Journal of Ambient Intelligence and Humanized Computing (2024) 15:1557–1572
https://doi.org/10.1007/s12652-018-0989-7

ORIGINAL RESEARCH

HandSense: smart multimodal hand gesture recognition based on deep neural networks

Zhenyuan Zhang[1] · **Zengshan Tian**[1] · **Mu Zhou**[1]

Received: 20 April 2017 / Accepted: 16 August 2018 / Published online: 23 August 2018

Abstract
Hand gesture recognition (HGR) is a promising enabler for human–computer interaction (HCI). Hand gestures are normally classified into multi-modal actions, including static gestures, fine-grained dynamic gestures, and coarse-grained dynamic gestures. Among them, the fine-grained action detection is limited under the small-scale image region condition. To solve this problem, we propose the HandSense, a new system for the multi-modal HGR based on a combined RGB and depth cameras to improve the fine-grained action descriptors as well as preserve the ability to perform general action recognition. First of all, two interconnected 3D convolutional neural network (3D-CNN) are employed to extract the spatial–temporal features from the RGB and depth images. Second, these spatial–temporal features are integrated into a fusion feature. Finally, the Support Vector Machine (SVM) is used to recognize different gestures based on the fusion feature. To validate the effectiveness of the HandSense, the extensive experiments are conducted on the public gesture dataset, namely the SKIG hand gesture dataset. In addition, the feasibility of the proposed system is also demonstrated by using a challenging multi-modal RGB-Depth hand gesture dataset.

Keywords Hand gesture recognition · HandSense · Fine-grained gestures · Spatial–temporal features

1 Introduction

The articulated hand gesture recognition (HGR) is one of the core technologies for human–computer interaction in the augmented reality and virtual reality applications since this technology provides a natural way for the users to interact with virtual environment (Priyal and Bora 2013). Recently, the HGR system has been applied in many applications such as computer games (Li 2012), sign language communication, robotics, and interactive interview systems (Tsai and Lee 2011). In this circumstance, a smart multi-modal HGR system is developed in this paper with the goal of robustly recognizing the hand gestures performed by different users under the challenging real-world settings.

✉ Zhenyuan Zhang
zhangzhenyuangm@gmail.com

Zengshan Tian
tianzs@cqupt.edu.cn

Mu Zhou
zhoumu@cqupt.edu.cn

[1] Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Motivation for multi-modal gestural interfaces Among the touch, tactile, and gestural human–computer interfaces (HCI), the gestural HCI is recognized to have certain advantages compared to the other two, such as the low visual load, high level of user acceptability, and low operational error rate (Parada-Loira et al. 2014). For the driver assistance system, in particular, the reduction in visual load and non-intrusive nature stimulates many automotive companies to investigate the HCI systems to alleviate the growing concern of distraction from the complicated interfaces in the vehicles (Jahn et al. 2009).

In this paper, we focus on developing a new vision-based HGR system to classify various hand gestures performed by different users with good robustness under the challenging environments. As far as we know, the dynamic hand gestures is normally divided into the fine-grained and coarse-grained hand gestures according to the range and speed of hand movement. The fine-grained gestures are defined as the very brief and subtle hand movements which are primarily performed by driving the fingers and articulating the wrist, such as the press and click, rather than the ones from the large muscle groups. The fine-grained hand gestures have been widely used in daily life to provide more information

compared with the coarse-grained hand gestures, especially for the non-verbal applications. For example, wagging index finger means 'No' and moving index finger up and down can be mapped to the meaning of pressing virtual button in the somatic games. Besides, the fine-grained gestures has medical value since it can be used to provide a window into precise telesurgery. In addition, the study in Singh et al. (2015) claims that the fine-grained gestures provide the patients with mobility impairment with good control over the environment without the fatigue over the time.

In addition to the general study of robust descriptors and fast classification schemes for the coarse-grained hand gestures (Liu and Kehtarnavaz 2016; Pal and Kakade 2016; Rao et al. 2018), we are also motivated to show the advantages of the fine-grained hand gestures (Prakash et al. 2017; Wen et al. 2016; Sharp et al. 2015)over the traditional HGR systems.

Challenges for vision-based multi-modal HGR system There are some general problems of the HGR systems. These systems are required to generalize over the users and variation of the gestures. The associated algorithms should be robust to the varying global illumination changes and shadow artifacts. For example, the HGR in volatile environment such as the vehicle interior differs from the one in indoor environment in terms of the illumination condition. In addition, recognizing the fine-grained gestures is a complicated and challenging task. First of all, detecting the fine-grained gestures defined by the low inter-class variability is challenging. Second, for the fine-grained gestures, the finger commonly self-occludes itself throughout the performance of the gestures. Third, the fine-grained gestures are much difficult to be detected under the small-scale image region condition.

In this paper, we collect a multi-modal RGB-depth (RGBD) hand gesture dataset, including 10 static gestures, 10 fine-grained gestures, and 8 coarse-grained gestures, for the testing. Examples of hand samples are shown in Figs. 13, 14 and 15. The collected dataset allows to study the orientation invariance, impact of occlusion, and illumination variability. In addition, different feature extraction methods are also compared on this dataset.

Up to now, a set of common spatial–temporal descriptors (Vieriu et al. 2011; Simonyan and Zisserman 2014a; Ji et al. 2013; Wang et al. 2012; Karpathy et al. 2014) for action recognition are evaluated in terms of recognition rate. Different descriptors are compared over the modalities with different classification schemes, such as the Hidden Markov Model (HMM) and Convolutional Neural Network (CNN) used for finding the optimal combination and gaining insight into the strength and limitation of different approaches. In this paper, we propose a new HGR system, namely HandSense, based on the combination of depth and RGB videos. The HandSense adopts the 3D-CNN architecture which selects multiple continuous frames as the input and relies on alternating the 3D convolutional and pooling layers to extract the spatial–temporal features. In addition, the heterogeneous information from the RGBD sensors is integrated to achieve high recognition accuracy under different lighting conditions. Finally, we introduce a new multi-modal hand gesture dataset, including the static, fine-grained, and coarse-grained hand gestures, to validate the performance of the HandSense. In all, the three main contributions of this paper are summarized as follows.

- The 3D-CNN architecture in the HandSense has the ability of effectively learning the spatial–temporal features from the continuous frames.
- The proposed method runs fast with over 160 frames per second on a single GPU, and thereby the real-time capacity for the HGR can be well guaranteed.
- To validate the HandSense, the proposed method is systematically evaluated on a public gesture dataset, namely SKIG, and meanwhile a multi-modal RGBD hand gesture dataset is also collected for the testing.

The rest of paper is organized as follows. Section 2 presents some related work on hand gesture recognition. The architecture of proposed system is described in Sect. 3. Section 4 evaluates the performance of HandSense. Finally, the conclusion and some future directions are given in Sect. 5.

2 Related work

Since the quality of RGBD output from the cameras improves and the hardware cost declines, a wide array of applications have spurred significant interests in the HGR. In this section, we will focus our literature review on the fine-grained HGR systems as well as the associated HGR methods.

2.1 Fine-grained HGR systems

Although a variety of methods have been proposed in recent years to recognize hand gestures, most of them focus on the videos with coarse gestures (Ji et al. 2013; Xue et al. 2018; Ge et al. 2016; Hu et al. 2014). In recent years, as the smartphones and many other mobile devices spread, many application scenarios, such as mobile HCI systems, require to recognize the fine-grained hand gestures. The fine-grained HGR has drawn significant interest due to its potential to improve gestures resolution (Kojima et al. 2017; Ma et al. 2018; Zhao et al. 2017; Rohrbach et al. 2016; Yamada et al. 2017). In Kojima et al. (2017), the authors employ the spatial–temporal extension of histogram of the oriented gradient variation to represent the appearance and temporal change

of the fine-grained hand gestures. The authors in Ma et al. (2018) utilize the deep learning method to improve fine-grained gestures discrimination while preserving the ability to perform the general gestures recognition. In Zhao et al. (2017), the authors consider the performance of the traditional hand segmentation methods with the abundant manually labeled training data to propose an online-learning hand segmentation approach for the fine-grained HGR without any manual labeled data for the training. The authors in Rohrbach et al. (2016) show the benefit of hand-centric approach for the fine-grained gestures classification and detection. In addition, the authors in Yamada et al. (2017) apply the dense trajectory information to recognize the similar and fine-grained hand gestures.

2.2 HGR methods

Most of the vision-based HGR systems are based on the sparse or dense extraction of the spatial–temporal handcrafted features. These systems normally consist of the feature descriptor phase followed by feature encoding. The authors in Wang et al. (2012) propose a HMM-based system for the dynamic gesture trajectory modeling and recognition by using the Histograms of Gradient (HOG) feature. For the sake of reducing the high dimensionality of the HOG feature, the authors in Klaser et al. (2008) propose an accurate system for the HGR by applying the Principal Component Analysis (PCA) to compress the HOG feature. In addition, the authors propose a new local descriptor based on the histograms of oriented 3D spatial–temporal HOG for the HGR. However, it turns out that most of the current methods are dataset-dependent and there is no all-embracing method that can surpass all the others. Besides, there is a growing trend in learning low and middle level features either in the supervised or unsupervised way.

Late advancements in hardware development, particularly the powerful GPUs, are important in the revival of deep learning. The CNN architecture has become an effective tool for extracting the high-level features and shown outstanding success in the classification task (Simonyan and Zisserman 2014b; Simonyan et al. 2013; Chung et al. 2016). Different from the traditional image classification task, the most challenging aspect of the HGR is the spatial–temporal variability since different hand gestures are with different shape, duration, and integrality. Recently, the deep learning methods have been adapted for the HGR. For example, the authors in Simonyan and Zisserman (2014a) propose a 2D-CNN based two-stream convolutional network architecture, which incorporates the spatial and temporal networks for the HGR in the videos. In Baccouche et al. (2011), the authors extend the 2D-CNN to the video domain by treating the space and time as two separate dimensions of the input for the convolution. In addition, the unsupervised learning scheme based on the convolutional gated Restricted Boltzmann Machines (RBM) for training the spatial–temporal features is proposed in Taylor et al. (2010) to learn the latent representation of successive image sequences. At the same time, the 3D-CNN is also proposed to recognize hand gestures (Molchanov et al. 2015). The proposed 3D-CNN architecture contains two sub-networks: high-resolution network and low-resolution network, with independent network parameter. The final class-membership probabilities for the gesture classifier are computed by multiplying the class-membership probabilities from the two networks. Different from the traditional 2D-CNN, the input of 3D-CNN is defined as image sequences, which contain not only the spatial information but also the relevant information between adjacent images. Inspired by the 3D-CNN, we propose a new vision-based model for the spatial–temporal features learning to recognize the fine-grained hand gestures in a robust manner.

3 System description

3.1 Architecture of HandSense

The architecture of HandSense is shown in Fig. 1. The system is composed of a Kinect device and a PC. Kinect is actually a 3D camera with low price. For Kinect, there is a color camera, an Infrared Ray (IR) projector and an IR

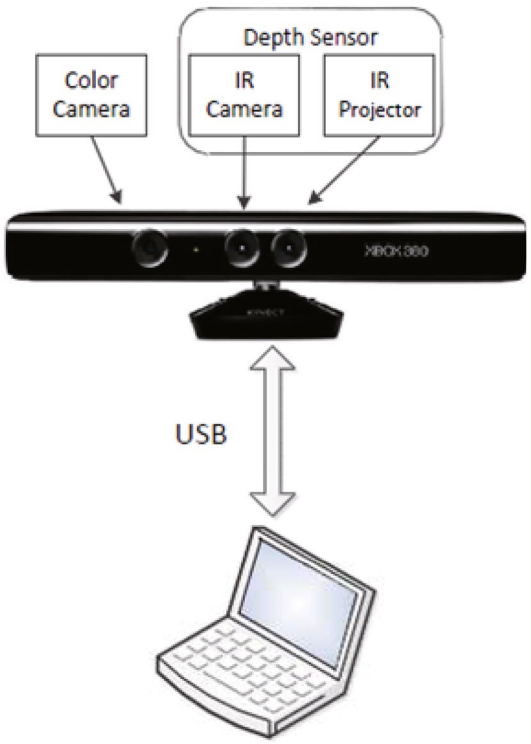

Fig. 1 Architecture of HandSense

camera. To obtain 3D images, the IR projector emits IR, and then the IR camera detects the reflected IR from the object to calculate the Time of Flight (TOF). After that, the TOF is sent to a PC via USB transmission for further processing. Kinect provides 640×480 RGB and depth images (about 30 frames/s). For the depth image, its effective depth ranges from 0 to 4096 mm. Besides, Kinect is capable of calibrating the sensors automatically under the presence of obstacles.

In general, hand gestures can be classified into three categories, static, fine-grained, and coarse-grained gestures. Each of them demands proper recognition means by which they can be properly defined to machines. As far as we know, there is no general method to be used to extract features from diverse types of hand gestures. Thus, HandSense is a promising HGR architecture. The workflow of HandSense is shown in Fig. 2. The process of HandSense incorporates three main components as follows.

Video data preprocessing Due to the fact that videos of diverse hand gesture datasets are with variable duration, we re-sample each hand gesture video into 30 frames to normalize the temporal length of gestures. Besides, all the video frames are resized into 320×240 pixels.

Hand gesture features extraction Different from traditional method, HandSense extracts spatial and temporal features of hand gestures by using 3D-CNN architecture.

Hand gestures recognition Based on the features extracted by 3D-CNN, HandSense applies SVM algorithm to recognize hand gestures.

3.2 Hand gesture features extraction

Diverse dynamic hand gestures differ in velocity, shape, and duration, which make it more difficult to recognize dynamic hand gestures compared with static ones. It is a critical problem for dynamic gestures recognition to extract spatial and temporal features simultaneously. The most popular algorithm used for dynamic gestures recognition is called as HOG-HMM which is based on handcrafted features sensitive to the sampling period.

Recently, inspired by the rapid development of deep learning in the image domain, various convolutional network models based on 2D-CNN are considered to extract image features automatically. However, these image based deep features are not suitable for dynamic hand gestures recognition due to the lack of motion modeling. To address this problem, we propose a new algorithm for multimodal hand gesture recognition based on 3D-CNN architecture, which can be used to extract spatial and motion features simultaneously.

Traditional 2D-CNN includes two parts, 2D convolution and 2D pooling operations, as shown in Fig. 3. In the 2D convolution operation, the image in the previous layer is convolved with the convolution kernel, and then put though the

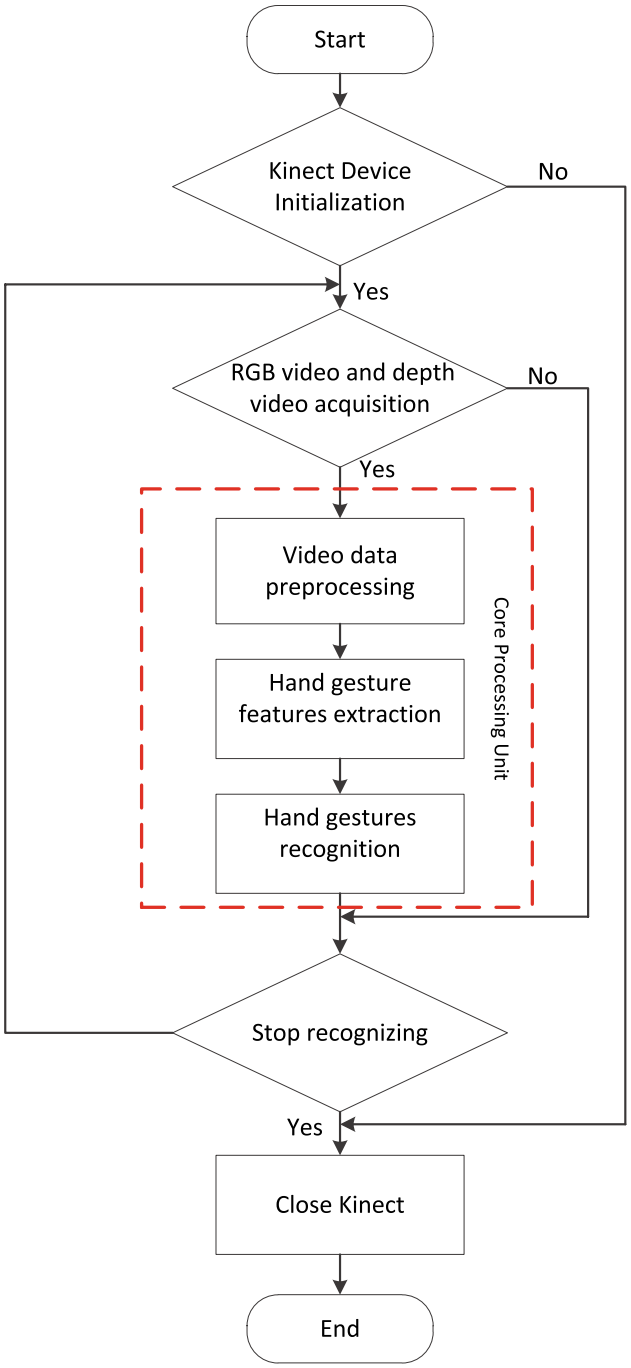

Fig. 2 Workflow of HandSense

activation function to generate the output feature map. This process can be described as

$$Y(m,n) = X(m,n) \otimes H(m,n) = \sum_{i=0}^{K_1-1} \sum_{j=0}^{K_2-1} X(m+i, n+j) H(i,j) \quad (1)$$

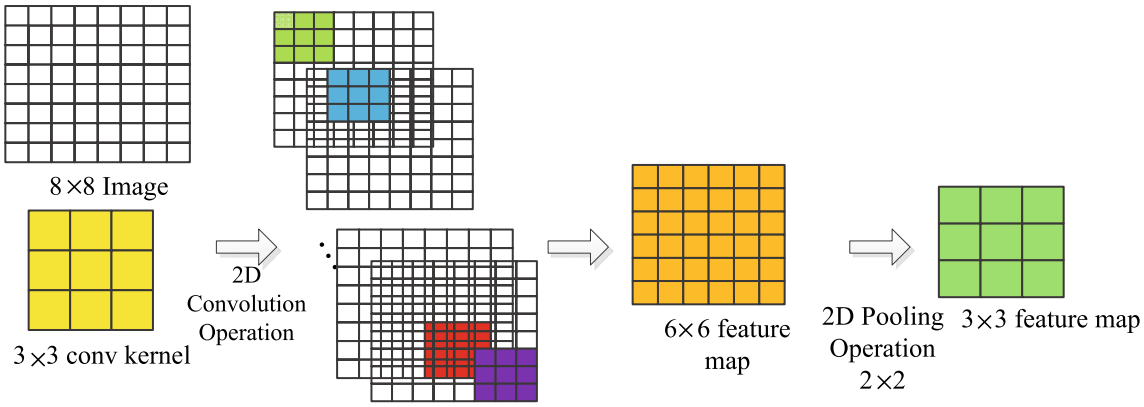

Fig. 3 Framework of 2D-CNN

where $Y(m,n)$, $X(m,n)$, and $H(m,n)$ stand for the value at position (m,n) in the feature map, image in the previous layer, and convolution kernel respectively. K_1 and K_2 denote the row and column number of the convolution kernel. The notation "$\otimes$" denotes the convolution operation. If the input is an $M \times N$ image and convolution kernel is $K_1 \times K_2$, the result of convolution operation will be a $(M - K_1 + 1) \times (N - K_2 + 1)$ feature map. To reduce the computational complexity, we perform the pooling operation after the convolution operation. The purpose of pooling operation is to generate the down-sampled version of feature map. If the factor of down-sampling is P and input feature map for the pooling operation is $M \times N$, the pooling operation will generate a $(M/P) \times (N/P)$ feature map. The 2D-CNN preserves the spatial features, but lost the temporal information of input image. However, for the dynamic hand gesture recognition problem, it is expected to capture the motion features which are involved in multiple consecutive frames. The framework of 2D-CNN is shown in Fig. 3.

Compared with 2D-CNN, 3D-CNN is capable of extracting both the spatial and temporal features by 3D convolution and pooling operations. The framework of 3D-CNN is shown in Fig. 4. Different from 2D-CNN, the input of 3D-CNN is contiguous frames, and meanwhile it utilizes a convolution kernel cube to perform the convolution operation instead of the 2D convolution kernel in 2D-CNN. In this framework, the feature cube in the convolution layer is connected to multiple contiguous frames in the previous layer to capture motion information. The value at position (x, y, z) in the feature cube can be expressed as

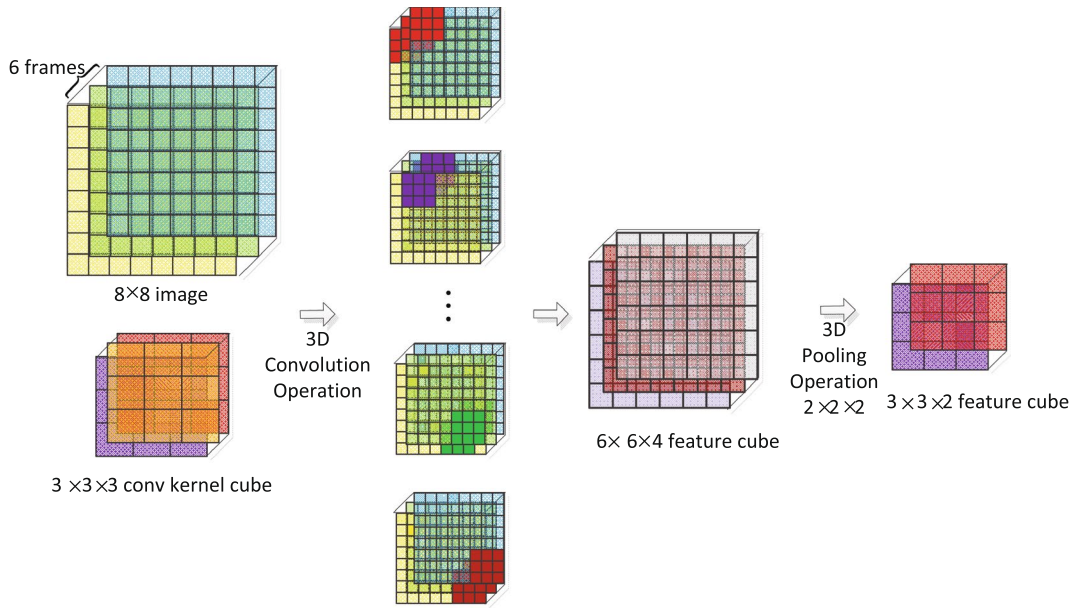

Fig. 4 Framework of 3D-CNN

$$Y(x,y,z) = X(x,y,z) \otimes H(x,y,z) = \sum_{i=0}^{K_1-1} \sum_{j=0}^{K_2-1} \sum_{k=0}^{K_3-1} X(x+i, y+j, z+k)\, H(i,j,k) \quad (2)$$

where $Y(x,y,z)$, $X(x,y,z)$, and $H(x,y,z)$ stand for the value at position (x,y,z) in the feature cube, previous contiguous frame, and convolution kernel cube. K_1, K_2 and K_3 denote the length, width, and height of the convolution kernel cube. The notation "$\otimes$" denotes the convolution operation. If the input includes P pieces of $M \times N$ images along the temporal dimension and convolution kernel cube is $K_1 \times K_2 \times K_3$, the result of convolution operation will be an $(M - K_1 + 1) \times (N - K_2 + 1) \times (P - K_3 + 1)$ feature cube. Similar to 2D-CNN, 3D-CNN applies the 3D pooling operation to subsample the feature cube, which leads to the same number of feature cubes with reduced spatial and temporal resolution.

As discussed in Hinton et al. (2012), in each convolution layer, a 2D convolution kernel can extract only one type of features, such as rough sketch, corners, and edge/color conjunctions. Thus, several convolution kernels can be used in each convolution kernel to extract various features. HandSense applies the multiple 3D convolution kernel cubes to obtain spatial and temporal features. In addition, the number of feature cubes in latter layers is increased to form more types of features from the same set of low-level feature cubes.

Based on the 3D-CNN framework described above, various 3D-CNN architectures can be designed to be applied to different applications. In our system, we propose a new 3D-CNN architecture which is developed to recognize multimodal hand gestures in HandSense. The architecture for HGR based on RGB video is shown in Fig. 5. In this architecture, the input is several sequential images extracted from a hand gesture video. Considering operation efficiency, the video is resized into the frame size of 320×240 and cropped into several crops of 16 pieces of frames before training the 3D-CNN architecture. We then apply the 3D convolution with the kernel size of $3 \times 3 \times 3$ (3×3 in the spatial dimension and 3 in the temporal dimension) in layer 1. To increase the number of feature cubes, 64 convolution kernel cubes are utilized resulting in 64 feature cubes. In the subsequent pooling operation, we apply $2 \times 2 \times 1$ (2×2 in the spatial dimension and 1 in the temporal dimension) subsampling of each feature cube, which leads to the same number of feature cubes with the reduced spatial resolution. The next layer is obtained by applying the 3D convolution through 128 kernel cubes with the kernel size of $4 \times 4 \times 3$ (4×4

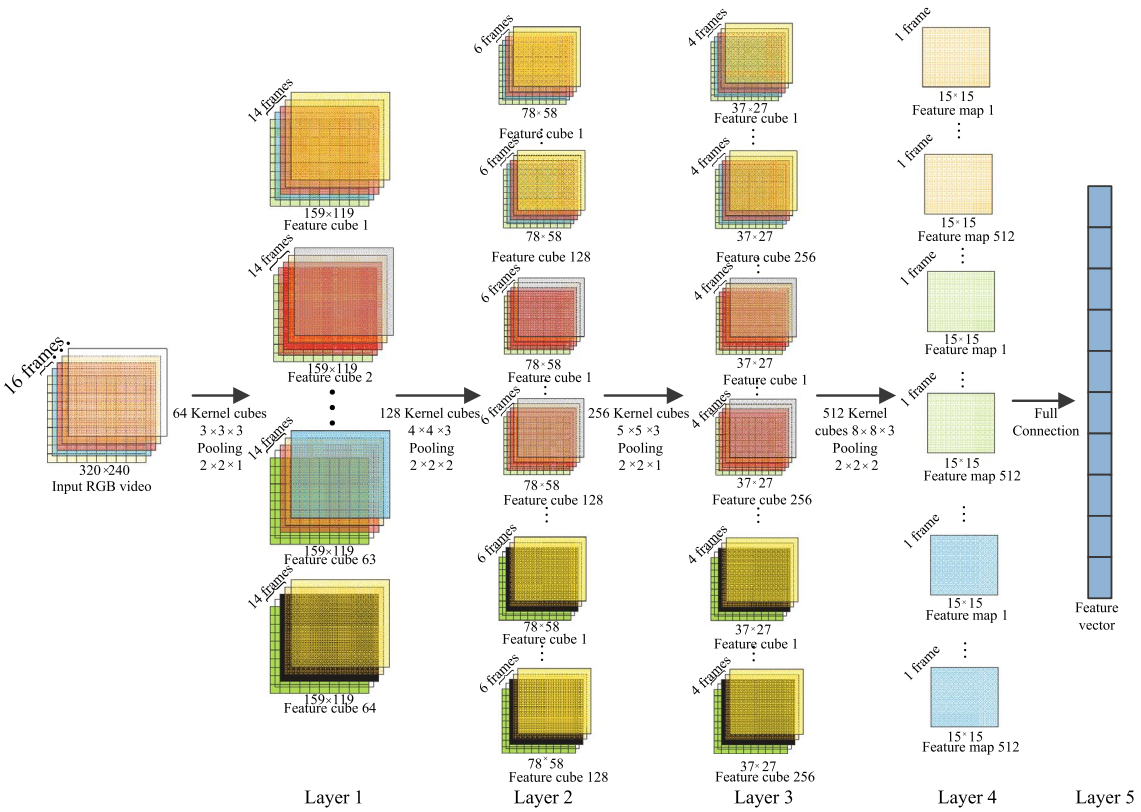

Fig. 5 3D-CNN architecture for HGR based on RGB video

in the spatial dimension and 3 in the temporal dimension) and $2 \times 2 \times 2$ (2×2 in the spatial dimension and 2 in the temporal dimension) pooling operation. The general design principle of CNN is that the number of feature maps should be increased in latter layers by generating multiple types of features from the set of lower-level feature maps. By utilizing 256 kernel cubes with the kernel size of $5 \times 5 \times 3$ and $2 \times 2 \times 1$ pooling operation, we can obtain the feature cubes in layer 3. To extract spatial and temporal features further, we use 512 kernel convolution cubes and $2 \times 2 \times 2$ pooling operation to obtain the feature maps in layer 4. At last, the full connection operation is performed to obtain a feature vector which contains the spatial and temporal information from a dynamic hand gesture video. HandSense takes the advantage of various feature vectors to recognize hand gestures.

Except for the softmax layers, all the other layers in the architecture utilize Rectified Linear Unit (ReLU) activation function, which is described as

$$f(\mathbf{z}) = \max(0, \mathbf{z}) \tag{3}$$

where $\mathbf{z}$ denotes the input vector of activation function.

Currently, the great majority of technologies for HGR are based on ordinary cameras (Liu and Wang 2011). However, RGB color images from ordinary cameras have the disadvantage that they are susceptible to the illumination changes in surroundings. In addition, single sensor system cannot provide enough information for tracking hands in 3D space since a mass of spatial position information has to be constructed from the 2D into 3D mappings. Considering the above reasons, HandSense takes the advantage of the combination of RGB and depth cameras to realize HGR function with avoiding the spatial information loss. The architecture for extracting hand gesture features in HandSense is shown in Fig. 6.

In HandSense, the architecture for extracting hand gesture features can be divided into two processing channel, RGB video channel and depth video channel. RGB and depth videos are captured synchronously by using Kinect. Then, these two types of videos are sent to the corresponding processing channel to obtain different types of features. The depth video channel is much similar to the RGB video channel. Considering the variety of convolution networks, the 3D-CNN architecture based on depth video is slightly changed compared with the one based on RGB video, mainly including the number of layers and convolution kernel parameters. At last, we fuse the features extracted from RGB and depth videos respectively to form a fusion feature vector. The fusion feature vector is a high-level abstraction in comparison with hand-designed features, and meanwhile it contains abundant spatial and temporal information extracted from hand gesture videos, which can be utilized for hand gesture recognition.

3.3 Hand gesture recognition

HandSense trains a 3D convolution network and a SVM classifier for each hand gesture class. SVM algorithm tries to find a hyperplane to separate the samples of different hand gesture classes and maximize the distance between each decision boundary and any of the samples. During the training, to learn the SVM parameters for each hand gesture action category, HandSense selects the training examples of target action category as the positive samples (with the label of target action category $y_i = +1$) and the training examples of other action categories as the negative samples (with the label of target action category $y_i = -1$). Then, we minimize the SVM objection function (Joachims 2002) over the parameter vector $\mathbf{w}$ and slack variables ξ_i.

We make use of different penalty factors for the positive and negative action categories to solve the data imbalance problem. In general, the number of negative examples p is larger than that of positive examples q. Thus, we distribute a larger penalty factor M_+ to the positive examples, which attaches more importance to the positive samples. On the contrary, the negative examples are with smaller penalty

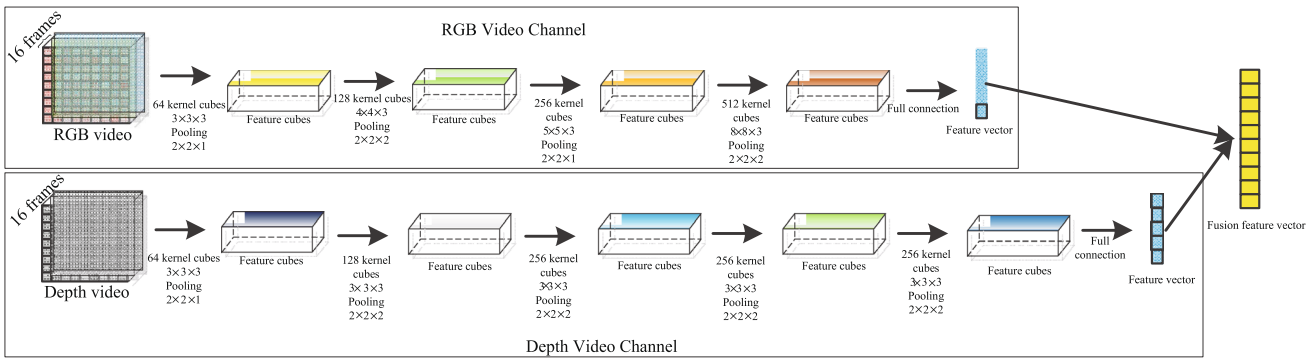

Fig. 6 Architecture for extracting hand gesture features in HandSense

factor M_-. The SVM objective function in HandSense system is described as

$$\begin{aligned} &\min_{\omega,\xi} \ \frac{1}{2}\|w\|^2 + M_+ \sum_{i=1}^{q} \xi_i + M_- \sum_{j=q+1}^{p+q} \xi_j \\ &s.t.\ y_i\left[\left(w^T H_i\right) + b\right] \geq 1 - \xi_i,\ (i = 1, 2, \ldots p+q) \\ &\quad \xi \geq 0 \end{aligned} \tag{4}$$

where H_i is the fused spatial and temporal feature vector with respect to the ith action example extracted by 3D-CNN from the RGB and depth videos. $\left(H_i, y_i\right)$ is the input vector of SVM classifier. $p+q$ is the number of total hand gesture action examples. By applying the SMO algorithm (Platt 1999), we utilize the LIBSVM (Chang and Lin 2011) to perform hand gesture classification.

4 Experiments

In this section, extensive experiments will be performed to validate the proposed hand gesture recognition system. We test HandSense on a well-known hand gesture recognition benchmark, SKIG hand gesture dataset (Liu and Shao 2013). In addition, we propose to use a new challenging dataset, which contains static, fine-grained, and coarse-grained hand gestures, to test the performance of HandSense under different scenarios. Besides, we also show the performance comparison between the classic HGR systems and proposed HandSense.

4.1 Training

We train HandSense based on SKIG hand gesture dataset, which contains 2160 hand gesture videos(including 1080 RGB and 1080 depth videos). All these videos are synchronously captured by Kinect. As shown in Fig. 7, this dataset contains in total 10 categories of hand gestures, circle (clockwise), triangle, up–down, right–left, wave, 'Z', cross, comehere, turn-around, and pat. All these 10 categories of hand gestures involve three different hand postures, fist, index, and flat.

SKIG dataset contains less than 3000 hand gesture videos, which are not enough to prevent overfitting. To solve this problem, we perform data augmentation (Krizhevsky et al. 2012) before training. As shown in Fig. 8, the data augmentation is comprised of three operations, horizontal mirroring, vertical mirroring, and random cropping.

To further avoid the overfitting and increase the generalization performance of hand gesture classifier, we also augment the dataset during the training. Different from the data augmentation before the training, the one during the training includes the rotation, scaling, and spatial elastic deformation. Besides, we also rely on a dropout regularization approach (Hinton et al. 2012) to reduce the impact of overfitting.

During the training, we select the output of softmax function as the cost function, which is described as

$$F_{\text{cost\_function}} = \frac{\exp\left(z_c\right)}{\sum_q \exp\left(z_q\right)} \tag{5}$$

where z_q is the output of neuron q.

In HandSense, we perform optimization via the stochastic gradient descent with the mini-batches of 40. We update the parameter ω in 3D-CNN by using Nesterov accelerated gradient (Sutskever et al. 2013) in each iteration i , as shown below.

$$\nabla\omega_i = \left\langle \frac{\delta F_{\text{cost\_function}}}{\delta\left(w_{i-1}\right)} \right\rangle_{batch} \tag{6}$$

$$v_i = uv_{i-1} - \lambda\nabla w_i \tag{7}$$

$$w_i = w_{i-1} + uv_i - \lambda\nabla w_i \tag{8}$$

where λ is the learning rate, which is initialized as 0.003 and with 90 percentages decrease after 10,000 iterations. u is the

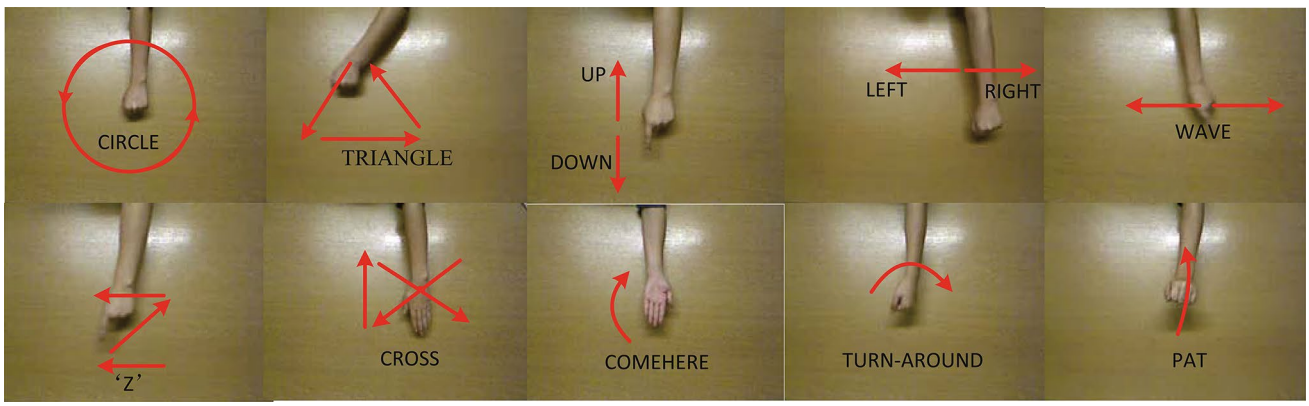

Fig. 7 SKIG dataset

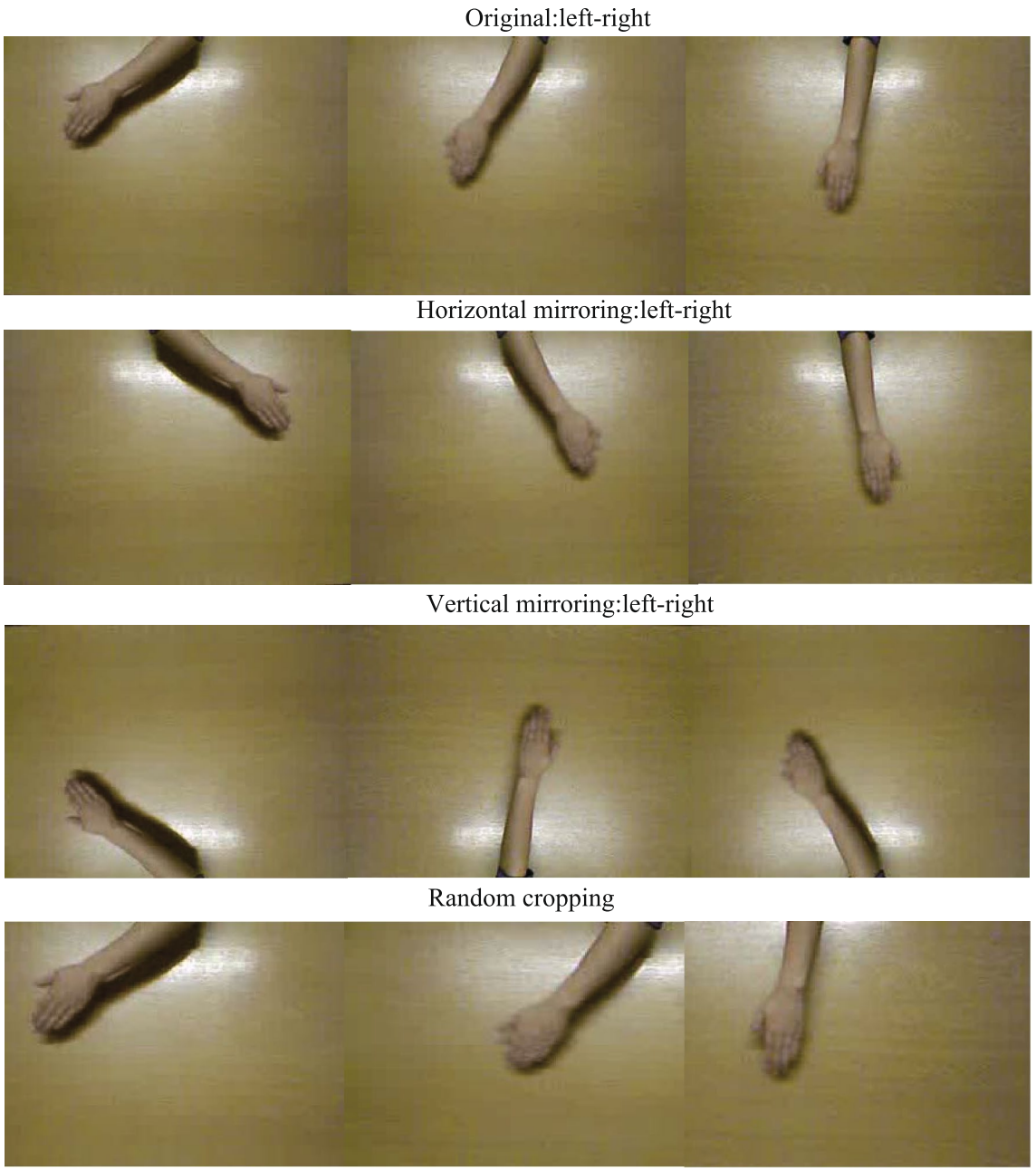

Fig. 8 Data augmentation operations

momentum coefficient, which is set as 0.9. ∇w_i is the value of gradient of cost function with respect to the parameter ω_i, which is averaged across 40 mini-batches.

In deep learning, loss curves is calculated on training and testing sets and shows how well the model is doing for these two sets. It is a summation of the errors made for each example in training or testing sets. Figure 9 plots the loss curves with respect to the training and testing process before 20,000 iterations. We can find that the loss value generally declines with the increase of number of iterations. Besides, the values of training and testing loss significantly decrease as the iteration increases to 2000, while the loss values will maintain around 0.0001 and 0.2 respectively when the number of iterations is larger than 12,000.

In HandSense, we use the public available cuda-convnet package to train our models on a single NVIDIA GTX 1080, which has 8GB memory and 2560 CUDA cores. The training process based on SKIG dataset takes roughly 3 days.

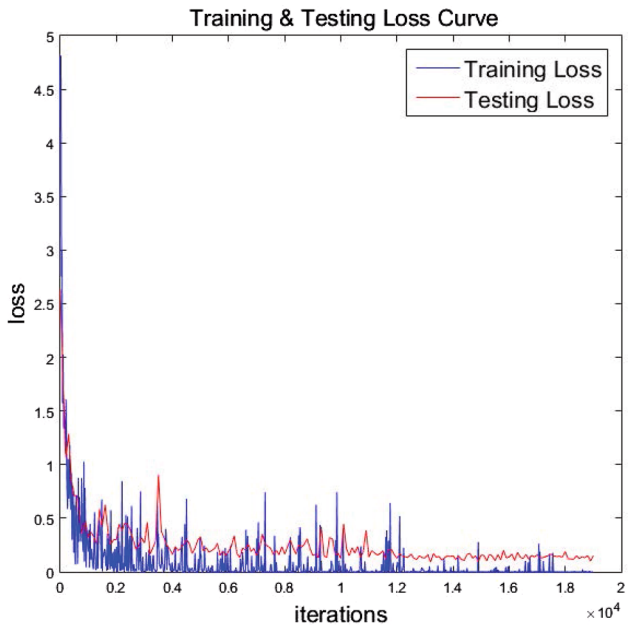

Fig. 9 Training and testing loss curve

Table 1 TP = true positive, TN = true negative, FP = false positive, FN = false negative

	Positive	Negative
Truth: positive	TP	FN
Truth: negative	FP	TN

4.2 Results of HGR based on the public hand gesture dataset

To test the accuracy of HGR, SKIG dataset is randomly divided into training (70% of the total) and testing (30% of the total) sets, resulting in 2100 training and 900 testing hand gesture videos respectively. In machine learning, *TP*, *TN*, *FP* and *FN* stand for the true positive, true negative, false positive and false negative results respectively, which are described in Table 1.

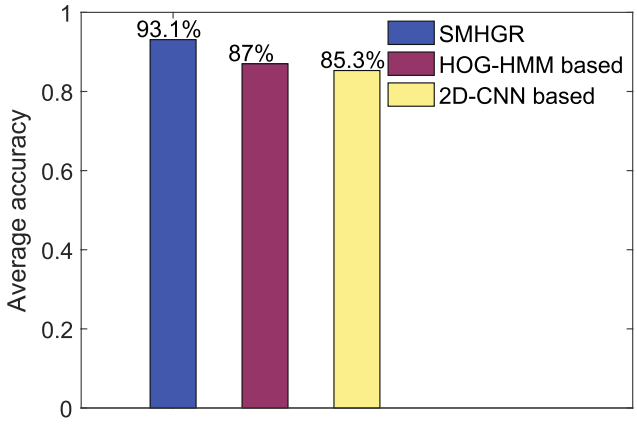

Fig. 11 Comparison result between HandSense and the current state-of-the-art HGR systems based on SKIG dataset

We evaluate the experimental results in terms of average accuracy for each category of hand gestures, which is calculated by

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

In Fig. 10, we use a confusion matrix to illustrate the accuracy for different categories of hand gestures. From this figure, we can find that the average accuracy is 93.1% and some categories of hand gestures, such as up–down and cross, can be more easily to be recognized compared to the others.

We continue to compare proposed HandSense system with the current state-of-the-art HGR systems. The HOG-HMM based HGR (Wang et al. 2012) uses color image sequences

Fig. 10 Confusion matrix with respect to SKIG dataset

	Circle	Triangle	Up-down	Left-right	Wave	'Z'	Cross	Comehere	Turn-around	Pat
Circle	91%	3%	0%	0%	0%	0%	6%	0%	0%	0%
Triangle	2%	95%	0%	0%	0%	3%	0%	0%	0%	0%
Up-down	0%	0%	98%	0%	0%	0%	0%	0%	0%	2%
Left-right	0%	0%	0%	90%	6%	4%	0%	0%	0%	0%
Wave	0%	1%	0%	7%	92%	0%	0%	0%	0%	0%
'Z'	0%	3%	0%	0%	0%	90%	5%	2%	0%	0%
Cross	0%	0%	0%	0%	0%	4%	96%	0%	0%	0%
Comehere	0%	0%	6%	0%	0%	0%	0%	91%	0%	3%
Turn-around	0%	0%	0%	3%	4%	0%	0%	0%	93%	0%
Pat	0%	0%	4%	0%	0%	0%	0%	1%	0%	95%

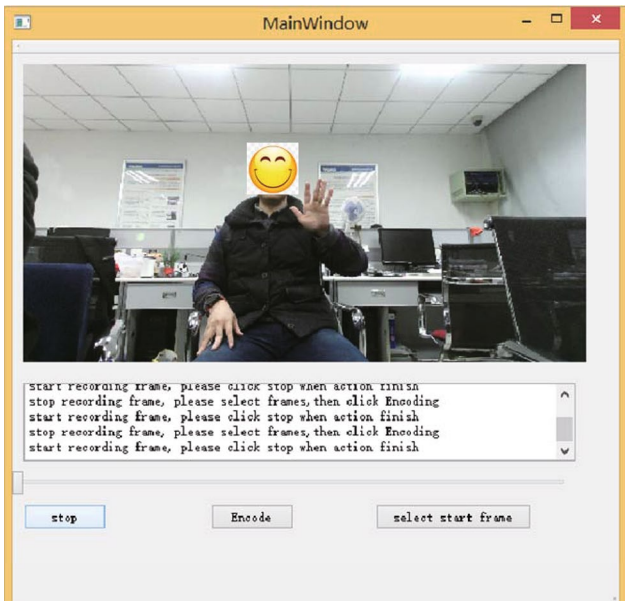

Fig. 12 Data acquisition software

to recognize 10 different digit gestures generated by motion trajectories of a single hand. The 2D-CNN based motion recognition (Karpathy et al. 2014) extends 2D-CNN from the spatial into spatial -temporal dimensions. By using SKIG dataset, the comparison result of HandSense and these HGR systems is shown in Fig. 11. From this result, we can find that the average accuracy of the HandSense is higher than the one by the HOG-HMM and 2D-CNN based HGR systems.

4.2.1 Results of HGR based on collected dataset

1. Data collection

To evaluate HandSense with real data, we collect a new hand gesture dataset using Kinect under different illumination backgrounds. As shown in Fig. 12, the data acquisition software is based on a PC and programmed by Visual C++. A set of 4200 samples of hand gestures is constructed for 28 categories.

Different from traditional hand gesture datasets which contain signal-mode hand gestures solely, the collected dataset divides the hand gestures into static, fine-grained, and coarse-grained hand gestures. The static hand gestures contain 10 different hand gestures (labeled with number from 0 to 9), as shown in Fig. 13. According to the range of movement, the dynamic hand gestures are segmented into the fine-grained and coarse-grained hand gestures. The fine-grained hand gestures are with the small amplitude of movement of finger, which include toggle, pinch, scratch, screw, hook finger, 'OK', 'good', 'victory', 'knock', and 'press', as shown in Fig. 14. The coarse-grained hand gestures involve the movement of fingers and arm with large amplitude, which contain up–down, left–right, back–forth, wave, circle, 'V', cross, and grasp, as shown in Fig. 15.

2. Result of HGR based on collected dataset

Confusion matrices in Figs. 16, 17, and 18 illustrate the average recognition rates with respect to the static, fine-grained, and coarse-grained hand gestures, 92.8%, 86.2%, and 91.3%. Obviously, the average accuracy with respect to fine-grained hand gestures is lower than the one by the other two types of hand gestures due to the fact that small range movements of other parts of a body, such as the head slightly shaking and the arm slightly moving, can be easily detected. Furthermore, different backgrounds have no significant impact on hand gesture recognition since 3D-CNN mainly focuses on the key spatial and temporal features and ignores the interference from the backgrounds.

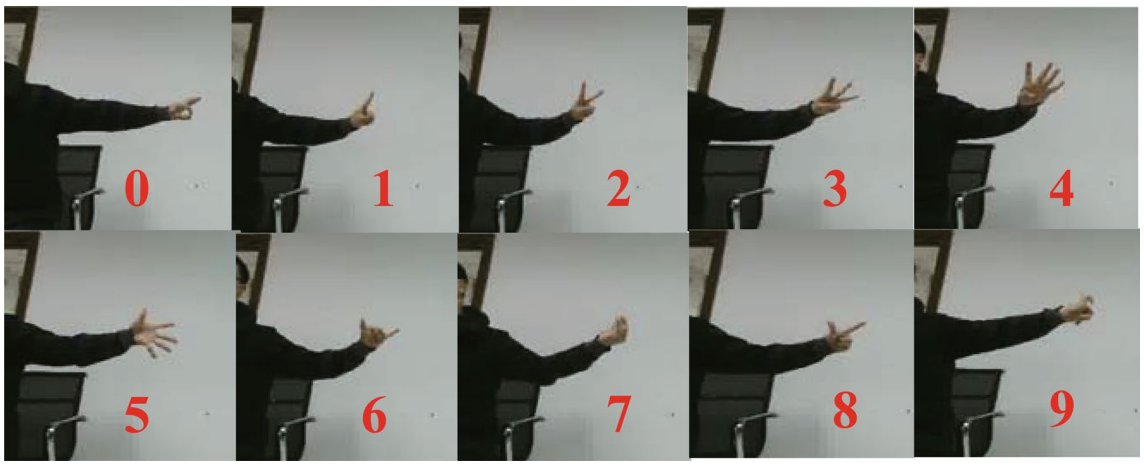

Fig. 13 Static hand gestures

Fig. 14 Fine hand gestures

3. Result of HGR based on the fused RGB and depth videos

Considering the fact that depth cameras can work well in low lighting and different illumination conditions, we combine depth and RGB videos to recognize hand gestures. Figure 19 shows an example of the depth videos for static hand gestures.

Figure 20 shows the result of HGR based on the fused RGB and depth videos. From this figure, we can find that the average recognition accuracy is increased by 3.1%, 2.9%, and 1.3% respectively by using the depth camera. This result is due to the fact that the depth video is able to provide range information for HGR, which is beneficial in low lighting and different illumination conditions.

4. Performance comparison

In this section, we evaluate proposed HandSense against the existing HOG-HMM and 2D-CNN based systems. Figure 21 shows the average accuracy of HandSense and HGR based on our collected dataset. Since HOG features are extracted

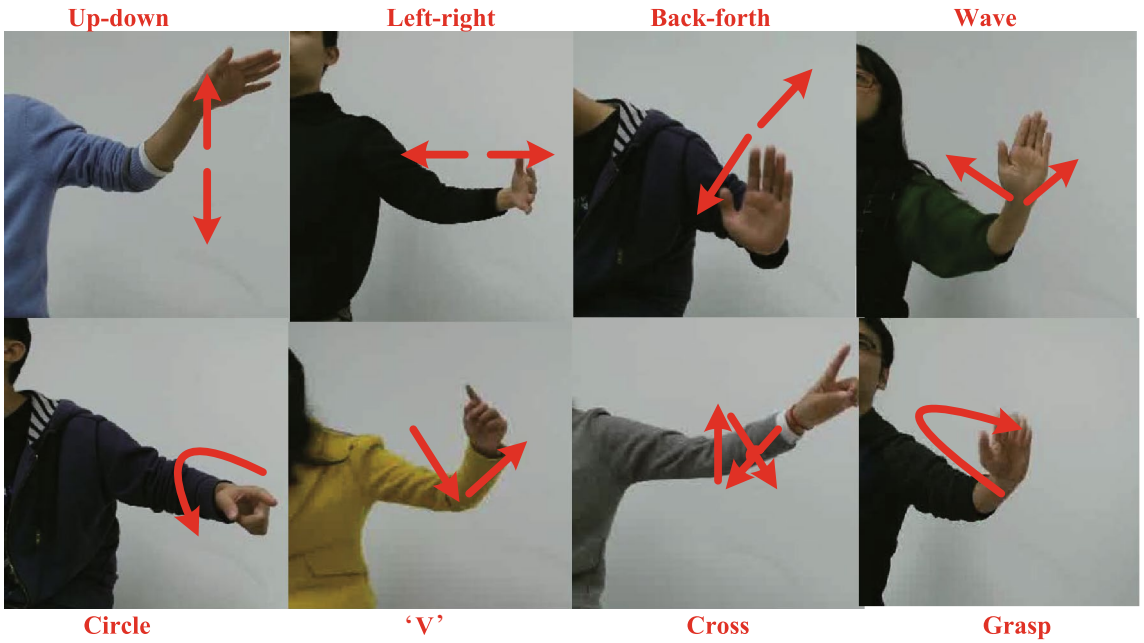

Fig. 15 Coarse hand gestures

	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'
'0'	90%	0%	0%	0%	1%	0%	1%	3%	0%	5%
'1'	2%	92%	0%	0%	0%	3%	0%	0%	0%	3%
'2'	0%	3%	91%	2%	0%	2%	0%	0%	0%	2%
'3'	0%	0%	2%	94%	0%	4%	0%	0%	0%	0%
'4'	1%	0%	0%	3%	94%	0%	0%	2%	0%	0%
'5'	0%	2%	2%	0%	3%	93%	0%	0%	0%	0%
'6'	0%	0%	0%	2%	0%	0%	98%	0%	0%	0%
'7'	5%	0%	0%	1%	0%	0%	0%	94%	0%	0%
'8'	0%	0%	2%	3%	0%	0%	3%	0%	92%	0%
'9'	0%	8%	0%	1%	0%	0%	0%	1%	0%	90%

Fig. 16 Confusion matrix with respect to static hand gestures

from RGB video, the HOG-HMM based system does not fuse RGB and depth videos. As can be seen from this figure, HandSense performs better than the HOG-HMM and 2D-CNN based systems especially for fine-grained hand gesture recognition due to the superior performance of 3D-CNN in extracting spatial and temporal features. Therefore, we can conclude that features extracted by using deep learning methods have significant advantage over handcraft features.

5 Conclusion

In this paper, we propose the HandSense for multi-modal HGR, which combines different scales of image appearance and video motion information. We build on our system for multi-modal HGR in RGBD video sequences, which are insensitive to the change of illumination environment. This approach makes better use of spatial–temporal

	Toggle	Pinch	Scratch	Screw	Hook finger	'OK'	'Good'	Victory	Knock	Press
Toggle	86%	2%	8%	3%	0%	0%	0%	0%	1%	0%
Pinch	0%	90%	4%	0%	2%	0%	0%	4%	0%	0%
Scratch	0%	4%	88%	0%	5%	0%	3%	0%	0%	0%
Screw	5%	0%	4%	80%	0%	1%	5%	0%	5%	0%
Hook finger	0%	6%	4%	0%	83%	0%	0%	0%	0%	7%
'OK'	0%	3%	2%	0%	0%	86%	5%	4%	0%	0%
'Good'	0%	5%	2%	0%	0%	0%	85%	0%	0%	8%
Victory	0%	1%	3%	0%	3%	0%	0%	91%	0%	2%
Knock	5%	0%	0%	10%	0%	0%	0%	0%	82%	3%
Press	0%	2%	2%	0%	0%	0%	5%	0%	0%	91%

Fig. 17 Confusion matrix with respect to fine hand gestures

	Up-down	Left-right	Back-forth	Wave	Circle	'V'	Cross	Grasp
Up-down	93%	0%	0%	0%	3%	0%	4%	0%
Left-right	0%	95%	0%	1%	1%	3%	0%	0%
Back-forth	0%	0%	90%	0%	5%	3%	0%	2%
Wave	1%	5%	0%	90%	0%	1%	0%	3%
Circle	0%	1%	0%	7%	92%	0%	0%	0%
'V'	0%	0%	0%	8%	4%	86%	2%	0%
Cross	1%	0%	0%	3%	2%	0%	94%	0%
Grasp	0%	0%	0%	5%	4%	0%	0%	91%

Fig. 18 Confusion matrix with respect to coarse hand gestures

Fig. 19 Confusion matrix with respect to coarse hand gestures

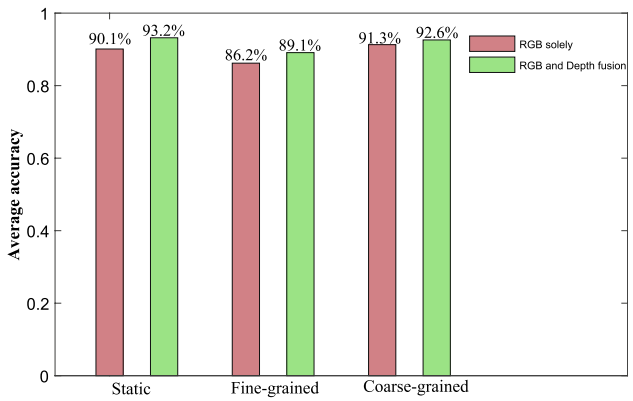

Fig. 20 Result of HGR based on the fused RGB and depth videos

information, which enables the fine grained gestures to be distinguished. To validate the effectiveness of the HandSense, we propose a newly-constructed dataset, which includes the static, fine grained, and coarse grained hand gestures. Compared with the existing HOG-HMM and 2D-CNN, the HandSense performs better in reliable HGR with respect to the challengeable hand gestures under low lighting and different illumination conditions. Considering the requirement of the real-time HGR system design, we will design the real-time HGR system by replanting the deep learning network into the FPGA platform as a part of our future work. In addition, the multi-hand gestures recognition forms an interesting topic in the HGR. Here, although the range information can be measured by the Depth sensor, the difficulty lies in how to distinguish

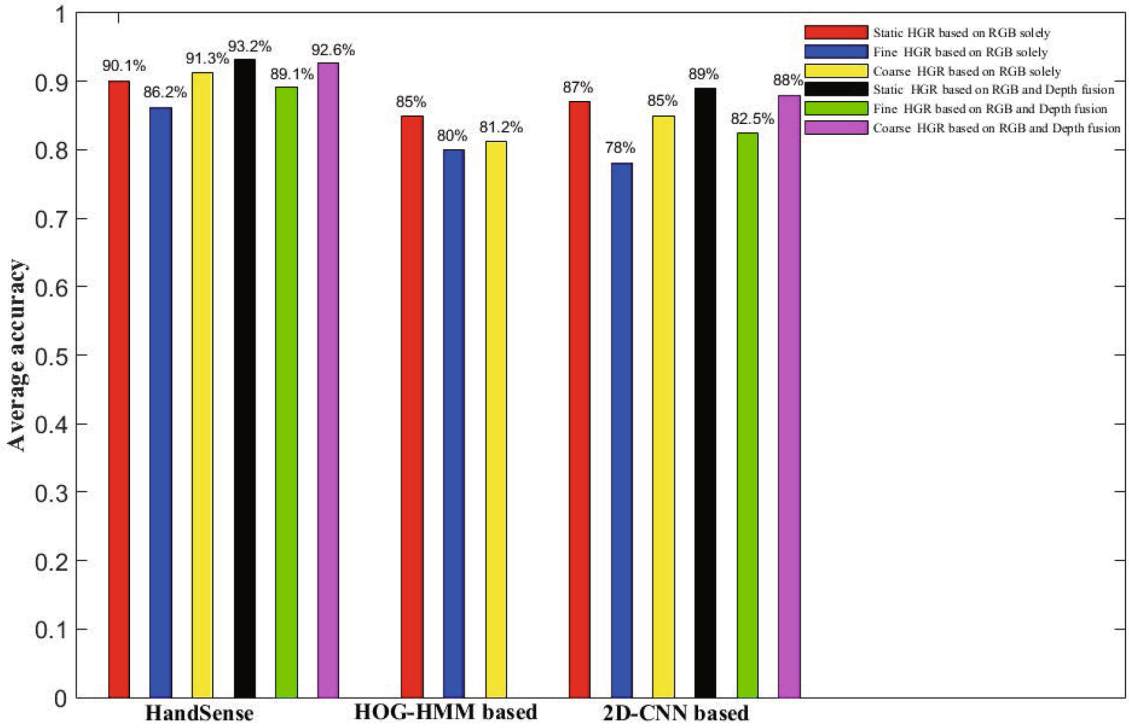

Fig. 21 Comparison result between HandSense and HGR based on collected dataset

different hand movement trajectories with the same range. Thus, we will also plan to study the multi-object tracing technology to solve the problem of multi-hand gestures recognition in future.

Acknowledgements Many thanks are given to the reviewers for the careful review and valuable suggestions. This work was supported in part by the National Natural Science Foundation of China (61301126, 61471077), Program for Changjiang Scholars and Innovative Research Team in University (IRT1299), Special Fund of Chongqing Key Laboratory (CSTC), Fundamental and Frontier Research Project of Chongqing (cstc2017jcyjAX0380, cstc2015jcyjBX0065), and University Outstanding Achievement Transformation Project of Chongqing (KJZH17117).

References

Baccouche M, Mamalet F, Wolf C, Garcia C, Baskurt A (2011) Sequential deep learning for human action recognition. In: International workshop on human behavior understanding. Springer, pp 29–39

Chang CC, Lin CJ (2011) Libsvm: a library for support vector machines. ACM Trans Intel Syst Technol (TIST) 2(3):27

Chung S, Park C, Suh S, Kang K, Choo J, Kwon BC (2016) Re-vacnn: Steering convolutional neural network via real-time visual analytics. In: Future of interactive learning machines workshop at the 30th annual conference on neural information processing systems (NIPS)

Ge L, Liang H, Yuan J, Thalmann D (2016) Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3593–3601

Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012) Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:12070580

Hu M, Shen F, Zhao J (2014) Hidden markov models based dynamic hand gesture recognition with incremental learning method. In: 2014 international joint conference on neural networks (IJCNN), IEEE, pp 3108–3115

Jahn G, Krems JF, Gelau C (2009) Skill acquisition while operating in-vehicle information systems: interface design determines the level of safety-relevant distractions. Hum Factors 51(2):136–151

Ji S, Xu W, Yang M, Yu K (2013) 3d convolutional neural networks for human action recognition. IEEE Trans Pattern Anal Mach Intell 35(1):221–231

Joachims T (2002) Optimizing search engines using clickthrough data. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp 133–142

Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Fei-Fei L (2014) Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1725–1732

Klaser A, Marszałek M, Schmid C (2008) A spatio-temporal descriptor based on 3d-gradients. In: BMVC 2008-19th British machine vision conference, British machine vision association, pp 1–10

Kojima S, Ohyama W, Wakabayashi T (2017) Gesture recognition based on spatiotemporal histogram of oriented gradient variation. In: Informatics, electronics and vision and 2017 7th international symposium in computational medical and health technology (ICIEV-ISCMHT), IEEE, pp 1–4

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems 25, Curran Associates, Inc, pp 1097–1105

Li Y (2012) Hand gesture recognition using kinect. In: 2012 IEEE 3rd international conference on software engineering and service science (ICSESS), IEEE, pp 196–199

Liu K, Kehtarnavaz N (2016) Real-time robust vision-based hand gesture recognition using stereo images. J Real-Time Image Proc 11(1):201–209

Liu L, Shao L (2013) Learning discriminative representations from rgb-d video data. In: IJCAI, vol 1, p 3

Liu WM, Wang LH (2011) The soccer robot the auto-adapted threshold value method based on hsi and rgb. In: 2011 International Conference on Intelligent computation technology and automation (ICICTA), IEEE, vol 1, pp 283–286

Ma M, Marturi N, Li Y, Leonardis A, Stolkin R (2018) Region-sequence based six-stream cnn features for general and fine-grained human action recognition in videos. Pattern Recogn 76:506–521

Molchanov P, Gupta S, Kim K, Kautz J (2015) Hand gesture recognition with 3d convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 1–7

Pal DH, Kakade S (2016) Dynamic hand gesture recognition using kinect sensor. In: 2016 international conference on global trends in signal processing, information computing and communication (ICGTSPICC), IEEE, pp 448–453

Parada-Loira F, González-Agulla E, Alba-Castro JL (2014) Hand gestures to control infotainment equipment in cars. In: IEEE Intelligent Vehicles Symposium Proceedings. IEEE, pp 1–6

Platt JC (1999) 12 fast training of support vector machines using sequential minimal optimization. In: Advances in kernel methods, pp 185–208

Prakash RM, Deepa T, Gunasundari T, Kasthuri N (2017) Gesture recognition and finger tip detection for human computer interaction. In: 2017 international conference on innovations in information, embedded and communication systems (ICIIECS), IEEE, pp 1–4

Priyal SP, Bora PK (2013) A robust static hand gesture recognition system using geometry based normalizations and krawtchouk moments. Pattern Recogn 46(8):2202–2219

Rao GA, Syamala K, Kishore P, Sastry A (2018) Deep convolutional neural networks for sign language recognition. In: 2018 conference on signal processing and communication engineering systems (SPACES), IEEE, pp 194–197

Rohrbach M, Rohrbach A, Regneri M, Amin S, Andriluka M, Pinkal M, Schiele B (2016) Recognizing fine-grained and composite activities using hand-centric features and script data. Int J Comput Vision 119(3):346–373

Sharp T, Keskin C, Robertson D, Taylor J, Shotton J, Kim D, Rhemann C, Leichter I, Vinnikov A, Wei Y, et al. (2015) Accurate, robust, and flexible real-time hand tracking. In: Proceedings of the 33rd annual ACM conference on human factors in computing systems, ACM, pp 3633–3642

Simonyan K, Zisserman A (2014a) Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems, pp 568–576

Simonyan K, Zisserman A (2014b) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556

Simonyan K, Vedaldi A, Zisserman A (2013) Deep inside convolutional networks: Visualising image classification models and saliency maps. 2013. arXiv preprint arXiv:13126034

Singh G, Nelson A, Robucci R, Patel C, Banerjee N (2015) Inviz: Low-power personalized gesture recognition using wearable textile capacitive sensor arrays. In: 2015 IEEE international conference on pervasive computing and communications (PerCom), IEEE, pp 198–206

Sutskever I, Martens J, Dahl G, Hinton G (2013) On the importance of initialization and momentum in deep learning. In: International conference on machine learning, pp 1139–1147

Taylor GW, Fergus R, LeCun Y, Bregler C (2010) Convolutional learning of spatio-temporal features. In: European conference on computer vision. Springer, Berlin, pp 140–153

Tsai CY, Lee YH (2011) The parameters effect on performance in ann for hand gesture recognition system. Expert Syst Appl 38(7):7980–7983

Vieriu RL, Goraş B, Goraş L (2011) On hmm static hand gesture recognition. In: 2011 10th international symposium on signals, circuits and systems (ISSCS), IEEE, pp 1–4

Wang X, Xia M, Cai H, Gao Y, Cattani C (2012) Hidden–Markov-models-based dynamic hand gesture recognition. Math Probl Eng 2012:986134

Wen H, Ramos Rojas J, Dey AK (2016) Serendipity: Finger gesture recognition using an off-the-shelf smartwatch. In: Proceedings of the 2016 CHI conference on human factors in computing systems, ACM, pp 3847–3851

Xue Y, Ju Z, Xiang K, Chen J, Liu H (2018) Multimodal human hand motion sensing and analysis-a review. In: IEEE Transactions on Cognitive and Developmental Systems. IEEE, pp 1–14

Yamada K, Yoshida T, Sumi K, Habe H, Mitsugami I (2017) Spatial and temporal segmented dense trajectories for gesture recognition. In: Thirteenth international conference on quality control by artificial vision 2017, International society for optics and photonics, vol 10338, p 103380F

Zhao Y, Luo Z, Quan C (2017) Unsupervised online learning for fine-grained hand segmentation in egocentric video. In: 2017 14th conference on computer and robot vision (CRV), IEEE, pp 248–255