Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

Dynamic Hand Gesture Recognition for Robot Arm Teaching based on Improved LRCN Model

Kaixiang Luan
Graduate School of Information, Production and Systems
Waseda University
Kitakyushu, Fukuoka, Japan
luankaixiang@fuji.waseda.jp

Takafumi Matsumaru
Graduate School of Information, Production and Systems
Waseda University
Kitakyushu, Fukuoka, Japan
matsumaru@waseda.jp

***Abstract* – In this research, we focus on finding a new method of human-robot interaction in industrial environment. A vision-based dynamic hand gestures recognition system has been proposed for robot arm picking task. 8 dynamic hand gestures are captured for this task with a 100fps high speed camera. Based on the LRCN model, we combine the MobileNets (V2) and LSTM for this task, the MobileNets (V2) for extracting the image features and recognize the gestures, then, Long Short-Term Memory (LSTM) architecture for interpreting the features across time steps. Around 100 samples are taken for each gesture for training at first, then, the samples are augmented to 200 samples per gesture by data augmentation. Result shows that the model is able to learn the gestures varying in duration and complexity and gestures can be recognized in 88ms with 90.62% accuracy in the experiment on our hand gesture dataset.**

Index Terms -Deep Learning, LSTM, Robotics picking, Gesture recognition, Robot teaching

I. Introduction

With the development of the automation technology and computer vision, more and more robots are appearing in industry field. Therefore, the concept of human-robot interaction (HRI) has raised more and more research interests. In today's industrial environment, robots are often carried out with robot controller and control panels which are not often user-friendly. In the recent years, human-robot interaction (HRI) is developing towards simplification and humanization, there is no doubt that an effective interaction method between human and robot can greatly release the heavy task of human workers and increase the work efficiency. Gesture is an intuitive body language for human communication, together with the oral language, it can fully express the human's emotion, purposes and ideas. In past research, methods based on static hand gesture have been successfully tested in real scenarios. However, the challenge of recognition with dynamic hand gesture is still existed, because of the low recognition accuracy and speed.

The rise of deep learning has provided a new development direction for HRI with dynamic hand gesture. Deep learning methods can automatically learn the hidden essential features in images without the prior knowledge, build the learning model by continuously iteratively learning and training and make feature judgement or attribute classification for the new data, which provide stronger self-adaptability and generalization ability.

In this research, we proposed a vision-based hand gesture recognition system to classify 8 dynamic gestures of real-time for teaching the robot arm to do the picking task, in order to improve the performance of speed and accuracy, we will use a 100fps high speed camera for capturing the data.

The main contributions to this work are as follows: Firstly, we proposed and evaluated an improved LRCN architecture by combining the MobileNets (V2) and LSTM for the dynamic gesture recognition. Secondly, using the high-speed camera for quicker and better recognition. Thirdly, generate a novel dataset with 8 dynamic gestures using the high-speed camera for the robot picking task.

The rest of paper is organized as follows: We begin with the literature review of previous researches, after that, conclude the description of our methodology. Finally, closing off with conclusion of experiment performance and result.

II. Related Researches

A. Current robot control method

Robot control is one of the most challenging problems in the industry fields. At present, the mechanical interaction methods like handhold control panel and mechanical robot controller are widely used in industry between human and robot. For example, the IRC5 developed by ABB [1], as shown in Fig. 1. The R-30iA controller developed by FANUC. [2] Our goal is to find a method which is more intuitive and user-friendly.

Fig. 1 IRC5 Industrial Robot Controller
by ABB [1]

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

B. Hand gesture recognition

Recently, with the achievement of the computer vision and deep learning, several methods about hand gesture were proposed to control the robot. These methods can be divided into three categories: hand gesture with devices, static hand gesture recognition and dynamic hand gesture recognition.

1) *Hand gesture with devices:* This kind of method needs operator to wear the wearable devices such as gloves and sensors. Jain et al. Proposed the method which using the accelerator and Bluetooth inside the smart phone to recognize the hand gesture of operator and control the robot.[3]. Kruse et al. Developed a tele-robotic interaction system to control the industrial robot by the Kinect sensor and autonomous force controller [4].

2) *Static hand gesture recognition:* In the past, a lot of methods have been utilized in HRI based on static hand gestures. JIANG Suifeng et al. Developed an operation method for industrial robots based on static hand gestures extracted by Kinect sensor [5]. Harish Kumar Kaura et al. Proposed a system that user can navigate the wireless robot by various static gestures. [6]

3) *Dynamic hand gesture recognition:* With the development of depth-relative camera like 3D Kinect [7], 3D dynamic hand gesture recognition become another hot research topic. Ni Tao et al. Achieved to control the mechanical arm position and pose by using real-time dynamic gesture recognition based on Kinect sensor. [8]. Many recent researches based on CNN and deep learning also achieved good performance on the recognition of dynamic gestures because of their robustness at learning visual sequence features. J. A. Castro-Vargas et al. Using the 3DCNN to recognize the dynamic gestures applied on robot arm interaction. [9]. Barros et al. apply the deep neural model on the dynamic gesture recognition for the humanoid robot control [10]. Tsironi et al. They apply the ConvLSTM for the human-robot interaction.[11]

Summarize the above methods, we can conclude that using wearable devices are not suitable for the industrial environment because that it is too complex and time-consuming to wear and also uncomfortable for working. Also, when the operator is changed, the complicated calibration process must be repeated. In addition, the static hand gesture recognition methods limit the movements that the user can perform in front of the robot, as some gesture may trigger the system and cause erroneous actions. For these reasons, our approach will use the dynamic hand gestures to provide greater security when interacting with the robot, minimizing the risk of erroneous operations, and also making human manipulation more intuitive and user friendly.

The entire process of dynamic hand gesture control for HRI has been shown in Fig.2. We will focus on dotted part in this research.

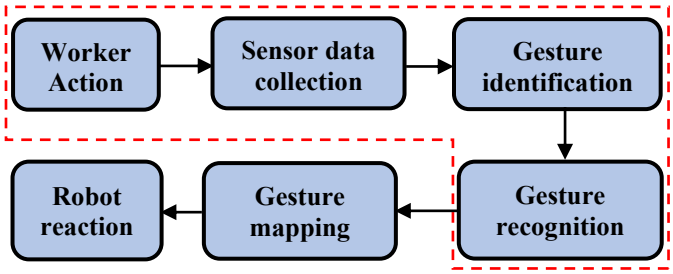


Fig.2. Process of gesture control for Human-robot interaction

C. Deep Learning model

In the models of deep learning, RNN (Recurrent Neural Network) is often used for processing sequence data. The hidden layer nodes of RNN are connected and the information about the previous sequence is memorized and applied to the current output calculation. However, with the increase of time sequence, the relative information correlation between the later and former RNN will gradually decrease and result in gradient vanishing and exploding problem. For this, S. HOCHREITER et al [12] proposed the LSTM (long short-term memory network) to solve the problem of long-term dependencies. Based on this, J. DONAHUE et al [13] combined the CNN and LSTM and propose the LRCN (long-term recurrent convolutional network), CNN for extracting the image features, LSTM for analyzing the correlation of feature sequences in time dimension.

III. PROPOSED METHOD

In our research, we propose a vision-based hand gesture recognition system for the task of robot picking based on improved LRCN model which combined the MobileNet (V2) and LSTM. The MobileNet (V2) is used for extracting the features of the hand gesture images, and LSTM is used for analyzing the correlation between image sequences. The whole process are divided into three stages: Data collection, Features extraction and Sequence learning. The input is the image sequences, the feature of each frame is extracted by CNN as an input to the LSTM. The LSTM can predict a label of the entire sequence.

A. Data collection

Some existing gesture datasets such as Microsoft Kinect, Leap Motion Dataset [14] and Creative Senz3D Dataset [15] have been generated by using RGBD cameras such as Microsoft Kinect camera and Creative Senz3D camera. These datasets cannot be used for our task for following reasons: These gestures are not designed for robot teaching. They are captured with the low-speed camera(30fps), which cannot clearly capture the intermedia we want. In addition, the depth images in these datasets are not required in our mission. Therefore, we decided to collect our own frames data which are useful for the robot picking task.

Fig.3. Experimental environment for data collection (0.5m)

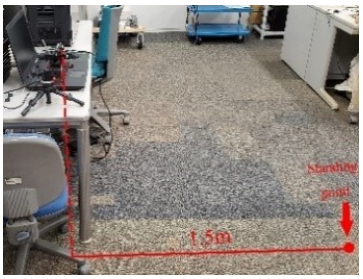


Fig.4. Experimental environment for data collection (1.5m)

The dataset contains 8 gestures (backward, down, forward, grasp, left, loose, right, up) captured by the ZED Mini camera [16], manufactured by STEREO LABS which can capture 100fps videos. The distance is set with 0.5m and 1.5m which can capture the upper body and whole body. Each category has 100 samples, and each sample has one label, which also means that every frame in the same sample has the same label. Each gesture performs 50 continuous single frames. The recording of gestures haves taken these influencing factors into account:

1) Camera angle: In the case where the entire gesture can be captured clearly, we adjust camera position to get the gesture data in different angle.

2) Different hand shapes: The entire dataset was captured by two different people in different environment. We also change the two people's clothes regularly.

After that the data augmentation is also being applied, because the gesture direction needs to be recognized in our task, some common data augmentation method like flip, translation cannot be applied. Two methods are applied as follows:

1) Illumination: For extracting the features better, we do the data augmentation to enrich our dataset by adjusting the lighting conditions of sequence images.

2) Add random noise: We add gaussian and pepper noise randomly to improve the generalization ability of model.

The entire dataset is collected in a static environment and the experimental environment is shown in Fig.3 and Fig.4. The image size is captured at a resolution of 224 X 224 pixels. Two fragments of 9 frames are shown as sample in Fig.5, they represent left and right movement respectively.

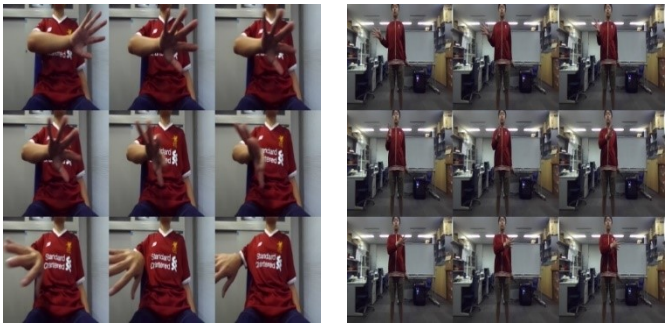


Fig.5. A sample of "right" and "left" in our dataset (0.5m and 1.5 m)

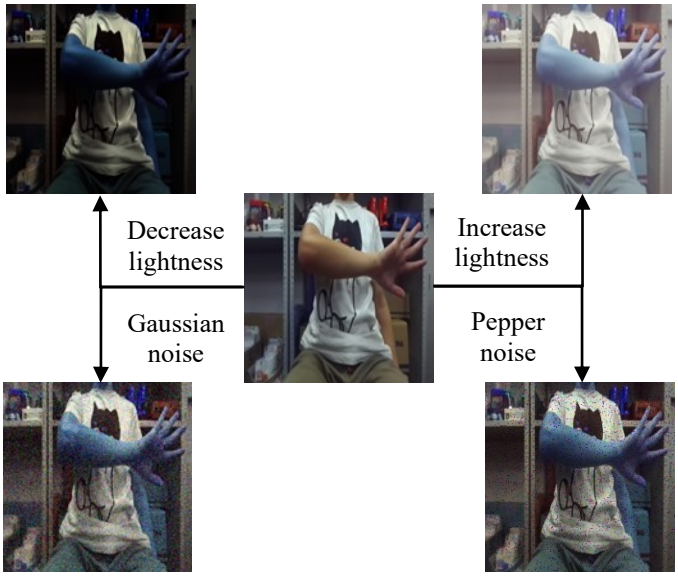


Fig.6 Data augmentation for recognition

For comparison, we also show a sample captured by the same camera at low-speed mode (30fps) as Fig.7. Clearly shows that compared with the high-speed camera; low-speed camera cannot capture clear intermediate gestures for the subsequent processing. It is not conducive to our later learning of sequence features.

Fig. 7. A sample captured by same camera with 30fps

B. Feature extraction

With the AlexNet [17] shined in 2012, Nowadays, various CNN have been developed, such as ResNet [18], VGG [19], ShuffleNet [20], ShuffleNet (V2) [21] and GoogleNet [22]. Although the performance is increasing, but with ResNet, the network has reached to 152 layers and the model size has reached 300MB+. This huge storage and computational overheard has severely limited the use of CNN in certain low-power areas.

Therefore, in our task, we choose MobileNet(V2) [23] for the visual feature extraction. This model takes the advantages of the limited resources of mobile devices and embedded applications to effectively maximize the accuracy of the model to meet a variety of application cases under limited resources. The main idea of this model is splitting the convolution process into two parts: depthwise and pointwise. Based on some structural adjustment of MobileNet (V1) [24], MobileNet (V2) has two major improvements:

1) Linear Bottlenecks, the nonlinear activation layer behind the small dimension output layer is removed, in order to ensure the expression ability of the model.

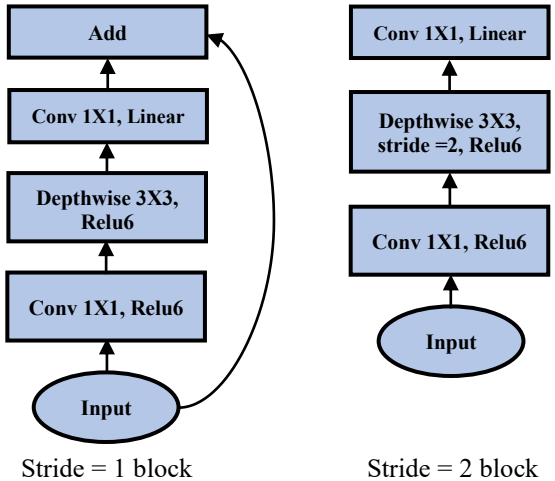
2) Inverted Residual block. This structure is exactly the opposite of the dimension reduction and re-amplification in the traditional residual block, so the shortcut turns to connect the feature map with the dimension reduced.

The Linear Bottlenecks + Inverted residual block is the basic structure of MobileNet V2. The architecture of is shown in Fig.8.

The reasons why we choose the MobileNet (V2) in our task are as follows:

1) Compared with others, MobileNet(V2) maintains the basically same performance in the process of greatly reducing the amount of network computing, it is the most suitable for the embedded device just like the industrial robot in our task.

2) The real-time operation is the goal that we want to achieve in the task, the MobileNet (V2) is also faster than other architecture, it can recognize the gesture faster while maintain the accuracy.



Stride = 1 block Stride = 2 block

Fig. 8. The architecture of MobileNet V2

In our CNN architecture, we add a global average pooling (GAP) layer to calculate the average value of each feature map after the last fully connection layer of MobileNet (V2). It reduces the dimension of the feature and regularize the structure of entire network to prevent the overfitting. The details of this part are shown in Fig.10. At first, each video frame comes into the CNN model and extracted with 3-dimension feature maps. Because the input size of LSTM is (batch size, timesteps, input dimension), so we use the global average pooling layer to transform the feature map into a 1-dimension vector. Compared with the fully connection layer, it greatly reduces the parameters of network.

C. Sequence learning

Then, the features come to the sequence learning stage. The characteristic of sequence learning is that the output of one step is not only depending on the input of this step, but also depend on the input and output of other steps. Traditional machine learning methods of the field of sequence learning includes HMM (Hidden Markov Model) and CRF (Conditional Random Field). In recent years, the deep learning algorithm RNN (Recurrent Neural Network) becomes more and more popular to solve the sequence learning problem, but it still has problems. The classical RNN architecture cannot learn long-term dependencies due to the problem of gradient vanishing and exploding problem.

Therefore, we choose Long Short-Term Memory (LSTM) is a special RNN architecture, it is good at processing the data related to time and also capable of learning long-term dependencies from the sequence. The LSTM unit architecture is combined with a memory cell, an input gate, an output gate and a forget gate. The structure pf LSTM has been shown in Fig.9.

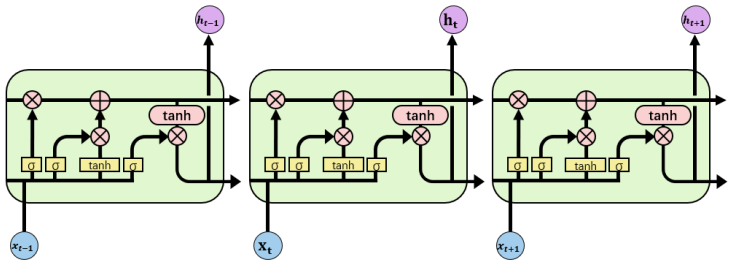


Fig.9. LSTM structure

The equations of LSTM unit architecture are defined as follows:

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$
$$i_t = \sigma_g(W_i x_t + U_i h_{h-1} + b_i)$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_x x_t + U_c h_{h-1} + b_c)$$
$$h_t = o_t \circ \sigma_h(c_t)$$

Here *t* is the timestep in time *t*. At time *t,* the feature x^t is extracted from the MobileNet (V2). After that, put the

processed feature into a LSTM block with 50 units. f_t is the forget gate's activation vector, i_t is the input gate's activation vector, o_t is the output gate's activation vector, the input of them are all $[x_t, h_{t-1}]$. H_t is the hidden state vector also known as output vector of the LSTM unit, c_t is the cell state vector, obtained from c_{t-1} at the previous moment and the input? *W, U, b* is the output of the LSTM unit.

The key point of this model is adding CNN architecture for each step of LSTM. We can implement this by encapsulating each layer of CNN into a TimeDistributed layer and add it into the main model. Through this layer we can realize the transition from two-dimension to three-dimension and realize the transformation from image classification to video classification. In this case, CNN is applied s to multiple input time steps and sequentially provides a series of image features to the LSTM model. Finally, we use the SoftMax function to get the categorical classification of 8 gestures. The entire architecture is shown in Fig.10.

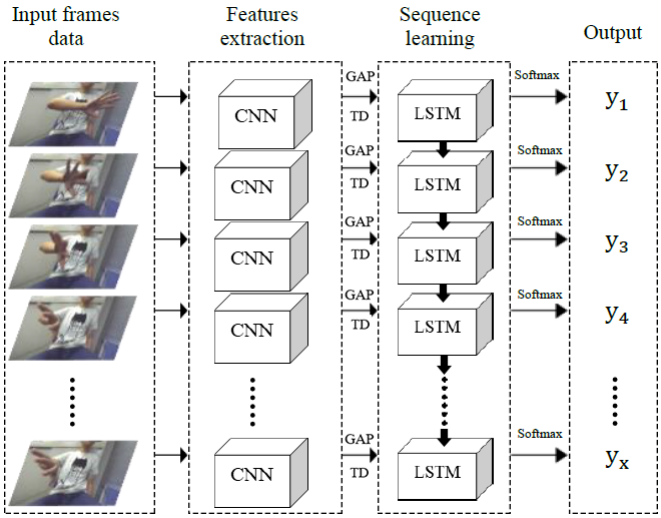


Fig.10. The architecture of entire model

IV. Result

Based on the method mentioned in the previous section, the experiment is performed. Firstly, we will describe the training result. After that, the evaluation result will be described.

A. *Training result*

In this research, LSTM is used for gesture recognition task as mentioned in previous chapter. There are eight dynamic hand gestures designed as experiment subject which is Backward, Down, Forward, Grasp, Left, Loose, Right, Up.

During the training stage, 80% of these data are used for training and rest 20% data are used for validation.

Due to the dataset is not no big, the MobileNet (V2) is not being trained, it is transfer learning from ImageNet and Finetuning on the top five layers. We wish to train it by backpropagating the errors of LSTM from multiple input sequence images to CNN.

The platform used for training the network is TITAN RTX which has 24GB memory. After some experiment, the GPU can fit with batch size 32. The parameter of training stage is set after experiment and be shown in TABLE I.

TABLE I

Input size	50 X 224 X 224 X 3
Steps/Epoch	150
Batch Size	32
Leaning Rate	0.01
Optimizer	SGD

For the multiple classification problem, we choose the categorical_crossentropy as the loss function. Assuming the output vector of the network is y, the target label is t, the function is shown as followed:

$$Loss = -\sum_{i=1}^{8}\sum_{t=1}^{c}(y_{i,t} * \log \hat{y}_{i,t}\ )).$$

8 is the number of samples and c is the category of label. The training losses and validation losses are shown in Fig.11

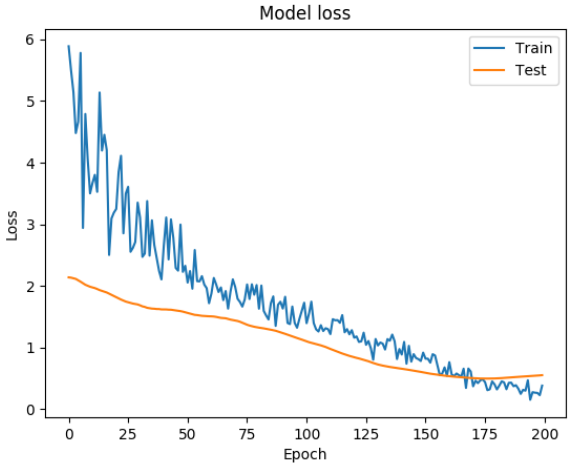


Fig.11. Training loss validation loss

After the training begun, the loss starts to shock down until 50th epoch and convergence to stability at 175th epoch.

B. *Recognition result*

20% of the entire dataset is selected as the validation data, this part of data hasn't appeared in the training data and also haven't do the data augmentation.

As shown in Fig.12, through the entire training process, the test accuracy is at a rising trend and convergence to 90.62% at 50th epoch. Because the validation data are not applied the data augmentation, until about 120th epoch, the training accuracy is lower than test accuracy.

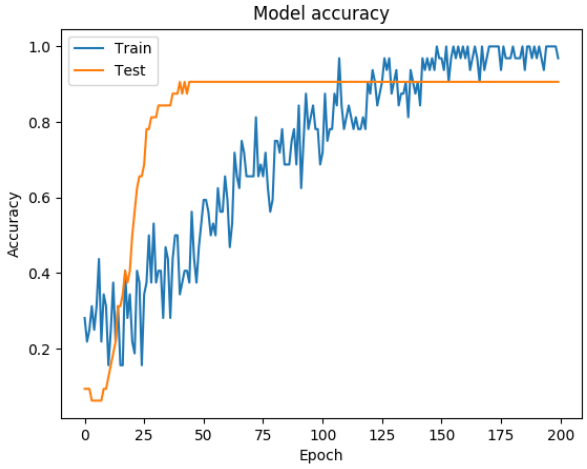


Fig.12. Training accuracy and validation accuracy

The entire operation procedure is shown as Fig.13. The operator should finish an entire gesture within 0.5s which equals to 50fps in 100fps camera. After that, each gesture will take 88ms to be recognized on our network.

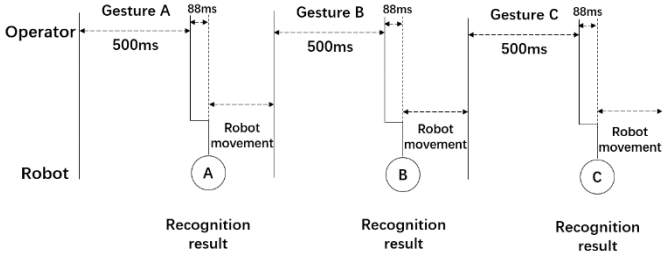


Fig.13. Operation procedure

V. Conclusion

In this work, the combination between MobileNet (V2) and LSTM have been tested in order to classify 8 gestures for operating the robots to do the picking work. The result shows that each gesture can be recognized in 88ms with at least 90.62% accuracy.

In the future, we plan to enrich the datasets with different environment to increase the robustness of the model and accuracy in more complex background. In addition, attention mechanism is an interesting research direction and we plan to introduce it into our model in order to further improve the generalization ability of the model.

Acknowledgment

This research is supported by Japan Society for The Promotion of Science (KAKENHI-PROJECT-17K06277), to which we would like to express our sincere gratitude.

References

[1] ABB IRC5, https://new.abb.com/products/robotics/controllers/irc5

[2] FANUC R-30iA http://rab.ict.pwr.wroc.pl/~malewicz/fanuc/fanuc/r-30ia_karel_reference_manual_%5bver%5b1%5d.7.50%5d%5bmarrc75kr07091e_rev.d%5d.pdf

[3] Jain M, Aditi A L, Khan M F, et al. Wireless gesture control robot: an analysis [J]. International Journal of Research in Computer and Communication Engineering, 2012, 1(10): 855 – 857.

[4] Kruse D, Wen J T, Radke R J. A sensor-based dual-arm tele-robotic system [J]. IEEE Transactions on Automation Science and Engineering, 2014, 12(1): 4 – 18.

[5] Yun Liu, Zhijie Gan, and Yu Sun. 2008. Static Hand Gesture Recognition and its Application based on Support Vector Machines. In Ninth Acis International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/ distributed Computing. 517–521.

[6] Kaura H K, HONRAO V, PATIL S, et al. Gesture controlled robot using image processing[J]. International Journal of Advanced Research in Artificial Intelligence, 2013, 2(5): 65-77.

[7] Jesus Suarez and Robin R.Murphy. 2012. Hand gesture recognition with depth images: A review. In Ro-Man. 411-417

[8] NI Tao, ZHAO Yongjia, ZHANG Hongyan, LIU Xianfu, HUANG Lingtao. Real-time mechanical arm position and pose control system by dynamic hand gesture recognition based on Kinect device. Doi: 10.6041/j.ison. 1000-1298.2017.10.053

[9] J. A. Castro-Vargas1, B. S. Zapata-Impata1, P. Gil1, J. Garcia-Rodriguez, F. Torres PREDICTIVE HAND GESTURE CLASSIFICATION FOR REAL TIME ROBOT CONTROL

[10] Barros, P., Parisi, G. I., Jirak, D., and Wermter, S. (2014). Real-time Gesture Recognition Using a Humanoid Robot with a Deep Neural Architecture.

[11] Tsironi, E., Barros, P., and Wermter, S. Gesture recognition with a convolutional long short-term memory recurrent neural network. Bruges, Belgium, 2 (2016)

[12] Hocheretter S, Schmidhuber J. Long short-term memory[J]. Neural Computation, 2013, 9(8): 1735 – 1780.

[13] Donahue J, Hendricks L A, Guadarrama S, et al. Long-term recurrent convolutional networks for visualvrecognition and description[C] // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. [S. I.]: Association for Computational Linguistics, 2014; 1724-1734.

[14] Giulio Marin, Fabio Dominio, and Pietro Zanuttigh. 2015. Hand gesture recognition with leap motion and kinect devices. In IEEE International Conference on Image Processing. 1565–1569.

[15] Alvise Memo, Ludovico Minto, and Pietro Zanuttigh. 2015. Exploiting Silhouette Descriptors and Synthetic Data for Hand Gesture Recognition. (2015).

[16] StereoLabs, Zed Mini. https://www.stereolabs.com/zed-mini/

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 1097–1105.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition. 770–778.

[19] Karen Simonyan, Andrew Zisserman. Very deep convolutional networks for large-scale image recognition.arXiv:1409.1556v6 [cs.CV] 10 Apr 2015.

[20] Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. arXiv preprint arXiv:1707.01083 (2017)

[21] Ma N, Zhang X, Zheng H T, et al. Shufflenet v2: Practical guidelines for efficient cnn architecture design[J]. arXiv preprint arXiv:1807.11164, 2018.

[22] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke and Andrew Rabinovich. 2015. Going deeper with convolutions. In Computer Vision and Pattern Recognition. 1-9

[23] Mark Sandler Andrew Howard Menglong Zhu Andrey Zhmoginov Liang-Chieh Chen Google Inc. MobileNetV2: Inverted Residuals and Linear Bottlenecks. arXiv:1801.04381v4 [cs.CV] 21 Mar 2019.

[24] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. (2017).