Proceedings of the ASME 2020 International Symposium on Flexible Automation
ISFA 2020
July 5-9, 2020, Chicago, USA

ISFA-12345

DYNAMIC GESTURE DESIGN AND RECOGNITION FOR HUMAN-ROBOT COLLABORATION WITH CONVOLUTIONAL NEURAL NETWORKS

Haodong Chen*, Wenjin Tao, Ming C. Leu
Department of Mechanical and Aerospace Engineering
Missouri University of Science and Technology
Rolla, MO 65409, USA

Zhaozheng Yin
Department of Biomedical Informatics&
Department of Computer Science
Stony Brook University
Stony Brook, NY 11794, USA

ABSTRACT

Human-robot collaboration (HRC) is a challenging task in modern industry and gesture communication in HRC has attracted much interest. This paper proposes and demonstrates a dynamic gesture recognition system based on Motion History Image (MHI) and Convolutional Neural Networks (CNN). Firstly, ten dynamic gestures are designed for a human worker to communicate with an industrial robot. Secondly, the MHI method is adopted to extract the gesture features and generate static images of dynamic gestures as inputs. Then an augmented gesture dataset is established, which includes the activities of six human subjects. Finally, a CNN model is constructed for gesture recognition. The experimental results show very good classification accuracy using this method. False recognition cases are analyzed and discussed.

Keywords: Human-robot collaboration, Dynamic gesture recognition, Motion History Image, Convolutional Neural Networks

1. INTRODUCTION

With the development of industrial intelligence, robotic systems are becoming an essential part of factory production. Meanwhile, the concept of human-robot collaboration (HRC) has attracted more and more interest in the industrial field. Literature suggests that in the industry with a high degree of automation, the HRC system can increase human-robot collaboration efficiency and also provide more flexibility in the work environment [1]. In 2012, Shi et al. [2] proposed different degrees of work-sharing. At the lowest level, the robot and the human operator do not have any contact and they work in two different spaces, but without any barriers between them. In 2014, Morato et al. [3] designed a framework to address safety and efficiency during assembly operations involving humans and robots. In 2019, Li et al. [4] proposed a method of human-robot collaboration planning, which considered human fatigue in assigning disassembly tasks to humans and robots.

Ideally, an HRC system should be similar to human-human collaboration in the industry. However, in the application of HRC, space-separation and time-separation of workers and robots result in lower productivity. To improve this situation and realize higher-efficieny collaboration, different communication channels between humans and robots should be established [5]. In the limited communication channels between human workers and industrial robots, gesture recognition has been effectively applied for use as an interface between humans and robots [6]. In 2010, Riek et al. [7] conducted a video-based lab experiment to measure time for a human to cooperate with a robot using gestures. Three gestures (beckon, give, shake hands) were designed in that experiment. In 2015, Chen et al. [8] proposed an approach for recognizing the gestures of a human worker during an assembly task in the HRC. In 2016, Liu et al. [9] established an interactive astronaut-robot system, which applied wearable glove and American Sign Language in the collaboration of the astronaut with a robot co-worker. In 2018,

*Corresponding author, Email address:h.chen@mst.edu

Islam et al. [10] presented a set of robust gestures for a diver to control an underwater robot in collaborative task execution. There are other prior works in gesture communication [11], but using standardized sets of gestures is limited.

To collaborate with human workers, robots need to understand human gestures correctly. In this regard, deep learning methods have demonstrated impressive performance in the generalization ability. For example, convolutional neural networks (CNN) have better performance in the action recognition with a limited dataset than traditional methods, especially when the dataset is limited [12–15]. Unlike the common feature extraction, which focuses on specific patterns, the deep learning extractor is trained to obtain the most discriminative features from given data. In 2018, Du et al. [16] combined the skeletonization algorithm and CNN method to realize the gesture recognition. In 2019, Wu [17] selected hand images and the edge images of a hand to design a double-channel CNN for the hand recognition task.

In this paper, a method of dynamic gesture recognition based on the Motion History Image (MHI) approach and depth CNN algorithm is proposed and demonstrated. This paper is organized as follows. Section 2 describes ten dynamic sign gestures and the corresponding robotic arm motions in our HRC system. Section 3 combines an image preprocessing algorithm and an MHI method to extract the features of dynamic gestures, and the dynamic gesture videos are converted into static images. Section 4 illustrates the construction of the CNN framework. The experimental setups and results are described in Sections 5. Section 6 provides the conclusion.

2. DESIGN OF DYNAMIC GESTURE SET

For the gestures used in the communication between human workers and robots, they should be easy to sign and remember, socially acceptable, and minimizing the cognitive workload. McNeil [18] proposed a classification scheme of gestures with four categories: Iconic (gestures present images of concrete entities and/or actions), Metaphoric (gestures are not limited to depictions of concrete events), Deictic (the prototypical deictic gesture is an extended 'index' finger, but almost any extensible body part or held object can be used), and Beats (gestures use hands to generate time beats). Based on these principles and categories, we design gestures that are mainly Iconic or Deictic for the HRC.

2.1. Gesture Set

Based on the real cases of human collaboration with a six-degrees-of-freedom (6 DoF) robotic arm (with a gripper as the end effector), we defined some essential commands to communicate with the robot. It consists of to a basic set of ten gestures, which are shown in Fig. 1. All the gestures are dynamic gestures. They are more natural than static gestures and can be combined together generate more commands.

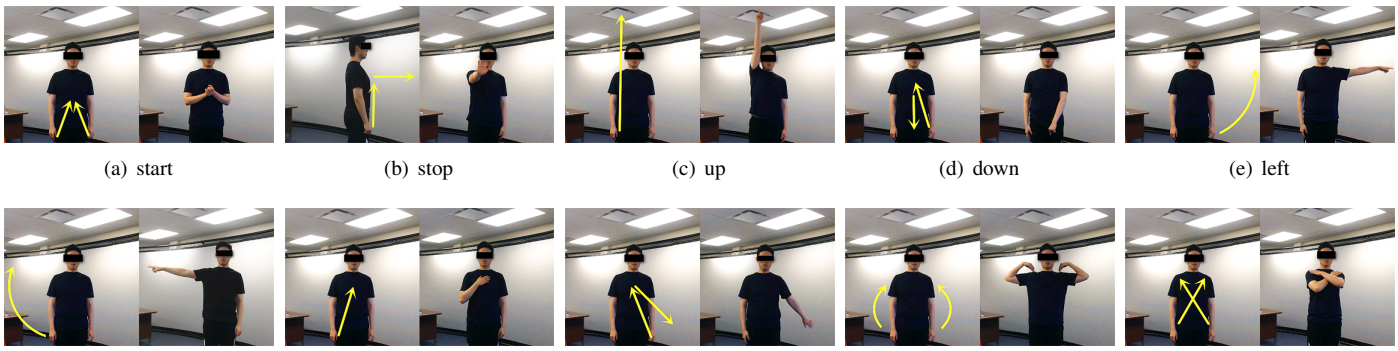
In Fig. 1, the left image of each gesture illustrates the start position: The person stands up with arms straight down. Hands are in a natural pose and pinky fingers are to the back. Then, the person can follow the direction of yellow arrows to carry out the gestures with their hands and arms. The right image of each gesture illustrates the end position. These various gestures can be carried out as follows:

- *Start*: Fully clap in front of the chest.
- *Stop*: Raise the right arm until the hand reaches the shoulder level and extend the arm with the palm facing the front, like a 'stop' sign in the traffic direction gesture.
- *Up*: Extend the right arm straight up with the index finger pointing up.
- *Down*: Bend the left hand and raise its wrist to the chest level. Then extend the left hand straight down with the index finger pointing down.
- *Left*: Swing the left arm straight out and up to the side with the index finger extended until the arm reaches the shoulder level.
- *Right*: Swing the right arm straight out and up to the side with the index finger extended until the arm reaches the shoulder level.
- *Inward*: Rotate the right forearm up around the right elbow joint with the hand open, until the right hand reaches the chest level and the palm faces back.
- *Outward*: Rotate the left forearm up around the left elbow joint with the left hand open until the hand reaches the chest level. Then rotate the left arm down around the left elbow joint until the arm is straight with about 30 $^\circ$ from the bodyline and its palm faces back.
- *Open gripper*: Bend each of the two arms up against its shoulder and the elbow until its fingers touch the same side of shoulder and its pinky finger at the front.
- *Close gripper*: Bend the two arms and cross them in front of the chest with the two hands on the different sides of shoulders. The palms face backward and the fingers are open.

2.2. Robot Movement

The next step is to consider the corresponding robot movement for each gesture above. The robot is a 6 DoF robot, with six joints and a gripper as the end-effector. The six DoF is needed in order to reach a volume of space from any orientation. The robot is free to change position of its end-effector as forward/backward (surge), up/down (heave), and left/right (sway) translations in three orthogonal (x-y-z) axes, as well as changes in the orientation of the end-effector. through rotation about three perpendicular axes.

The position and orientation changes of the end-effector within the robot's workspace can be realized by converting the



(a) start (b) stop (c) up (d) down (e) left

(f) right (g) inward (h) outward (i) open gripper (j) close gripper

FIGURE 1: The ten dynamic gestures.

end-effector position and orientation changes into the changes in the linear and angular displacements of the six joints. This is an inverse kinematics problem that can be readily solved for most of industrial robots. In order to have more intuitive human-robot interaction, the movement of the robot and the movement of the human will have a mirrored relationship since the human will face the robot in signing the gestures in human-robot collaboration. Thus when the human signs the robot to move right, the robot should move left and vice versa. However, when the human signs the robot to move up, down, inward, or outward, to start or stop, and to open gripper or close gripper, the robot should move accordingly (i.e., no mirror images on these commands).

3. FEATURE ACQUISITION BASED ON MOTION HISTORY IMAGE

The Motion History Image (MHI) approach is adopted to realize the feature extraction of human movements. This approach is a view-based template method that records the temporal history of a movement and converts it into static images. The MHI H_τ (x, y, t) can be obtained from an update function Ψ (x, y, t) using the following formula:

$$H_\tau(x,y,t) = \begin{cases} \tau & \text{if } \Psi(x,y,t) = 1 \\ \max(0, H_\tau(x,y,t-1) - \delta)) & \text{otherwise} \end{cases} \quad (1)$$

where x and y are the image pixel coordinates and t is time. Ψ (x, y, t) represents the movement of an object in the current video frame, the duration τ denotes the temporal extent of a movement, and δ is the decay parameter. This function Ψ (x, y, t) is called for every new video frame analyzed in the sequence. The result of this computation is a scalar-valued image where more recently moving pixels are brighter and vice-versa.

Regarding the parameters in Eq. (1), an MHI with a τ smaller than the number of frames will lose prior motion information. When the value of τ is set too high, the brightness changes (changes of pixel values) in the MHIs will be less clear. So in the generation of MHIs, τ is set same as the number of frames. In all dynamic gestures, both arms and body change their locations in different ranges. In an MHI, most features of the human movement exist in the arm pixels. After some tests, the decay parameter δ is set as 4 to eliminate the extra movements of the body, and only the arm pixels are recorded in the final MHIs.

Figs. 2 and 3 demonstrate the generation of the MHIs for the gesture representing the $left$ movement. Generally, an MHI is obtained from binary images of the sequential frames in Fig. 2. Note that not all frames are shown in this figure. The binary images are generated using the frame subtraction:

$$\Psi(x,y,t) = \begin{cases} 1 & \text{if } D(x,y,t) \geq \xi \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where $\Psi(x,y,t)$ represents the binarized image, and ξ is a threshold. The threshold ξ can eliminate the background noise in the MHIs, and it is set as 85 based on the environment shown in Fig. 1. $D(x,y,t)$ is defined as:

$$D(x,y,t) = \mid I(x,y,t) - I(x,y,t \pm \triangle) \mid \quad (3)$$

where the $I(x,y,t)$ is the intensity value of pixel location with the coordinate (x, y) at the tth frame of the image sequence. $\triangle$ is the difference in distance between two pixels at the same location but at different times [19].

Now that all the parameters in the MHI approach are set, the MHI of each dynamic gesture can be obtained. Fig. 4 shows the final MHIs for the ten gestures. All arm pixels during the movements are recorded. The brightness of pixels denotes the

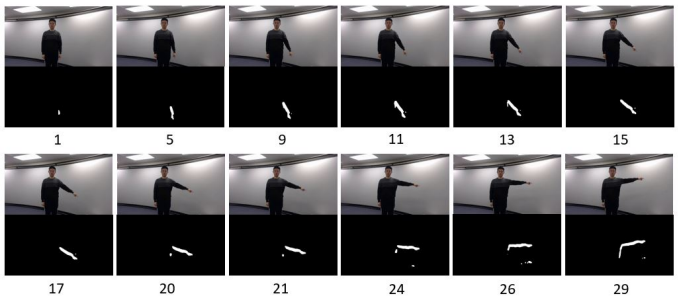


FIGURE 2: Binary images of different frames.

FIGURE 3: The MHI of the $left$ gesture.

gesture sequence. More recently moving pixels are brighter and vice-versa.

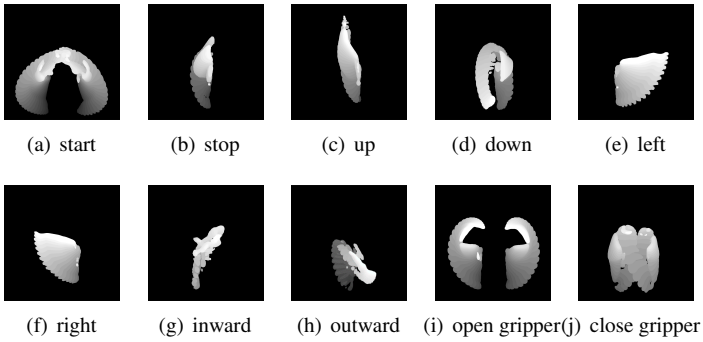


FIGURE 4: The MHIs of the ten dynamic gestures.

4. CONVOLUTIONAL NEURAL NETWORK MODEL

The overall architecture of our CNN model is shown in Fig. 5. The input images are MHIs with the depth of one. The input MHIs are resized to 32 × 32 ($width \times height$). The CNN consists of two convolution layers, each of which is followed by the max-pooling layers. The sizes of the convolution kernels, volumes at each layer, and the pooling operators are all shown in Fig. 5. A 5 × 5 × 40 feature map is obtained after the second pooling. Next, it is flattened as a 1000 feature vector. Then, a fully connected layer with 128 neurons is obtained. The output of this network is a softmax layer, which produces the class-membership probabilities for the 10 gestures.

In each convolution layer, the Recitified Linear Unit (RELU) is applied as the activation function [20]. The output of the softmax layer is computed as:

$$P(C \mid x) = \frac{exp(z_C)}{\sum_{C=1}^{10} z_C} \tag{4}$$

where $P(C \mid x)$ is the predicted probability of being class C for sample x, z_C is the output feature vector of the neuron q in the softmax layer, and 10 is the number of gestures.

The dropout is carried out after the second pooling layer, which randomly drops units from the neural network during training. It has been proven to be a powerful regularization technique used to avoid overfitting [21].

5. EXPERIMENT AND RESULT

5.1. Datasets

The raw training dataset includes 10 dynamic gestures signed by 5 human subjects. For each subject, every gesture was recorded 50 times. After the MHI extraction, there are 2500 MHI samples and each gesture class has 250 samples.

For deep learning, training data including a large number of samples can achieve good performance. To build a powerful image classifier using limited training data, image augmentation is applied to boost the performance of the network model.

Image augmentation artificially increases the variations of images in training data by using flips, rotation, variations in brightness and shifts, etc. [22]. In Fig. 4, it is obvious that the gestures $left$ and *right* are the same movements of the mirrored direction with a different arm. Hence the flip and rotation transformations will reduce the separability of these two images. The brightness change and horizontal/vertical shifts are finally carried out to enlarge the size of the training dataset to 10000 images, and there are 1000 images in every gesture class. Fig. 6 shows some samples of the augmented images. The inputs are the first images in both Fig. 6(a) and 6(b).

The test dataset is constructed with another human subject whose gestures are not recorded in the training dataset. There are 500 gestures without image augmentation in the test dataset. Each gesture class includes 50 samples.

5.2. Preprocessing

The image size of samples obtained from Section 5.1 is 1920 × 1080 ($width \times height$). In the CNN model described in Section 4, the input images need to have a uniform size of 32 × 32 ($width \times height$). Therefore, a preprocessing procedure of

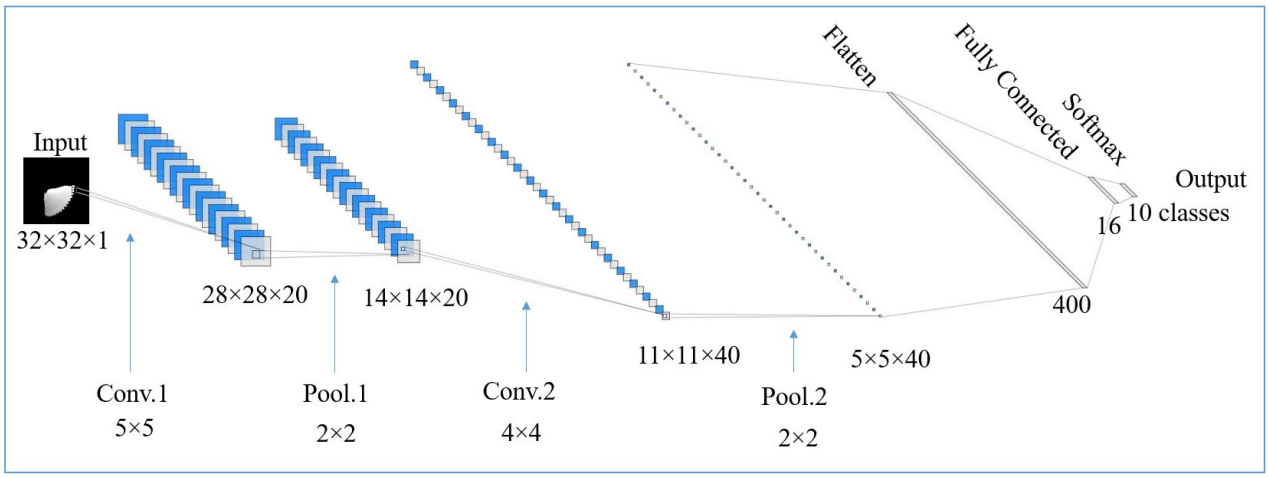
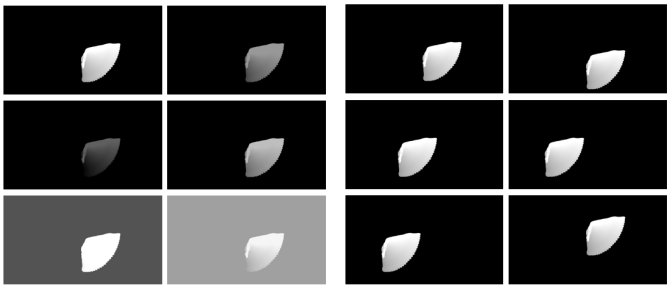


FIGURE 5: The overall architecture of the CNN model. (The 'Conv.' and 'Pool.' denote the operations of convolution and pooling, respectively).



(a) brightness (b) shift

FIGURE 6: Samples of the augmented images.

the size is implemented to resize the image data for the CNN model.

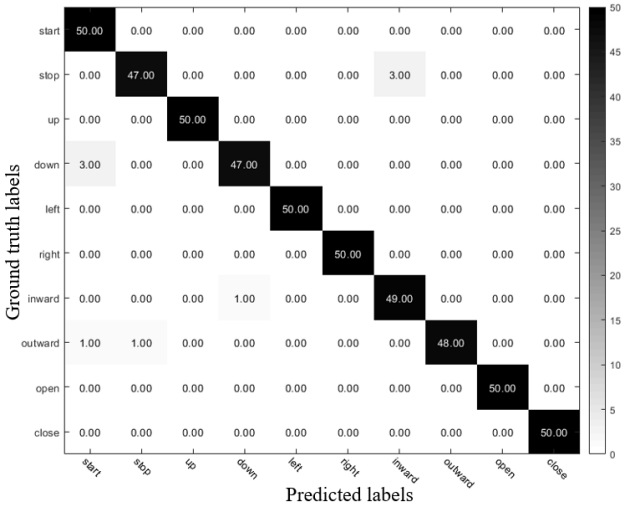


FIGURE 7: The confusion matrix of classification.

5.3. Result and Discussion

The confusion matrix of classification is shown in Fig. 7. The confusion matrix is also known as an error matrix. It realizes visualization of the classification performance. Each column of the matrix represents the instances in a predicted class while each row represents the instances in a ground truth class.

Some commonly used metrics are adopted to evaluate the classification performance:

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \tag{5}$$

$$Precision = \frac{TP}{TP+FP} \tag{6}$$

$$Recall = \frac{TP}{TP+FN} \tag{7}$$

$$F-score = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{8}$$

where in Eq. (5), (6) and (7), the True Positive (TP) describes a

sample x from a certain class C that is correctly classified as C. The True Negative (TN) means a sample x from a 'not C' class is correctly classified as the 'not C' class. The False Positive (FP) is defined as a sample x from a 'not C' class is incorrectly classified as C. The False Negative (FN) describes a sample x of class C is misclassified as other 'not C' classes. They are the four basic combinations of actual data category and assigned category in the classification [23, 24].

In Eq. (8), the $F-score$ is a measure that considers both the precision and the recall of the test. It represents the harmonic mean of the precision and recall [25].

Table 1 shows the values of the metrics of the classification results. Take the *start* gesture as an example, in the first cell, 50 *start* gestures (ground truth label) are predicted as the *start* (predicted label), so the TP is 50. In other diagonal cells, a total of 446 samples from 'not *start*' gestures are predicted as 'not *start*' gestures, so the TN is 446. In the first column, there are 4 'not *start*' gestures are predicted as '*start*' gesture, so the FP is 4. In the first row, 0 *start* gestures are predicted as other gestures, so the FN is 0.

TABLE 1: The metrics of classification evaluation.

Classes	TP	TN	FP	FN	Accuracy	Precision	Recall	F-score
start	50	446	4	0	0.992	0.925	1.00	0.961
stop	47	449	1	3	0.992	0.979	0.940	0.959
up	50	450	0	0	1.00	1.00	1.00	1.00
down	47	449	1	3	0.992	0.979	0.940	0.959
left	50	450	0	0	1.00	1.00	1.00	1.00
right	50	450	0	0	1.00	1.00	1.00	1.00
inward	49	447	3	1	0.992	0.942	1.00	0.970
outward	48	450	0	2	0.996	1.00	0.960	0.980
open	50	450	0	0	1.00	1.00	1.00	1.00
close	50	450	0	0	1.00	1.00	1.00	1.00

In the evaluation, the Precision describes the exactness or quality of the method, whereas Recall can be seen as a measure of completeness or quantity. The F-score can provide a more realistic measure of a test's performance by using both Precision and Recall. From Fig. 7 and Table 1, it can be seen that most gestures are recognized completely correctly. All the metrics are higher than 90% and the values of Accuracy and F-score are even higher than 95%, which shows how well the trained model could be generalized to a new subject.

On failure recognition, two cases in Fig. 7 should be particularly mentioned. Three *stop* gestures are recognized as the *inward* and three *down* gestures are classified as the *start* gesture. To find out the reason, we checked the training dataset and the test dataset. In Fig. 8, images in the first column are the correctly classified images of the test dataset. The middle three columns show the samples recognized as wrong gestures. The last column demonstrates samples of the gesture *start* and *inward* in the training dataset. For the *down* gesture, we can see that all right edges of gestures in the middle three images are more similar to the circle-shape with the *start* gesture, but different from the real *down* gesture with the straight edge on the right. For the false *inward* gestures, they share almost the same convex shape on the upper right corner with the real *inward* gesture, yet the actual *stop* gestures do not have that feature.

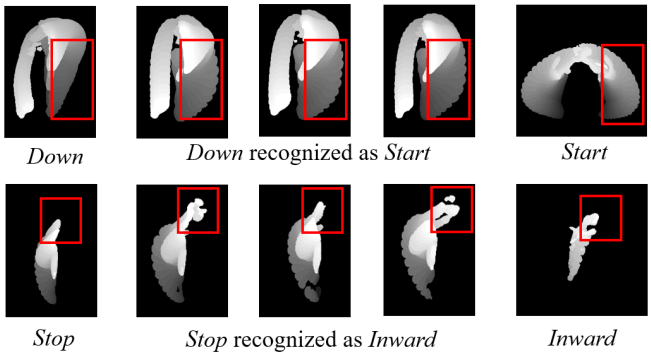


FIGURE 8: The failure recognition gestures.

Therefore, based on the reasons for resulting in the incorrect classifications as discussed above, two possible remedies are as follows. First, the instructions of some gestures can be more detailed, in order to avoid similar features existing in gestures of different classes. Second, the dataset can be more diverse, in order to cover more features extracted. In the future, we will improve instructions for signing all the gestures and will enlarge the dataset for extraction of more gesture features. Also, a modified network structure of deep learning will be constructed to improve the classification result.

6. CONCLUSION

In this paper, we design a new set of dynamic gestures for human-robot collaboration and construct a Convolutional Neural Network (CNN) model for dynamic gesture recognition. Ten hand gestures are designed, each represents a different instruction to the robot. The Motion History Images (MHIs)

approach is applied to the feature extraction, and an image dataset is established from the video-based dataset. The gesture dataset has five training subjects and one test subject. The developed CNN model is evaluated on the test dataset and achieves a recognition accuracy of higher than 94 %, which shows that our method has very good practicability in classification and robust generalization ability.

In the future, we will apply our dynamic gesture system to real-time human collaboration with robots and design more dynamic gestures to improve the performance of our HRC system.

ACKNOWLEDGMENT

This research work was supported by the National Science Foundation, grant no. 1830479, entitled *Intelligent Human-Robot Collaboration for Smart Factory*, and also by the Intelligent Systems Center at Missouri University of Science and Technology. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Haodong Chen, Zhiqiang Teng, Zheng Guo, and Ping Zhao. An integrated target acquisition approach and graphical user interface tool for parallel manipulator assembly. *Journal of Computing and Information Science in Engineering*, 20(2), 2020.

[2] Jane Shi, Glenn Jimmerson, Tom Pearson, and Roland Menassa. Levels of human and robot collaboration for automotive manufacturing. In *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*, pages 95–100. ACM, 2012.

[3] Carlos W Morato, Krishnanand N Kaipa, Jiashun Liu, and Satyandra K Gupta. A framework for hybrid cells that support safe and efficient human-robot collaboration in assembly operations. In *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 1. American Society of Mechanical Engineers Digital Collection, 2014.

[4] Kai Li, Quan Liu, Wenjun Xu, Jiayi Liu, Zude Zhou, and Hao Feng. Sequence planning considering human fatigue for human-robot collaboration in disassembly. *Procedia CIRP*, 83:95–104, 2019.

[5] Hao-dong Chen, Yi-fan Wang, Zheng Guo, Wen-xiu Chen, and Ping Zhao. A gui software for automatic assembly based on machine vision. In *2018 IEEE International Conference on Mechatronics, Robotics and Automation (ICMRA)*, pages 105–111. IEEE, 2018.

[6] Hongyi Liu and Lihui Wang. Gesture recognition for human-robot collaboration: A review. *International Journal of Industrial Ergonomics*, 68:355–367, 2018.

[7] Laurel D Riek, Tal-Chen Rabinowitch, Paul Bremner, Anthony G Pipe, Mike Fraser, and Peter Robinson. Cooperative gestures: Effective signaling for humanoid robots. In *Proceedings of the 5th ACM/IEEE international conference on Human-robot interaction*, pages 61–68. IEEE Press, 2010.

[8] Anca D Dragan, Shira Bauman, Jodi Forlizzi, and Siddhartha S Srinivasa. Effects of robot motion on human-robot collaboration. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, pages 51–58. ACM, 2015.

[9] Chenguang Yang, Chao Zeng, Peidong Liang, Zhijun Li, Ruifeng Li, and Chun-Yi Su. Interface design of a physical human–robot interaction system for human impedance adaptive skill transfer. *IEEE Transactions on Automation Science and Engineering*, 15(1):329–340, 2017.

[10] Md Jahidul Islam, Marc Ho, and Junaed Sattar. Understanding human motion and gestures for underwater human–robot collaboration. *Journal of Field Robotics*, 36(5):851–873, 2019.

[11] Michael Van den Bergh, Daniel Carton, Roderick De Nijs, Nikos Mitsou, Christian Landsiedel, Kolja Kuehnlenz, Dirk Wollherr, Luc Van Gool, and Martin Buss. Real-time 3d hand gesture interaction with a robot for understanding directions from humans. In *2011 Ro-Man*, pages 357–362. IEEE, 2011.

[12] Thomas Wesley Holmes, Kevin Ma, and Amir Pourmorteza. Combination of ct motion simulation and deep convolutional neural networks with transfer learning to recover agatston scores. In *15th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, volume 11072, page 110721Z. International Society for Optics and Photonics, 2019.

[13] Wenjin Tao, Ze-Hao Lai, Ming C Leu, and Zhaozheng Yin. Worker activity recognition in smart manufacturing using imu and semg signals with convolutional neural networks. *Procedia Manufacturing*, 26:1159–1166, 2018.

[14] Wenjin Tao, Ming C Leu, and Zhaozheng Yin. American sign language alphabet recognition using convolutional neural networks with multiview augmentation and inference fusion. *Engineering Applications of Artificial Intelligence*, 76:202–213, 2018.

[15] Md Al-Amin, Ruwen Qin, Wenjin Tao, and Ming C Leu. Sensor data based models for workforce management in smart manufacturing. In *Proceedings of the 2018 Institute of Industrial and Systems Engineers Annual Conference (IISE 2018)*, 2018.

[16] Du Jiang, Gongfa Li, Ying Sun, Jianyi Kong, and Bo Tao.

Gesture recognition based on skeletonization algorithm and cnn with asl database. *Multimedia Tools and Applications*, 78(21):29953–29970, 2019.

[17] Xiao Yan Wu. A hand gesture recognition algorithm based on dc-cnn. *Multimedia Tools and Applications*, pages 1–13, 2019.

[18] David McNeill. *Gesture and thought*. University of Chicago press, 2008.

[19] Md Atiqur Rahman Ahad, Joo Kooi Tan, Hyoungseop Kim, and Seiji Ishikawa. Motion history image: its variants and applications. *Machine Vision and Applications*, 23(2):255–281, 2012.

[20] Hasib Zunair, Aimon Rahman, and Nabeel Mohammed. Estimating severity from ct scans of tuberculosis patients using 3d convolutional nets and slice selection. *CLEF2019 Working Notes*, 2380:9–12, 2019.

[21] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[22] Justin Salamon and Juan Pablo Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3):279–283, 2017.

[23] Md Al-Amin, Wenjin Tao, David Doell, Ravon Lingard, Zhaozheng Yin, Ming C Leu, and Ruwen Qin. Action recognition in manufacturing assembly using multimodal sensor fusion. In *The 25th International Conference on Production Research (ICPR'19).*, 2019.

[24] Wenjin Tao, Ze-Hao Lai, Ming C Leu, Zhaozheng Yin, and Ruwen Qin. A self-aware and active-guiding training & assistant system for worker-centered intelligent manufacturing. *Manufacturing letters*, 21:45–49, 2019.

[25] Ingrid Visentini, Lauro Snidaro, and Gian Luca Foresti. Diversity-aware classifier ensemble selection via f-score. *Information Fusion*, 28:24–43, 2016.