the random down-sampling strategy lose some effective motion information of fast and tiny movements after length normalization of inputs, so gestures with fast and tiny movements of hands are difficult to recognize. 2) Uniform down-sampling cannot reserve all the key motion information when most of the frames only contain meaningless static gesture, thus such kind of gestures are also difficult to recognize. 3) Gesture sequences with terrible illumination cannot be well recognized in the experiments. 4) Very similar gestures are also hard to distinguish. Multi-scale features can improve the recognition accuracy of gestures with fast and tiny movements, if global and local features can be learned at the same time. Down-sampling or normalization according to the effectiveness of movements may be an optional skill to reserve useful motion information for gesture recognition. Dynamic recurrent neural networks may be a better choice to learn effective spatiotemporal features for gestures which have various length and random performing time and speed.

## V. Conclusion

In this paper, we present a multimodal gesture recognition method based on 3D convolutional neural networks and convolutional Long-Short-Term-Memory (LSTM) networks. The evaluation results demonstrate that learning spatiotemporal features simultaneously is more suitable than learning spatial and temporal features consecutively or separately for gesture recognition. Spatiotemporal features are more robust to complex backgrounds of gestures. 3D convolutional neural networks are good options to learn short-term spatiotemporal features, and convolutional LSTM networks are better choices for long-term spatiotemporal learning. In the future, we will try to learn the dynamic image represented by one spatiotemporal feature map for each gesture, by replacing the spatial pyramid pooling layer of the proposed deep architecture with convolutional networks. Gestures always have various lengths, so dynamic recurrent neural networks can be used for continuous gesture recognition in our future works.

## References

[1] 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 IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[2] L. Liu and L. Shao, "Learning discriminative representations from RGB-D video data," in 23rd International Joint Conference on Artificial Intelligence, 2013, pp. 1493-1500.

[3] H. Choi and H. Park, "A hierarchical structure for gesture recognition using RGB-D sensor," in the second international conference on Human-agent interaction, 2014.

[4] P. Wang, W. Li, S. Liu, Z. Gao, C. Tang, and P. Ogunbona, "Large-scale Isolated Gesture Recognition Using Convolutional Neural Networks," in 2016 23rd International Conference on Pattern Recognition (ICPR), 2016, pp. 7-12.

[5] S. Mitra and T. Acharya, "Gesture recognition: A survey," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 37, pp. 311-324, 2007.

[6] S. Escalera, V. Athitsos, and I. Guyon, "Challenges in multimodal gesture recognition," Journal of Machine Learning Research, vol. 17, pp. 1-54, 2016.

[7] G. Zhu, L. Zhang, L. Mei, J. Shao, J. Song, and P. Shen, "Large-Scale Isolated Gesture Recognition Using Pyramidal 3D Convolutional Networks," in 2016 23rd International Conference on Pattern Recognition (ICPR), 2016, pp. 19-24.

[8] P. Cirujeda and X. Binefa, "4DCov: A Nested Covariance Descriptor of Spatio-Temporal Features for Gesture Recognition in Depth Sequences," in 2014 2nd International Conference on 3D Vision, 2014, pp. 657-664.

[9] M. Liu and H. Liu, "Depth Context: a new descriptor for human activity recognition by using sole depth sequences," Neurocomputing, vol. 175, pp. 747-758, 2016.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436-444, 2015.

[11] P. T. Tung and L. Q. Ngoc, "Elliptical density shape model for hand gesture recognition," in the Proceedings of the Fifth Symposium on Information and Communication Technology, 2014.

[12] Y. Li, Q. Miao, K. Tian, Y. Fan, X. Xu, R. Li, et al., "Large-scale Gesture Recognition with a Fusion of RGB-D Data Based on the C3D model," in 2016 23rd International Conference on Pattern Recognition (ICPR), 2016, pp. 25-30.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems, 2012, pp. 1097-1105.

[14] N. Nishida and H. Nakayama, "Multimodal Gesture Recognition Using Multi-stream Recurrent Neural Network," in pacific-rim symposium on image and video technology, 2015, pp. 682-694.

[15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431-3440.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580-587.

[17] 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, pp. 1-20, 2016.

[18] M. Liang, X. Hu, and B. Zhang, "Convolutional Neural Networks with Intra-layer Recurrent Connections for Scene Labeling," in Advances in Neural Information Processing Systems, 2015, pp. 937-945.

[19] Y. Sun, D. Liang, X. Wang, and X. Tang, "Deepid3: Face recognition with very deep neural networks," arXiv preprint arXiv:1502.00873, 2015.

[20] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, and D. Lin, "Temporal Segment Networks: Towards Good Practices for Deep Action Recognition," arXiv preprint arXiv:1608.00859, 2016.

[21] G. Lefebvre, S. Berlemont, F. Mamalet, and C. Garcia, "Inertial Gesture Recognition with BLSTM-RNN," in Artificial Neural Networks, ed: Springer, 2015, pp. 393-410.

[22] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in Advances in Neural Information Processing Systems(NIPS), 2014, pp. 568-576.

[23] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, et al., "Long-term recurrent convolutional networks for visual recognition and description," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2625-2634.

[24] Z. Li, E. Gavves, M. Jain, and C. G. M. Snoek, "Videolstm convolves, attends and flows for action recognition," arXiv preprint arXiv:1607.01794, 2016.

[25] 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 IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4207-4215.

[26] 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 Advances in Neural Information Processing Systems, 2015, pp. 802--810.

[27] J. Konecny and M. Hagara, "One-shot-learning gesture recognition using HOG-HOF features," Journal of Machine Learning Research, vol. 15, pp. 2513-2532, 2014.

[28] D. Wu, F. Zhu, and L. Shao, "One shot learning gesture recognition from RGBD images," in IEEE Conference on Computer Vision and Pattern Recognition, 2012.

[29] Y. M. Lui, "Human gesture recognition on product manifolds," Journal of Machine Learning Research, vol. 13, pp. 3297-3321, 2012.

[30] J. Wan, Q. Ruan, W. Li, G. An, and R. Zhao, "3D SMoSIFT: three-dimensional sparse motion scale invariant feature transform for activity recognition from RGB-D videos," Journal of Electronic Imaging, vol. 23, p. 023017, 2014.

[31] J. Wan, G. Guo, and S. Z. Li, "Explore Efficient Local Features from RGB-D Data for One-Shot Learning Gesture Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, pp. 1626-1639, 2016.

[32] H. J. Escalante, V. Ponce-Lpez, J. Wan, M. A. Riegler, B. Chen, A. Claps, et al., "Chalearn joint contest on multimedia challenges beyond visual analysis: An overview," in Proceedings of ICPRW, 2016.

[33] 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, vol. abs/1611.06689, 2016.

[34] L. Pigou, A. v. d. Oord, S. Dieleman, M. Van Herreweghe, and J. Dambre, "Beyond Temporal Pooling: Recurrence and Temporal

Convolutions for Gesture Recognition in Video," arXiv preprint arXiv:1506.01911, 2015.

[35] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, pp. 221-231, 2013.

[36] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 4489-4497.

[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[38] D. Gavrila, "The Visual Analysis of Human Movement: A Survey," Computer Vision and Image Understanding, vol. 73, pp. 82-98, 1999.

[39] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in International Conference on Machine Learning, 2015, pp. 448-456.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, pp. 1904-1916, 2015.

[41] I. Guyon, V. Athitsos, P. Jangyodsuk, and H. J. Escalante, "The ChaLearn gesture dataset (CGD 2011)," Machine Vision Applications, vol. 25, pp. 1929-1951, 2014.

[42] H. S. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from RGB-D videos," International Journal of Robotics Research, vol. 32, pp. 951-970, 2013.

[43] S. P. Priyal and P. K. Bora, "A robust static hand gesture recognition system using geometry based normalizations and Krawtchouk moments," Pattern Recognition, vol. 46, pp. 2202-2219, 2013.

[44] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," Artificial Intelligence Review, vol. 43, pp. 1-54, 2015.