The Journal of Engineering

The 2nd 2018 Asian Conference on Artificial Intelligence Technology (ACAIT 2018)

IET Journals
The Institution of Engineering and Technology

RGB-D static gesture recognition based on convolutional neural network

eISSN 2051-3305
Received on 19th July 2018
Revised 04th August 2018
Accepted on 06th August 2018
doi: 10.1049/joe.2018.8327
www.ietdl.org

Bin Xie[1,2], Xiaoyu He[1,2], Yi Li[1,2] ✉

[1]*School of Information Science and Engineering, Central South University, Changsha, People's Republic of China*
[2]*Mobile Health Ministry of Education, China Mobile Joint Laboratory, Xiangya Hospital Central South University, Changsha, People's Republic of China*
✉ *E-mail: liyi1002@csu.edu.cn*

Abstract: In the area of human–computer interaction (HCI) and computer vision, gesture recognition has always been a research hotspot. With the appearance of depth camera, gesture recognition using RGB-D camera has gradually become mainstream in this field. However, how to effectively use depth information to construct a robust gesture recognition system is still a problem. In this paper, an RGB-D static gesture recognition method based on fine-tuning Inception V3 is proposed, which can eliminate the steps of gesture segmentation and feature extraction in traditional algorithms. Compared with general CNN algorithms, the authors adopt a two-stage training strategy to fine-tune the model. This method sets a feature concatenate layer of RGB and depth images in the CNN structure, using depth information to promote the performance of gesture recognition. Finally, on the American Sign Language (ASL) Recognition dataset, the authors compared their method with other traditional machine learning methods, CNN algorithms, and the RGB input only method. Among three groups of comparative experiments, the authors' method reached the highest accuracy of 91.35%, reaching the state-of-the-art currently on ASL dataset.

1 Introduction

With the rapid development of computer technology, communication technology, and hardware equipment, HCI has become more and more frequent in our lives. People's requirements for user experience is becoming higher and higher. In recent years, the technology of gesture recognition has received more and more attention as a friendly interactive method. In some gesture interaction scenes, people can directly interact through the palm and finger changes in the direct operation, without the need to use the mouse and keyboard. This kind of HCI is more natural and in line with the current trend of HCI development [1]. For example, in emerging areas such as virtual reality, wearable devices, smart operating rooms, drone control, and distance learning, gesture interaction has its unique advantages [2].

Gesture recognition can be roughly classified into two categories: dynamic gesture recognition and static gesture recognition. In this paper, we focus on static gesture recognition. At present, a typical disadvantage of traditional static gesture recognition methods is that they are not robust to the recognition performance under complex background and lighting conditions. This is because severe lighting changes and complex backgrounds seriously affect the result of gesture segmentation. Gesture segmentation is a vital step in gesture recognition. Inaccurate gesture segmentation naturally affects the accuracy of gesture classification.

Recently, rapid-developing depth cameras have gradually been applied in the field of gesture recognition [3]. Although depth cameras have been used in the field of computer vision for years, there are still several limits due to their high price and poor quality. In 2010, Microsoft launched the low-cost RGB-D camera Kinect, which can provide high-quality range images, making the depth camera being widely used in the field of gesture recognition [4]. In this paper, the ASL dataset that we used was collected by Kinect as well.

In the last few years, deep learning has been developed rapidly. Particularly, CNN has achieved effective results in image classification, natural language understanding and other fields. Currently, CNN models such as GoogLeNet [5], VGG16, VGG19 [6], and Inception V3 [7] have achieved excellent results in ImageNet Large-scale Visual Recognition Challenge (ILSVRC).

Transfer learning is a popular method in the current field of deep learning. In simple terms, transfer learning is the transfer of trained model parameters to another model to help another model training. The advantage is that transfer learning can avoid model's overfitting and greatly reduce the training time, making the model more easily to be converged. A large number of research results have also confirmed the good performance of transfer learning. For example, GoogLeNet pre-trained with ImageNet image library based on a large number of real images can predict diabetic retinopathy [8]; Inception V3 pre-trained in the ImageNet image library is used to distinguish keratinocyte malignant melanoma from benign nevus, and in recognition, accuracy has exceeded that of dermatologists [9].

2 Related work

Pugeault and Bowden used the Microsoft Kinect to collect colour and depth images of ASL 24 letters (J and Z involving dynamic gestures, not included), and then used OpenNI + NITE framework for gesture detection and tracking. The method used for feature extraction is to use a set of Gabor filters for the intensity of images, and finally classify the features using the random decision forest (RDF) method [10]. The conclusion of the paper is that the average recognition accuracy is 73% when using only colour image computing features, and the average recognition accuracy is 69% when using only depth images. The average recognition accuracy of the combined features of the two is 75%. The paper proposed a large American RGB-D dataset for recognition, but feature extraction was too simple and the recognition accuracy was not so high.

Estrela *et al.* also performed two groups of comparative experiments using the ASL dataset [11]. The first group of experiments compared Partial Least Squares (PLS) classifiers and Support Vector Machines (SVM) classifiers. They ensured consistent training data and feature extraction using the Binary Appearance and Shape Elements (BASE) feature. The accuracy rate of the PLS classifier is 66.27%, which is slightly higher than 62.85% of the SVM. The second set of experiments compares two feature extraction methods. The feature extraction method is replaced with the SIFT feature. The accuracy of the experiments using the PLS and SVM classifiers is 71.51 and 65.55%.

This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

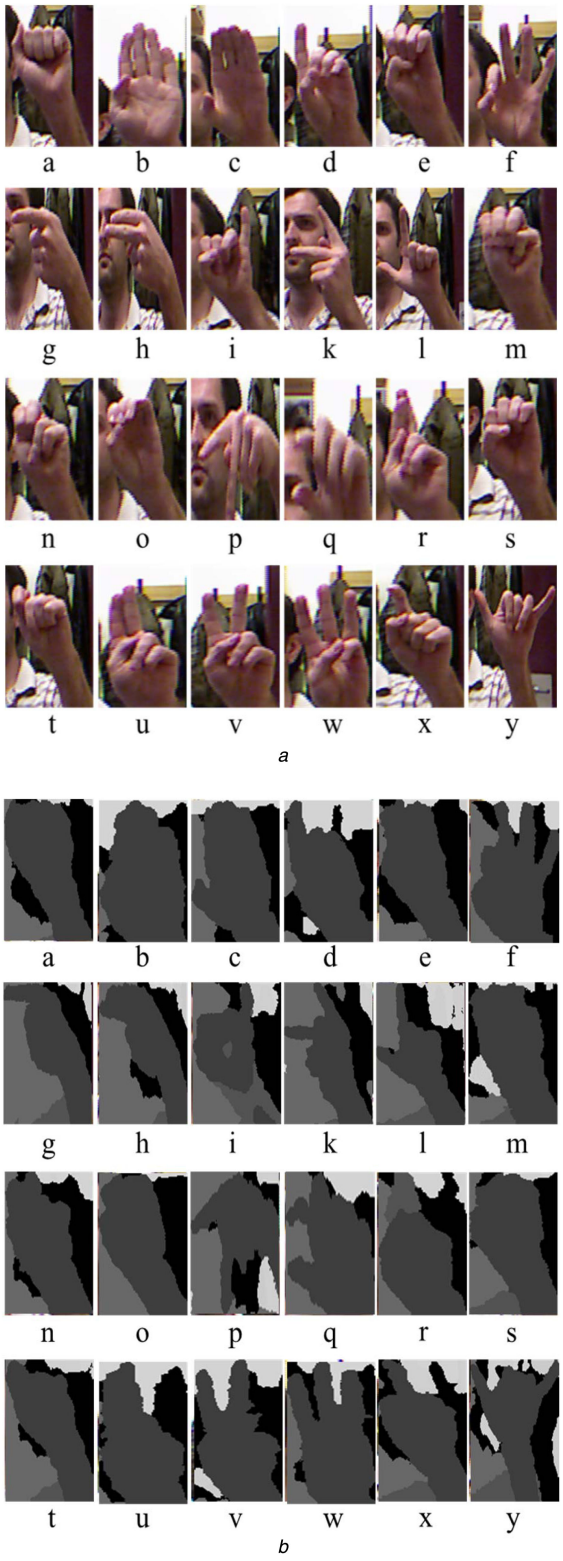

Fig. 1 *Part of images in the ASL dataset:*
(a) RGB images; ***(b)*** depth images

Therefore, the author concludes that using SIFT features and PLS classifiers can achieve optimal recognition results. The paper compares two kinds of classifiers and two kinds of feature extraction methods, but the recognition rate is still not good enough.

Zhang proposed a method called H3DF + SVM and conducted a classification experiment on the ASL dataset [12]. The paper uses the 3D plane histogram as a feature to extract, and then uses the SVM to classify it. The recognition rate can reach 73.3 and 98.9%. The author also uses the same method to test the ChaLearn Gesture Data dynamic gesture dataset, and its recognition rate can reach 95.5% and 99.2%. The 2 accuracy rates here represent the separation and non-separation of training and test sets. In the following experiments, we completely separated the training set and test set. Therefore, here we take the accuracy of this method as 73.3% on the ASL dataset.

Han *et al*. proposed a CNN method for static gesture recognition [13]. First of all, the author has collected a data set of ten gestures for a total of 12,000 images. In the image preprocessing stage, Gaussian skin model and background subtraction are used to obtain CNN training and test data. The experiment constructed a simple six-layer CNN (including convolution layer, average pooling layer, downsampling layer, Dropout layer, full connection layer, and last Softmax layer) achieving an average classification rate of 93.8%. However, the ten types of gesture datasets in this paper are not too hard to identify on the hand shape, and the background is relatively simple. Therefore, the author can only use the simple CNN to classify the RGB images to obtain higher recognition accuracy.

From the research status introduced above, the research trend of static gesture recognition evolves from the traditional method to deep learning, from manually extracting features to learning features using CNN. Using a deep CNN is able to learn deeper features of images and finish gesture recognition task under complex background and changing lighting conditions. Due to the appearance of RGB-D cameras, some gesture recognition methods that fuse RGB images with depth images have been implemented [14–16].

3 Datasets

3.1 Original dataset

The ASL dataset [10] is a static image dataset published by Pugeault and Bowden in 2011. It provides different gesture expressions of 24 English letters in the form of images (except for the English letters y and z). The ASL dataset is recorded by the five operators under different lighting conditions and background conditions using Kinect. Each gesture provides RGB images and corresponding depth images. There are about 500 hand gesture images corresponding to each letter in each operator in the database. Therefore, there are ∼60,000 RGB images and depth images in the ASL data set. Fig. 1 shows part of images in the ASL dataset.

3.2 Data preprocessing

3.2.1 Data cleaning: After research on the datasets, we found that the number of depth maps and colour maps in some folders is not equal. Therefore, we must be very careful when we use colour images and kinect depth images. Not all colour images have their corresponding depth images. Therefore, we need to perform data cleaning. Our criterion is to retain the colour image corresponding to the depth image and remove the rest of the image.

3.2.2 Depth image processing: Since the depth image in the ASL data set is a single-channel image, and the Inception V3 network input is $299 \times 299 \times 3$, our approach here is to convert the single-channel image to three channels, preserving one channel data as depth information, and set the other two channels to 0.

3.2.3 Data augmentation: The general method of data augmentation is to expand the data volume by means of image rotation, translation, and scaling. The data augmentation can avoid overfitting of the model and help train a scale-insensitive deep neural network, see Section 5.1.2 of this paper for more details.

3.3 Final dataset

After data cleaning and depth image processing, we divided the dataset into two sub-dataset: colour map and depth map. Each sub-dataset was then non-overlapped and divided into three categories: training set, verification set, and test set. The final data distribution was as shown in Fig. 2.

This is an open access article published by the IET under the Creative Commons Attribution License
(http://creativecommons.org/licenses/by/3.0/)

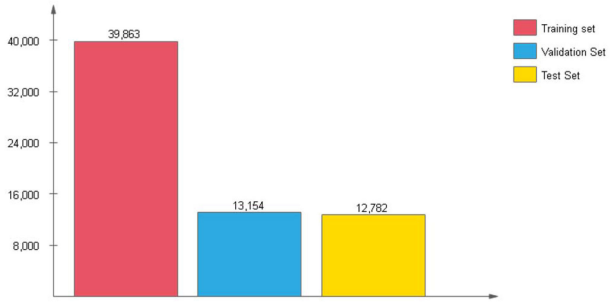

Fig. 2 *Data distribution of colour and depth images*

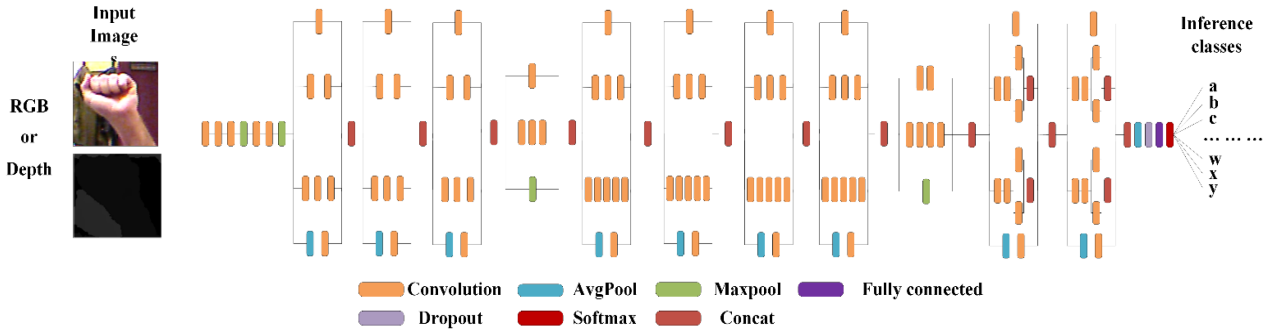

Fig. 3 *Inception V3 network structure*

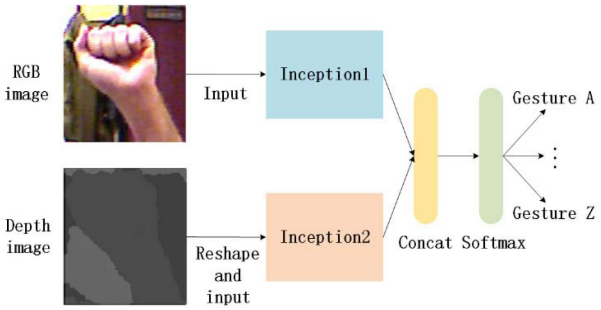

Fig. 4 *Feature concatenation diagram*

4 Methodology

4.1 Inception V3 model

The model used in the experiment was Inception V3, which was proposed by Szegedy *et al.*[7]. Inception V3 uses a volume integral solution based on Inception V1 and V2, decomposing the original 5×5 convolution kernel into two 3×3 convolution kernels, and using 3×3 convolution kernels with two 1×3 convolution kernels. Instead, the model creator applies the network proximity technology to the three Inception modules to change the output grid dimension to $17 \times 17 \times 768$, and then introduces the Inception module with five volume integral solutions. After the grid interval gets 8×8 the mesh dimension of $\times 1280$, and finally introduces two volume integral solutions for Inception, and simply merges two $8 \times 8 \times 1024$ grids in the channel dimension into a new grid with 2014 filters. The network structure of our Inception V3 can be seen from Fig. 3.

4.2 Two-stage training strategy

We divide the training process into two phases. In the first stage, we only trained the top layer and freeze other layers on our dataset. This step is set to adapt the original ImageNet pre-trained model to our classification task. In the second stage, we load the weights output from the first stage and continue training. We can make full use of the weight parameters pre-trained on the ImageNet dataset. The specific methods are described in Section 5.1.3 of this paper.

4.3 Concatenation of colour features and depth features

From our analysis of the research status of static gesture recognition, we can know that obtaining a high recognition rate for the finger spelling dataset still need to combine depth information. Colour and depth images are input into our fine-tuned Inception V3, and the feature vectors of them are merged at the last smoothing layer. Finally, the merged feature vector is input into the Softmax for classification. Fig. 4 shows the specific process.

4.4 Our Inception V3 fine-tuning model

Combining the contents of Sections 4.1–4.3 above, we propose a model based on the Inception V3 fine-tuning model. Our model directly accepts RGB and depth images as input, implementing the fusion of feature vectors at the high-level features of the network. In the training phase, we imported ImageNet pre-training weights and adopted a two-stage training strategy to fine-tune the model, speeding up the convergence of the model. After the training, we can save the optimal model and weights. In the testing stage, we only need to load them, and input the RGB image as well as its corresponding depth image to obtain the recognition result.

5 Experiments and results

5.1 Implementation details

Experiments were conducted using comparative experiments. We compared the fine-tuning models of Inception V3 proposed in Section 4.5 with advanced machine learning methods, deep learning algorithms and the RGB image input only method. In order to ensure the fairness of the experiment, all of our datasets are obtained in Section 3.3. The operating system of our experimental device is Ubuntu 16.04, and the GPU is GTX 1060. We use Keras as our framework for CNN implementation and use Tensorflow as the backend. Keras is an advanced neural network API written in Python.

5.1.1 Training parameters and training strategies: In the experiment, we set the top epochs and the fit epochs to 50, and the batch size to 100. In order to avoid overfitting, we use early-stopping in the training process. Early-stopping is a monitor for stopping the training in advance. In this paper, we set the accuracy of the validation set (validation accuracy) as the monitor. If there is

This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

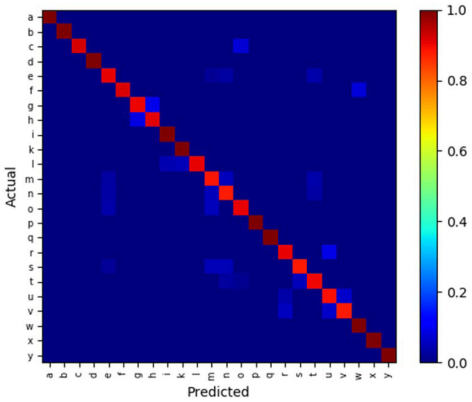

Fig. 5 *Confusion matrix of the ASL dataset in the fine-tuned Inception V3 classification experiment*

Table 1 Comparison with three kinds of traditional machine learning methods

Recognition methods	Gabor + RDF [10]	SIFT + PLS [11]	H3DF + SVM [12]	Our model
recognition rate	75%	71.51%	73.3%	91.35%

Table 2 Comparison with three convolutional neural networks

Recognition methods	CaffeNet	VGG16	VGG19	Inception V3	Our model
recognition rate	73.75%	83.44%	87.37%	88.15%	91.35%

a decrease in validation accuracy, then stop training after some consecutive epochs.

5.1.2 Implementation of data augmentation: The data enhancement tool uses Keras' ImageDataGenerator function. We set the parameters like rotation_range, width_shift_range, height_shift_range, and zoom_range to implement data augmentation. These transformations of the images greatly increase the amount of training data, and strengthen the model's recognition ability for gesture scale transformation as well.

5.1.3 Implementation of two-stage training: The first stage: we import the original model of Inception V3, and use Imagenet as the initial weight. Since this weight is a pre-trained weight for 1000 kinds of daily objects in ImageNet competition, the number of classifications is inconsistent with our goal. Here, we removed the top layer, re-established the full connection layer for our experiments. In order to obtain the appropriate top layer weights, we first randomly initialise the top layer network, freeze all convolutional layers of Inception V3, train multiple rounds on the ASL dataset, and finally save the top layer weights. The second stage: in the first stage, we determined the pre-training network weights that include the top level. Since there are relatively few data in this experiment (there are only about 500 RGB or depth images for each type of gesture), training with the entire network will most likely lead to overfitting. Therefore, we only fine-tune the two inception modules on the basis of the original weight, and freeze the previous inception module. We use low learning rate SGD to retrain our model and save the final model weights.

5.1.4 Implementation of feature concatenation: Using the Concatenate method in the Keras framework, we can easily implement feature concatenation. We only need to input both RGB and depth images into the fine-tuned Inception V3 network, and then set a concatenate function in front of the topmost Softmax layer to splice the RGB and depth feature vectors together and then input them to the Softmax layer for classification.

5.2 Results

We did our experiments on the ASL dataset, trained on 39,863 images and tested on 13,154 images. Through experiments, the average accuracy of training set obtained by our method is 96.23%, and the average accuracy of the test set is 91.35%.

In order to evaluate the specific performance of our method, we made a confusion matrix based on the recognition results of each class according to the actual and predicted values of the gesture tags. From the confusion matrix of Fig. 5, we can see that the darker red cubes are centred on the diagonal of the square and the recognition rate of many gestures is above 90%. The minimum recognition rate is the letters m and n, which are 85.37 and 87.80, respectively. We also found that the error of the classifier basically concentrated in two groups of gestures. One group was the letters e, m, n, s, and t. The hand styles of these letters were similar to the fist pose; the other group was the letters r, u, and v. These alphabetic gestures are stuck out with two fingers.

5.3 Comparison with traditional machine learning methods

We compare the experimental results with the Gabor + RDF, SIFT + PLS, and H3DF + SVM introduced in Section 2. We did not specifically implement these machine learning algorithms. We just compare the accuracy of these methods mentioned by the papers with our method. From Table 1, we can see that the recognition rate of our model is the highest. Additionally, it eliminates the process of gesture segmentation and manual selection of features.

5.4 Comparison with other CNN models

In order to ensure complete comparison experiments, we used the Keras framework to implement the four popular CNN models of CaffeNet, VGG16, VGG19, and Inception V3 without fine-tuning, and conducted experiments on the same dataset. We ensure that the rest of the experimental parameters are identical except for the network model. For example, we all use RGB-D input, and we have performed feature fusion processing. From the experimental results in Table 2, we can see that the highest accuracy of other four models is 88.15% of Inception V3, which is lower than 91.35% after fine-tuning. This also shows the superiority of the fine-tuning strategy of our method.

5.5 Comparison experiment of RGB input and RGB-D input

In the previous experiments, we have compared our method with traditional machine learning methods and other CNN methods. In this experiment, we compare the test results of RGB and RGB-D inputs. We want to find out how the accuracy of gesture recognition improves with the addition of depth information. Therefore, in this experiment, we only used the depth image input

This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

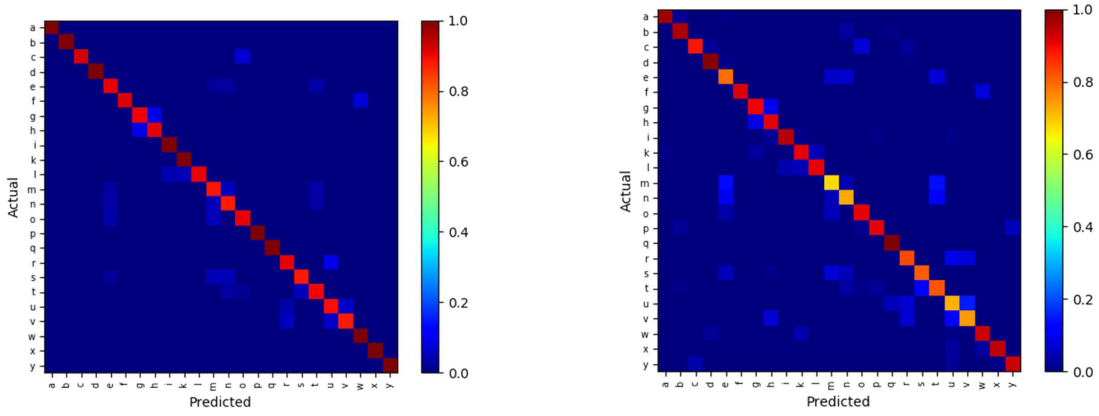

Fig. 6 *Comparison of two models' confusion matrices:*
(a) RGB-D input model confusion matrix; ***(b)*** RGB input model confusion matrix

Table 3 Comparison of recognition rates for two similar groups of gestures under RGB-D and RGB inputs

Similar gestures	Alphabet gestures	RGB-D input, %	RGB input, %	Compare to RGB, %
similar fist poses	e	91	79	+12
	m	88.3	75.3	+13
	n	88	78.2	+9.8
	s	87.8	81	+6.8
	t	90	82	+8
two fingers stick out	r	90.8	83.8	+7
	u	89.3	79.3	+10
	v	86.1	78	+8.1

as a variable, and all adopted the Inception V3 fine-tuning model to keep other variables consistent. After training and testing, the average accuracy of the test set of the final RGB input model was 83.78%, which was lower than 91.35% of the RGB-D input model. In order to judge the specific performance of the two classifiers, we made the confusion matrix of the RGB and the RGB-D input models in Fig. 6.

As we can see from Fig. 6, the darker colour blocks of the confusion matrix are still focused on the diagonal of the matrix, and most gesture recognition rates remain above 80%, and some easily recognised letters a, b, d, f, i etc. remain above 90%. The gesture with the lowest recognition rate is the letter m, which is only 75.3%. By laterally comparing (a) and (b), we find that the recognition rates of the two groups of similar gestures mentioned in Section 5.2 have risen from RGB to RGB-D, and the rate of increase has varied from 6.8 to 13%. The results of the comparison are recorded in Table 3. We can conclude that combining the features of depth information helps the model to classify similar gestures and helps to improve the overall recognition rate as well.

6 Conclusion

We propose a fine-tuning method based on Inception V3 for static gesture recognition. Unlike other static gesture recognition methods relying on artificial extraction features, our method can automatically extract features for classification. Our innovation is that we have adopted a two-stage training strategy for Inception V3, making full use of the weights pre-trained on the ImageNet to speed up our training process. Finally, we test our own method on the ASL dataset and the recognition rate reaches 91.35%.

In addition, we also conducted three groups of comparative experiments. In experiment 1, we compared three traditional machine learning gesture recognition algorithms, and proved that our method is superior to [10, 11, 12] in the recognition rate. Experiment 2 compares the four CNN models including CaffeNet, VGG16, VGG19, and Inception V3 without fine-tuning operation. The highest recognition rate among four models is 88.15% of Inception V3, which is lower than 91.35% of our fine-tuned method. This result explains the superiority of the fine-tuning strategy of this method. Experiment 3 compares the test results of CNN using RGB and RGB-D inputs. The conclusions prove that the combination of RGB + depth features help to improve the recognition rate of some similar gestures in the dataset, and the overall recognition rate is also increased from 83.78 to 91.35%. As for future work, we will focus on studying the optimisation of model parameters and using CNN for complex dynamic gesture recognition.

7 References

[1] Pisharady, P.K., Saerbeck, M.: 'Recent methods and databases in vision-based hand gesture recognition: a review[J]', *Comput. Vis. Image Underst.*, 2015, **141**, pp. 152–165

[2] Hasan, H., Abdul-Kareem, S.: 'Human–computer interaction using vision-based hand gesture recognition systems: a survey[J]', *Neural Comput. Appl.*, 2014, **25**, (2), pp. 251–261

[3] Khan, R.Z., Ibraheem, N.A.: 'Hand gesture recognition: a literature review[J]', *Int. J. Artif. Intell. Appl.*, 2012, **3**, (4), p. 161

[4] Zhang, Z.: 'Microsoft kinect sensor and its effect[J]', *IEEE Multimed.*, 2012, **19**, (2), pp. 4–10

[5] Szegedy, C., Liu, W., Jia, Y., *et al.*: 'Going deeper with convolutions[C]'. *IEEE Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9

Q2 [6] Simonyan, K., Zisserman, A.: 'Very deep convolutional networks for large-scale image recognition[J]', arXiv preprint arXiv:1409.1556, 2014

[7] Szegedy, C., Vanhoucke, V., Ioffe, S., *et al.*: 'Rethinking the inception architecture for computer vision[C]'. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826

Q3 [8] Gulshan, V., Peng, L., Coram, M., *et al.*: 'Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs[J]', *JAMA*, 2016, **316**, (22), pp. 2402–2410

[9] Esteva, A., Kuprel, B., Novoa, R.A., *et al.*: 'Dermatologist-level classification of skin cancer with deep neural networks[J]', *Nature*, 2017, **542**, (7639), p. 115

[10] Pugeault, N., Bowden, R.: 'Spelling it out: real-time ASL fingerspelling recognition[C]'. IEEE Int. Conf. on Computer Vision Workshops, 2011, pp. 1114–1119

[11] Estrela, B., Camarachavez, G., Campos, M.F., *et al.*: 'Sign language recognition using partial least squares and RGB-D information[C]'. Space Vehicle Orbital Motion, 1924

[12] Zhang, C., Yang, X., Tian, Y.L.: 'Histogram of 3D facets: a characteristic descriptor for hand gesture recognition[C]'. IEEE Int. Conf. and Workshops on Automatic Face and Gesture Recognition, 2013, pp. 1–8

[13] Han, M., Chen, J., Li, L., *et al.*: 'Visual hand gesture recognition with convolution neural network[C]'. IEEE/acis Int. Conf. on Software Engineering, Artificial Intelligence, NETWORKING and Parallel/distributed Computing, 2016, pp. 287–291

This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

[14] Shih, H.C., Liu, E.R.: 'Machine-to-machine interaction based on remote 3D arm pointing using single RGBD camera[J]', *Lect. Notes Electr. Eng.*, 2014, **260**, pp. 1109–1114

[15] Chen, X., Koskela, M.: 'Using appearance-based hand features for dynamic RGB-D gesture recognition[C]'. IEEE Int. Conf. on Pattern Recognition, 2014, pp. 411–416

[16] Li, Y., Wang, X., Liu, W., *et al.*: 'Deep attention network for joint hand gesture localization and recognition using static RGB-D images[J]', *Inf. Sci.*, 2018

This is an open access article published by the IET under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

JOE20188327

Author Queries

Q Please make sure the supplied images are correct for both online (colour) and print (black and white). If changes are required please supply corrected source files along with any other corrections needed for the paper.
Q1 Please check the sentence "After the grid..." for clarity.
Q2 Please provide volume number in Ref. [5].
Q3 Please provide place of conference in Refs. [7, 10, 11, 12, 13, 15].
Q4 Please provide volume number, page range in Ref. [16].

This is an open access article published by the IET under the Creative Commons Attribution License
(http://creativecommons.org/licenses/by/3.0/)