Continuous Gesture Recognition with Hand-oriented Spatiotemporal Feature

Zhipeng Liu, Xiujuan Chai, Zhuang Liu, Xilin Chen
Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, 100190, China
University of Chinese Academy of Sciences, Beijing, 100049, China
Cooperative Medianet Innovation Center, China
{zhipeng.liu, xiujuan.chai, zhuang.liu, xilin.chen}@vipl.ict.ac.cn

Abstract

In this paper, an efficient spotting-recognition framework is proposed to tackle the large scale continuous gesture recognition problem with the RGB-D data input. Concretely, continuous gestures are firstly segmented into isolated gestures based on the accurate hand positions obtained by two streams Faster R-CNN hand detector. In the subsequent recognition stage, firstly, towards the gesture representation, a specific hand-oriented spatiotemporal (ST) feature is extracted for each isolated gesture video by 3D convolutional network (C3D). In this feature, only the hand regions and face location are considered, which can effectively block the negative influence of the distractors, such as the background, cloth and the body and so on. Next, the extracted features from calibrated RGB and depth channels are fused to boost the representative power and the final classification is achieved by using the simple linear SVM. Extensive experiments are conducted on the validation and testing sets of the Continuous Gesture Datasets (ConGD) to validate the effectiveness of the proposed recognition framework. Our method achieves the promising performance with the mean Jaccard Index of 0.6103 and outperforms other results in the ChaLearn LAP Large-scale Continuous Gesture Recognition Challenge.

1. Introduction

In recent years, gesture recognition has gained a great deal of attention because of its great potential applications, such as sign language translation, human computer interactions, robotics, virtual reality and so on. However, it still remains challenging due to its complexity of the gesture activities from the large scale body motion to tiny finger motion and also various of hand postures.

The continuous evolution of gesture recognition technique is accompanied by the development of the data capture sensors. From the literatures, three kinds of visual data capture sensors are used for gesture recognition, which are data glove, video camera and depth camera. In the early stage, researchers utilize data glove equipped with 3D trackers and accelerator sensors to collect the various information of hand shape and position [24, 13]. Although the data gloves can provide accurate hand data, it is very expensive and inconvenient for the user, which limits the wide use of the data glove in our daily life. Therefore, some researchers replace data glove with normal video cameras to make the process of collecting hand data more convenient. Wang et al. [26] collect hand data with web cameras and develop sign retrieval system with a vocabulary of 1113 signs. However, it is difficult for pure video based method to obtain accurate hand tracking and segmentation due to the complicated illuminations and backgrounds. With the emergency of novel sensors, depth information is obtained easily. Microsoft Kinect [30] frees signer from data glove by providing accurate depth information as well as color images simultaneously. Intrinsically, depth and color information characterizes the change of limbs' distance and the static appearance of limbs respectively. The multi-channel data form more powerful gesture representation than single modality. Therefore, more and more researchers focus on how to use the RGB-D data to boost the performance of gesture recognition and several RGB-D gesture databases are released. Among them, ChaLearn LAP RGB-D Continuous Gesture Dataset (ConGD) is a large dataset with clear testing protocols and a challenge is organized based on it [23, 5].

In this paper, a spotting-recognition framework is proposed to solve the continuous gesture recognition problem with the RGB-D data input. Given a continuous gesture sequence, the contained isolated gestures are segmented first with the precise hand detection. Then for each isolated gesture, the specific spatiotemporal feature toward gesture representation is extracted by C3D model, which only considers the hand regions and face location in each image frames.

Compared with the feature from the whole image input, this hand-oriented feature alleviates the non-meaningful distractor regions, such as the background, clothing and body and so on. Besides the hand regions, the face region is also considered as a reference location to characterize the relative motion pattern of the hands. Finally, the fused feature from RGB and depth channels is fed into linear SVM classifier to get the gesture label.

The contribution of our work mainly lies in the following three aspects. Firstly, a novel two streams Faster R-CNN (2S Faster R-CNN) hand detector is developed to get accurate hand regions by integrating RGB and depth inputs. Secondly, the hand-oriented C3D feature effectively characterizes gestures, including the hand postures and the motion trajectories. Finally, the whole spotting-recognition framework is validated in the Continuous Gesture Datasets (ConGD) and shows promising performance.

The remainder of this paper is organized as follows: Section 2 briefly reviews the related work on continuous gesture recognition. Section 3 introduces our proposed method on continuous gesture recognition problem. The experimental results and discussion are presented in Section 4 and Section 5 concludes the paper.

2. Related Work

Generally speaking, continuous gesture recognition is more challenging than isolated gesture recognition for the vague boundaries of the contained isolated gestures in the unsegmented sequences. Therefore, the temporal segmentation is one of the primary problems in continuous gesture recognition task.

There are broadly two kinds of solutions for temporal segmentation. One is to tackle the segmentation and recognition problems simultaneously. The other is to detect the boundaries directly to realize the segmentation. In the first category, the dynamic programming [2] and viterbi decoding [17] were commonly used to segment continuous gesture sequences. Celebi et al. [2] proposed a weighted dynamic time warping (DTW) for gesture recognition, which time-warps an oberverd motion sequence of weighted body joints to pre-stored gesture sequences. In addition, Pitsikalis et al. [17] extracted time boundary information in sign language via the statistical sub-unit construction and decoding. With the rapid development of deep learning, Koller et al. [12] proposed a multilayer bi-directional Long Short Term Memory (BLSTM) that was trained end-to-end with a deep Convolutional Neural Network (CNN). The joint model was embedded into a Hidden Markov Model (HMM) for iterative refinement and final continuous gesture recognition. As for the second category, the main strategy is to tackle the temporal segmentation and recognition in separated steps, i.e. the segmented gestures will be fed into recognition module and output the recognition results. Such approaches are usually based on the hypothesis that the certain boundaries can be determined by some rules in continuous gesture sequence. Movement Epenthesis (ME) is a transition period between two adjacent signs. Gao et al. [6] realized ME detection for gesture segmentation. Instead of the explicitly ME modeling, Yang et al. [28] proposed an adaptive threshold method to filter out the ME periods. With the assumption that significant motion will occur when a true gesture begins, Molchanov et al. [15] employed the radar sensor to collect velocity of movement and segment dynamic gestures. Similarly, Jiang et al. [9] proposed a method based on quantity of movement (QOM) by assuming the same start pose among different gestures. In a summary, the second category of methods will work well if the transition is explicit. The key advantage is to simplify continuous gesture recognition problem into isolated gesture recognition. In this paper, we also adopt the scheme to realize the temporal segmentation with the hand positions as illustrated in Chai et al [3].

Apart from temporal segmentation, the feature of gesture also plays an important role in continuous gesture recognition. In early works, the traditional handcrafted features like Local Binary Pattern (LBP) and Histogram of Oriented Gradients (HOG) are widely used to characterize the shape of hand or body [11, 27]. Then spatiotemporal domain are taken into count to design more effective features for video data. For example, Wan et al. [22] extended the scale invariant feature transform (SIFT) into three dimensions and proposed three-dimensional sparse motion scale invariant feature transform for activity recognition from RGB-D videos. Wang et al. [25] proposed the Grassmann Covariance Matrix (GCM) to model the gesture videos, in which representation, covariance matrix was used to calculate the distance between gesture samples in the Grassmannian Manifold. Recently, convolutional neural network has made a great breakthrough on computer vision related tasks for its powerful feature extraction ability, such as image classification, object detection and semantic segmentation. Thus many researchers used CNN features instead of the handcrafted features for better performance. Simonyan et al. [19] proposed a two-stream convolutional network architecture which incorporated spatial and temporal networks. Their two-stream network took static image and optical flow as input to capture the complementary information on appearance from static frames and motion between frames. Then, the extracted temporal and spatial features were fused as final feature for action classification. Tran et al. [21] extended 2D convolution into 3D convolution, which was capable of learning both the spatial and the temporal aspect of videos. The 3D convolutional networks were also exploited in the gesture recognition area [14, 16]. Considering on the good performance, we further propose a hand-oriented C3D features in this paper for effective gesture representation.

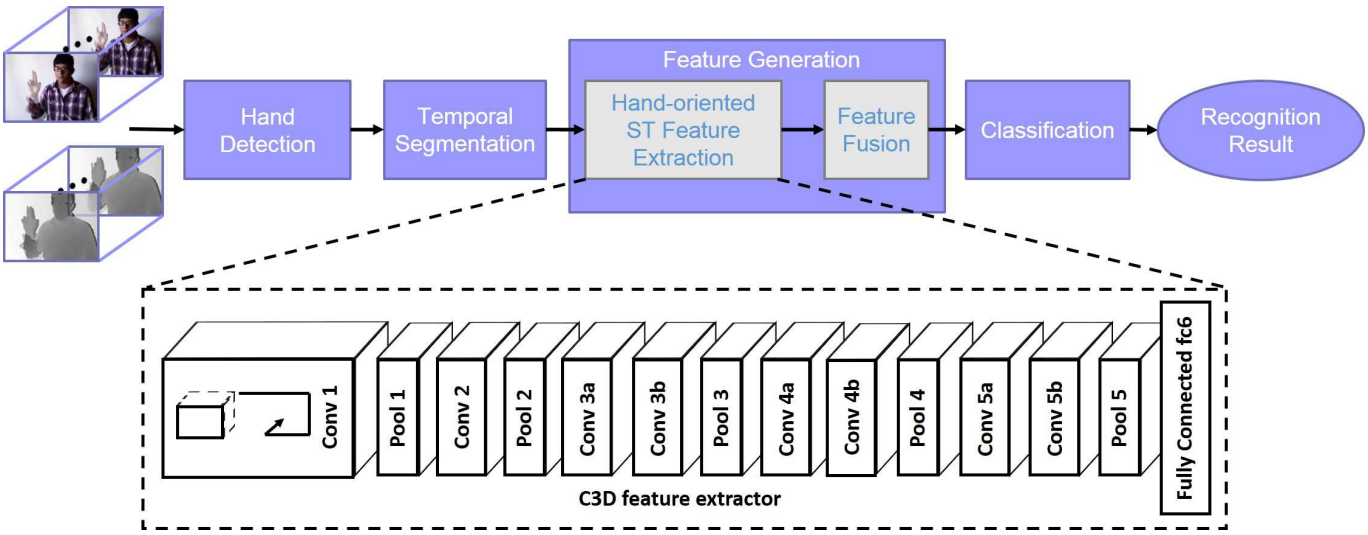

Figure 1. The pipeline of our continuous gesture recognition framework.

3. Methodology

In this section, we will describe the spotting-recognition framework for large scale continuous gesture recognition. Figure 1 shows the pipeline of our whole framework.

Simply speaking, our method is mainly composed of the following four modules. First is a key preprocessing step, i.e. hand detection with two streams Faster R-CNN. Second is the temporal segmentation based on the hand positions. Third is the feature generation module, including the hand-oriented spatiotemporal (ST) feature extraction with C3D model for each segmented gestures and the following feature fusion. Finally is the gesture classification module with simple linear SVM classifier.

3.1. Hand Detection with Two Streams Faster R-CNN

Hand detection is very crucial for our temporal segmentation, and also for the subsequent recognition module. In order to effectively utilize the visual information provided by different channels, i.e. RGB and depth, we propose a two streams Faster R-CNN detection method. This section mainly describes the detection framework. Its entire testing process is shown in Fig. 2.

Although RGB and depth videos are obtained concurrently, they maybe not well registered. Therefore, the original depth pixels are first aligned to the corresponding color coordinate space by the mapping relationship between the color and the depth coordinate spaces. Camera calibration technique is used in this procedure [1]. Figure 3 shows an example of coordinate alignment. Once the alignment is done, the features corresponding to RGB and aligned depth images can be extracted by a fully convolutional network respectively. These two extracted feature maps are concatenated and considered as the final feature representation. Then region proposal network (RPN) [18] is used to generate high quality regions of interest (ROI). After that, the classification and bounding box regression will be done for each ROI as in Fast R-CNN [7]. Finally, the non-maximum suppression is performed independently for each class using the algorithm and settings from [7].

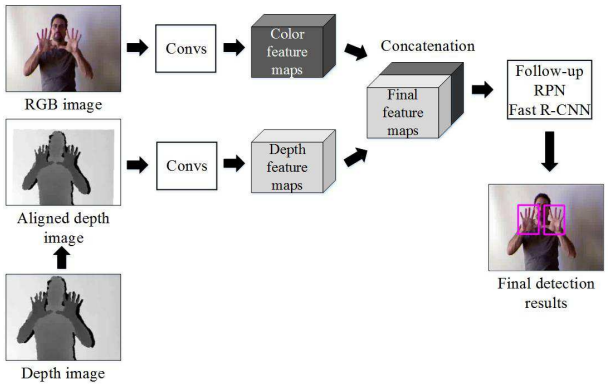

Figure 2. Hand detection pipeline of two-streams Faster R-CNN.

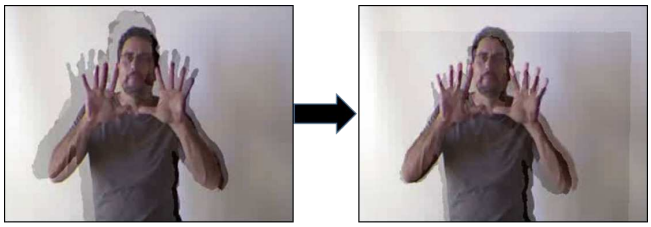

Figure 3. An example of coordinate alignment between RGB and depth channels.

3.2. Temporal Segmentation

Through our observation, we find the fact that the subject raises hands up when beginning to sign a gesture and puts hands down after performing one gesture in the Continuous Gesture Dataset (ConGD). In addition, accurate hand positions can be obtained by our proposed two streams Faster R-CNN. Therefore, we use hand positions to realize the temporal segmentation of the continuous gesture sequences. Firstly, we get a stable hand position from the initial several frames. Then by adding with an empirical value, a height threshold is fixed to determine the gesture boundaries. If one hand is first higher than the height threshold, it indicates the subject begins to sign a new gesture. If both hands are lower than the height threshold, it indicates the signed gesture is ending. Figure 4 shows an example of the temporal segmentation result for a continuous gesture sequence.

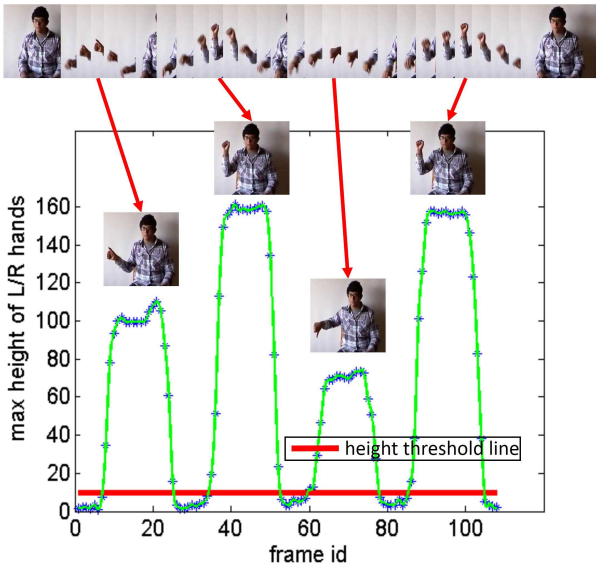

Figure 4. An example of the temporal segmentation result for a continuous gesture sequence.

3.3. Feature Generation

3.3.1 Hand-oriented Spatiotemporal Feature Extraction

It is no doubt that feature extraction is a crucial step in pattern recognition. As for the specific gesture recognition, the feature should characterize both appearance and motion pattern of the hands. To grasp the powerful feature of gestures, more attention should be paid to hand regions. Therefore, in each image frame, we keep the hand regions and block other regions based on the detected hand positions, which can effectively alleviate the distractor regions, such as background, clothing, body and so on. Considering on the different location of the signer in the image, the relative hand motion should be better. Thus the face is also detected and the location is encoded in the feature extraction. CNN features have been proved more effective than traditional handcrafted features, such as HOG. While normal 2D convolutional kernels only target spatial correlative information in an image, we employ 3D convolutional kernels, which can exploit temporal pattern besides spatial information, while eliminating the need for secondary temporal modeling techniques. Therefore, C3D model [21] is utilized to extract the hand-oriented spatiotemporal information based on continuous video input with only hand regions and face location maintained in each frame. As illustrated in Fig. 1, C3D feature extractor is a part of C3D model. The whole architecture of the C3D consists of 8 convolutional layers (with 64, 128, 256, 256, 512, 512, 512, 512 filters), 5 pooling layers, 2 fully connected layers of size 4096 and final softmax layer to output predicted label. All the kernels of 3D convolutional layers are of size $3 \times 3 \times 3$, the stride of the 3D convolutional layers are all of size $1 \times 1 \times 1$, and each convolutional layer is followed by a rectified linear unit (ReLU). All pooling kernels are of size $2 \times 2 \times 2$ except for the first pooling layer is $1 \times 2 \times 2$, of which the kernel is $1 \times 2 \times 2$ to keep more temporal information in the early stage of the network.

C3D is set up with the video frames which are resized into 128×171. We also use jittering by using random crops with a size of 112×112, which can be considered data augmentation. The network in original C3D model [21] takes 16-frames clips as input. However, Li et al. [14] found that most isolated gesture videos in ConGD are with $29 - 39$ frames by statistics. And they took 32 frames clips as input because more frames help with increasing the information of inputs and making it easier to track the detail path of gestures. Therefore, the strategy, 32 frames input, is also employed in this paper.

Segmented videos with more than 32 frames are sampled with the dynamic ratio according to their frame number, while videos with less than 32 frames are extended by interpolation. In order to train the C3D effectively with ConGD, which only has 249 categories of gestures and 14314 continuous gesture videos, C3D model is first pre-trained on Sports-1M dataset, which is the largest video classification benchmark with 1.1 million sports videos in 487 categories. After that, two C3D models are fine-tuned by processed RGB and depth videos respectively, which only contain hand regions and face location. Finally, removing the last softmax and fully connected layer, we obtain C3D feature extractor. The feature of fc6, with 4096 dimensions, is the generated hand-oriented spatiotemporal feature. Figure 5 illustrates the whole procedure of the feature generation.

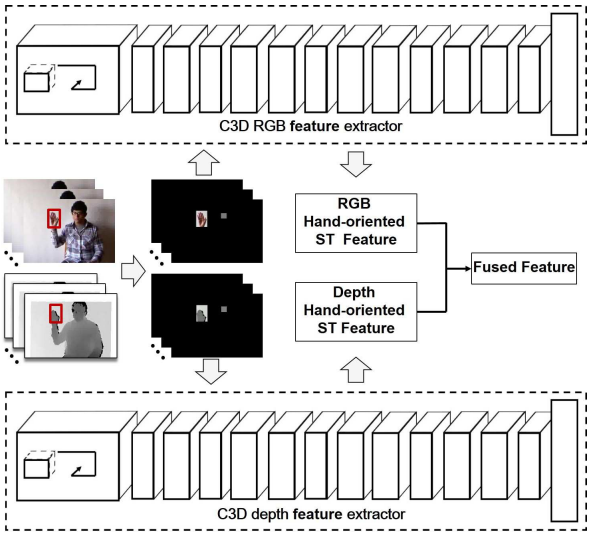

Figure 5. The process of feature generation.

3.3.2 Feature Fusion

As illustrated in Fig. 5, for each segmented gesture, we have the processed RGB and depth videos, in which only the hand regions and face location maintained. In order to deeply exploit the complementary multi-channel data, we fuse them to improve the final recognition performance. There are two fusion schemes. One is to directly fuse processed RGB and depth videos. The other is to fuse the extracted hand spatiotemporal features. The processed RGB and depth videos are not registered precisely because of the coarse calibration. In addition, the sum of the RGB and depth channel is 6. The original C3D input channel is 3. If the first fusion scheme is adopted, parameters of the first pre-trained C3D layer can not be employed. Therefore, in this paper, the fusion in feature level is adopted. The feature vectors of RGB and depth channels are concatenated and the fusion process is formulated as follows:

$$fc6_{rgb} = L2_{norm}\left(C3D(v_{prgb})\right) \tag{1}$$

$$fc6_{depth} = L2_{norm}\left(C3D(v_{pdepth})\right) \tag{2}$$

$$F = fc6_{rgb} \oplus fc6_{depth} \tag{3}$$

where v_{prgb} and v_{pdepth} denote processed RGB and depth gesture videos, $L2_{norm}$ is L2-normalization, $fc6_{rgb}$ and $fc6_{depth}$ denote the fc6 feature of C3D model according to corresponding rgb and depth channels, $\oplus$ is the concatenation operator, F is the final fused feature. The whole feature generation procedure can be summarized into the following three steps. Firstly, two fine-tuned C3D models take the processed RGB and depth videos as input and output fc6 features respectively. After that, fc6 features is followed by an L2-normalization. Finally, fc6 features corresponding to RGB and depth channels are integrated by simple concatenation.

3.4. Classification

Given a continuous gesture sequence, it is processed by four steps, which are hand detection, temporal segmentation, feature generation and classification. Finally, its corresponding sequence label will be predicted.

To well train the classifier, a training set Ω is prepared first, which contains plenty of continuous gesture samples and the corresponding class label. For example, a sequence with 120 frames consists of three gestures labels $[2, 4, 8]$ and with begin and end frame index $[1\ 30; 31\ 66; 67\ 120]$. First, A continuous gesture sequence is segmented into isolated gestures according to the given segmentation information. Then, These isolated gestures are used to fine-tune C3D. After that, we get hand-oriented spatiotemporal features as described in section 3.3. At last, the fused features are used to train a linear SVM classifier.

The testing procedure is similar with the key steps in training process. Given a continuous gesture sequence with unknown segmentation, the temporal segmentation is achieved with the detected hand positions by 2S Faster R-CNN. Next for each isolated gesture, the hand-oriented spatiotemporal features are extracted and fused, which is sent to the well trained SVM classifier and then output the corresponding labels.

3.5. Implementation Details

In the hand-oriented spatiotemporal feature extraction procedure, the fine-tuned C3D model is trained using the mini-batch stochastic gradient descent with the momentum of 0.9 and the weight decay of 0.00005. In each time of iteration, a mini-batch of 10 shuffled video clips are sent into the network. The initial learning rate is set as 0.0001, and it decreases at the ratio of 0.9 after every 5000 iterations. The training process stops after 100000 iterations. In addition, the penalty parameter C of the error term in linear SVM is set as 0.1 empirically. As for programming platform, both two streams Faster R-CNN hand detection and C3D are implemented in Caffe [8].

4. Experimental Results

In this section, we demonstrate the effectiveness of our method by groups of experiments on the Large-scale Continuous Gesture Recognition Dataset of the ChaLearn LAP challenge 2017 (ChaLearn LAP ConGD Dataset). First, ConGD and its evaluation protocol are briefly introduced. Second, we verify the effectiveness of 2S Faster R-CNN. Third, the quantitative analysis on our temporal segmentation is given. Then, widely experiments are conducted to verify the effectiveness of our hand-oriented spatiotempo-

Sets	# of Provided	# of Gestures	# of RGB Videos	# of Depth Videos	# of Performers	Label Provided	Temporal Segmention Provided
Training	249	30442	14314	14314	17	Yes	Yes
Validation	249	8889	4179	4179	2	Yes	Yes
Testing	249	8602	4042	4042	2	No	No

Table 1. Information of the ChaLearn LAP ConGD Dataset

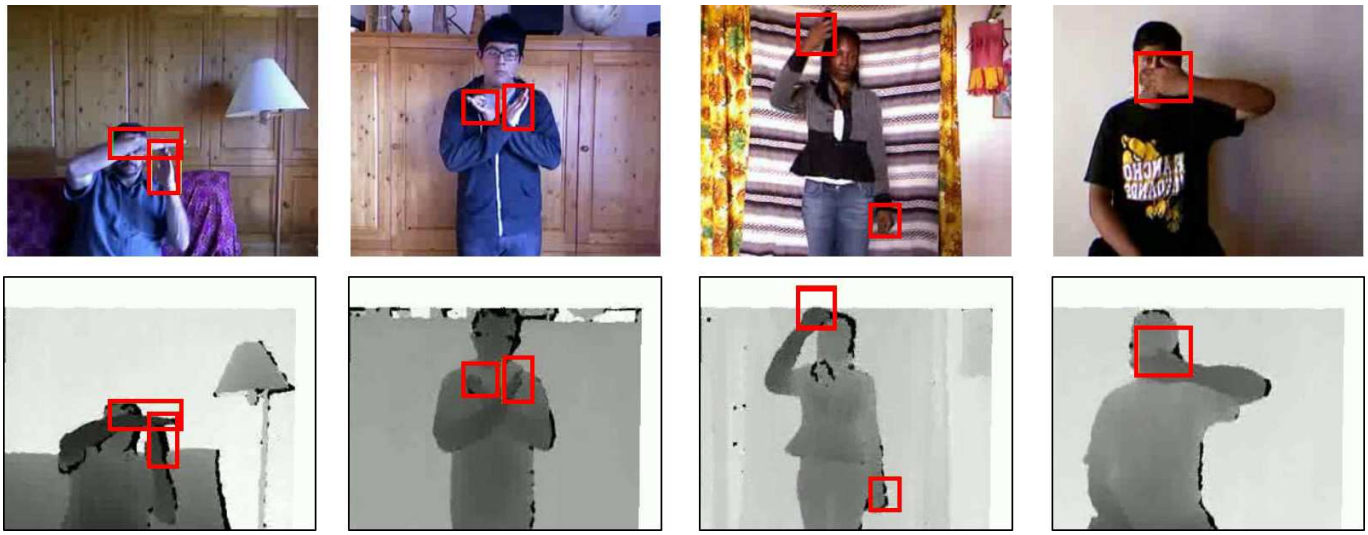

Figure 6. Some examples of the visualized hand detection results

ral feature and also the fusion strategy. Finally, we give the comparison with other state-of-the-art methods.

4.1. Dataset and Evaluation Protocol

Totally, ConGD dataset includes 47933 RGB-D gestures from 22535 RGB-D continuous gesture videos. The data is performed by 21 different signers and split into three mutually exclusive subsets, i.e. the training, validation and testing sets. The detailed information of the database is shown in Table 1. In order to measure the performance of different methods, the mean Jaccard Index (mJI) [23] is adopted as the evaluation criteria for the recognition algorithms. This score measures the average relative overlap between predicted and ground-truth labels for all given continuous video sequences.

Sets	# of RGB Images	# of Hand Regions
Training	50842	83022
Testing	3155	5006

Table 2. Information of our collected Hand Dataset

4.2. Evaluation on 2S Faster R-CNN

In this section, we will evaluate the 2S Faster R-CNN on the hand detection task. In this experiment, first we collect some image frames from the training data of ConGD. The hand regions in these images are labeled manually. The hand detection dataset is divided randomly into training and testing sets, whose detailed information is shown in Table 2.

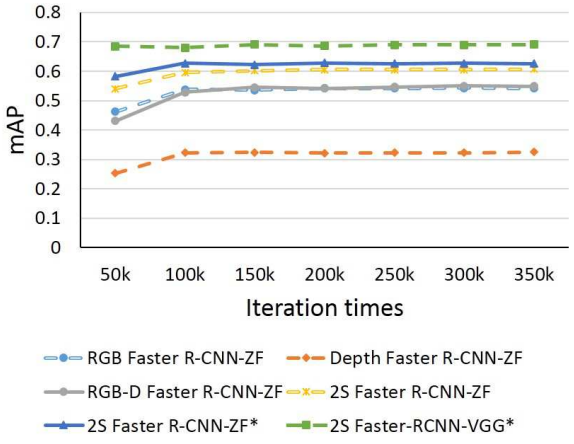

Figure 7. The comparison of the hand detection results

To evaluate the performance of our hand detection algorithm, mean Average Precision (mAP) is employed as the criterion. Here we compared the proposed method with three baseline models, which are "RGB Faster R-CNN-

ZF", "Depth Faster R-CNN-ZF" and "RGB-D Faster R-CNN-ZF". They correspond to the Faster R-CNN methods with the input of only RGB, depth and concatenated RGB and depth data respectively. In these implementations, the ZF [29] model is used to extract CNN feature. For 2S Faster R-CNN, ZF model is first trained by using the training data of our collected Hand Dataset (recorded briefly as "2S Faster R-CNN-ZF") and fine-tuned with Imagenet [4] (recorded briefly as "2S Faster R-CNN-ZF*") respectively. To boost the performance, we further use the VGG [20] model instead of ZF model, and the experiment is denoted as "2S Faster R-CNN-VGG*". The above mentioned six groups of models are all trained using the mini-batch stochastic gradient descent. The initial learning rate is set as 0.001, and it decreases at the ratio of 0.1 after every 70000 iterations. The training process stops after 350000 iterations.

The comparison of the hand detection results are shown in Fig. 7. The hand detection result of "2S Faster R-CNN-ZF" outperforms "RGB Faster R-CNN-ZF", "depth Faster R-CNN-ZF" and "RGB-D Faster R-CNN-ZF", which verifies the effectiveness of our proposed 2S Faster R-CNN. In addition, the fine-tuned models can improve the performance slightly as illustrated by the results of "2S Faster R-CNN-ZF" and "2S Faster R-CNN-ZF*". Finally, the performance of "2S Faster R-CNN-VGG*" is significantly superior to "2S Faster R-CNN-ZF*", which maybe because that VGG model can extract more powerful CNN feature than ZF. Therefore, "2S Faster R-CNN-VGG*" is used as our hand detector. Figure 6 gives some examples of the visualized hand detection results. We can see that hand detection result seems not good enough in depth channel, which is mainly caused by the coarse alignment.

4.3. Evaluation on Temporal Segmentation

Temporal segmentation plays a great role in continuous gesture recognition. So in this section, we will give the quantitative evaluation of our segmentation strategy and also the comparison with quantity of movement (QOM) [9], which is a widely used method for gesture segmentation. QOM tries to determine the gesture boundaries with the potential hand region information, which is derived by motion detection with depth input. While our segmentation is realized based on our detected hand introduced in section 3.2.

To quantitatively measure the segmentation performance, we define a spotting score to denote the proportion of correct segmented frames. For the k^{th} sequence , let $g_{k,i} = [s_i, e_i]$ and $p_{k,i} = [s_i, e_i]$ where s_i and e_i denote the start and end frames for the i^{th} segmented fragment in the continuous sequence, $g_{k,i}$ and $p_{k,i}$ are the i^{th} segmented fragment in true and predicted sequences respectively. The detail process of spotting score computation is described in Algorithm 1.

Algorithm 1 Compute Spotting Score

Input: true segment sequence set $G = \{g_1, g_2, \ldots, g_n\}$, predicted segment sequence set $P = \{p_1, p_2, \ldots, p_n\}$, the number of true segmented gestures in k^{th} sequence L_k^g, the number of predicted segmented gestures in k^{th} sequence L_k^p, where $k = 1, 2, \ldots, n$.
Output: spotting score S.

$S \leftarrow 0$
$count \leftarrow 0$ ▷ record the number of segmented isolated gestures from continuous gestures.
for k = 1 to n **do**
  for i = 1 to L_k^g **do**
    $count \leftarrow count + 1$
    $s_i^k \leftarrow 0$ ▷ s_i^k store the max intersection-over-union (IOU) among $g_{k,i}$ and $p_{k,j}, j = 1, 2, \ldots, L_k^p$.
    $dic_i^k \leftarrow 0$
    for j = 1 to L_k^p **do**
      $IOU \leftarrow \frac{g_{k,i} \cap p_{k,j}}{g_{k,i} \cup p_{k,j}}$
      if $IOU > s_i^k$ **then**
        $s_i^k \leftarrow IOU$
        $dic_i^k \leftarrow j$
  for i = 1 to L_k^g **do**
    $Stack\ sta$ ▷ declare a stack sta
    $max_s \leftarrow 0$
    for j = i + 1 to L_k^g **do**
      if $dic_i^k == dic_i^j\ and\ s_j^k\ != 0$ **then**
        PUSH(sta, j)
        $max_s = max(max_s, s_j^k)$
    if EMPTY(sta) == true **then** ▷ indicate there is not same IOU in a predicted segment if sta is empty.
      $S \leftarrow S + s_i^k$
    else ▷ get the maximus IOU and set others as 0.
      $S \leftarrow S + max_s$
      while EMPTY(sta) == false **do**
        $sameIdx \leftarrow TOP(sta)$
        $POP(sta)$
        $s_{sameIdx}^k \leftarrow 0$
$S \leftarrow S/count$
return S

Since the ground-truth segmentation results on testing set are unavailable, we conduct the experiment on validation set. There are totally 4179 continuous sequences and 8889 gestures on the validation set. The spotting scores of our segmentation and QOM are 0.8910 and 0.7732 respectively. It is obvious that our segmentation is superior to QOM for the stable and accurate hand positions. While the QOM is easily to be influenced by the complex illumination and other non-dominant motions, such as the arm movement et al.

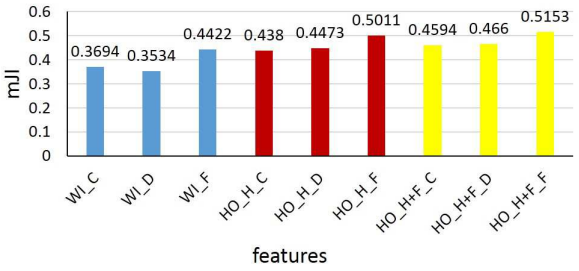

Figure 8. The performance of different features

4.4. Evaluation on Hand-oriented Spatiotemporal Feature

In this section, we will give the thorough evaluation on our hand-oriented feature. It has been mentioned above, in the proposed feature, the face location is taken as a complementary to the hand regions. Thus we give the comparison on the hand-oriented feature between only hand input (HO_H) and hand plus face input(HO_H+F). We carry out the experiments in the color (shorten as C), depth (shorten as D) and fusion (shorten as F) channels respectively. In addition, we perform the recognition experiments by using the C3D feature extracted from the whole image frames, which is denoted as WI. The experiments are also conducted on the color and depth modals separately, which are denoted as WI_C and WI_D respectively. All the experimental results are shown in Fig. 8.

From the results, it can be seen that the specific designed hand-oriented feature is significantly superior to the original C3D feature with whole image input. While in our proposed hand-oriented feature, the encoding of face location can slightly improve the performance compared with only hand regions input.

4.5. Evaluation on Different Channels

In above section, we show the experimental results on separated channel. While in this section, we will evaluate the recognition performance with fused features.

In gesture recognition, different channel features reveal different aspects of signs. For example, feature in RGB channel mainly characterizes the detailed texture and feature in depth channel mainly focuses on the geometric shape in the whole. Intuitively, these two kinds of features complement each other. Figure 8 gives the experimental results with single channel fused features.

From this figure, we can see that the fusions obtain the performance gains of 5 to 9 percentage points in mean Jaccard Index. These improvement show that the fusion scheme is extremely effective compared with any single channel feature.

Rank	Team	Score
1	**ICT_NHCI**	**0.6103**
2	AMRL	0.5950
3	PaFiFa	0.3744
4	Deepgesture	0.3164

Table 3. Performance comparison with other methods on testing set of ConGD

4.6. Comparison with Other Methods

In this section, we show the performance comparison with other methods on the ChaLearn LAP Large-scale Continuous Gesture Recognition Challenges. All results are run by the organizer on the testing set of ConGD, and only the data from the training set are used for algorithm training. Table 3 lists the mean Jaccard Index score of first four teams and our group won the first place [10].

5. Conclusion

This paper presents an effective spotting-recognition framework for large-scale continuous gesture recognition. Targeting on the gesture analysis task, first we need to determine the hand regions by a two-streams Faster R-CNN method. With the accurate hand positions, the input continuous gesture sequence can be segmented into several isolated gestures effectively. To generate more representative feature, a hand-oriented spatiotemporal feature is proposed, which characterizes the hand postures and motion trajectories for each gesture by 3D convolutional network. To boost the performance, the features in color and depth channels are fused further. Extensive experiments are conducted and show the impressive performance of our method. We also won the first place in the ChaLearn LAP large scale continuous gesture recognition challenge.

6. Acknowledgements

This work was partially supported by 973 Program under contract No 2015CB351802, Natural Science Foundation of China under contracts Nos.61390511, 61472398, 61532018, and the Youth Innovation Promotion Association CAS.

References

[1] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[2] S. Celebi, A. S. Aydin, T. T. Temiz, and T. Arici. Gesture recognition using skeleton data with weighted dynamic time warping. In *VISAPP*, 2013.

[3] X. Chai, Z. Liu, F. Yin, Z. Liu, and X. Chen. Two streams recurrent neural networks for large-scale continuous gesture

recognition. In *International Conference Pattern Recognition Workshops*, 2016.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[5] J. Duan, J. Wan, S. Zhou, X. Guo, and S. Li. A unified framework for multi-modal isolated gesture recognition. In *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM),(Accept)*, 2017.

[6] W. Gao, G. Fang, D. Zhao, and Y. Chen. Transition movement models for large vocabulary continuous sign language recognition. In *Automatic Face and Gesture Recognition*, 2004.

[7] R. Girshick. Fast R-CNN. In *ICCV*, 2015.

[8] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, 2014.

[9] F. Jiang, S. Zhang, S. Wu, Y. Gao, and D. Zhao. Multi-layered gesture recognition with kinect. *The Journal of Machine Learning Research*, 16(1):227–254, 2015.

[10] W. Jun, S. Escalera, A. Gholamreza, H. J. Escalante, X. Baró, I. Guyon, M. Madadi, A. Juri, G. Jelena, L. Chi, and X. Yiliang. Results and analysis of chalearn lap multi-modal isolated and continuous gesture recognition, and real versus fake expressed emotions challenges. In *ICCV Workshops*, 2017.

[11] M. B. Kaaniche and F. Brémond. Tracking hog descriptors for gesture recognition. In *Advanced Video and Signal Based Surveillance*, 2009.

[12] O. Koller, S. Zargaran, and H. Ney. Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent cnn-hmms. In *CVPR*, 2017.

[13] W. Kong and S. Ranganath. Automatic hand trajectory segmentation and phoneme transcription for sign language. In *Automatic Face and Gesture Recognition*, 2008.

[14] 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 *International Conference Pattern Recognition Workshops*, 2016.

[15] P. Molchanov, S. Gupta, K. Kim, and K. Pulli. Multi-sensor system for driver's hand-gesture recognition. In *Automatic Face and Gesture Recognition*, 2015.

[16] 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.

[17] V. Pitsikalis, S. Theodorakis, C. Vogler, and P. Maragos. Advances in phonetics-based sub-unit modeling for transcription alignment and sign language recognition. In *CVPR Workshops*, 2011.

[18] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

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

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

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

[22] 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*, 23(2):023017–023017, 2014.

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

[24] C. Wang, W. Gao, and S. Shan. An approach based on phonemes to large vocabulary chinese sign language recognition. In *Automatic Face and Gesture Recognition*, 2002.

[25] H. Wang, X. Chai, X. Hong, G. Zhao, and X. Chen. Isolated sign language recognition with grassmann covariance matrices. *ACM Transactions on Accessible Computing (TACCESS)*, 8(4):14, 2016.

[26] H. Wang, A. Stefan, S. Moradi, V. Athitsos, C. Neidle, and F. Kamangar. A system for large vocabulary sign search. In *ECCV Workshops*, 2010.

[27] H. Wang, Q. Wang, and X. Chen. Hand posture recognition from disparity cost map. In *Asian Conference on Computer Vision*, 2012.

[28] H.-D. Yang and S.-W. Lee. Robust sign language recognition by combining manual and non-manual features based on conditional random field and support vector machine. *Pattern Recognition Letters*, 34(16):2051–2056, 2013.

[29] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

[30] Z. Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012.