ChaLearn Looking at People: IsoGD and ConGD
Large-Scale RGB-D Gesture Recognition

Jun Wan, Senior Member, IEEE, Chi Lin, Member, IEEE, Longyin Wen, Yunan Li, Qiguang Miao, Senior Member, IEEE, Sergio Escalera, Member, IEEE, Gholamreza Anbarjafari, Senior Member, IEEE, Isabelle Guyon, Guodong Guo, Senior Member, IEEE, and Stan Z. Li, Fellow, IEEE

Abstract—The ChaLearn large-scale gesture recognition challenge has run twice in two workshops in conjunction with the International Conference on Pattern Recognition (ICPR) 2016 and International Conference on Computer Vision (ICCV) 2017, attracting more than 200 teams around the world. This challenge has two tracks, focusing on isolated and continuous gesture recognition, respectively. It describes the creation of both benchmark datasets and analyzes the advances in large-scale gesture recognition based on these two datasets. In this article, we discuss the challenges of collecting large-scale ground-truth annotations of gesture recognition and provide a detailed analysis of the current methods for large-scale isolated and continuous gesture recognition. In addition to the recognition rate and mean Jaccard index (MJI) as evaluation metrics used in previous challenges, we introduce the corrected segmentation rate (CSR) metric to evaluate the performance of temporal segmentation for continuous gesture recognition. Furthermore, we propose a bidirectional long short-term memory (Bi-LSTM) method, determining video division points based on skeleton points. Experiments show that the proposed Bi-LSTM outperforms state-of-the-art methods with an absolute improvement of 8.1% (from 0.8917 to 0.9639) of CSR.

Index Terms—Bidirectional long short-term memory (Bi-LSTM), gesture recognition, RGB-D.

I. INTRODUCTION

HUMAN action and gesture recognition have received a lot of attention from the computer vision community. In the past few years, several famous action and gesture datasets have been released, such as the NTU RGB+D dataset [7], MSR-Action3D dataset [8], CAD-60 [9] and CAD-120 dataset [10], DHG2016 dataset [11], SHREC17 dataset [12], RGBD-HuDaAct dataset [13], SYSU 3-D human–object interaction (3DHOI) dataset [14], HMDB51 dataset [15], UCF101 dataset [16] and Kinetics datasets [17], CGD [1], multimodal gesture dataset [2], Sheffield gesture dataset [4], NIVIDIA gesture dataset [5], and EgoGesture dataset [6]. These datasets pushed the advance of the state-of-the-art research for action and gesture recognition [18], [19]. Interestingly, the data size of action datasets is much larger than gesture datasets in terms of both the amount of data and the number of classes. For example, the Kinetics dataset includes 600 action classes and 500,000 video clips while only limited data are provided for gesture recognition (i.e., the Sheffield dataset contains about ten classes, 1080 videos). The main reasons for this are: 1) actions can be more easily captured than gestures, such as the Kinects dataset from YouTube videos; 2) the gesture can be seen as a semiotic sign highly dependent on the cultural context (i.e., Chinese number) while the action is goal-directed motion sequence (i.e., play football). Therefore, if a large amount of gestures is needed, it requires a lot of human-labor costs; and 3) actions tend to focus on the body information with large motions (such as hugging and sports) while gestures are produced as part of deliberate actions and signs, involving the motion of the...
up body, especially the arms, hands, and fingers. Furthermore, facial expressions are also considered to be involved in gesture recognition. The previous leads to high complexity of collecting large gesture recognition datasets. In addition, because of the different nature between signs and gestures, there is no guarantee existing action datasets are suitable for training systems for gesture-based recognition scenarios, such as sign language recognition and human–computer interaction, where not only the body part motion is needed but also the semantic language and facial expressions. Due to the above reasons, one can see that the released gesture datasets in Table I are very limited in size. This hinders further developments of deep learning-based methods in gesture recognition.

To this end, we present two large-scale datasets with RGB-D video sequences, namely, the ChaLearn-isolated gesture dataset (IsoGD) and the continuous gesture dataset (ConGD) for the tasks of isolated and continuous gesture recognition, respectively. Both of them consist of more than 47,000 gestures fallen into 249 classes performed by 21 performers. Besides, we organized two ChaLearn large-scale gesture recognition challenge workshops in conjunction with the International Conference on Pattern Recognition (ICPR) 2016 [20] and International Conference on Computer Vision (ICCV) 2017 [21]. The datasets allowed for the development and comparison of different algorithms, and the competition and workshop provided a way to track the progress and discuss advantages and disadvantages learned from the most successful and innovative entries.

The main contributions are summarized as follows.

1) We discuss the challenges of creating two large-scale gesture benchmark datasets, namely, the IsoGD and ConGD, and highlight developments in both isolated and continuous gesture recognition fields by creating the benchmark and holding the challenges. We analyze the submitted results in both challenges and review the published algorithms in the last three years.

2) A new temporal segmentation algorithm called the bidirectional long short-term memory (Bi-LSTM) segmentation network is proposed, which is used to determine the start and end frames of each gesture in the continuous gesture video. Compared with the existing methods, the main advantage of the proposed method is to avoid the need for prior assumptions.

3) A new evaluation metric called the corrected segmentation rate (CSR) is introduced and used to evaluate the performance of temporal segmentation. Compared with the published methods, the proposed Bi-LSTM method improves state-of-the-art results. The superiority of temporal segmentation is about 8.1% (from 0.8917 to 0.9639) by CSR on the testing sets of the ConGD dataset.

The remainder of this article is organized as follows. We describe datasets, evaluation metrics, and organized challenges in Section II. In Section III, we review the state-of-the-art methods focusing on both datasets. We propose a new algorithm for temporal segmentation in Section IV and present experimental results on the two proposed datasets in Section V. Finally, we conclude this article in Section VI.

### II. Dataset Introduction and Challenge Tasks

#### A. Motivation

Benchmark datasets can greatly promote the research developments in their respective fields. For example, the ImageNet Large-Scale Visual-Recognition Challenge [22] (ILSVRC) is held every year from 2010 to 2017, which includes several challenging tasks, including image classification, single-object localization, and object detection. The dataset presented in this challenge contains 1000 object classes with approximately 1.2 million training images, 50,000 validation images, and 100,000 testing images, which greatly promotes the development of new techniques, particularly those based on deep learning architectures, for image classification and object localization. Several other datasets have also been designed to evaluate different computer vision tasks, such as human pose recovery, action and gesture recognition, and face analysis [2], [23], [24].

Nevertheless, there are very few annotated datasets with a large number of samples and gesture categories for the task of RGB-D gesture recognition. Table I lists the publicly available RGB-D gesture datasets released from 2011 to 2017. Most datasets include less than 20 gesture classes (e.g., [3] and [4]). Although the CGD dataset [1] has about 54,000 gestures, it is designed for the one-shot learning task (only one training sample per class). The multimodal gesture dataset [2], [23] contains about 13,000 gestures with 387 training samples per class, but it only has 20 classes.

In order to provide the community with a large dataset for RGD gesture recognition, here we take benefit of the previous CGD dataset [1] by integrating all gesture categories and samples to design two new large RGB-D datasets for gesture spotting and classification. In Table I, the new IsoGD and ConGD datasets show a significant increase in size in terms of both the number of categories and the number of samples in comparison to state-of-the-art alternatives.

#### B. Dataset Introduction

As previously mentioned, our datasets were derived from the CGD dataset [1] which was designed for the “one-shot learning” task. The CGD dataset contained 540 batches (or subfolders), and all batches of CGD had 289 gestures from 30 lexicons and a large number of gestures in total (54,000 gestures in about 23,000 RGB-D video clips), which makes it a very valuable material to carve out different tasks. This is what
TABLE II
SUMMARY OF ISOGD AND CONGD DATASETS

<table>
<thead>
<tr>
<th>Sets</th>
<th>the ISOGD dataset</th>
<th>the CONGD dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#gestures #videos #subjects</td>
<td>#gestures #videos #subjects</td>
</tr>
<tr>
<td>Train</td>
<td>35878</td>
<td>35878</td>
</tr>
<tr>
<td>Valid</td>
<td>5764</td>
<td>3784</td>
</tr>
<tr>
<td>Test</td>
<td>6271</td>
<td>6271</td>
</tr>
<tr>
<td>All</td>
<td>47933</td>
<td>47933</td>
</tr>
</tbody>
</table>

TABLE III
SUMMARY OF PARTICIPATION FOR BOTH CHALLENGES

<table>
<thead>
<tr>
<th>Challenge</th>
<th>#Round</th>
<th>#Teams1</th>
<th>#Teams2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isolated Gesture Challenge</td>
<td>1</td>
<td>67</td>
<td>20</td>
</tr>
<tr>
<td>Continuous Gesture Challenge</td>
<td>2</td>
<td>44</td>
<td>17</td>
</tr>
<tr>
<td>Isolated Gesture Challenge</td>
<td>1</td>
<td>65</td>
<td>6</td>
</tr>
<tr>
<td>Continuous Gesture Challenge</td>
<td>2</td>
<td>39</td>
<td>11</td>
</tr>
</tbody>
</table>

1) total number of teams; 2) teams that submitted the predicted results on test sets.

TABLE IV
SUMMARY OF THE RESULTS IN OUR CHALLENGES

<table>
<thead>
<tr>
<th></th>
<th>Isolated Gesture</th>
<th>Continuous Gesture</th>
</tr>
</thead>
<tbody>
<tr>
<td>rank by test</td>
<td>evaluation</td>
<td>evaluation</td>
</tr>
<tr>
<td></td>
<td>Team</td>
<td>Team</td>
</tr>
<tr>
<td></td>
<td>valid (n)</td>
<td>valid (m)</td>
</tr>
<tr>
<td>1</td>
<td>ASU [28]</td>
<td>61.40%</td>
</tr>
<tr>
<td>2</td>
<td>SYUC [26]</td>
<td>59.76%</td>
</tr>
<tr>
<td>3</td>
<td>CST [31]</td>
<td>62.05%</td>
</tr>
<tr>
<td>4</td>
<td>AMIR [29]</td>
<td>60.81%</td>
</tr>
<tr>
<td>5</td>
<td>XHT [32]</td>
<td>59.60%</td>
</tr>
<tr>
<td>6</td>
<td>baseline [25]</td>
<td>49.19%</td>
</tr>
<tr>
<td>7</td>
<td>MUSR [25]</td>
<td>56.94%</td>
</tr>
<tr>
<td>8</td>
<td>FMY [25]</td>
<td>56.94%</td>
</tr>
</tbody>
</table>

Authorized licensed use limited to: INSTITUTE OF AUTOMATION CAS. Downloaded on June 08,2022 at 06:10:31 UTC from IEEE Xplore. Restrictions apply.
Fig. 1. Gesture lexicons in IsoGD and ConGD. The pink boxes indicate different gesture classes. The gray boxes are not used in our datasets because the number of samples from the MotorcycleSignals is very limited. In order to keep data balance, we omitted this lexicon. Other boxes with the same color indicate the same gesture we have merged from CGD.

evaluation phase (August 7, 2016 to August 17, 2016 for the first round, and June 23, 2017 to July 2, 2017 for the second round). Table III shows the summary of the participation for both gesture challenges. The total number of registered participants of both challenges is more than 200, and 54 teams have submitted their predicted results.

For each round, training, validation, and test datasets were provided. Training data were released with labels, validation data were used to provide feedback to participants in the leaderboard, and test data were used to determine the winners. Note that each track had its own evaluation metrics. The four tracks were run in the CodaLab platform. The top three ranked participants for each track were eligible for prizes. The performances of winners are shown in Table IV.

III. REVIEW OF STATE-OF-THE-ART METHODS

In recent years, the commercialization of affordable RGB-D sensors, such as Kinect, made it available depth maps, in addition to classical RGB, which are robust against illumination variations and contain abundant 3-D structure information. Based on this technology, we created the IsoGD and ConGD datasets [25], which has been already used by several researchers to evaluate the performance of gesture recognition models. In this section, we provide a review of state-of-the-art methods using both datasets, pointing out
advantages and disadvantages, comparing them, and providing discussion for future research directions.

We summarize the recognition task into two main categories, namely, isolated and continuous gesture recognition, which are shown in Fig. 2. For the task of isolated gesture recognition, 2-D CNN-based methods [41], [42] learn spatial features while 3-D CNN-based methods [27]–[29], [31], [33], [34], [39], [43] learn spatiotemporal features. In order to obtain a tradeoff between highly discriminative features (such as 3-D spatiotemporal features) and computation complexity, 2-D CNN over dynamic images encoding spatiotemporal information have been explored in [21], [28], [32], [35], [40], and [44]. Recurrent neural networks (RNNs) [45] or its variant long short-term memory (LSTM) [46] have been also applied to analyze sequential information in videos [29], [35], [37], [38], [43]. However, the task of continuous gesture recognition involves additional challenges to the ones of classical isolated gesture recognition. In continuous gesture recognition, there may be several gestures in a video to be recognized in time. This task uses to be addressed either in a frame-by-frame fashion [36], [39] or by a temporal segmentation strategy [34], [35], [38], [40]. A detailed comparison of methods and their basic features evaluated on isolated and continuous gesture recognition datasets is shown in Tables V and VI, respectively. Below, we briefly review common techniques used in gesture recognition, such as preprocessing strategies, CNN models, and fusion strategies [47].

A. Preprocessing

In the case of the RGB modality, it is highly affected by illumination changes, while the depth modality is insensitive to illumination variations, though it may present some noisy readings based on environmental factors, such as surface reflections. In order to overcome previous issues, Miao et al. [27] implemented the Retinex theory [60] to normalize illumination of RGB videos, and used a median filter to denoise depth maps. Asadi-Aghbolaghi et al. [61] utilized a hybrid median filter and an inpainting technique to enhance depth videos.

The second category of preprocessing is based on frame unification and video calibration. The reason for frame unification is to fix the same dimension for all inputs in CNNs. After a statistical analysis of frame number distribution of training data on the IsoGD dataset, Li et al. [31] fixed the frame number of each clip as 32 to minimize the loss of motion

![Fig. 2. Graphical representation of categories of RGB-D-based gesture recognition methods.](image-url)
path in the temporal dimension. The same criterion has been used by most subsequent methods [27], [29], [31], [34], [43]. Meanwhile, although the RGB and depth videos are captured concurrently by many devices, for example, the Kinect sensor, RGB, and depth images are not accurately registered spatially. Such a spatial misalignment may affect multimodality fusion. Therefore, Wang et al. [49] proposed a self-calibration strategy based on a pinhole model to register multimodal data. Similarly, Asadi-Aghbolaghi et al. [61] exploited the intrinsic and extrinsic parameters of cameras to warp the RGB image to fit the depth one.

For continuous gesture recognition, temporal segmentation is widely applied to split continuous gestures into several isolated gestures. For example, Chai et al. [38] first took such a segmentation strategy for continuous gesture recognition. It assumes all gestures begin and end with performers' hands down. Then, the video can be characterized as successive gesture parts and transition parts. A similar idea is used in [34], [35], and [40]. Camgoz et al. [36] conducted such a temporal segmentation in a different way. They treat the segmentation process as a feature to learn and use the likelihood to split the videos into multiple isolated segments, which is done by localizing the silence regions, where there is no motion. Then, the motion of the hand palm and finger movements are fed into a Bi-LSTM network for gesture spotting [59]. Although the work of Hoang et al. [59] uses Bi-LSTM, similar to us, our architecture and inputs are different, and we achieve higher recognition performance, as shown in Table VI.

B. Deep Learning-Based Methods

Due to the overwhelming advantage of deep learning versus handcrafted feature-based methods, next, we briefly review deep learning-based methods, which are grouped into 2-D CNNs, 3-D CNNs, and RNN/LSTM for gesture recognition, and Faster R-CNN for hand detection.

2-D CNNs: 2-D CNNs, such as AlexNet [41], VGG [42], and ResNet [62], have shown great performance dealing with still-image recognition tasks. There are several methods [32], [49], [63] that implement the 2-D CNN to extract spatial features. In order to extend 2-D CNN to consider temporal information, Wang et al. [32], [35] used rank pooling [64] to generate dynamic depth images (DDIs) and computed dynamic depth normal images (DDNIs) and dynamic depth motion normal images (DDMNIs) to wrap both the motion information and the static posture in an image. The counterpart work of Wang et al. [51] uses RGB videos to generate visual dynamic images (VDIs). The work of Wang et al. [28] extends the DDIs for both body and hand level representation, which are called body-level DDIs (BDDIs) and hand-level DDIs (HDDIs), respectively. Zhang et al. [44] used an enhanced depth motion map (eDMM) to describe depth videos and a static pose map (SPM) for postures. Then, two CNNs are used to extract features from these representations. Wang et al. [49] used the scene flow vector, which is obtained by registered RGB-D data, as a descriptor to generate an action map, which is subsequently fed into AlexNet for classification.

3-D CNNs: 3-D CNNs such as C3D [65] were proposed to extend 2-D CNNs to compute spatiotemporal features. Li et al. [31], [78] utilized 3-D CNN to extract features from RGB-D, saliency, and optical-flow videos. Zhu et al. [33], [43] proposed a pyramid 3-D CNN model, in which the videos are divided into three 16 frame clips, performing prediction in each of them. Final recognition is obtained by means of score fusion. Such a pyramid 3-D CNN model is also employed by [28], [29], and [35]. Liu et al. [34] and Wang et al. [35] extended 3-D CNN in a similar fashion for continuous gesture recognition, first splitting the continuous gestures into isolated ones using temporal segmentation. In [36], 3-D CNN is used in a framewise fashion, with the final classification given by posterior estimation after several iterations.

RNN/LSTM: The RNN [45], or its variation, LSTM [46] is a kind of network where connections between units form a directed cycle. The special structure of RNN-like models allows for sequential input analysis. Chai et al. [38] used two streams of RNN to represent features of RGB-D videos and used LSTM to model the context. Pigou et al. [37] first used a ResNet to extract features of gray-scale video, and then used a bidirectional LSTM [66] to process both temporal directions. Zhu et al. [43] used convolutional LSTM (ConvLSTM) with 3-D CNN input to model the sequential relations between small video clips. The 3-D CNN + LSTM scheme is also employed in [29], [35], and [52].

Faster R-CNN: The faster R-CNN [67] was initially proposed for object detection tasks. Some works in gesture recognition used object detection algorithms for hand detection [21], [28], [34], [35], [38]. Chai et al. [38] used faster R-CNN to detect hands for recognizing begin–end gesture instances. The same strategy is applied in related works [34], [35]. Some methods further combined global and local hand regions to boost recognition performance [21], [28].

Attention Models: Some attention-aware methods [54], [68] have been applied for gesture recognition. For example, Narayana et al. [54] proposed a focus of the attention network (FOANet) which introduced a separate channel for every focus region (global, right/left hand) and modality (RGB, depth, and flow). Zhang et al. [68] proposed an attention mechanism embedding into the ConvLSTM network, including attention analysis in ConvLSTM gates for spatial global average pooling and fully connected operations. Li et al. [57] proposed a spatiotemporal attention-based Res3D network to focus on the gesture itself. These attention-based methods have shown high performances for gesture recognition, as shown in Table V.

C. Multimodality Fusion Scheme

For the task of RGB-D gesture recognition, fusion mechanisms are widely considered [28], [32], [33], [35], [43], [51]. This kind of scheme consolidates the scores generated by networks that are fed with different modalities. Among these methods, the averaging [28], [33], [35], [43] and multiply [32], [51] score fusions are two of the most frequently applied. Li et al. [31], Zhu et al. [33], and Miao et al. [27] adopted feature-level fusion. The former methods [31], [33] directly blend the features of RGB and
depth modalities in a parallel or serial way, which simply average or concatenate. Considering the relationship between features from different modalities that share the same label, Miao et al. [27] adopted a statistical analysis-based fusion method—canonical correlation analysis, and Li et al. [31] adopted an extension version of discriminative correlation analysis, which tries to maximize the inner class pairwise correlations across modalities and intraclass differences within one feature set. Hu et al. [50] paid more attention to the fusion scheme and designed a new layer comprised of a group of networks called the adaptive hidden layer, which serves as a selector to weight features from different modalities of data. Lin et al. [52] developed an adaptive scheme for setting weights of each voting subclassifier via a fusion layer, which can be learned directly by the CNNs.

D. Other Techniques to Boost Performance

Multiple Modalities: Based on the available RGB and depth data modalities in the proposed datasets, additional data modalities have been considered by researchers. Li et al. [78] generated saliency maps to focus on image parts relevant to gesture recognition and used optical flow [31] to learn features from RGB image motion vectors. Wang et al. [49] and Asadi-Aghbolaghi et al. [61] extended optical flow from RGB videos to construct RGB and depth optical flow from RGB and depth videos, respectively. Another strategy uses the skeleton information as an extra modality via regional multiperson pose estimation (RMPE) [21], [69].

Data Augmentation: Data augmentation is another common way to boost performance. Miao et al. [27] focused on data augmentation to increase overall dataset size while Zhang et al. [44] mainly augmented data to balance the number of samples among different categories, including translation, rotation, the Gaussian smoothing, and contrast adjustment.

Pretrained Models: Some C3D-implemented methods [31], [33], [43] are pretrained on external datasets, such as sports-1M [70]. In terms of cross-modality finetuning, Zhu et al. [43] first trained the networks with RGB and depth data from scratch and then finetuned the depth one with the model trained from RGB data. The same process is done for the RGB model. The result of cross-modality finetuning [43] showed an improvement of 6% and 8% for RGB and depth inputs, respectively.

E. Summary of State-of-the-Art Methods

We summarize the previous methods features and compare them in Tables V and VI for isoGD and conGD gesture recognition datasets, respectively. All methods are published in the last three years. For isolated gesture recognition on the IsoGD dataset, Table V, all methods except [25] are based on deep learning. The recognition rate is improved by 58% from the 24.19% of the handcrafted method of [25] to the 82.17% of the 2-D CNN fusion strategy of [54]. For continuous gesture recognition on the ConGD dataset, Table VI, the performance has also been improved considerably for both MJJ and CSR metrics since 2017.

F. Discussion

In this section, we review the techniques on both isolated and continuous gesture recognition based on RGB-D data. After the release of the large-scale IsoGD and ConGD datasets, new methods have pushed the development of gesture recognition algorithms. However, there are challenges faced by the available methods that allow us to outline several future research directions for the development of deep learning-based methods for gesture recognition.

Fusion of RGB-D Modalities: Most methods [28], [32], [33], [35], [43] considered RGB and depth modality as a separate channel and fused them at a later stage by concatenation or score voting, without fully exploiting the complementary properties of both visual modalities. Therefore, cooperative training using RGB-D data would be a promising and interesting research direction.

Attention-Based Mechanism: Some methods [34], [38], [54] used hand detectors to first detect hand regions and then designed different strategies to extract local and global features for gesture recognition. However, these attention-based methods need hard to train specialized detectors to find hand regions properly. It would be more reasonable to consider sequence modeling self-attention [71], [72] and exploit it for dynamic gesture recognition.

Simultaneous Gesture Segmentation and Recognition: The existing continuous gesture recognition works [28], [34], [40], [48] first detect the first and end points of each isolated gesture, and then train/test each segment separately. This procedure is not suitable for many real applications. Therefore, simultaneous gesture segmentation and recognition would be an interesting line to be explored.

Joint Structure Learning: A structure attention mechanism can be further explored. Some works [52], [54] train each attention part (i.e., arm, gesture) separately and fuse the scores of several waterworks to obtain the final recognition result. However, it cannot consider the structure information among the relationships of body parts. We believe gesture recognition will benefit from joint structure learning (i.e., body, hand, arm, and face).

Efficient and Fast Networks: We discussed different works that benefited from the fusion and combination of different trained models and modalities. However, these are highly complex strategies from a computational perspective. For real applications, lightweight networks would be preferred.

Besides, there is one criticism encountered from the challenges. In real gesture-based applications (i.e., sign language recognition), the interval of neighbor gestures sometimes would be not obvious. However, in the ConGD and CGD datasets, the presence of silence between neighbor gestures always happens. In future work, the release of the annotated continuous sign language datasets without nonobvious silences would push further the research in the field.

IV. TEMPORAL SEGMENTATION BENCHMARK

Here, we propose a benchmark method, namely, the Bi-LSTM network, for temporal segmentation. Before it, we
illustrate the drawbacks of the current temporal segmentation methods.

A. Drawbacks of Temporal Segmentation Methods

1) Handcrafted Hand Motion Extraction: Some methods [25], [28], [40], [48] first measure the quantity of movement (QoM) for each frame in a multigesture sequence and threshold the QoM to obtain candidate boundaries. Then, a sliding window is adopted to refine the candidate boundaries to produce the final boundaries of the segmented gesture sequences in a multigesture sequence. However, it captures not only hand motions but also the background movements that may be harmful to temporal segmentation.

2) Unstable Hand Detector: Some methods [34], [38] used the faster R-CNN [73] to build the hand detector. Due to the high degree of freedom of human hands, it is very hard to tackle some intractable environments, such as hand-self occlusion and drastically hand shape changing. The errors of hand detection would considerably reduce the performance of temporal segmentation.

3) Strong Prior Knowledge Requirement: Most previous methods (e.g., [25], [34], [38], [40], and [48]) use prior knowledge (e.g., a performer always raises hands to start a gesture and puts hands down after performing another gesture) is not practical for real applications.

In contrast to the previous methods, we did not only use human hands but also the arm/body information [34], [38]. Moreover, we designed a Bi-LSTM segmentation network to determine the start–end frames of each gesture automatically without requiring specific prior knowledge.

B. Proposed Bi-LSTM Method

We treat the temporal segmentation as a binary classification problem. The flowchart is shown in Fig. 3. We first use the convolutional pose machine (CPM) algorithm [73]–[76] to estimate the human pose, which consists of 60 keypoints (18 keypoints for human body and 21 keypoints for left and right hands, respectively). The keypoints are shown in the left part of Fig. 3. Therefore, the human gesture/body from an image is estimated by these keypoints. For the $t$-th frame of a video, the gesture is represented by a 120-D (2 × 60) vector $V_t$ in the following:

$$ V_t = \{(x_i - \bar{x}, y_i - \bar{y}), i = 1, \ldots, 60\} $$  \hspace{1cm} (4)

where the coordinate of the $i$th keypoint is represented by $(x_i, y_i)$, the average coordinate of all detected keypoints is denoted by $(\bar{x}, \bar{y})$, and $\bar{y} = (1/n') \sum_{k=1}^{n'} y_k$, $n'$ is the number of detected keypoints of frame $t$.

We use the data $\{(V_t, g_t) | t = 1, \ldots, m\}$ to train the Bi-LSTM network [66], where $m$ is the total number of frames in the video, and $g_t$ is the start and end frames indicator of a gesture, that is, $g_t = 1$ indicates the start and last frames of a gesture, and $g_t = 0$ for other frames. The Bi-LSTM network combines the bidirectional RNNs (BRNNs) [77] and the LSTM, which captures long-range information in bidirectional inputs. The LSTM unit $H$ is implemented by the following composite function:

$$ i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) $$
$$ f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) $$
$$ c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) $$
$$ o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) $$
$$ h_t = o_t \tanh(c_t) $$ \hspace{1cm} (5)

where $\sigma$ is the activation function, and $i_t, f_t, c_t, o_t, h_t$ are the input gate, forget gate, output gate, cell activation vector, and the hidden vector at time $t$, respectively. For the Bi-LSTM, the network computes both the forward and backward hidden vectors $\tilde{h}_t$ and $\hat{h}_t$ at time $t$, and the output sequence $y_t$ as

$$ y_t = W_{h_x} \tilde{h}_t + W_{h_y} \hat{h}_t + b_y $$ \hspace{1cm} (6)

We design four hidden layers in the Bi-LSTM network in Fig. 3. Notably, if a frame is within the segment of a gesture, we annotate it as the positive sample; otherwise, it is treated as negative. To mitigate the class imbalance issue, we assign different weights to the positive and negative samples. The objective function is defined as

$$ J(\theta) = -\frac{1}{m} \sum_{t=0}^{m-1} \sum_{i=0}^{1} \omega_i \log \frac{e^{\theta^T x_t}}{\sum_{j=0}^{k} e^{\theta^T x_t}} $$ \hspace{1cm} (7)

where $\theta$ is the parameter matrix of the softmax function and $\omega_i$ is the weight used to mitigate the class imbalance issue. According to our statistics, the ratio of the positive and negative samples is approximately 1:40. Thus, we set $(w_0/w_1) = 40$ ($w_0$ is the weight penalty of positive samples) to balance the loss terms of positive and negative samples in the training phase.
Fig. 4. Comparison of the first ten classes of the ConGD dataset for MJJ metric. For most of the classes (six out of ten classes), our method achieves the best performance. From Table VI, the MJJ of our method, Liu et al. [34] and Zhu et al. [58] for all classes are 0.7179, 0.6103, and 0.7163, respectively. Note that we only show MJJs of the first ten classes.

Fig. 5. Examples of failure cases by the CPM method [76] for keypoint detection, especially for hand keypoints. Gesture ID: (a) 1, (b) 6, (c) 7, and (d) 10.

Fig. 6. CSR curve of the proposed Bi-LSTM method on the ConGD dataset. Up: test set. Bottom: validation set.

The gradient of the objective function is computed as

$$\nabla_{\theta} J(\theta) = -\frac{1}{m} \sum_{t=0}^{m-1} \sum_{i=0}^{k} \omega_{i} x_{t} \left( I_{[v=j]} - \frac{e^{\theta_{i} x_{t}}}{\sum_{j=0}^{k} e^{\theta_{j} x_{t}}} \right)$$

(8)

where $\nabla_{\theta} J(\theta)$ is the gradient with respect to the parameter $\theta_{v}$, and $I_{[v=j]}$ is the indicator function, that is, $I_{[v=j]} = 1$ if and only if $v = j$.

In this way, we use the learned model by the Bi-LSTM network to predict the probability of each frame whether it is the start or end frames. If the probability value of a frame is large than 0.5, this frame is treated as start or end frames.

V. EXPERIMENTS

In this section, we evaluate and compare our proposed gesture recognition by the segmentation strategy on the ConGD dataset. First, the experimental setup is presented, including the running environments and settings. Then, the performances and comparisons on the ConGD dataset are given.

Our experiments are conducted on an NVIDIA Titan Xp GPU. The input of the Bi-LSTM network is a 120-D vector. We use the Adam algorithm to optimize the model with the batch size 120. The learning rate starts from 0.01 and the models are trained for up to 50 epochs.

The performance of the proposed Bi-LSTM method for temporal segmentation is shown in Table VII, which achieves 0.9668 and 0.9639 for CSR@IoU = 0.7 on both validation and testing sets of ConGD. After temporal segmentation, we use the model of [52] to perform final gesture recognition. The results are also shown in Table VI, where MJJ = 0.6830 and 0.7179 on the validation and test sets, respectively. Based on MJJ and CSR, our method achieves the best performance. Although the metric of MJJ depends on both temporal segmentation and final classification, the recognition performance of MJJ can still benefit from an accurate temporal segmentation, such as the proposed Bi-LSTM method.

We also provide comparisons for each category on the ConGD dataset in Fig. 4. Here, our method (overall MJJ: 0.6830 for the validation set and 0.7179 for the test set) is compared with two state-of-the-art methods [34] (overall MJJ: 0.5163 for the validation set and 0.6103 for the test set) and [58] (overall MJJ: 0.5368 for the validation set and 0.7163 for the test set) for each category.

For illustration purposes, we show MJJs scores of the first ten classes in Fig. 4. It shows that our method achieves the best performance among competing methods for most classes (six out of ten classes). Some cases of failure of our approach are shown in Fig. 5. They are mainly produced by self-occlusions and hand keypoints detection failures, which lead to inaccurate inputs used by our models. As future work, more accurate hand pose detection strategies would enhance the overall performance of the proposed solution.

Fig. 6 shows the CSR curve in each epoch under different IoU thresholds from 0.5 to 0.9. One can see that when the IoU threshold is between 0.5 and 0.8, the CSR is very stable after three epochs. When IoU is equal to 0.9, the training epochs for the CSR increases. This is because the correct condition is more strict (> 90% overlapped region will be treated as the correct one) and it will cost more time to seek the best CSR. Our proposed Bi-LSTM method can obtain very stable results under different IoU thresholds. For example, even the IoU is equal to 0.9, the CSR of our method still is higher than 0.9. Alternative temporal segmentation methods [34], [36], [37], [40] are relatively inferior (the best is about 0.81 in [34]) on the validation set. Also, our Bi-LSTM can obtain the best performance on the test set of the ConGD dataset.

Then, we randomly select 1000 video sequences in the ConGD datasets to check for computational requirements. It required about 0.4 s under the GPU environment [NVIDIA TITAN X (Pascal)] and 6 s on the CPU environment [Intel Core i7-5820K@3.30 GHz] for the proposed Bi-LSTM method. It demonstrates the proposed Bi-LSTM method is ultrahigh-speed processing (~0.4 ms/video-GPU, ~6 ms/video-CPU). We note that the processing time of the CPM algorithm is about 283 ms/f on the GPU environment.
Finally, we selected the three longest video sequences of the ConGD dataset, and the segmentation results of the proposed Bi-LSTM method are shown in Fig. 7. The green points are the ground truth of the segmentation point, while the blue line is the confidence of positive responses. These three videos have more than 100 frames and contain at least five gestures. Compared with the videos with a fewer number of gestures, the dense gestures make it hard to find the segment points accurately. However, our Bi-LSTM method can mark the start and end points of each gesture, and the segmentation for all the gestures are with confidence over 0.8.

VI. CONCLUSION
In this article, we proposed IsoGD and ConGD datasets for the task of isolated and continuous gesture recognition, respectively. Both datasets are the current largest datasets for dynamic gesture recognition. Based on both datasets, we have run challenges in ICPR 2016 and ICCV 2017 workshops, which attracted more than 200 teams around the world and pushed the state of the art for gesture recognition. Then, we reviewed the last 3-years methods for gesture recognition based on the provided datasets. Besides, we proposed the Bi-LSTM method for temporal segmentation. We expect the proposed datasets to push the research in gesture recognition.

REFERENCES
Longin Wen received the B.Eng. degree in automation from the University of Electronic Science and Technology of China, Chengdu, China, in 2010, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2015. He is a Staff Scientist with the AI Lab, JD Finance, Mountain View, CA, USA. After that, he moved to the University at Albany, State University of New York, Albany, NY, USA, for postdoctoral research. From 2016 to 2018, he was a Computer Vision Scientist with GE Global Research, Niskayuna, NY, USA.

Yunan Li received the Ph.D. degree from the School of Computer Science and Technology, Xidian University, Xi’an, China, in 2019. He currently serves as a Lecturer with Xidian University. His research interests include machine learning, computer vision, and gesture recognition.

Gholamreza Anbarjafari (Senior Member, IEEE) received the Ph.D. degree in electrical and electronic engineering (major in image processing) from Eastern Mediterranean University, Famagusta, Cyprus, in 2010. He is currently Director and Chief Data Scientist at PwC Advisory in Finland. He is also with Faculty of Engineering at Hasan Kalyoncu University, Turkey. He is the Founder of the Intelligent Computer Vision Lab, University of Tartu, Tartu, Estonia. He was also the Deputy Scientific Coordinator of the European Network on Integrating Vision and Language (IV&L Net) ICT COST Action IC1307. He has been a representative of Estonia for several other COST Actions. Dr. Anbarjafari is an Associate Editor and a Guest Lead Editor of several journals, Special Issues and Book projects. He is the Chair of Signal Processing/Circuits and Systems/Solid-State Circuits Joint Societies Chapter of IEEE Estonian section.

Sergio Escalera (Member, IEEE) received the Ph.D. degree in multiclass visual categorization systems from the Computer Vision Center, University of Alabama at Birmingham (UAB), Birmingham, AL, USA, in 2008. He is an Associate Professor with the Department of Mathematics and Informatics, Universitat de Barcelona, Barcelona, Spain. He is an Adjunct Professor with the Universitat Oberta de Catalunya, Barcelona, Aalborg University, Aalborg, Denmark, and Dalhousie University, Halifax, NS, Canada. Dr. Escalera obtained the 2008 Best Thesis Award on Computer Science at Universitat Autònoma de Barcelona. He leads the Human Pose Recovery and Behavior Analysis Group at UB and CVC. He is also a member of the Computer Vision Center at Campus UAB.

Qiguang Miao (Senior Member, IEEE) received the Doctoral degree in computer application technology from Xidian University, Xi’an, China, in 2005. He is a Professor and the Ph.D. Student Supervisor with the School of Computer Science and Technology, Xidian University. He has published over 100 papers in the significant domestic and international journals or conferences. His research interests include machine learning, intelligent image processing, and malware behavior analysis and understanding.

Guodong Guo (Senior Member, IEEE) received the B.E. degree in automation from Tsinghua University, Beijing, China, the first Ph.D. degree in pattern recognition and intelligent control from the Chinese Academy of Sciences, Beijing, China, and the second Ph.D. degree in computer science from the University of Wisconsin–Madison, Madison, WI, USA, in 2006. In 2018, he joined with Baidu Research, Beijing. He is an Associate Professor with the Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV, USA.

Isabelle Guyon received the Ph.D. degree in physical sciences from University Pierre, Paris, France, in 1988. She is a Chaired Professor of big data with the Universite Paris-Saclay (UPSud/INRIA), Saint-Aubin, France, specialized in statistical data analysis, pattern recognition, and machine learning. Prior to joining Paris-Saclay, she worked as an Independent Consultant and was a Researcher with AT&T Bell Laboratories, Murray Hill, NJ, USA. Her areas of expertise include computer vision and bioinformatics. Her recent interest is in applications of machine learning to the discovery of causal relationships.

Stan Z. Li (Fellow, IEEE) received the B.Eng. degree from Hunan University, Changsha, China, in 1982, the M.Eng. degree from the National University of Defense Technology, Changsha, in 1985, and the Ph.D. degree from Surrey University, Guildford, U.K., in 1991. He is a Chair Professor of artificial intelligence with Westlake University, Hangzhou, China. He was a Researcher and the Director of the Center for Biometrics and Security Research with Institute of Automation, Chinese Academy of Sciences, Beijing, China. He was a Researcher with Microsoft Research Asia, Beijing, and an Associate Professor with Nanyang Technological University, Singapore. He has published over 500 papers with Google scholar index of over 42 000 and h-index of 95.