# EgoGesture: A New Dataset and Benchmark for Egocentric Hand Gesture Recognition

Yifan Zhang, *Member, IEEE*, Congqi Cao, Jian Cheng, *Member, IEEE*, and Hanqing Lu, *Senior Member, IEEE*

***Abstract*—Gesture is a natural interface in human–computer interaction, especially interacting with wearable devices, such as VR/AR helmet and glasses. However, in the gesture recognition community, it lacks of suitable datasets for developing egocentric (first-person view) gesture recognition methods, in particular in the deep learning era. In this paper, we introduce a new benchmark dataset named EgoGesture with sufficient size, variation, and reality to be able to train deep neural networks. This dataset contains more than 24 000 gesture samples and 3 000 000 frames for both color and depth modalities from 50 distinct subjects. We design 83 different static and dynamic gestures focused on interaction with wearable devices and collect them from six diverse indoor and outdoor scenes, respectively, with variation in background and illumination. We also consider the scenario when people perform gestures while they are walking. The performances of several representative approaches are systematically evaluated on two tasks: gesture classification in segmented data and gesture spotting and recognition in continuous data. Our empirical study also provides an in-depth analysis on input modality selection and domain adaptation between different scenes.**

***Index Terms*—Benchmark, dataset, egocentric vision, gesture recognition, first-person view.**

## I. Introduction

VISION-BASED gesture recognition [1], [2] is an important and active field of computer vision. Most of the methods are in a strongly supervised learning paradigm. Hence, the availability of a large number of training data is the base of the work. With the development of deep leaning technique, the lack of large scale and high quality datasets has become a vital problem and limits the exploring of many data-hungry deep neural networks algorithms.

In the domain of vision-based gesture recognition, there exist some established datasets, such as Cambridge hand gesture dataset [3], Sheffield KInect Gesture (SKIG) Dataset [4], MSRGesture3D [5] and LTTM Creative Senz3D dataset [6], but with only a few gesture classes (no more than 12) and limited number of samples (no more than 1400). Since 2011, ChaLearn Gesture Challenge has been launched every year and provided several large scale gesture datasets: Multi-modal gesture dataset [7], ChaLearn LAP IsoGD and ConGD datasets [8]. However, the gestures in these datasets are all captured in the second-person view, which are not suitable for egocentric gesture recognition task. Here we give our definitions on the three views in the gesture recognition domain: 1) **First-person view**: the camera as a performer. The view are obtained by the camera mounted on a wearable device of the performer. 2) **Second-person view**: the camera as a receiver. The performer performs gestures actively like interacting with the camera. One faces the camera in a relative near distance. This usually happens in a human machine interaction scenario. 3) **Third-person view**: the camera as an observer. The performer performs gestures spontaneously without the intention to interact with the camera. One could be far from the camera and not face to the camera. This usually happens in a surveillance scenario.

To interact with wearable smart devices such as VR/AR helmet and glasses, using hand gesture is a natural and intuitive way. The gestures can be captured by egocentric cameras mounted on the devices (typically near the head of the user). First-person vision provides a new perspective of the visual world that is inherently human-centric, and thus brings its unique characteristics to gesture recognition: 1) Egocentric motion: since the camera is mounted on the device near the head of the user, camera motion can be significant with the movement of the head of the user, in particular when the users perform gestures while they are walking. 2) Hands in close range: due to the short distance from the camera to the hands and the narrow field-of-view of the egocentric camera, hands could be partly or even totally out of the field-of-view.

Currently, it is not easy to find a benchmark dataset for egocentric gesture recognition. Most of the egocentric hand-related datasets like EgoHands [9], EgoFinger [10] and GUN-71 [11], are built for developing the techniques on hand detection and segmentation [9], finger detection [10], or understanding a specific action [11]. They do not explicitly design gestures for interaction with wearable devices. To the best of our knowledge, the Interactive Museum database presented by Baraldi [12] with a goal of enhancing museum experience is the only public dataset for egocentric gesture recognition. However, it contains only 7 gesture classes performed by five subjects with 700 video sequences, which can not satisfy the data size demand for training deep neural networks. We believe one reason for egocentric

Manuscript received May 5, 2017; revised December 29, 2017; accepted February 4, 2018. Date of publication February 21, 2018; date of current version April 17, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61332016 and Grant 61572500, and in part by the Youth Innovation Promotion Association CAS. *(Yifan Zhang and Congqi Cao are co-first authors.)* The Guest editor coordinating the review of this manuscript and approving it for publication was Prof. Hari Kalva. *(Corresponding author: Jian Cheng.)*

The authors are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences and University of Chinese Academy of Sciences, Beijing 100190, China (e-mail: yfzhang@nlpr.ia.ac.cn; congqi.cao@nlpr.ia.ac.cn; jcheng@nlpr.ia.ac.cn; luhq@nlpr.ia.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2018.2808769

gesture recognition being less explored is a shortage of fully annotated large scale ground truth data.

In this paper, we introduce up-to-date the largest dataset called EgoGesture for the task of egocentric gesture recognition. The dataset, which has been already publicly available[1], contains more than 24 thousand RGB-D video samples and 3 million frames from 50 distinct subjects. We carefully design 83 classes of static or dynamic gestures specifically for interaction with wearable devices. Our dataset has the largest number of data, gesture classes and subjects than other egocentric gesture recognition datasets. It is more complex as our data is collected from more diverse yet representative scenes with large variation including clutter background, strong and weak illumination condition, shadow, indoor and outdoor environment. We specially design two scenarios where the subjects perform gestures while they are walking.

Given our dataset, we systematically evaluate the state-of-the-art methods based on both hand-crafted and deep learned features on two tasks: gesture classification in segmented data and gesture spotting and recognition in continuous data. Which to be the better video representation, either single-frame-based representation learned from the 2D CNN or spatiotemporal features learned from 3D CNN, is investigated. Our empirical study also provides an in-depth analysis on input modality selection from RGB and depth modalities and domain adaptation between different subjectes and scenes. We believe the proposed dataset can be used as a benchmark and help the community to move steps forward in egocentric gesture recognition, making it possible to apply data-hungry methods such as deep neural networks for this task.

## II. Related Work

### A. Datasets

In the field of gesture recognition, most of the established datasets are captured in second-person view. Cambridge hand gesture dataset [3] consists of 900 image sequences of 9 gesture classes, which are defined by 3 primitive hand shapes and 3 primitive motions. Sheffield KInect Gesture (SKIG) Dataset [4] contains 1080 RGB-D sequences collected from 6 subjects by a Kinect sensor. It collects 10 classes of hand gestures in total. Since 2011, ChaLearn Gesture Challenge has provided several large scale gesture datasets: Multi-modal Gesture Dataset (MMGD) [7], ChaLearn LAP IsoGD and ConGD datasets [8]. Multi-modal Gesture Dataset [7] contains only 20 classes. ChaLearn LAP IsoGD and ConGD datasets [8] provides the largest number of subjects and samples, but it is not specially designed for human computer interaction, with gestures from various application domains such as sign language, signals to machinery or vehicle, pantomimes, etc. CVRR-HAND 3D [15] and nvGesture [16] are two gesture datasets captured under real-world or stimulated driving settings. CVRR-HAND 3D [15] provides 19 classes of driver hand gestures performed by 8 subjects against a plain and clean background. nvGesture [16] acquired a larger dataset of 25 gesture types from 20 subjects, recorded by color, depth and stereo-IR sensors.

For first-person view hand-related dataset, EgoHands [9], which is used for hand detection and segmentation, contains images captured by Google Glass with manually labeled pixel-wise hand regions annotation. EgoFinger [10] captures 93,729 hand color frames, collected and labeled by 24 subjects for finger detection and tracking. GUN-71 [11] provides 71 classes of fine grained grasp actions to deal with object manipulation. Some datasets focus on recognizing activities of daily living from the first-person view for studies of Dementia [18]–[20]. These datasets collected 8-18 classes of instrumental activities of daily living captured by a camera mounted on the shoulder or chest of the patients or healthy volunteers.

The tasks in the datasets mentioned above are different from gesture recognition. The datasets proposed in [17] and [12] are most similar to our work. Starner[17] propose an egocentric gesture dataset which defines 40 American sign language gestures captured by a camera mounted on the hat of the only 1 subject. But the dataset currently is not online available. The Interactive Museum database [12] contains only 7 gesture classes performed by 5 subjects with 700 video sequences. To the best of our knowledge, our proposed EgoGesture dataset is the largest one for the use of egocentric gesture recognition. Detailed comparison between our dataset and some related gesture datasets can be found in Table I.

### B. Algorithms

Many efforts have been dedicated to hand related action, pose or gesture recognition. Karaman *et al.* [18] propose a Hierarchical Hidden Markov Model (HHMM) to detect activities of daily living (ADL) such as making coffee, washing dishes in videos. Jiang *et al.* [21] introduce a unified deep learning framework that jointly exploits feature and class relationships for action recognition. Hand pose estimation is another hot topic. Sharp *et al.* [22] present a system for reconstructing the complex articulated pose of the hand using a depth camera by combining fast learned reinitialization with model fitting based on stochastic optimization. Wan *et al.* [23] present a conditioned regression forest for estimating hand joint positions from single depth images based on local surface normals.

In this work, we focus on gesture recognition rather than explicit hand pose estimation, as we believe they are different tasks. Gesture recognition aims to understand the semantic of the gestures. Hand pose estimation aims to estimate the 2d/3d position of hand key points. Hence, in the following, we focus to present a brief overview on the approaches on two basic tasks in gesture recognition: gesture classification in segmented data and gesture spotting and recognition in continuous data.

*1) Gesture Classification in Segmented Data:* The key point for this task is to find a compact descriptor to represent the spatiotemporal content of the gesture. Traditional methods are based on hand-crafted features such as improved Dense Trajectories (iDT) [24] in RGB channels, Super Normal Vector (SNV) [25] in depth channel, and MFSK [26] in RGB-D channels. Most of the sophisticated designed features are derived or consist of

[1]http://www.nlpr.ia.ac.cn/iva/yfzhang/datasets/egogesture.html

TABLE I
COMPARISON OF THE PUBLIC GESTURE DATASETS

| Datasets | Samples | Labels | Subjects | Scenes | Modalities | Task | View |
|---|---|---|---|---|---|---|---|
| Cambridge Hand Gesture Dataset 2007 [3] | 900 | 9 | 2 | 1 | RGB | classification | second-person |
| MSRGesture3D 2012 [5] | 336 | 12 | 10 | 1 | RGB-D | classification | second-person |
| ChAirGest 2013 [13] | 1,200 | 10 | 10 | 1 | RGB-D, IMU | classification | second-person |
| SKIG 2013 [4] | 1,080 | 10 | 6 | 3 | RGB-D | classification | second-person |
| ChaLearn MMGR 2013, 2014 [7], [14] | 13,858 | 20 | 27 | - | RGB-D | classification, detection | second-person |
| CVRR-HAND 3D Dataset 2014 [15] | 886 | 19 | 8 | 2 | RGB-D | classification | second-person |
| LTTM Senz3D 2015 [6] | 1,320 | 11 | 4 | 1 | RGB-D | classification | second-person |
| ChaLearn Iso/ConGD 2016 [8] | 47,933 | 249 | 21 | - | RGB-D | classification, detection | second-person |
| nvGesture 2016 [16] | 1532 | 25 | 20 | 1 | RGB-D stereo-IR | classification, detection | second-person |
| ASL with wearable computer system 1998 [17] | 2500 | 40 | 1 | 1 | RGB | classification | first-person |
| Interactive Museum Dataset 2014 [12] | 700 | 7 | 5 | 1 | RGB | classification | first-person |
| EgoGesture the proposed dataset | 24,161 | 83 | 50 | 6 | RGB-D | classification, detection | first-person |

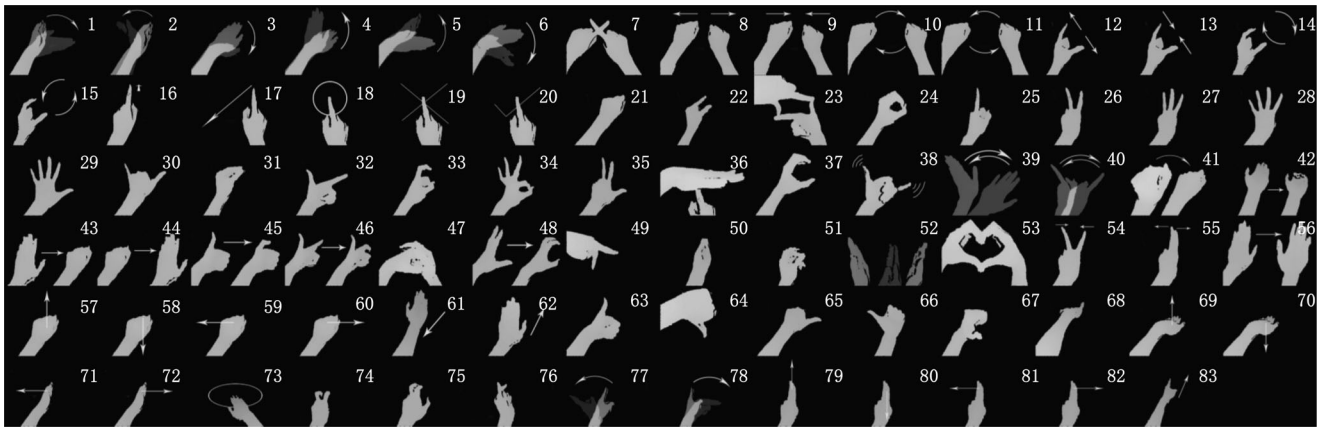


Fig. 1. The 83 classes of hand gesture designed in our proposed EgoGesture dataset.

Fig. 2. (Left) A subject wearing our data acquiring system to perform a gesture. (Right-top) The RealSense camera mounted on the head. (Right-mid) The image captured by the color sensor. (Right-bottom) The image captured by the depth sensor.

HOG, HOF, MBH and SIFT features which can represent the appearance, shape and motion changes corresponding to the gesture performance. They can be extracted from the single frame or consecutive frame sequence locally at spatiotemporal interest points [27] or densely sampled in the whole frame [28]. Ohn-Bar and Trivedi [15] evaluate several hand-crafted features for gesture recognition. A number of video classification systems successfully employ iDT [24] feature with Fisher vector [29] aggregation technique, which are widely regarded as state-of-the-art method for video analysis. Depth channel features are usually specifically designed for the characteristics of the depth information. Super normal vectors [25] employ surface normals. Random occupancy patterns [30] and layered shape pattern [31] are extracted in point clouds.

Recently, deep learning methods have become the main stream in computer vision tasks. Generally, there are mainly four frameworks to utilize deep learning methods for spatiotemporal modeling: 1) Use 2D ConvNets [32], [33] to extract features of single frames. By encoding frame features to video descriptors, classifiers are trained to predict the labels of videos. 2) Use 3D ConvNets [34], [35] to extract features of video clips. Then aggregate clip features into video descriptors. 3) Make use of recurrent neural networks (RNN) [36], [37] to model the temporal evolution of sequences based on convolutional features. 4) Represent a video as one or multiple compact images and then input it to a neural network for classification [38].

*2) Gesture Spotting and Recognition in Continuous Data:* This task aims to locate the starting and ending points of a

TABLE II
DESCRIPTIONS OF THE 83 GESTURES IN OUR PROPOSED EGOGESTURE DATASET

| | | | | |
|---|---|---|---|---|
| Manipulative | | Move | 1 Wave palm towards right | 2 Wave palm towards left |
| | | | 3 Wave palm downward | 4 Wave palm upward |
| | | | 5 Wave palm forward | 6 Wave palm backward |
| | | | 77 Wave finger towards left | 78 Wave finger towards right |
| | | | 57 Move fist upward | 58 Move fist downward |
| | | | 59 Move fist towards left | 60 Move fist towards right |
| | | | 61 Move palm backward | 62 Move palm forward |
| | | | 69 Move palm upward | 70 Move palm downward |
| | | | 71 Move palm towards left | 72 Move palm towards right |
| | | | 79 Move fingers upward | 80 Move fingers downward |
| | | | 81 Move fingers toward left | 82 Move fingers toward right |
| | | | 83 Move fingers forward | |
| | | Zoom | 8 Zoom in with two fists | 9 Zoom out with two fists |
| | | | 12 Zoom in with two fingers | 13 Zoom out with two fingers |
| | | Rotate | 10 Rotate fists clockwise | 11 Rotate fists counter-clockwise |
| | | | 14 Rotate fingers clockwise | 15 Rotate fingers counter-clockwise |
| | | | 56 Turn over palm | 73 Rotate with palm |
| | | Open/close | 43 Palm to fist | 44 Fist to Palm |
| | | | 54 Put two fingers together | 55 Take two fingers apart |
| Communicative | Symbols | Number | 24 Number 0 | 25 Number 1 |
| | | | 26 Number 2 | 27 Number 3 |
| | | | 28 Number 4 | 29 Number 5 |
| | | | 30 Number 6 | 31 Number 7 |
| | | | 32 Number 8 | 33 Number 9 |
| | | | 35 Another number 3 | |
| | | Direction | 63 Thumb upward | 64 Thumb downward |
| | | | 65 Thumb towards right | 66 Thumb towards left |
| | | | 67 Thumbs backward | 68 Thumbs forward |
| | | Others | 7 Cross index fingers | 19 Sweep cross |
| | | | 20 Sweep checkmark | 21 Static fist |
| | | | 34 OK | 36 Pause |
| | | | 37 Shape C | 47 Hold fist in the other hand |
| | | | 53 Dual hands heart | 74 Bent two fingers |
| | | | 75 Bent three fingers | 76 Dual fingers heart |
| | Acts | Mimetic | 16 Click with index finger | 17 Sweep diagonal |
| | | | 22 Measure (distance) | 18 Sweep circle |
| | | | 23 take a picture | 38 Make a phone call |
| | | | 39 Wave hand | 40 Wave finger |
| | | | 41 Knock | 42 Beckon |
| | | | 45 Trigger with thumb | 46 Trigger with index finger |
| | | | 48 Grab (bend all five fingers) | 49 Walk |
| | | | 50 Gather fingers | 51 Snap fingers |
| | | | 52 Applaud | |

specific gesture in continuous stream, which may be addressed by two strategies :

1) Perform temporal segmentation and classification sequentially. For automatic segmentation, appearance-based method [39], [40] are used to find candidate cuts based on the amount of motion or the similarities with respect to the neutral pose. Jiang *et al.* [39] measured the quantity of movement (QOM) of each frame and then got the candidate cuts when QOM is below a threshold, and finally refined the candidate cuts using sliding windows. In [40], hands are assumed to return to a neutral pose between two gestures, and the correlation coefficient is calculated between the neutral pose and the rest frames. Then the gesture segments can be localized by identifying the peak locations from the correlations. After temporal segment, different features can be extracted from each segmented gesture clip.

2) Perform temporal segmentation and classification simultaneously. Sliding window is a straightforward way to predict labels of truncated data in a series of fix-length window sliding along the video stream. The classifier is trained with an extra non-gesture class to handle the non-gesture part [41]. Another way is to employ sequence labeling models such as RNNs [16] and HMMs [42], [43] to predict the label for the sequence. Molchanov *et al.* [16] employ a recurrent three-dimensional convolutional neural network that performs simultaneous detection and classification of dynamic hand gestures from multimodal data. In [42], a multiple channel HMM (mcHMM) is used, where each channel is represented as a distribution over the visual words corresponding to that channel.

## III. THE EGOGESTURE DATASET

### A. Data Collection

To collect the dataset, we select Intel RealSense SR300 as our egocentric camera due to its small size and integrating both RGB and depth modules. The two modality videos are recorded in the resolution of $640 \times 480$ with the frame rate of 30 fps. As shown in Fig. 2, the subject wears the RealSense camera with a strap mount belt on their heads. They are asked to perform

all the gestures in 4 indoor scenes and 2 outdoor scenes. In the room, the four scenes are defined as follows: 1) the subject in a stationary state with a static clutter background; 2) the subject in a stationary state with a dynamic background; 3) the subject in a stationary state facing a window with strong sunlight; 4) the subject in a walking state. When outside, the two scenes are defined as follows: 1) the subject in a stationary state with a dynamic background; 2) the subject in a walking state with a dynamic background. We hope to simulate all possible using scenarios of wearable devices in our dataset. When collecting data, we first teach the subjects how to perform each gesture and tell them the gesture names (short descriptions). Then we generate a gesture name list with random order for each subject. Thus, the subject is told the gesture name and performs the gesture accordingly. They are asked to continuously perform 9–12 gestures as a session which is recorded as a video.

### B. Dataset Characteristics

*1) Gesture Classes:* Pavlovic [44] classified gestures into two categories: manipulative and communicative. Communicative gestures are further classified into symbols and acts. We design gestures in our dataset following this categorization. Since the gestures are used for human computer interaction, they should be meaningful, natural and easy to remember by the users. Under this principle, we design 83 gestures (shown in Fig. 1) which is currently the largest number of classes in the existing egocentric gesture datasets with the aim to cover most kinds of manipulation and communication operations to the wearable devices.

For manipulative gestures, we define four basic operations: zoom, rotate, open/close, and move. The "move" and "rotate" operations are defined along different directions. In each operation, we design gestures with different hand shapes (e.g., palm, finger, fist) to represent hierarchical operations, which can hierarchically correspond to virtual objects, windows, abstractions of computer-controlled physical objects, such as joystick. For communicative gestures, we define symbol gestures to represent number, direction and several popular symbols such as OK, Pause, etc. We design act gestures to imitate some actions, such as taking a picture, making a phone call, etc. Table II provides the description of each gesture in our dataset.

*2) Subjects:* The small number of subjects could make the intra-class variation very limited. Hence, we invited 50 subjects for our data collection which is also currently the largest number of subjects in the existing gesture datasets. In the 50 subjects, there are 18 females and 32 males. The average age of the subjects is 25.8, where the minimum age is 20, the maximum age is 41. The hand pose [shown in Fig. 3(A)], movement speed and range, using either right or left hand [Fig. 3(B)] vary significantly from different subjects in our dataset.

*3) Egocentric Motion:* When people use wearable device, they are often in a walking state, which can cause severe egocentric motion. It results in view angle change and motion blur in both RGB and depth channels [Fig. 3(C), (E)]. Hands are probably outside of the field-of-view in this situation (Fig. 3(D)]. In our dataset, we specially design two walking scenes in indoor and outdoor environments to collect data.

*4) Illumination and Shadow:* To evaluate the robustness to the illumination change of the baseline methods, we have data collected under extreme conditions such as facing to a window with strong sunlight where the brightness of the hand image is very low; backing to strong sunlight where the brightness of the hand image is very high and the shadow of the body projects on the hand [Fig. 3(F)]. When outside of the room, the depth image could be very blur due to the noisy environmental infrared light [Fig. 3(G)].

*5) Clutter Background:* We design scenes with static background placed with daily-life stuffs [Fig. 3(H)]; and dynamic background with walking people appearing in the camera [Fig. 3(I)].

### C. Dataset Statistics

We invited 50 distinct subjects to perform 83 classes of gestures in 6 diverse scenes. Totally 24,161 video gesture samples and 2,953,224 frames are collected in RGB and depth modality, respectively. Fig. 4 demonstrate the sample distribution on each subject in the 6 scenes. In the figure, the horizontal axis and the vertical axis indicate the subject ID and the number of the samples, respectively. We use different colors to represent different scenes. The numeral on each color bar represents the number of gesture samples in the corresponding scene recorded with the subject corresponding to the ID in the horizontal axis. There are 3 subjects (i.e., Subject 3, Subject 7 and Subject 23) who did not record videos in all the 6 scenarios. The total number of gesture samples of each subject is also listed above the stacked bars.

For each gesture class, there are up to 300 samples with large intra-class variety. When data collection, around 12 gestures are considered as a session and recorded as a video. Thus, it forms 2,081 RGB-D videos. Note that the order of the gestures performed is randomly generated. Hence, the videos can be used to evaluate gesture detection in continuous stream. The start and end frame index of each gesture sample in the video are also manually labeled, which providing the test-bed for segmented gesture classification. In the dataset, the minimum length of a gesture is 3 frames. The maximum length of a gesture is 196 frames. There are 437 gesture samples with a length less than 16 frames and 38 gesture samples with a length less than 8 frames.

In Table III, we show the data statistics of our dataset and compare it to other gesture datasets which are currently available on the web. Since we cannot download Cambridge Hand Gesture Dataset 2007 [3], MSRGesture3D 2012 [5], ASL 1998 [17], their statistics are not provided. The test data of some datasets are also not available. The dataset statistics include number of total frames, mean of the gesture sample durations, standard deviation of the gesture sample durations, the percentage of the training data. The mean and standard deviation of the gesture sample durations is calculated over the samples from all gesture classes in the dataset.

To further demonstrate the complexity of the data, we employ two types of objective criteria. We use normalized standard deviation for gesture duration in each gesture class to describe the

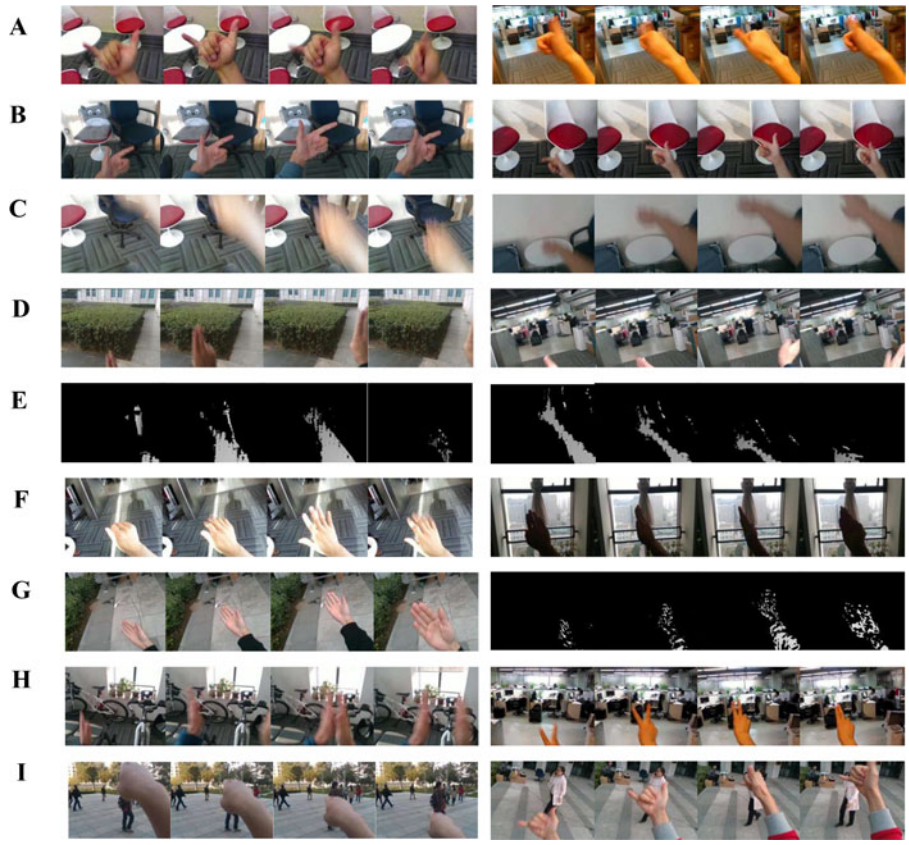



Fig. 3. Some examples to demonstrate the complexity of our gesture dataset. (A) Pose variation in same class; (B) Left-right hand change in same class; (C) Motion blur in RGB channels; (D) Hand out of field-of-view; (E) Motion blur in depth channel; (F) Illumination change and shadow; (G) Blur in depth channel in outdoor environment; (H) Clutter background; (I) Dynamic background with walking people.

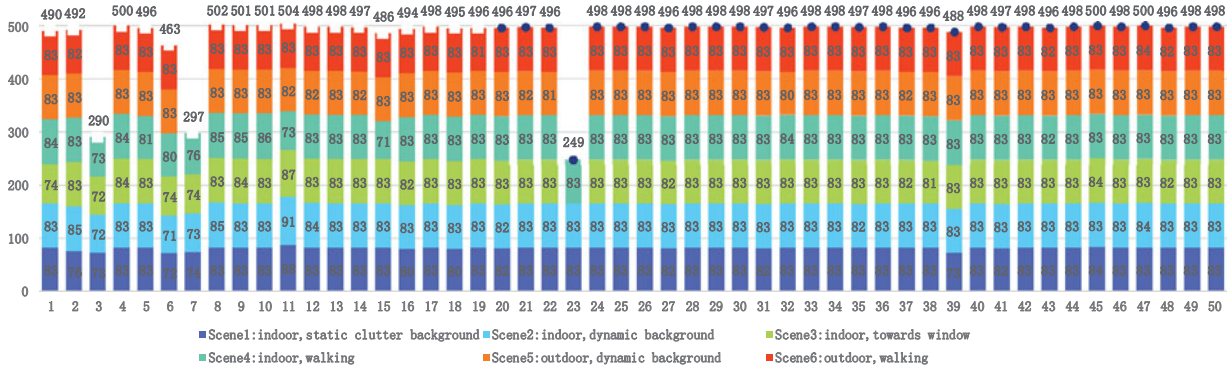



Fig. 4. The distribution of the gesture samples on each subject in EgoGesture dataset. The horizontal axis and the vertical axis indicate the subject ID and sample numbers, respectively.

TABLE III
STATISTICS OF THE PUBLIC GESTURE DATASETS

| Datasets | Frames | Mean duration | Duration std | Duration $\overline{nstd_k}$ | Edge density | % of train |
|---|---|---|---|---|---|---|
| ChAirGest 2013 [13] | 55,988 | 63 | 20.8 | 0.19 | 0.026 | 0.750 |
| SKIG 2013 [4] | 156,753 | 145 | 60.9 | 0.19 | 0.022 | 0.667 |
| ChaLearn MMGR 2013, 2014 [7], [14] | 1,720,800 | 52 | 17.0 | 0.32 | 0.102 | 0.560 |
| CVRR-HAND 3D Dataset 2014 [15] | 27,794 | 31 | 11.9 | 0.31 | 0.066 | 0.875 |
| LTTM Senz3D 2015 [6] | 1,320 | 30 | 0 | 0 | 0.077 | - |
| ChaLearn Iso/ConGD 2016 [8] | 1,714,629 | 41 | 18.5 | 0.37 | 0.110 | 0.635 |
| nvGesture 2016 [16] | 122,560 | 80 | 0 | 0 | 0.084 | 0.685 |
| Interactive Museum Dataset 2014 [12] | 25,584 | 37 | 12.8 | 0.28 | 0.025 | 0.100 |
| EgoGesture the proposed dataset | 2,953,224 | 38 | 13.9 | 0.33 | 0.127 | 0.595 |

speed variation of different subjects when they performing the same gesture. The normalized standard deviation of durations in gesture class $k$ is calculated as:

$$nstd_k = \frac{1}{\bar{l^k}} \sqrt{\frac{\sum_i^N (l_i^k - \bar{l^k})^2}{N}} \tag{1}$$

where in gesture class $k$, $l_i^k$ is the duration of the $i$th sample, $\bar{l^k}$ is the average duration of samples, $N$ is the number of samples. For the whole dataset, we get the average $\overline{nstd_k}$ over all gesture classes. From Table III, we can find that our EgoGesture datset has the 2nd largest duration $\overline{nstd_k}$ (0.33). This demonstrates our datset has large speed variation for different subjects when they performing the same gesture.

We use edge density to describe the texture complexity of the frames in the dataset. Edge is found by applying Sobel operator on the entire frame. The edge magnitude of a pixel is a combination of the edge strength along the horizontal and vertical directions:

$$E(x,y) = \sqrt{E_h^2(x,y) + E_v^2(x,y)} \tag{2}$$

The edge density of a frame is calculated as:

$$D = \frac{\sum_x^M \sum_y^N \mathbf{1}_{\{E(x,y)>T\}}}{MN} \tag{3}$$

where $\mathbf{1}_{\{\cdot\}}$ denotes the indicator function: if $a = 1$, then $\mathbf{1}_{\{a\}} = 1$, otherwise $\mathbf{1}_{\{a\}} = 0$. $M$ and $N$ are the width and height of the frame. $T$ is a threshold which is set as 100. We use the average edge density over the dataset as the criteria. Our dataset has the largest edge density (0.127), which means it has the largest texture complexity comparing to other datasets.

## IV. Benchmark Evaluation

In our newly created EgoGesture Dataset, we systematically evaluate state-of-the-art methods based on both hand-crafted features and deep networks as baselines on two tasks: gesture classification and gesture detection.

### A. Experimental Setup

We randomly split the data by subject into training (60%), validation (20%) and testing (20%) sets, resulting in 1,239 training, 411 validation and 431 testing videos. The numbers of gesture samples in training, validation and testing splits are 14416, 4768 and 4977 respectively. The subject IDs we use for testing are: 2, 9, 11, 14, 18, 19, 28, 31, 41, 47. The subject IDs for validation are: 1, 7, 12, 13, 24, 29, 33, 34, 35, 37.

### B. Gesture Classification in Segmented Data

For classification, we segment the video sequences into isolated gesture samples based on the beginning and ending frames annotated in advance. The learning task is to predict class labels for each gesture sample. We use classification accuracy which is the percent of correctly labeled samples as the evaluation metric for this learning task.

*1) Hand-Crafted Features:* We select three representative hand-crafted features: iDT-FV [24], SNV [25] and MFSK-BoVW [26], which are suitable for RGB, depth and RGB-D channels respectively.

**iDT-FV** [24] is a well-known compact hand-crafted feature for local motion modeling where global camera motion is canceled out by optical flow estimation. We compute the Trajectory, HOG, HOF and MBH descriptors in the RGB videos. The dimensions of the descriptors are 30 for Trajectory, 96 for HOG, 108 for HOF, 192 for MBH (including 96 for MBHx and 96 for MBHy). After PCA [45], we train GMMs with 256 Gaussians to generate Fisher vectors (FV) for each type of the descriptor. Then, we concatenate the FVs after applying L2 normalization. Finally, we use a linear SVM for classification.

**SNV** [25] clusters hypersurface normals in a depth video to form the polynormal and aggregates the low-level polynormals into the super normal vector. We follow the setting of [25] to compute normals, learn dictionary, generate the descriptors of video sequences and train linear SVM classifiers.

**MFSK-BoVW** [26] is designed to Mix Features Around Sparse Keypoints (MFSK) from both RGB and depth channels. We follow the setting of [26] to extract features. The spatial pyramid as the scale space is built for every RGB and depth frame. Keypoint detection around the motion regions is applied in scale spaces via SURF detector and tracking techniques. Then 3D SMoSIFT, HOG, HOF, MBH features are calculated in local patches around keypoints. The bag of visual word (BoVW) framework is used to aggregate the local features. Limited to the size of physical memory, we sample 19 instances from each gesture class to generate the visual word dictionary with the size of 5000. Finally, a linear SVM is trained for classification.

*2) Deep Learned Features:* We choose VGG16, C3D, VGG16+LSTM and IDMM+CaffeNet as baselines which correspond to the four deep learning frameworks described in Section II-B to do classification on our dataset.

**VGG16** [32] is a 2D convolutional neural network which contains 13 convolutional layers and 3 fully-connected layers. We train a VGG16 model to classify single frames for RGB and depth videos respectively, where the parameters trained on ImageNet are used as initialization. We test with two outputs of VGG16: the activations of the softmax layer and the activations of the $fc6$ layer. To aggregate the frame-level outputs into a video-level descriptor, we sum the softmax outputs of each frames over the video, then the class with the highest probability is chosen as the label of the video sequence. For $fc6$ features, average pooling and $L2$ normalization are used for aggregation, while linear SVM is employed to do classification. For modality fusion, the scores of classification probability obtained from RGB and depth inputs are added with a weight which is chosen on the validation split.

**C3D** [34] is a 3D convolutional neural network with eight 3D convolutional layers, one 2D pooling layers, four 3D pooling layers and three fully-connected layers. The 3D layers take a volume as input and output a volume which can preserve the spatiotemporal information of the input. We train a C3D model for RGB and depth videos respectively. The model trained on

Sports-1M dataset is used as initialization. We follow the experimental settings in C3D [34] which uses $171 \times 128$ pixel 16-frame length video clips as input and utilized average pooling to aggregate $fc6$ features into video descriptors. Linear SVM is employed to do classification after L2 normalization. We also test the performance of C3D with 8-frame length input. Besides fc6 features, the performance of softmax layer output is also reported.

**C3D+hand mask**: besides using original RGB and depth frames as input directly sending to the C3D model, we also evaluate the performance of a hand segmentation based C3D method. Since close-range depth camera realsense can eliminate most of the background information and the captured depth frame can be roughly considered as a hand mask, we use it to perform hand segmentation on the RGB frame. Then the segmented hand region is used as the input of the C3D model.

**C3D+LSTM+RSTTM**: In [46], we propose a model by augmenting C3D with a recurrent spatiotemporal transform module (RSTTM). There are three parts in an RSTTM: a localization network, a grid generator and a sampler. The localization network predicts a set of transformation parameters conditioned on the input feature through a number of hidden layers. Then, the grid generator uses the predicted transformation parameters to construct a sampling grid, which is a set of points where the source map should be sampled to generate the target transformed output. Finally, the sampler takes the feature map to be transformed and the sampling grid as inputs, producing the target output map sampled from the input at the grid points. In C3D model, the 3D feature map is inserted with an RSTTM, which can actively warp the 3D feature map into a canonical view in both spatial and temporal dimensions. The RSTTM has recurrent connections between neighboring time slices, which means the transform parameters are predicted conditioned on the current input feature and the previous state. Finally, the output of $fc6$ layer in C3D is connected with a single-layer LSTM with 256 hidden units.

**VGG16+LSTM** makes use of the recurrent neural networks (RNN) to model the evolution of the sequence. With gate units, the long short term memory network (LSTM) [36] addresses the problem of gradient vanishing and explosion in RNN. We connect a single-layer LSTM with 256 hidden units after the first fully-connected layer of VGG16 to process sequence inputs. Videos are split into fixed-length clips and the VGG16+LSTM network predicts the label of each clip as described in [47]. The predictions of clips are averaged for video classification. We finally use non-overlapping $160 \times 120$ pixel 16-frame length video clips as input with the tradeoff between accuracy and computational complexity. We also test the performance of $lstm7$ layer features plus linear SVM.

**IDMM+CaffeNet [38]** encodes both spatial and temporal information of a video into an image called improved depth motion map (IDMM) which allows the use of the existing 2D ConvNets for classification. We construct IDMMs as introduced in [38] by accumulating the absolute depth difference between current frame and the starting frame for each pre-segmented gesture samples. We use IDMMs to train a CaffeNet with five convolutional layers and three fully-connected layers. The classification result of an IDMM represents the prediction of the whole gesture sample.

TABLE IV
GESTURE CLASSIFICATION ACCURACY OF THE BASELINES IN SEGMENTED EGOGESTURE DATA

| Method | RGB | depth | RGB-D |
|---|---|---|---|
| iDT-FV [24] | 0.643 | - | - |
| SNV [25] | - | 0.569 | - |
| MFSK-BoVW [26] | - | - | 0.464 |
| IDMM+CaffeNet [38] | - | 0.664 | - |
| VGG16 [32] softmax | 0.572 | 0.579 | 0.612 |
| VGG16 fc6 | 0.625 | 0.623 | 0.665 |
| VGG16+LSTM [36] softmax | 0.673 | 0.690 | 0.725 |
| VGG16+LSTM [36] lstm7 | 0.747 | 0.777 | 0.814 |
| C3D [34] fc6, 8 frames | 0.817 | 0.844 | 0.865 |
| C3D softmax, 16 frames | 0.851 | 0.868 | 0.887 |
| C3D fc6, 16 frames | **0.864** | **0.881** | **0.897** |
| C3D+HandMask | - | - | 0.872 |
| C3D+LSTM+RSTTM [46] | 0.893 | 0.906 | 0.922 |

**Training details of deep features:** We set the learning rate and the batch size as large as possible in our experiments. When the loss is steady, we reduce the learning rate with a fixed decay factor which is set to 10. Stochastic Gradient Descent (SGD) is used for optimization. More specifically, for learning rate: VGG16 (0.001), C3D (0.003), VGG16+LSTM (0.0001). For the step size of learning rate decay: VGG16 (5), C3D (5), VGG16+LSTM (10). For batch size: VGG16 (60), C3D (20), VGG16+LSTM (20).

*3) Results and Analysis:* The classification accuracies of the representative methods are listed in Table IV. In the method column, models with the suffix "softmax" means that the results are generated directly from the softmax layer of the network, which is an end-to-end fashion. Models with the suffix of other layers such as "fc6" and "lstm7" means that we use the output of the specified layer of the network as a feature vector to train a linear SVM classifier.

**Comparison between different methods:** As we can see, in most cases, deep learned features perform much better than hand-crafted features, i.e., iDT, SNV and MFSK. The hand-crafted features are usually computationally intensive and have a high cost in time and storage which are not suitable for large-scale dataset. For deep learned features, VGG16 does not perform as well as other approaches since it losses the temporal information seriously. Directly applying 2D ConvNets to individual frames of videos can only characterize the visual appearance. For example, it is impossible to distinguish between "zoom in" and "zoom out" just with the information of appearance. Benefit from the attached temporal model, VGG16+LSTM improves the performance of VGG16 significantly.

The performance of C3D based model is obviously superior to those of other methods with a margin more than 10%. It is probably because of the excellent spatiotemporal learning ability of C3D. C3D with 16-frame length input performed better than that with 8-frame length input on our dataset, which is inconsistent with the conclusion in [16]. It is probably because of the large variation of gesture duration in our dataset. The

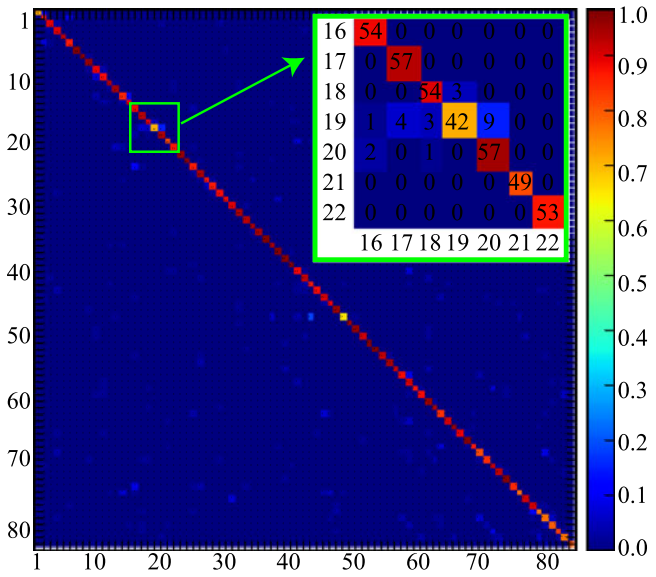



Fig. 5. The confusion matrix of C3D with 16-frame length input and RGB-D fusion on EgoGesture dataset.

duration of a single gesture varies from 3 to 196 in our dataset, while the length of a segmented gesture in nvGesture dataset [16] is 80 frames.

For the two methods built on top of C3D model, C3D+HandMask uses hand mask generated from the depth frame to perform hand segmentation on the RGB frame in order to get rid of the background noise. However, it performs worse than the C3D model with directly result-fusion from RGB and depth channels. We believe the performance is affected by the quality of hand masks. Inaccurate hand segmentation may lose important information. The model C3D+LSTM+RSTTM achieves consistent improvement against C3D in all of the three modalities. The RSTTM module can actively transform feature maps to a canonical view which is easier to be classified. This can tackle with the camera global motion which is often an issue in egocentric vision domain.

**Comparison between different settings:** The results on our dataset show that by adding an SVM on top of the neural networks, either 2D , 3D ConvNet or RNN, the performance is consistently superior to the direct softmax output of the neural networks, which prove that SVM can bring more discrimination power for classification.

**Comparison between different modalities:** Generally, the results on depth data are better than those on RGB data as the short-range depth sensor can eliminate most of the noise from the background. However, the depth sensor is easy to be affected in outdoor environment with strong illumination. Since the two modalities are complementary, the performance are further improved by fusing the results from the two modalities.

**Analysis of confusion matrix:** The confusion matrix of C3D by fusing the results obtained with RGB and depth inputs is shown in Fig. 5. The gesture classes with the highest accuracy are: “Draw circle with hand in horizontal surface” (Class 73), “Dual hands heart” (Class 53), “Applaud” (Class 52), “Wave finger” (Class 40), “Pause” (Class 36), “Zoom in with fists” (Class 8) and “Cross index fingers” (Class 7) which are all with an accuracy of 98.3%. The gesture classes with the lowest classification accuracies are: “Grasp” (66.1%), “Sweep cross” (71.2%), and “Scroll hand towards right” (72.4%). Specifically, the most confusing class of ”Grasp” (Class 48) is “Palm to fist” (Class 43), “Sweep cross” (Class 19) is easy to be classified as “Sweep checkmark” (Class 20), while “Scroll hand towards right” (Class 1) is likely to be regard as “Scroll hand towards left” (Class 2). It is reasonable since these gestures contain similar movements.

**Analysis of different scenes:** By analyzing the classification results of each scene (shown in Fig. 6), we can find several interesting facts: 1) the iDT feature is easy to be affected by global motion with worse performance in scene 4, 5 and 6 which contain egocentric motion or background dynamics. 2) In outdoor environment, deep learned features (i.e., VGG16 and C3D) from the depth channel is weaker than that from the RGB channel in most cases, which can be seen in the results of scene 5 and 6. The reason is that the depth sensor is easily to be affected by outdoor environmental lights. 3) The egocentric motion caused by walking hurts the performance for all the methods which can be seen in the results of scene 4 and 6. The results of scene 4 do not degenerate too much because the walking speed is low due to the space limit in an indoor environment. 4) Illumination changing affects the RGB feature more than depth feature. Evidence can be found in the results of VGG16+LSTM and C3D in scene 3 where the performers are facing to a window. 5) RGB and depth results fusion can consistently improve the model performance.

**Domain adaptation:** In our experimental setting, the data are split by subjects into training (60%), validation (20%) and testing (20%) sets. The training set and the testing set are from different subjects, where the data distributions are related but biased. This can be called cross-subject test, which is a common experimental setting in gesture recognition domain, as it can evaluate the domain adaptation ability of the methods. For comparison, we conduct another experiment without cross-subject setting. we split data on a video level. Video data from all subjects are collected together. 20% and 20% data are randomly sampled from the whole data for validation and testing, respectively. The rest data are used for training. Consequently, data from all subjects are included in both training and testing set. The random sampling results in 14511, 4828 and 4822 samples for training, validation and testing, respectively. The classification results of three representative models: VGG16, VGG16+LSTM and C3D are listed in Table V. In the table, the label “w/o CS” and “CS” correspond to without cross-subject and with cross-subject, respectively. Comparing to the results in Table IV with cross-subject setting, the performance (without cross-subject setting) of all the 3 methods in RGB, depth and RGB-D modalities has consistent improvement, with the maximum 0.075 and the minimum 0.024. This indicates that the distributions of data from different subjects have bias which causes the performance decrease when training and testing data are from different subjects. This also proves that in our dataset the same gesture performed by different subjects has certain diversity on hand pose, movement speed and range.

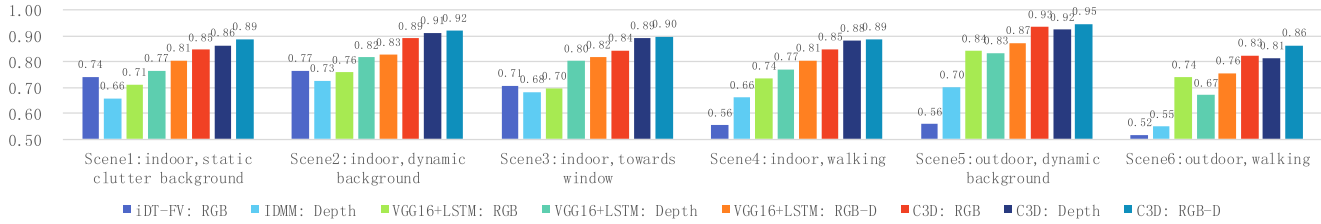



Fig. 6. Classification accuracy of baselines in 6 different scenes on EgoGesture dataset.

TABLE V
CLASSIFICATION ACCURACY WITH OR WITHOUT CROSS-SUBJECT SETTING

| Method | Modality | Accuracy (w/o CS) | Accuracy (CS) | $\delta$ |
|---|---|---|---|---|
| VGG16 fc6 | RGB | 0.667 | 0.625 | 0.042 |
| VGG16+LSTM lstm7 | RGB | 0.764 | 0.689 | 0.075 |
| C3D fc6, 16 frames | RGB | **0.892** | **0.864** | 0.028 |
| VGG16 fc6 | depth | 0.647 | 0.623 | 0.024 |
| VGG16+LSTM lstm7 | depth | 0.801 | 0.732 | 0.069 |
| C3D fc6, 16 frames | depth | **0.907** | **0.881** | 0.026 |
| VGG16 fc6 | RGB-D | 0.697 | 0.665 | 0.032 |
| VGG16+LSTM lstm7 | RGB-D | 0.826 | 0.753 | 0.073 |
| C3D fc6, 16 frames | RGB-D | **0.922** | **0.897** | 0.025 |

TABLE VI
CLASSIFICATION ACCURACY OF C3D WITH DOMAIN ADAPTATION ON DIFFERENT SCENES

| Configuration | Modality | Accuracy | |
|---|---|---|---|
| from stationary (scene1,2,3,5) to walking (scene4,6) | RGB | 0.773 | s4: 0.794<br>s6: 0.751 |
| | depth | 0.790 | s4: 0.870<br>s6: 0.711 |
| | RGB-D | 0.826 | s4: 0.880<br>s6: 0.773 |
| from indoor (scene1,2,3,4) to outdoor (scene5,6) | RGB | 0.820 | s5: 0.889<br>s6: 0.751 |
| | depth | 0.764 | s5: 0.880<br>s6: 0.649 |
| | RGB-D | 0.846 | s5: 0.911<br>s6: 0.781 |

Comparing to other two methods, C3D has the smallest performance decrease (mean: 0.026), which demonstrates it has the better domain adaptation ability on different subjects.

To further evaluate the domain adaptation ability of the winning method C3D on different scenes, we conduct experiments with two settings: 1) transferring the model trained on stationary scenes to walking scenes; 2) transferring the model trained on indoor scenes to outdoor scenes.

Table VI lists the results of C3D on the two settings with different modalities. We also report the classification accuracy on each testing scene and the performance degradation against the results in Fig. 6. For the 1st setting from stationary to walking scene, C3D using RGB channels input performs worse than the depth channel input (RGB: 0.773; depth: 0.790). For the 2nd setting from indoor to outdoor scene, C3D using depth channel input performs worse than the RGB channel input (RGB: 0.820; depth: 0.764). The best performance (RGB-D, 0.826) in the 1st setting is lower than the best performance (RGB-D, 0.846) in the 2nd setting. We can conclude that egocentric motion is the more critical factor in gesture recognition comparing to outdoor environment light interference. Scene 6 is the most challenging scene where subjects walk in an outdoor environment, which causes the largest dataset bias. However, this is a common usage scenario for wearable smart devices in daily life.

### C. Gesture Spotting and Recognition in Continuous Data

It is worthy to note that gesture classification in segmented data is a preliminary task to evaluate the performance of different feature representations for spatial and temporal modeling. In our dataset, the manual annotation of the beginning and ending frames of the gesture sample lead to a tight temporal segmentation of the video, where the non-gesture parts of the video are eliminated. In a practical hand gesture recognition system, gesture spotting and recognition in continuous data is the final task which is more challenging. It aims to perform temporal segmentation and classification in an unsegmented video stream. Performance of this task is evaluated by the Jaccard index used in ChaLearn LAP 2016 challenges [8]. This metric measures the average relative overlap between the ground truth and the predicted label sequences for a given input.

For sequence $s$, let $G_{s,i}$ and $P_{s,i}$ be binary indicator vectors in which 1-values correspond to frames where the $i$th gesture is being performed. The Jaccard index for the $i$th class is defined as:

$$J_{s,i} = \frac{G_{s,i} \cap P_{s,i}}{G_{s,i} \cup P_{s,i}} \tag{4}$$

where $G_{s,i}$ and $P_{s,i}$ are the ground truth and prediction of the $i$th gesture label at sequence $s$ respectively. When $G_{s,i}$ and $P_{s,i}$ are both empty, $J_{s,i}$ is defined to be 0.

The Jaccard index for the sequence $s$ with $l_s$ unique true labels is computed as:

$$J_s = \frac{1}{l_s} \sum_{i=1}^{L} J_{s,i} \tag{5}$$

where $L$ is the number of gesture classes.

TABLE VII
GESTURE SPOTTING AND RECOGNITION RESULTS OF THE BASELINES IN CONTINUOUS EGOGESTURE DATA

| Method | Modality | Jaccard | Runtime |
|---|---|---|---|
| sw+C3D [34]-$l16s16$ | RGB | 0.585 | 624fps |
| sw+C3D-$l16s8$ | RGB | 0.659 | 312fps |
| sw+C3D+STTM-$l16s8$ | RGB | **0.670** | 215fps |
| lstm+C3D-$l16s8$ | RGB | 0.619 | 219fps |
| QOM+IDMM [38] | depth | 0.430 | 30fps |
| sw+C3D-$l16s16$ | depth | 0.600 | 626fps |
| sw+C3D-$l16s8$ | depth | 0.678 | 313fps |
| sw+C3D+STTM-$l16s8$ | depth | 0.681 | 229fps |
| lstm+C3D-$l16s8$ | depth | **0.710** | 230fps |
| sw+C3D-$l16s16$ | RGB-D | 0.618 | 312fps |
| sw+C3D-$l16s8$ | RGB-D | 0.698 | 156fps |
| sw+C3D+STTM-$l16s8$ | RGB-D | 0.709 | 111fps |
| lstm+C3D-$l16s8$ | RGB-D | **0.718** | 112fps |

Finally, the mean Jaccard index of all the testing sequences is calculated as the final evaluation metric.

$$\bar{J}_S = \frac{1}{n}\sum_{j=1}^{n} J_{s_j} \tag{6}$$

*1) Baseline Methods:* We evaluate three strategies for temporal segmentation and classification in continuous EgoGesture data.

**Sliding windows**: we employ a length-fixed window sliding along the video stream, and perform classification within the window. Since C3D [34] has been tested to be the best classification model on this dataset, we use it as the classification method. We train a C3D model to classify 84 gestures (with an extra non-gesture class) for EgoGesture dataset. We collect training samples of the non-gesture class in the 16-frame intervals before the starting frame and after the ending frame of each gesture sample. For testing, a 16-frame length sliding window with 8 or 16 frame stride is used to slide through the whole sequence to generate video clips. The class probability of each clip predicted by C3D softmax layer is used to label all the frames in the clip. For the sliding windows with overlapping, the frame labels are predicted by accumulating the classification scores obtained from the two overlapped windows. The most possible class is chosen as the label for each single frame. The Jaccard index for detection is shown in Table VII. In the table, $l16s16$ denotes 16-frame length sliding window with 16-frame stride. We also evaluate the C3D model augmented with a spatiotemporal transform module (STTM) proposed in our previous work [46].

**Sequence labeling:** we employ an RNN to model the evolution of a complete video sequence. We choose to utilize a layer of LSTM to predict the class labels of each video clip based on the C3D features extracted at the current time slice and the hidden states of LSTM at the previous time slice. The whole model consisting of a C3D and a layer of LSTM with 256 units is end-to-end trainable. For training, we firstly generate a set of weakly segmented gesture samples that contain not only the valid gestures but also the non-gesture data. For efficiency, we constrain the maximum length of a weakly segmented gesture sample to be 120 frames. When testing, a whole video of arbitrary length is input to the unified model, generating a sequence of clip-level labels. At last the clip-level labels are converted to frame-level labels with the same operation of that used in the sliding window strategy.

**Temporal pre-segmentation:** this method is proposed in [38] which employs the quantity of movement (QOM) feature [39] to detect the starting and ending frames of each candidate gesture and pre-segment it from the video stream. QOM is calculated in the depth channel. It is assumed that all gestures starts from a similar pose, referred as neutral pose. The QOM measures the pixel-wise difference between a given depth image and the depth image of the neural pose. When the accumulated difference exceeds a threshold, the given frame is considered to be within a gesture interval. After pre-segmentation, a depth feature called Improved Depth Motion Map (IDMM) [38] is employed for classification. The IDMM, which converts the depth image sequence into one image, is constructed and fed to a CaffeNet to perform classification.

*2) Results and Analysis:* In Table VII, the prefix "sw" denotes sliding window strategy, the suffix "$l16s16$" denotes 16-frame length sliding window with 16-frame stride.We can see that the performances of C3D-$l16s8$ with overlapped sliding windows are better than those of C3D-$l16s16$ in all the modalities. The best performance (0.710) is achieved by lstm+C3D to model the temporal evolution from the RGB-D data. However, the performance of lstm+C3D on RGB data is even lower than that of C3D-$l16s8$ with sliding window strategy. It is probably because that the background in RGB data is much more complicate and dynamic than that of depth data. Hence, it is more difficult to model the long-term evolution of sequences from RGB data. We believe the detection results can be improved by reducing sliding window stride. However, the tradeoff between the accuracy and computational complexity should be considered. The runtime of the methods is also listed in the Table VII. A single K40 Tesla GPU and Intel i7-3770 CPU @3.4GHz are used. In QOM+IDMM method, the most time consuming step is to convert the depth sequence into one image with IDMM, making it less efficient than C3D model. Another disadvantage of it is that the detection performance heavily relies on the pre-segmentation which could be the bottleneck of the two-stage stratagem. From the results in Table VII, we can find that the performance on the task of gesture spotting and recognition in continuous data is far from satisfactory. To realize real-time interaction, extensive efforts have to be dedicated.

## V. CONCLUSION AND OUTLOOK

In this work, we have introduced up-to-date the largest dataset called EgoGesture for the task of egocentric gesture recognition with sufficient size, variation and reality, to successfully train deep networks. Our dataset is more complex than any existing datasets as our data is collected from the most diverse scenes. By evaluating several representative methods on our dataset, we obtain these conclusions: 1) the 3D ConvNet is more suitable for gesture modeling than 2D ConvNet and hand-crafted features;

2) Depth modality is more discriminative than RGB modality in most cases as background noise is eliminated. But it could degenerate in outdoor scene (see C3D) as the depth sensor may be affected by environmental lights. Multimodality fusion can boost the performance. 3) The egocentric motion caused by subject walking is the most critical factor which results in the largest dataset bias; 4) Compared to gesture classification in segmented data, the performance on gesture detection is far from satisfaction and has much more space to improve.

Based on our proposed dataset, there are several works can be further explored: 1) More data-hungry model for spatiotemporal modeling can be investigated. 2) By analyzing the attributes of our collected data, transfer learning between different views, locations or tasks is worthy to study to fit more usage scenarios. 3) Online gesture detection is another important task to make the gesture recognition technique applicable.

## ACKNOWLEDGMENT

The authors would like to thank L. Shi, Y. Gong, X. Li, Y. Li, X. Zhang, and Z. Li for their contributions to dataset construction.

## REFERENCES

[1] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 3, pp. 311–324, May 2007.

[2] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: A survey," *Artif. Intell. Rev.*, vol. 43, no. 1, pp. 1–54, 2015.

[3] T.-K. Kim and R. Cipolla, "Canonical correlation analysis of video volume tensors for action categorization and detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 8, pp. 1415–1428, Aug. 2009.

[4] L. Liu and L. Shao, "Learning discriminative representations from RGB-D video data," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, vol. 1, pp. 1493–1500.

[5] A. Kurakin, Z. Zhang, and Z. Liu, "A real time system for dynamic hand gesture recognition with a depth sensor," in *Proc. IEEE 20th Eur. Signal Process. Conf.*, 2012, pp. 1975–1979.

[6] A. Memo and P. Zanuttigh, "Head-mounted gesture controlled interface for human-computer interaction," *Multimedia Tools Appl.*, vol. 77, no. 1, pp. 27–53, 2018.

[7] S. Escalera *et al.*, "Chalearn multi-modal gesture recognition 2013: Grand challenge and workshop summary," in *Proc. 15th ACM Int. Conf. Multimodal Int*, 2013, pp. 365–368.

[8] J. Wan *et al.*, "Chalearn looking at people RGB-D isolated and continuous datasets for gesture recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshops*, 2016, pp. 56–64.

[9] S. Bambach, S. Lee, D. J. Crandall, and C. Yu, "Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions," in *Proc. IEEE Int. Conf. Comput. Vision*, Dec. 2015, pp. 1949–1957.

[10] Y. Huang, X. Liu, X. Zhang, and L. Jin, "A pointing gesture based egocentric interaction system: Dataset, approach and application," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshops*, 2016, pp. 16–23.

[11] G. Rogez, J. S. Supancic, and D. Ramanan, "Understanding everyday hands in action from RGB-D images," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 3889–3897.

[12] L. Baraldi, F. Paci, G. Serra, L. Benini, and R. Cucchiara, "Gesture recognition in ego-centric videos using dense trajectories and hand segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshops*, 2014, pp. 688–693.

[13] S. Ruffieux, D. Lalanne, and E. Mugellini, "Chairgest: A challenge for multimodal mid-air gesture recognition for close HCI," in *Proc. 15th ACM Int. Conf. Multimodal Interaction*, 2013, pp. 483–488.

[14] S. Escalera *et al.*, "ChaLearn looking at people challenge 2014: Dataset and results," in *Proc. Comput. Vision*, 2014, pp. 459–473.

[15] E. Ohn-Bar and M. M. Trivedi, "Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 6, pp. 2368–2377, Dec. 2014.

[16] P. Molchanov *et al.*, "Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 4207–4215.

[17] T. Starner, J. Weaver, and A. Pentland, "Real-time american sign language recognition using desk and wearable computer based video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 12, pp. 1371–1375, Dec. 1998.

[18] S. Karaman *et al.*, "Hierarchical hidden Markov model in detecting activities of daily living in wearable videos for studies of dementia," *Multimedia Tools Appl.*, vol. 69, no. 3, pp. 743–771, 2014.

[19] I. González-Díaz *et al.*, "Recognition of instrumental activities of daily living in egocentric video for activity monitoring of patients with dementia," in *Health Monitoring and Personalized Feedback Using Multimedia Data*. New York, NY, USA: Springer, 2015, pp. 161–178.

[20] C. F. Crispim-Junior *et al.*, "Semantic event fusion of different visual modality concepts for activity recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1598–1611, Aug. 2016.

[21] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang, "Exploiting feature and class relationships in video categorization with regularized deep neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 352–364, Feb. 2018.

[22] T. Sharp *et al.*, "Accurate, robust, and flexible real-time hand tracking," in *Proc. 33rd Annu. ACM Conf. Human Factors Comput. Syst*, 2015, pp. 3633–3642.

[23] C. Wan, A. Yao, and L. Van Gool, "Hand pose estimation from local surface normals," in *Proc. Eur. Conf. Comput. Vision*, 2016, pp. 554–569.

[24] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vision*, Dec. 2013, pp. 3551–3558. [Online]. Available: https://hal.inria.fr/hal-00873267

[25] X. Yang and Y. Tian, "Super normal vector for activity recognition using depth sequences," in *Proc. 2014 IEEE Conf. Comput. Vision Pattern Recognit.*, Columbus, OH, USA, Jun. 23-28, 2014, pp. 804–811.

[26] J. Wan, G. Guo, and S. Z. Li, "Explore efficient local features from RGB-D data for one-shot learning gesture recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1626–1639, Aug. 2016. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2015.2513479

[27] Y. Zhu, W. Chen, and G. Guo, "Evaluating spatiotemporal interest point features for depth-based action recognition," *Image Vision Comput.*, vol. 32, no. 8, pp. 453–464, 2014.

[28] Y.-G. Jiang, Q. Dai, W. Liu, X. Xue, and C.-W. Ngo, "Human action recognition in unconstrained videos by explicit motion modeling," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3781–3795, Nov. 2015.

[29] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *11th Eur. Conf. Comput. vision*, 2010, pp. 143–156.

[30] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3d action recognition with random occupancy patterns," in *Proc. 12th Eur. Conf. Comput. Vision*, 2012, pp. 872–885.

[31] Y. Jang, I. Jeon, T.-K. Kim, and W. Woo, "Metaphoric hand gestures for orientation-aware VR object manipulation with an egocentric viewpoint," *IEEE Trans. Human-Mach. Syst.*, vol. 47, no. 1, pp. 113–127, Feb. 2017.

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

[33] A. Karpathy *et al.*, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2014, pp. 1725–1732.

[34] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 4489–4497.

[35] C. Cao, Y. Zhang, C. Zhang, and H. Lu, "Action recognition with joints-pooled 3d deep convolutional descriptors," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 3324–3330.

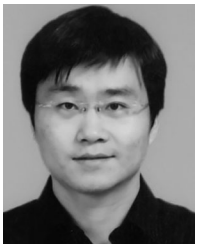
[36] A. Graves, "Generating sequences with recurrent neural networks," *CoRR*, arxiv:1308.0850, 2013.

[37] N. Nishida and H. Nakayama, "Multimodal gesture recognition using multi-stream recurrent neural network," in *Proc. Pacific-Rim Symp. Image Video Technol*, 2015, pp. 682–694.

[38] P. Wang *et al.*, "Large-scale continuous gesture recognition using convolutional neutral networks," in *Proc. 23rd Int. Conf. Pattern Recognit.*, arxiv:1608.06338, 2016, pp. 13–18.

[39] F. Jiang, S. Zhang, S. Wu, Y. Gao, and D. Zhao, "Multi-layered gesture recognition with kinect," *J. Mach. Learn. Res.*, vol. 16, pp. 227–254, 2015.

[40] Y. M. Lui, "Human gesture recognition on product manifolds," *J. Mach. Learn. Res.*, vol. 13, pp. 3297–3321, 2012.
[41] N. C. Camgoz, S. Hadfield, O. Koller, and R. Bowden, "Using convolutional 3d neural networks for user-independent continuous gesture recognition," in *Proc. 23rd Int. Conf. Pattern Recognit.*, 2016, pp. 49–54.
[42] M. R. Malgireddy, I. Nwogu, and V. Govindaraju, "Language-motivated approaches to action recognition," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 2189–2212, 2013.
[43] D. Wu and L. Shao, "Deep dynamic neural networks for gesture segmentation and recognition," in *Proc. Workshop Eur. Conf. Comput. Vis.*, vol. 19, no. 20, 2014, pp. 552–571.
[44] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 677–695, Jul. 1997.
[45] B.-K. Bao, G. Liu, C. Xu, and S. Yan, "Inductive robust principal component analysis," *IEEE Trans. Image Process.*, vol. 21, no. 8, pp. 3794–3800, Aug. 2012.
[46] C. Cao, Y. Zhang, Y. Wu, H. Lu, and J. Cheng, "Egocentric gesture recognition using recurrent 3d convolutional neural networks with spatiotemporal transformer modules," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 3763–3771.
[47] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 2625–2634.



**Yifan Zhang** (M'10) received the B.E. degree in automation from Southeast University, Nanjing, China, in 2004, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2010. Then, he has joined the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, where he is currently an Associate Professor. From 2011 to 2012, he was a Postdoctoral Research Fellow with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute (RPI), Troy, NY, USA. His research interests include machine learning, computer vision, probabilistic graphical models, and relative applications, especially on video content analysis, gesture recognition, action recognition, etc.

**Congqi Cao** received the B.E. degree in information and communication from Zhejiang University, Hangzhou, China, in 2013. She is currently working toward the Ph.D. degree in image and video analysis at the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. Her current research interests include machine learning, pattern recognition, and relative applications, especially on video-based action recognition and gesture recognition.

**Jian Cheng** (M'06) received the B.S. and M.S. degrees from Wuhan University, Wuhan, China, in 1998 and 2001, respectively, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2004. From 2004 to 2006, he was a Postdoctoral Fellow with the Nokia Research Center, Beijing, China. He is currently a Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His current research interests include machine learning, pattern recognition, computing architecture and chips, and data mining.

**Hanqing Lu** (SM'06) received the B.E. and M.E. degrees from the Harbin Institute of Technology, Harbin, China, in 1982 and 1985, respectively, and the Ph.D. degree from the Huazhong University of Sciences and Technology, Wuhan, China, in 1992. He is currently a Professor with the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include image and video analysis, pattern recognition, and object recognition. He has authored or coauthored more than 100 papers in these areas.