Proceedings of the 39th Chinese Control Conference
July 27-29, 2020, Shenyang, China

Real-Time Sign Language Recognition Based on Video Stream

Kai Zhao[1], Kejun Zhang[2], Yu Zhai[2], Daotong Wang[2], Jianbo Su[1,2]

1. Department of Automation, Shanghai Jiao Tong University, Shanghai 200240
E-mail: jbsu@sjtu.edu.cn

2. Shanghai Lingzhi High-Tech Corporation, Shanghai, 200240

Abstract: There are millions of deaf-dumb people in the world communicating by sign language, thus designing a sign language recognition system is very meaningful and valuable for normal people to understand them. In this paper, we investigate a real-time Chinese sign language recognition system. A Chinese sign language dataset is firstly created. Considering practical applications, RGB camera is used to collect video stream, instead of RGB-D camera. In order to improve the accuracy of recognition, we propose a 3D-CNN method combined with optical flow processing. The collected RGB video stream is processed by optimized dense optical flow, and then put into 3D-CNN to extract feature vectors. For practical considerations, a real-time sign language recognition system is designed, composed of artificial interaction interface, motion detection module, hand and head detection module, etc. Experimental results show the superb performance and the applicability of the proposed systems.

Key Words: Sign language recognition, RGB, optical flow, 3D-CNN, motion detection, hand and head detection

1. Introduction

Sign language recognition has gained a lot of attention in recent years for automatic explanations by computer or a robot. There are many application scenarios of this research, which can help deaf-dumb people communicate with others in public areas such as hospitals, banks, and train stations. The development of this research will greatly reduce the inconvenience of deaf-dumb people's lives.

However, one of the problems facing sign language recognition is the lack of available sign language datasets. It has greatly prohibited the development of the practical systems. Sign language, like ordinary language, has regional differences. Different sign languages are used between different countries and even different regions of the same country. Therefore, it is hard to create a public dataset for research. Currently, the available public datasets are Chalearn14 [1] and RWTH-PHOENIX-Weather [2]. Chalearn14 is a dataset provided by "Chalearn Looking at People Challenge 2014", which contains 20 Italian sign language vocabularies and a total of 7,754 gesture examples. RWTH-PHOENIX-Weather is a German sign language dataset that contains 7,000 weather forecast sentences from 9 sign language speakers. Furthermore, there is so far no open and complete Chinese sign language dataset. For all Chinese researchers, creating a Chinese sign language dataset is an important task for automatic recognition.

Another problem facing sign language recognition is the extraction of sign language features. The input of sign language recognition is an RGB video stream, so we have to extract not only the spatial features but also the temporal features in the video stream. Traditional sign language recognition relies on artificially designed features, but when it comes to large sign language datasets, there will be great challenges. With the application of convolutional neural networks (CNNs) in computer vision, sign language recognition has also begun to extract features through 3D convolutional neural networks (3D-CNNs). Compared to 2D-CNN, 3D-CNN adds a convolution operation in the time dimension. As a feature extractor, deep 3D-CNN can extract both space and time features. Molchanov et al. [3] propose a recurrent 3D-CNN for dynamic gesture recognition and achieve an accuracy of 83.8%. To better extract features, the authors use depth and grayscale data. However, there is currently no effective feature extraction method for RGB video streams only.

In recent years, many researchers also tried to extract other features of sign language through some special equipment. Almeida et al. [4] use RGB-D sensors to obtain RGB-D video streams, which can provide RGB video stream and depth map. Each pixel value in the depth map is the actual distance of the sensor from the object. This characteristic of RGB-D video stream can be used to achieve background segmentation to better extract spatial features. Lokhande et al. [5] implanted an accelerator and a flexible sensor inside the glove to obtain data on the degree of bending of the fingers. However, combined with practical applications, whether it is the acquisition of depth images or the wearing of equipment, it will cause inconvenience to the daily use of deaf-dumb people.

At present, the focus of sign language recognition research is on the issue of recognition accuracy. In recent years, many researchers have implemented sign language recognition based on traditional methods. Among them, Hidden Markov Model (HMM) has been widely used. Gao et al. [6] use a self-organizing feature map and HMM method to obtain a recognition accuracy of 82.9%. Huang et al. [7] obtained video streams from Microsoft Kinect. Multi-channels of video streams, including RGB information, depth clue, and human skeleton positions, are used as input to the 3D-CNN in order to integrate three modes of information. Finally, on a dataset has 25 vocabularies, 88.5% recognition accuracy was obtained through the gray channel, and 94.2% recognition accuracy was obtained through the multi-channel.

In order to improve the accuracy of sign language recognition without using complex feature extraction methods. We propose a sign language recognition based on RGB

*This work was supported by the National Natural Science Foundation (NNSF) of China under Grants 61533012 , 91748120 and 52041502.

video stream, which uses 3D-CNN combined with optical flow processing. In view of the fact that it is difficult to obtain high recognition accuracy by only using RGB video streams as input for 3D-CNN feature extraction, we introduced the optical flow method. After the video stream is preprocessed by optical flow calculation, the separation of people and background can be achieved. Because real-time related issues are not considered in current sign language research, we created the first real-time sign language recognition system. In order to meet the real-time requirements of real-time systems, it is necessary to reduce the time required for video stream preprocessing and feature extraction. Therefore, the frame difference method is used to reduce redundant frames in the RGB video stream, thereby reducing the number of frames in the video stream. Motion detection, hand detection and head detection are important steps in sign language recognition systems, which can enable the system to capture RGB video in real time. We comprehensively evaluated the performance of YOLO-V3 and Faster R-CNN, and finally chose YOLO-V3 to achieve hand and head detection.

The reminder of this paper is organized as follows. Section 2 introduces the creation process of the Chinese Sign Language dataset. In the following Sections 3, the pre-processing process of video stream before feature extraction is described. Section 4 explains the working principle of the real-time sign language recognition system. In the experimental part of Section 5, the performance of the model on the test set and the effect of the real-time recognition system are tested, followed by conclusions in Section 6.

2. Dataset Creation

We collected the Chinese standard sign language vocabulary, which contains more than 5000 words and their demos. Ten deaf and mute people were invited to demonstrate sign language, and each of them had certain differences in amplitude and habits of their movements. Finally, a data set called CSSL5000 was created, which has a large vocabulary and diversity.

2.1. Data Collection

The video capture device is a 720P RGB camera. The recorded video format is MP4, the resolution is set to 640 × 480, and the frame rate is 30fps. 10 deaf and mute people were invited to demonstrate each word 15 times. Each word can be demonstrated in about 5 seconds, so the length of each video was set to 5 seconds. We require deaf and mute people to have no unnecessary movements during the entire video collection process. The recording background should be pure white while the illumination condition guaranteed. The angle of view of the camera needs to be kept horizontal, and only the upper body (above the hand) of the actor is recorded. The actor must stand in the center of the screen, and the position of the camera needs to be adjusted in real time according to the height of the sampled person. Before sampling, the sampled person needs to learn the Chinese standard sign language vocabulary uniformly to ensure the standardization. Fig. 1 shows RGB video streams of sign language actions demonstrated by different people in the dataset.

Fig. 1: Video frames in CSSL5000 dataset

2.2. Dataset Partitioning and Annotation

For each sampled person, 10 high quality videos were selected from 15 videos of each word displayed. Six people's videos were used as training set, two people's videos as the verification set, and the rest as the test set. The video files of the same word in each dataset are put into the same folder, and numbering the folders in order. Create folders in different datasets, name them one after the other, and place related videos of the same word in the same folder. Write an annotation document, annotating the meaning of each word in the document. Finally, a Chinese sign language dataset containing more than 500,000 sign language examples was obtained.

3. Data Preprocessing

Before feature extraction, all video streams need to be pre-processed. Each video is 5 seconds long, and at a frame rate of 30fps, 150 images are usually generated. If all frames in the video stream are put into 3D-CNN to extract features, it will be a large amount of data. Also, every frame in a video is not necessarily useful. Therefore, the pre-processed video stream can filter out useful frames and reduce the number of frames to 40. Then, these 40 frames of images are processed into the form required by the network.

3.1. Frame Difference Processing

The video sequence collected by the camera is continuous. If there is no moving target in the scene, the change of continuous frames is slight. If there are moving targets, there will be significant changes between continuous frames. Because the target in the scene is in motion, the position of the target in different image frames is different. This algorithm performs a differential operation on two or three consecutive images in time, and subtracts the pixels corresponding to different frames to determine the absolute value of the gray difference. When the absolute value exceeds a certain threshold, it can be judged as a moving target. The operation of the frame difference method is shown in Fig. 2.

This method can be used to solve two problems we encountered. Firstly, our captured video stream has many motionless pictures or pictures with small motion amplitude. In order to improve the efficiency of feature extraction and reduce the time required for data preprocessing, this principle can be used to remove unnecessary redundant frames in each video streams. Thereby the purpose of reducing the number of frames in the video stream is achieved. Secondly, the method can also be used for motion detection. The change between consecutive frames can be used to determine whether the person in the picture starts to move, and the system sends a signal to start collecting information.

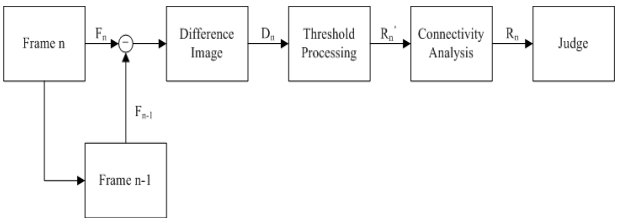


Fig. 2: The schematic diagram of frame difference method

Record the nth and n−1 frames of the video sequence as f_n and f_{n-1}, and the gray values of the corresponding pixels of the two frames as $f_n(x, y)$ and $f_{n-1}(x, y)$. Subtract the gray values of the corresponding pixels of the two frames of images according to formula (1), and take the absolute value to obtain the differential image D_n.

$$D_n(x,y) = \left| f_n(x,y) - f_{n-1}(x,y) \right| \qquad (1)$$

The threshold is set to T, and binarization of pixels one by one according to formula (2) to obtain a binarization image R_n'. Among them, a point with a gray value of 255 is a foreground (moving target) point, and a point with a gray value of 0 is a background point. The connectivity analysis of the image R_n' can finally obtain the image R_n containing the complete moving target.

$$R_n'(x,y) = \begin{cases} 255, D_n(x,y) > T \\ 0, else \end{cases} \qquad (2)$$

Threshold T needs to be constantly adjusted. There will be some static frames at the beginning and end of the video. By frame preprocessing, video frames that do not meet the conditions will be deleted, leaving only key action fragments, and controlling the number of frames of the video to 40 ~ 100 frames.

3.2. Image Processing

After each video is processed by frame difference, the number of pictures obtained is different, and the number of pictures needs to be processed into 40 pictures by an equal division method. The 40 images obtained for each video need to be converted to 320 × 240, and stored as jpg format files.

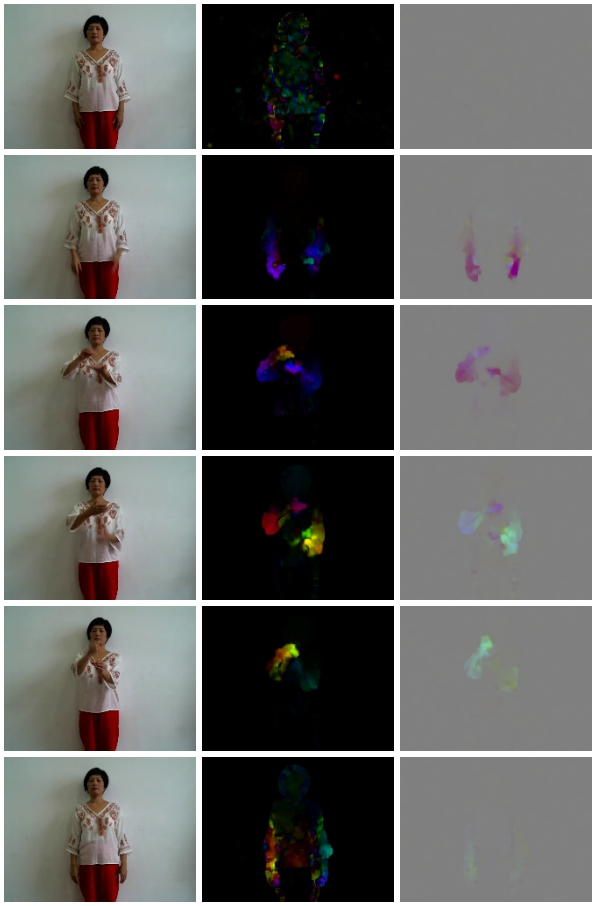
3.3. Optical Flow Calculation

Optical flow calculation is an important step in the video stream preprocessing stage. Optical flow is the instantaneous speed of pixel movement of a space moving object on the observation imaging plane. According to the degree of sparseness of the two-dimensional vectors in the formed optical flow field, the optical flow method is divided into two types: dense optical flow and sparse optical flow.

Dense optical flow is an image registration method that performs point-by-point matching for the entire image or a specified area in the image. It calculates the offset of all points on the image to form a dense optical flow field. Therefore, the calculation of dense optical flow requires a large amount of calculation.

In contrast to dense optical flow, sparse optical flow does not calculate point by point for each pixel of the image. It usually needs to specify a set of points for tracking. This set of points preferably has some obvious characteristics, such as Harris corner points. It can be seen that the calculation amount of sparse optical flow is much smaller than that of dense optical flow, but when the object moves too fast, the algorithm will produce a large error.

Therefore, dense optical flow is more suitable for sign language recognition. The image processed by sparse optical flow is not enough to represent the motion features of sign language. However, the calculation of dense optical flow will take a lot of time to pre-process the video stream, which will not meet the real-time requirements of the system. In Fig. 3, (a) represents the frames of the original video stream, (b) represents the image after the dense optical flow calculation, and (c) represents the image after the TV-L1 optical flow [8] calculation. It can be seen in (b) that after the calculation of dense optical flow, a lot of hot pixel will appear in the picture. In order to remove noise in the image, the dense optical flow algorithm needs to be optimized. After the dense optical flow algorithm is optimized by total variation (TV) regularization, it will remove unnecessary details in the image and retain important details. Under the condition that the accuracy of recognition is not affected, the optical flow calculation time can be adjusted to meet the system real-time requirements by continuously adjusting the optical flow parameters.



(a) (b) (c)

Fig. 3: Two optical flow calculation methods

4. Real-Time Recognition System

A real-time sign language recognition system is designed to automatically capture video streams and finally display recognition results in real time. Therefore, firstly, a model for prediction of results needs to be created through 3D-CNN, and then a human-computer interaction interface for

automatically collecting video streams needs to be designed.

4.1. Model Construction

Sign language recognition similar to action recognition is usually divided into three stages: locate the feature area of interest, extract and describe the features of the area of interest, and finally train a classifier based on the extracted features. Since 3D-CNN learns the features of the time dimension, the input of the model must be a segment of the video stream. Without prior knowledge, 3D-CNN can extract temporal features from the original RGB video stream. In order to improve the efficiency of sign language recognition, 40 frames of images obtained through video stream preprocessing are input to the network. Due to deeper neural networks have stronger learning capabilities, a deep 3D-CNN architecture called I3D [9] is adopted. I3D is inflated from the famous 2D-Inception CNN. After 3D-CNN extracts features, each video instance is represented by a feature vector. Based on these features extracted from the examples, the SVM is constructed for classification tasks.

The trained model is added to the system to predict the results. It takes a lot of time each time the model is loaded during prediction. Therefore, the system is designed with a model preloading method. When the system starts, the model is loaded first, and then it enters the video capture waiting state. After capturing the video and predicting the result, continue to enter the loop waiting state.

4.2. Human-Computer Interaction

Human-Computer interaction interface is one of the important parts of the real-time identification system. Firstly, video capture device needs to capture RGB video stream for recognition. Secondly, it is determined whether the user starts to input a sign language action according to the acquired video stream. Thirdly, the system also needs to determine whether the user has completed all sign language actions. Finally, the captured video stream can be pre-processed, feature extracted, predicted, and the recognition results displayed in real time.

Motion detection, hand and head detection modules are added to the system. The combination of these two modules can control the start and end of video stream acquisition. The motion detection module is mainly used to determine whether there is a moving target in the video picture, and it is implemented by the frame difference method. Frame difference method is widely used for motion detection, simple and efficient, and can meet the requirements of system real-time. If the threshold T is set too low, the machine will judge the small-scale motion of the human as the signal to start collecting. However, we only focus on the movements when the hand is about to be raised. Therefore, the threshold T needs to be set within a reasonable range in order to control the sensitivity of the machine to human motion.

The hand and head detection are used to detect whether there is a person in the video picture, and then further determine whether to start video collection. YOLO is a relatively commonly used method for item detection with high accuracy and detection speed. Therefore, YOLO-V3 was used for hand and head detection. First, we trained the YOLO model on the PASCAL VOC2012 human layout dataset, where three output units represent the hand, the head and the background. Then, we randomly selected 500 frames of images from the sign language data set and manually labeled the human hand and head. Finally, the labeled images are used for fine-tuning network training to obtain the fastest YOLO model.

5. Experiments

In this section, the experiment is divided into two parts: one is to test the accuracy of the model on the test set; the other is to test the effectiveness of the real-time sign language recognition system.

5.1. Results and Analysis on The Test Set

In order to improve the efficiency of the experiment, 1000 words in the CSSL5000 dataset were taken out for training and testing. The statistical information of the dataset used is shown in Table 1. In order to compare experimental results longitudinally, non-deep learning methods and deep learning methods are compared. In addition, whether the RGB video stream is preprocessed with optical flow is also used as a basis for comparison. The experimental results are shown in Table 2.

Table 1: Statistics for The Dataset

Category	1,000
Num. of singer	10
Total instance	100,000
Modality	RGB
Resolution	640×480
Video duration	2~5 seconds
Video format	MP4
FPS	30

Table 2: Experimental Results

Method	Ave Accuracy
HMM+RGB	61.3%
3D-CNN+RGB	67.8%
3D-CNN+RGB+Optcal flow	90.1%

It can be seen from the experimental results that HMM and 3D-CNN do not perform particularly well on RGB-based video streams. However, 3D-CNN has greatly improved the recognition effect on RGB video streams based on optical stream preprocessing. In the results of statistical experiments, sign language vocabularies that did not recognize well were mainly found in some words with smaller range of motion.

5.2. Testing of Real-Time Recognition System

To test the real-time identification system, first, the video capture program must be verified to work properly. The determination part of video acquisition is mainly composed

of a motion detection module and a hand and head detection module. The system uses a dual mechanism to determine whether to start capturing video streams. Figure 4 shows the results of motion detection on hand motion states. The left and right hands demonstrate the process from motionless to motion, and finally to motionless. The frame difference method is still used as an algorithm for motion detection. Constantly adjust the threshold T, and finally determine that a reasonable range between 40 and 80.

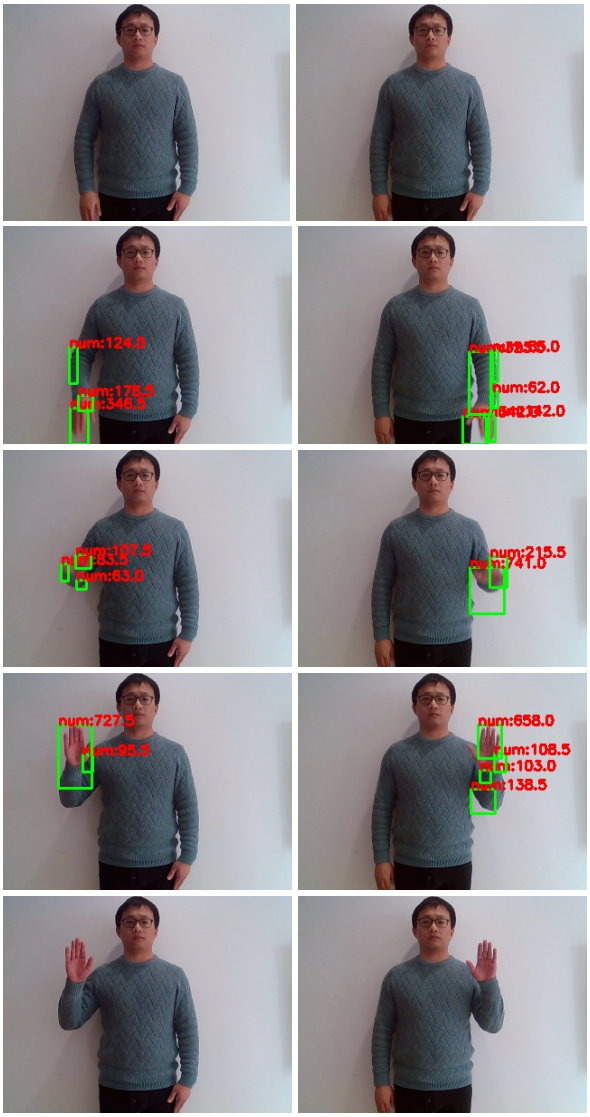


Fig. 4: Motion detection results for left and right hands

In order to obtain a better hand and head detection model, Faster R-CNN and YOLO-V3 are used for comparison. The dataset used is the PASCAL VOC2012 human layout dataset, which the training set contains 425 annotated pictures of various parts of the human body (head, hands, feet). First train R-CNN and YOLO with faster recognition speed through this dataset. Then randomly select 100 videos from the CSSL5000 data set, and then select 500 frames of pictures from the video. According to the data structure of VOC2010, each picture is annotated for fine-tuning the network. The experimental results are shown in Table 3. It can be seen that the IoU of Faster R-CNN is significantly higher than that of YOLO-V3. This shows that Faster R-CNN is higher than YOLO-V3 in detection accuracy. However, in terms of average recognition speed, YOLO-V3 is significantly faster than Faster R-CNN. Therefore, it can better meet the real-time requirements. Figure 5 shows the hand and head detection results achieved by the YOLO-V3 model.

Table 3: Experimental Results of Hand and Head Detection

Model Type	Ave IoU (Head)	Ave IoU (Hand)	Ave Time
Faster R-CNN	0.975	0.924	2.128
YOLO-V3	0.958	0.896	0.0189

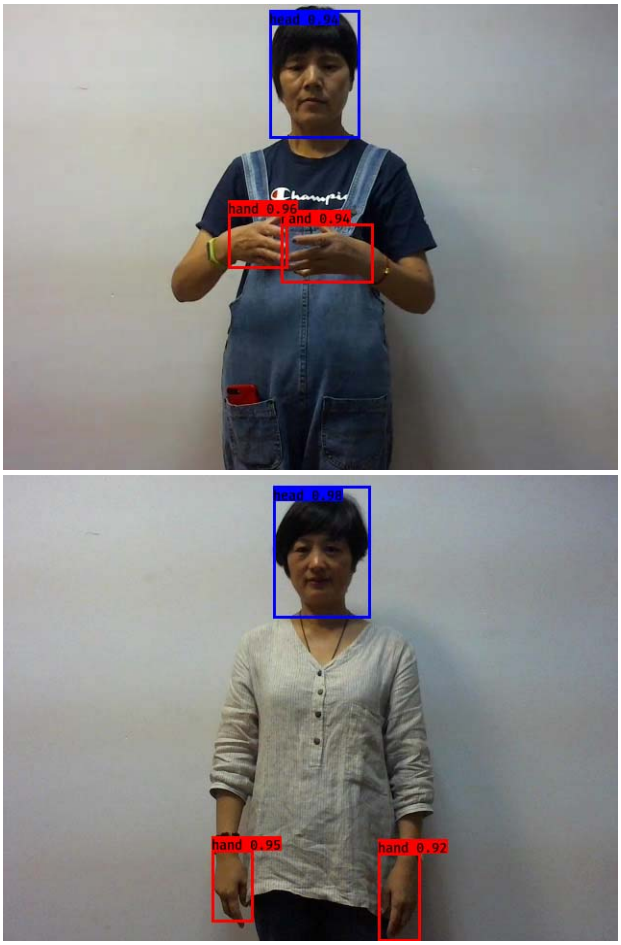


Fig. 5: YOLO-V3 hand and head detection results

After the above judgment, if a human body appears in the acquisition window and a moving object appears, the video stream will be captured. The obtained RGB video stream needs to be preprocessed before it can be put into 3D-CNN to extract feature vectors. After testing, the processing time of dense optical flow calculation for each frame is about 0.12 seconds. The optimized TV-L1 optical flow calculation only takes about 0.022 seconds for the processing time of each frame of image. Therefore, the preprocessing time for 40 frames is about 0.9 seconds. With the TensorFlow framework, it takes a long time to load the model for the first time, so the pre-loading method is used, which saves a lot of time. Finally, the recognition result is predicted by the pre-loaded model. According to the confi-

dence level of the recognition result, top1 is selected to be displayed as the final result. From obtaining the video stream to displaying the recognition result on the screen, the total time spent in the whole process is about 1 second, which basically meets the requirements of real-time performance. The recognition results of some words are shown in Fig. 6.

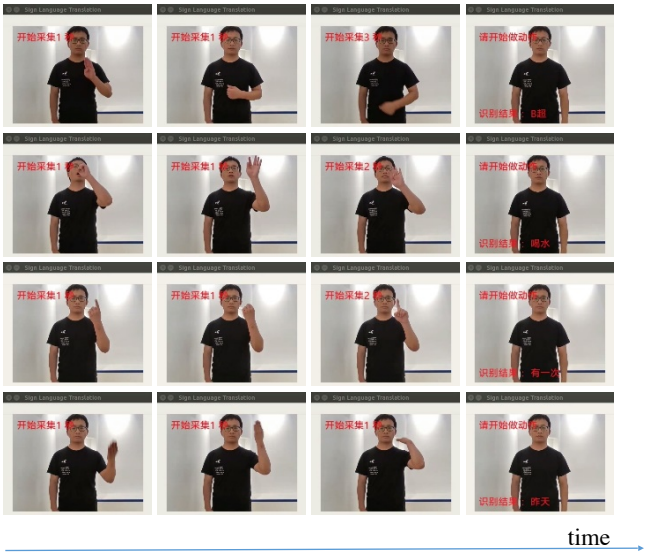


Fig. 6: Recognition results of real-time recognition system

6. Conclusion

In this paper, a real-time sign language system based on RGB video streams and 3D-CNN is proposed, and a Chinese sign language dataset is created, which provides the basis for a set of sign language recognition. We have shown that sign language recognition can be achieved by combining RGB video streams with TV-L1 optical flow calculations and extracting features through 3D-CNN. We have also shown the creation of a real-time sign language system. Motion detection and human detection are control centers of real-time systems. Frame difference method is a simple and effective motion detection algorithm, which has little effect on the real-time performance of the system. YOLO and Faster R-CNN were proposed for hand and head detection at the same time. We evaluated both networks and finally chose YOLO with better comprehensive performance. According to the evaluation of the experimental results, it is difficult for optical flow to capture some subtle movements of the hand. This makes it difficult for 3D-CNN to extract complete motion information.

In the future, we can add skin detection and human skeleton detection to improve the accuracy of motion detection. More effective feature detection and tracking from hand movement is also an interesting topic for further explorations.

References

1. Sergio Escalera, Xavier Baró, Jordi Gonzàlez, Miguel A. Bautista, & Isabelle Guyon. ChaLearn looking at people challenge 2014: Dataset and results. Workshop at the European Conference on Computer Vision. Springer International Publishing, 2014: 459–473.
2. Koller, Oscar, Forster, Jens, Ney, Hermann. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers[J]. Computer Vision & Image Understanding, 2015, 141:108-125.
3. Pavlo Molchanov, Xiaodong Yang, Shalini Gupta et al. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural networks[C]// IEEE Conference on Computer Vision and Pattern Recognition, 2016.
4. SGM Almeida, FG Guimarães, JA Ramírez. Feature extraction in Brazilian Sign Language Recognition based on phonological structure and using RGB-D sensors[J]. Expert Systems with Applications, 2014, 41(16):7259-7271.
5. Priyanka Lokhande, Riya Prajapati, Sandeep Pansare. Data gloves for sign language recognition system[J]. International Journal of Computer Applications, 2015, 11-14
6. Wen Gao, Gaolin Fang, Debin Zhao, Yiqiang Chen. A Chinese sign language recognition system based on SOFM/SRN/HMM[J]. Pattern Recognition, 2004, 37(12):2389-2402.
7. Jie Huang, Wengang Zhou, Houqiang Li, Weiping Li. Sign Language Recognition using 3D convolutional neural networks[C]// IEEE International Conference on Multimedia and Expo (ICME), 2015.
8. Christopher Zach, Thomas Pock, Horst Bischof. A Duality Based Approach for Realtime TV-L1 Optical Flow[C]// Proceedings of the 29th DAGM conference on Pattern recognition. 2007.
9. Carreira, Joao, Zisserman, Andrew. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset[J]. IEEE Conference on Computer Vision and Pattern Recognition, 2017.