Article │ Received 18 February 2024; Accepted 21 May 2024; Published 23 May 2024
https://doi.org/10.55092/sc20240005

Human–machine interaction controlling system for teleoperation of robotic arms in prefabrication assembly

Cheng Zhou [1,2], Rui Chen [1,2], Przemyslaw Sekula [3,4], Bin Tang [1,2,5,*] and Yu Qu [1,2]

[1] National Center of Technology Innovation for Digital Construction, Huazhong University of Science and Technology, Wuhan, China

[2] School of Civil and Hydraulic Engineering, Huazhong University of Science and Technology, Wuhan, China

[3] Department of Civil and Environmental Engineering, University of Maryland, College Park, USA

[4] Faculty of Informatics and Communication, University of Economics in Katowice, Katowice, Poland

[5] Institute of Artificial Intelligence, Huazhong University of Science & Technology, Wuhan, China

* Correspondence author; E-mail: thomastang@hust.edu.cn.

Abstract: Prefabrication assembly has been a widely used method in the construction industry in recent years. A controlling system for teleoperation of robotic arms in the prefabrication assembly with hand gesture recognition based on transfer learning is described in this study. A deep convolutional neural network with Xception model was used to recognize 13 different hand gesture types in the prefabrication assembly process with robotic arm in a laboratory setting. The proposed system provides safety and convenience to operators in construction sites. Results demonstrated that the proposed system has satisfactory performance and the developed algorithm can be used for teleoperation of robotic arms in prefabrication assemblies to provide feasible support for prefabricated construction.

Keywords: prefabrication assembly; hand gesture recognition; teleoperation; human–machine interaction; transfer learning

1. Introduction

Traditional construction methods are limited by increased production costs and decreased supply of labor force [1,2]. The emergence of prefabricated construction has alleviated these problems to some extent. Buildings are typically divided into construction components, which are a combination of parts and connectors. In the last few decades, assemblies using prefabricated components have become increasingly popular in the construction industry.

The prefabricated assembly process can be divided into three stages: production, transport and construction [3]. Prefabricated components offer faster production, lower cost, and more efficient assembly of elements than in situ construction [4]. Replacing in situ concrete casting panels with prefabricated elements has reduced construction time and labor cost by 70% and 43%, respectively [5]. However, as a prerequisite for this construction method is robotic automation. Construction site environments are complex and task-intensive, and the 'division of labour' between human-robot cooperation allows humans and robots to perform tasks based on their respective expertise and to take on more responsibilities in the future construction industry. Therefore, this paper proposes an automated construction method based on human-robot interaction with a robotic arm remotely operating prefabricated assemblies.

Due to the complex and changeable site environment, there are certain requirements for the timeliness and accuracy of remote operation. Human-computer interaction, which is characterised by two-way exchange of information and incorporates human participation and initiative, can effectively regulate remote operation to adapt to the complex site environment. Several robotic arm teleoperation methods have been proposed, controlled through virtual reality (e.g., data gloves), visual reality, and augmented reality [6,7]. This natural way of interaction is used in many other areas. For instance, a robot that can provide human service and object delivery was designed to achieve communication between human beings and service robots [8]. A real humanoid service robot applying the natural human–robot interaction was designed in the last decade [9]. Functioning in the natural way of human–robot interaction, a cooperative surgical robot system guided by hand gestures and supported by AR-based surgical field for robot-assisted percutaneous treatment was proposed [10]. Human–machine interaction includes speech, electroencephalogram (EEG) signal, hand gesture, face, and eye movement-tracking recognition. However, only some methods can be used for prefabrication assembly in construction sites. Given the complexity of the construction environment, construction sites are noisy and machines fail to identify the complete meaning of the speaker accurately. Speech recognition fails to satisfy requirements of prefabrication assembly because the mood of workers can also affect the effectiveness [11]. Face recognition, primarily used in service robots, is also unsuitable in construction sites [12]. Due to the necessity for workers to wear safety helmets on construction sites to meet relevant regulatory requirements, it is not very practical to wear neural-signal EEG devices at the construction sites [13]. Similarly, eye movement-tracking recognition is unsuitable because of the high complexity of construction sites. Sunlight, rain, noise, and other factors can affect the recognition of eye movement tracking [14]. The Hand gesture recognition can effectively overcome the shortcomings of the above recognition techniques, and it is more convenient for workers to operate. Therefore, Hand gesture recognition was chosen to control the robotic arm in this study.

External devices, such as Kinect or Leap Motion, are commonly used to recognize hand gestures [15–17]. Deep learning, on the other hand, has a strong learning ability and good adaptability, even surpassing human performance in areas such as image recognition and natural language processing [18]. Deep learning algorithms are trained using large datasets, and applying the knowledge learnt from the dataset to gesture recognition can effectively reduce the false recognition rate.

Deep learning has been widely used for speech recognition, image recognition and text processing [19,20]. At present, research on hand gestures for teleoperation of robotic arms in prefabrication assembly is lacking. However, acquiring tens of millions of images for the dataset in the traditional way of deep learning is difficult. Even if the dataset can be filled with enough images, the considerable amount of time needed to train the model fails to improve efficiency. By comparison, transfer learning, which is widely applied in small samples and has been recently used with deep learning architectures, can appropriately solve these problems and improve the outcome of problems with limited data [21,22]. Therefore, this study proposes the use of transfer learning in the controlling system of teleoperation of robotic arms in the prefabrication assembly. This study proposes a transfer learning based gesture recognition prefabricated assembly robot arm remote operation control system.

2. Literature review

2.1. Teleoperation for prefabrication assembly in the construction industry

Remote operation can enhance worker safety in extreme and hazardous construction environments. Over the years, research has proposed various methods for remotely operating construction equipment for excavation, aiming to reduce injury rates and effectively manage hazardous on-site tasks [23]. At present, robotic systems have been increasingly used in prefabricated construction [24]. Teleoperation has also been widely used in various fields and primarily designed for robot assistance or replacement of humans in extreme districts or difficult-to-reach places, such as nuclear facilities, low-temperature districts, and disaster areas. Several teleoperation methods for prefabrication assembly have been proposed in this century. NASA developed a teleoperation method for the controlling system of truss structure assembly [25]. ROCCO proposed a robotic system for assembly tasks that controls the robotic arm for erecting walls in residential buildings [26]. A European project, FutureHome, developed a system called AUTOMOD3 to generate assembly sequences and motion paths for robotic arms and build houses automatically with prefabricated components [27]. Several automated robot systems were developed by the group of Gramazio Kohler Research at ETH Zurich, including a mobile robotic brickwork system [28], to implement the construction process in the assembly of building components using a robotic arm.

Controlling with teleoperation of robotic arms in prefabrication assembly is the trend in the construction industry that offers convenience and safety to human operators. Several researchers have already proposed this idea by focusing on solving the time-delay issue that typically occurs during remote operation. In terms of approaches, the robotic arm can be controlled via manual operation, programmed control, AR, visual reality, hand gesture control, and voice control [29,30]. Human-centred research to understand and develop relevant tele-tele-operation technologies from a human perspective is necessary because tele-operation requires human-computer interaction and co-operation, and the existing tele-operation human-computer co-operation level is low.

2.2. Human–machine natural interaction for construction

Human–machine interaction includes speech, EEG signal, hand gesture, face, and eye movement-tracking recognition. Different methods are used in various fields. For instance, EEG signals can be used to identify resting EEG signals and visual-evoked potential of human beings [31–33]. At the same time, face recognition has a broad application prospect in public security, credit card verification, medical science, and human–machine interaction systems. These technologies can also be used in construction sites. For example, EEG signal recognition has been utilized in assessing mental fatigue in workers performing tasks and mental workload changes in construction workers during installation tasks [34,35]. Eye movement tracking has been used to view patterns of workers, understand their hazard recognition performance further, trigger self-reflection, and subsequently improve their hazard recognition performance [36]. However, some natural interaction approaches are unsuitable for construction sites because of the complex, noisy, and disordered environment. Given the different kinds of noises in construction sites, machines can have difficulty in recognizing correct directions from the commander with the use of voice control and the information transmitted to machines can also be interrupted by noise. Face recognition is limited by dynamic facial expressions that can be difficult to recognize, especially in tiny changes. EEG signal recognition is neither sufficiently accurate nor realistic because it disregards the use of safety helmets requisite in construction sites by requiring the operator to wear a head device. Moreover, natural factors, such as sunlight, rain, and noise, can affect the result. Hand gesture recognition is the optimal choice under these circumstances. The development of gesture recognition has progressed considerably. Using hand gestures to control is easy and convenient for anyone with professional knowledge, even those who lack relevant experience.

Hand gesture recognition can be conducted by using wearable devices, such as data gloves, and vision-based hand gesture recognition. A data glove is equipped with sensors that detect the flex of finger joints and the correlation between various hand positions, and then these discrete hand positions are translated into electrical signals represented by alphanumeric characters. However, wearable devices are inconvenient to use due to their large size, fragility, and high cost. Therefore, this study prefers vision-based hand gesture recognition, which has been discussed in some studies. A method for simultaneously detecting and tracking multiscale color features using particle filtering with an extension of layered sampling, which is referred to as hierarchical layered sampling, was proposed to recognize static hand gestures [37]. A method for contactless hand gesture recognition of meanings of nine static gestures in a predefined Popular Gesture scenario was put forward using Microsoft Kinect for Xbox [38]. A robust part-based hand gesture recognition system equipped with a Kinect sensor was used to recognize static hand gestures by addressing noisy hand shapes obtained from the Kinect sensor and measuring dissimilarity between handshapes [39]. Deep learning can be applied to solve the limitations of this study, such as low precision, serious dimensionality, self-occlusions, low processing speed, uncontrolled environments, and noise [40].

2.3. Deep learning for hand gesture recognition

Deep learning is a recent approach of machine learning that involves neural networks with more than one hidden layer. Deep learning has been used successfully in many natural-language processing tasks [17,41,42]. Networks based on deep learning paradigms demonstrate more biologically inspired architectures and learning algorithms compared with conventional feedforward networks. Deep networks are generally trained in a layer-wise fashion and rely on distributed and hierarchical learning of features similar to the human visual cortex [43]. These features allow the representation of highly nonlinear functions, the discovery of additional interesting features in training data, and improved modeling of complex problems.

Some scholars have investigated deep learning for hand gesture recognition. For example, a method that uses deep learning with a dataset of 243,000 tuples of images constituted by color, depth, and mask of the hand region was presented [44], and a method applying deep learning was proposed to recognize 24 hand gestures [45]. These methods have common problems, such as having a large hand gesture dataset, costly algorithm, and complex computational burden. A large amount of data is commonly required to train large networks when using deep learning. Hence, the performance may not be maximized when the available data is limited. However, the study suffers from many challenges and limitations in the absence of a large-scale specialized dataset. Although a dataset for optical images is large and easy to obtain, data are limited because of laborious processing and organizing. Hence, transfer learning is the chosen method in this study. Transfer learning is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying the solution to a different but related problem. Compared with deep learning models trained on “big data” image datasets (e.g., ImageNet), transfer learning is cost effective and efficient to use by “transfering” their learning ability to new classification scenarios rather than training from the beginning [46]. Accuracy can improve by using a large dataset for other applications [20], and small datasets are needed to train the network.

Transfer learning has been investigated in civil engineering for years. For example, a DCNN was used to train the big-data ImageNet database, which contains millions of images, and that learning was transferred to detect cracks in surfaced pavement images of hot-mix asphalt and Portland cement concrete, including a variety of noncrack anomalies and defects [21]. A safety guardrail detection model based on a convolutional neural network (CNN) using transfer learning was proposed to improve the efficiency in identifying unsafe conditions in construction sites and safety performance [46]. Transfer learning, which has rarely been used in hand gesture recognition, can solve not only the problem of small datasets but also save time and resources.

3. Methods

3.1. Framework

The framework of the proposed method is illustrated in Figure 1. A certain amount of gestures is first defined according to the simulation of the robotic arm while considering the clarity of meaning of each gesture in the design of gestures. After the definition of gestures, the information is then transferred to the local server, which is responsible for understanding the meaning of gestures. Transfer learning is used to solve the problem of lacking image data. Hand gestures corresponding to commands are classified into several types, and the commands are conveyed in the form of signals to the line receiver of the robotic arm. The receiver sends the signal to the computer, and the computer analyzes the signal. The local computer converts the data to commands through a program and passes the commands to the robotic arm.

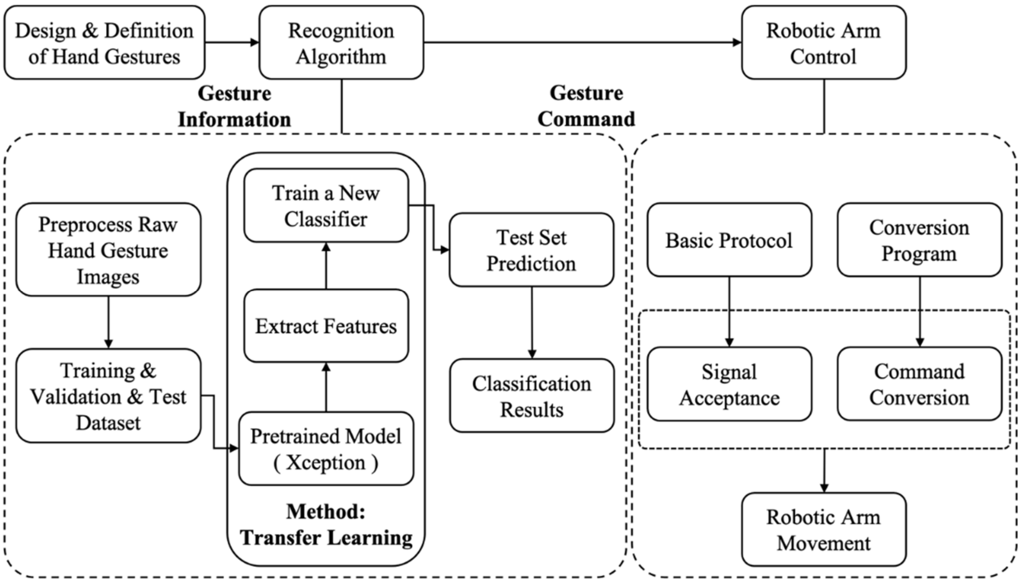


Figure 1. Framework of the proposed method.

3.2. Hand gesture recognition with transfer learning

3.2.1 Overview of transfer learning

Transfer learning can be used to train the Xception model [23] with a limited amount of hand gesture dataset and realize better recognition results than traditional machine learning with the same size of dataset. Figure 2 shows the classification of source and target domain. The domain, defined D = {χ, P(X)}, consists of two components, namely, feature space X and marginal probability distribution P (X), where X = {x1, x2, x3, …, xn} ∈χ. The task, defined T ={Y, f(·)}, consists of a label space Y = {y1, y2, y3, …, ym} and an objective predictive function f (·), which is not observed but is to be learned by pairs {xi, yi}. Finally, obtaining the source domain Ds, target domain Dt, and learning tasks Ts and Tt via transfer learning can improve the learning of the target predictive function f (·) in Dt by using the knowledge in Ds and Ts (Ds ≠ Dt or Ts ≠ Tt). Transfer learning uses the labeled source domain data to learn the calibration of the target domain data. The task of transfer learning allows the use of labeled source domain data to establish a reliable model for predicting data in the target area (source and target data have different probability distributions). Maximum mean discrepancy (MMD) is used in predicting data to calculate the distance between two domains as follows:

$$\mathrm{MMD}(\mathrm{X},\mathrm{Y}) = \left\| \frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}} \Theta(\mathrm{X}_{\mathrm{i}}) - \frac{1}{\mathrm{m}}\sum_{\mathrm{i}=1}^{\mathrm{m}} \Theta(\mathrm{X}_{\mathrm{i}}) \right\|_{\mathrm{H}}^{2} \tag{1}$$

where $\Theta(\cdot)$ is the mapping function, *X, Y* are two different distributions, *n, m* are the number of each distribution, and *H* is the distance to map the data in Reproducing Kernel Hilbert Space (RKHS) by $\Theta(\cdot)$.

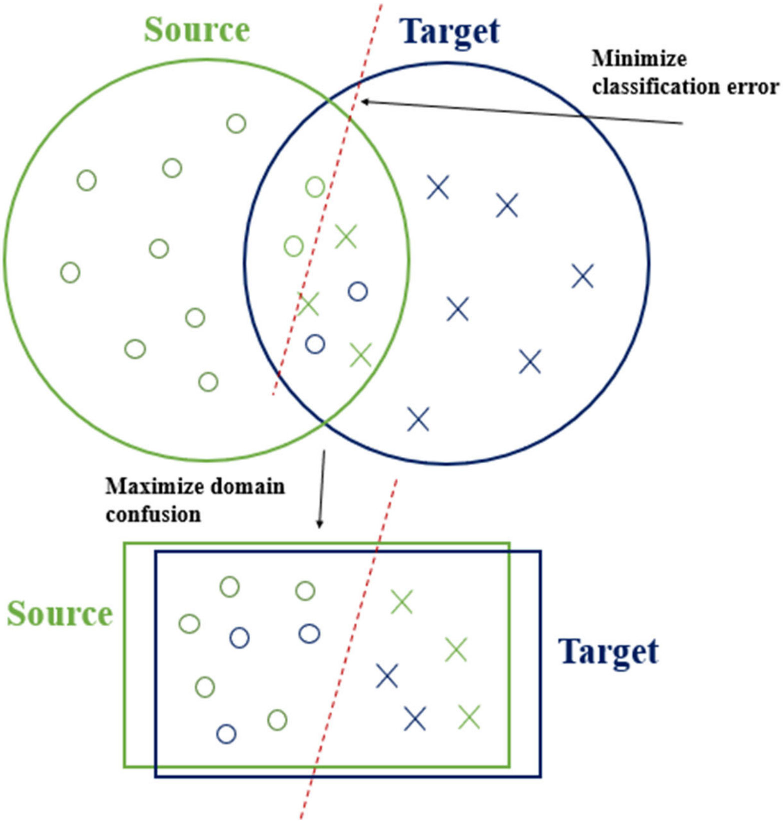


Figure 2. Classification of source and target domains.

Xception DCNN has been pretrained on ImageNet using general images for hand gesture recognition. A pretrained model may not be 100% accurate, but it saves a considerable amount of effort required in reinventing the model. Xception is then implemented on Keras framework with TensorFlow on the backend. The network in this step is “generalized.” Hand gesture images are input to feed the Xception DCNN, and then the network processes the dataset according to the defined epoch. The dataset is classified into several categories, and Keras will classify them by identifying the subcatalog. The direct training process has difficulty in obtaining a satisfactory result. Therefore, a fine-tuning method must be selected to optimize the entire Xception model. In summary, hand gesture recognition with the transfer learning method can be divided into two parts, namely, feature extractor and fine-tuned model.

3.2.2 Pretrained model of transfer learning

Given that deep learning is widely used for image recognition, several CNNs, including AlexNet, GoogLeNet, VGGNet, and ResNet, have been proposed. Updates of the above-mentioned off-the-shelf type of networks continued along with the growing depth of network layers to solve the problem of increasing the number of parameters while improving the performance of neural networks. This process may lead to overfitting and a massive amount of computation. Among these networks, the algorithm implemented in this study selected the Xception model of GoogLeNet as the network architecture.

The original model of Xception is the Inception model, which uses different sizes of convolutional kernels to observe the input data, including 1 × 1, 3 × 3, and 5 × 5 convolutions [23]. Networks expand and calculation times increase while using three kinds of kernels. Meanwhile, 1 × 1 convolutions are added into the network because they can increase or decrease the network dimension. This change leads to a CNN called Inception V1. Although the number of parameters is decreased to some extent, Inception V1 still requires a considerable amount of time to calculate. A small size of convolution kernels, such as 5 × 5 convolutions with 1 × 1 convolutions, can be used to replace the large size. In this way, the depth of the network increases with few parameters (Inception V3). On the basis of Inception V3 model, the researchers further improve the model via the method of depthwise separable convolution (Figure 3) to take the place of the former convolutions of Inception V3 and obtain the Xception module used in this study.

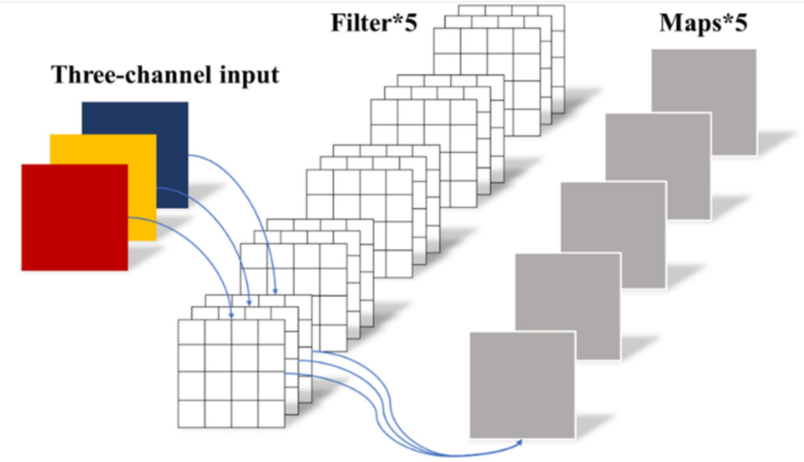


Figure 3. Depthwise seperable convolution.

Xception was proposed on the basis of depthwise separable convolutional layers under an influential hypothesis that determines cross-channel and spatial relations during the extraction of features in CNN. The Xception architecture consists of 36 convolutional layers, which are structured into 14 modules with linear residual connections around and beside the head and tail modules. In each module, the separable convolution layers come after ReLU while max-pooling layers reduce the image dimension that was transferred from the convolution layers. This extreme version of Inception first uses a 1 × 1 convolution to map the cross-channel correlations and then separately maps the spatial correlations of each output channel with the 3 × 3 convolution (Figure 4). Convolution kernels with certain weights extract information from the given image in the size of (s, t) and obtain an output of the changed image, which is the feature map. The convolution is expressed as follows:

$$\mathrm{P(s,t)} = \mathrm{(A * B)(s,t)} = \sum\nolimits_{\mathrm{k=0}} \sum\nolimits_{\mathrm{l=0}} \mathrm{A(s-k\,,t-l)B(k,l)} \tag{2}$$

where P is the convolution result, which is also the feature map; B is the discrete function of the kernel; and A is the discrete function of the convoluted matrix, which is the image. Gradient vanishing occurs with the traditional activation functions, such as standard logistics. The Xception model uses ReLU to prevent the situation mentioned above. The ReLU is expressed as follows:

$$\mathrm{ReLU}(x) = \max(x, 0) \tag{3}$$

where x is the value of the element.

The Xception model adds max-pooling layers in some modules to improve the image classification. These layers extract the maximum value in specific squares. A global average pooling layer is added at the end of the network to replace the traditional fully connected layers. The logistic regression for the output layer uses the following softmax as the activation function:

$$\mathrm{softmax}(x)_m = \frac{e^m}{\sum_{n=1}^{n} e^n} \tag{4}$$

where n is the total number of the elements and m is the order of the specified element.

Softmax can map the original output into the range (0,1), and these outputs are added up to 1, which corresponds to the probability. Several experiments on image classification tasks are conducted while training the models, including VGG-16, ResNet-152, and Inception V3, on both ImageNet and JFT to verify the performance of Xception. The results showed that the Xception architecture demonstrates better training accuracy than other architectures.

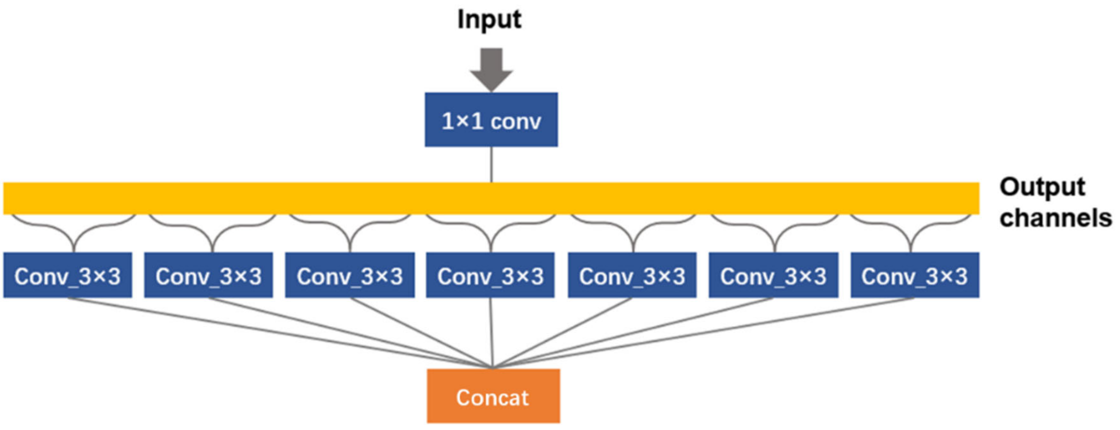


Figure 4. Extreme version of Inception.

3.3. Teleoperation system for the robotic arm

Figure 5 shows the kinematics model of an ABB IRB 6700-235/2.65 robotic arm. The model demonstrated in Figure 5 is based on the standard D–H parametric method, which is represented as the D–H coordinate system. The coordinate system of each arm joint is set up on the basis of the coordinate system of axis 1 in Figure 5. D–H parameters of the robotic arm are listed in Table 1.

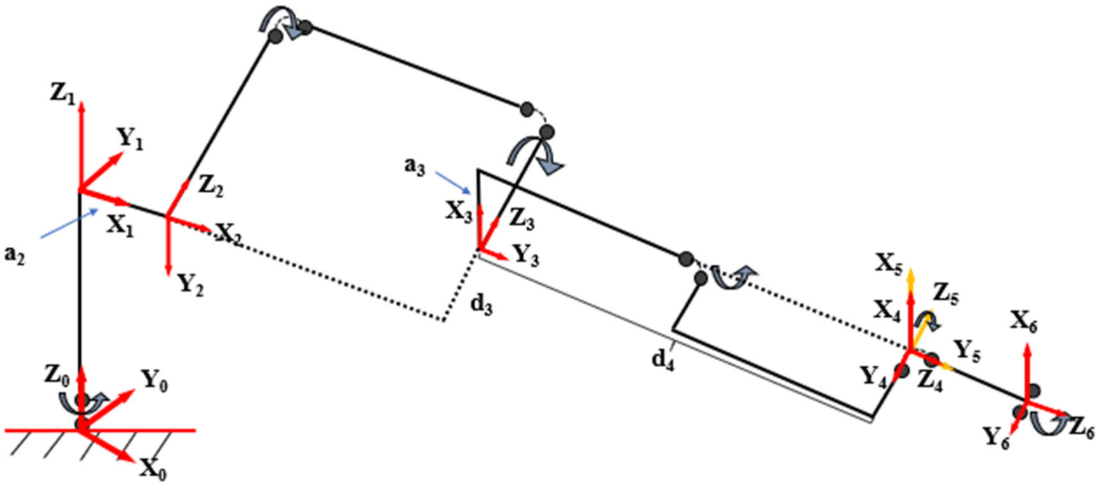


Figure 5. Diagram of the D–H coordinate system of ABB IRB 6700.

Table 1. D–H parameters of ABB IRB 6700-235/2.65.

Linkage i	Joint rotation angle (θ_i)/(°)	Distance (d_i)/mm	Length (l_i)/mm	Twist angle (α_i)/(°)
1	0	780	320	−90
2	−90	0	1135	0
3	0	0	200	−90
4	180	1182.5	0	−90
5	0	0	0	90
6	0	200	0	0

The parameters listed in Table 1 can be used to describe each joint's moving position in Equation (5). ${}^{i-1}B_i$ is the space description of a certain joint, θ_i is the rotation angle that the bridge between x_n and x_{n+1} moves along the Z_n-axis, d_i is the distance between x_n and x_{n+1} along the Z_n-axis, and α_i is the twist angle of the joint. The position of the last axis of the robotic arm is expressed as follows:

$$ {}^{i-1}B_i = R(\theta_i)Trans(d_i)Trans(l_i)R(\alpha_i) = \begin{bmatrix} cos\theta_i & -sin\theta_i cos\alpha_i & sin\theta_i cos\alpha_i & l_i cos\theta_i \\ sin\theta_i & cos\theta_i cos\alpha_i & -cos\theta_i sin\alpha_i & l_i sin\theta_i \\ 0 & sin\alpha_i & cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5) $$

$$ {}^0B_6 = {}^0B_1 \; {}^1B_2 \; {}^2B_3 \; {}^3B_4 \; {}^4B_5 \; {}^5B_6 = \begin{bmatrix} n_x & o_x & d_x & P_x \\ n_y & o_y & d_y & p_y \\ n_z & o_z & d_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6) $$

A diagram of the teleoperation system for robot arm control is illustrated in Figure 6. An operator shows a particular gesture that indicates a moving instruction in front of the PC, and the image is then captured and saved in JPG form. The prepared program will analyze the image and connect it with a predesigned number that maps each specific command. Meanwhile, the first part of the prepared program is the deep learning algorithm, which has been trained in the PC with many images and obtained satisfactory performance in the hand gesture recognition. A piece of code used for converting image recognition results into specific numbers is added after the recognition part of the deep learning algorithm. The correlation between the predesigned number and robotic arm movements is expressed in RAPID language to ensure that the robotic arm understands the meaning of each gesture. The number will be sent to the robotic arm in the base of interface protocol. The robotic arm receives the number and automatically compares the number to the preset movements and moves as requested.

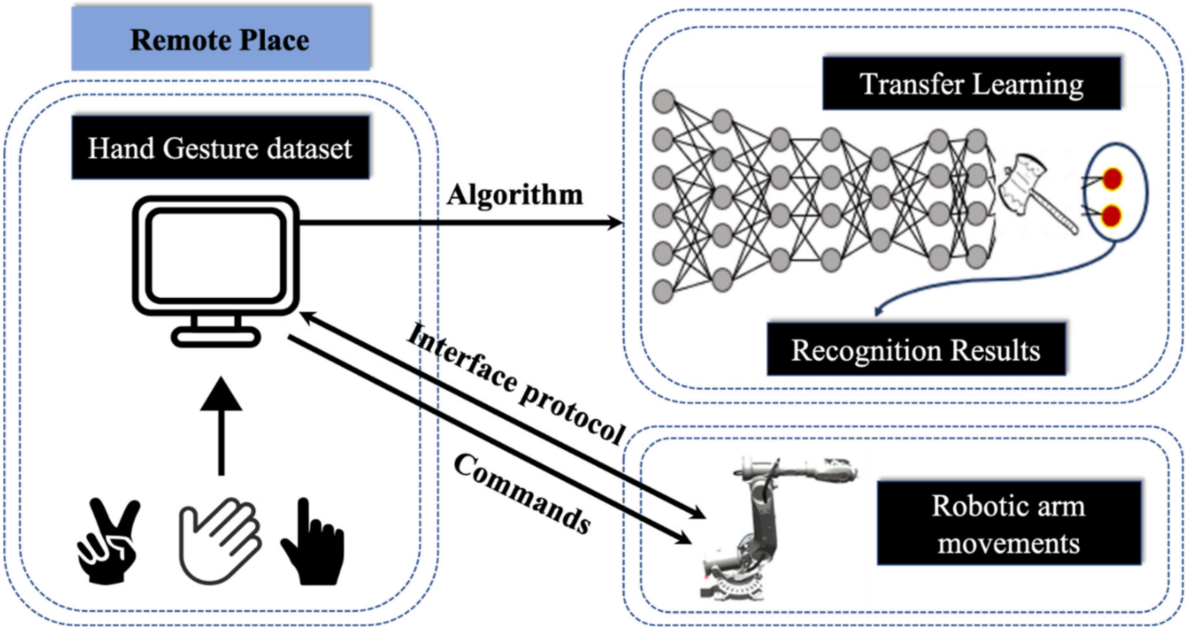
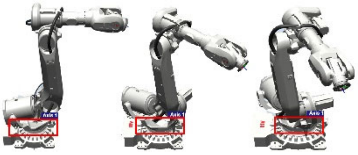
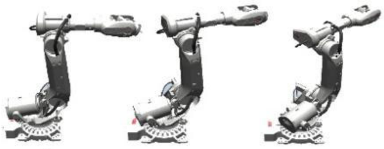
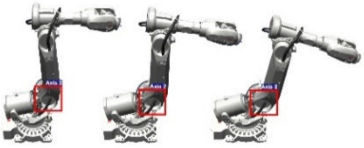
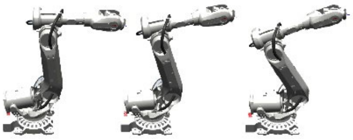
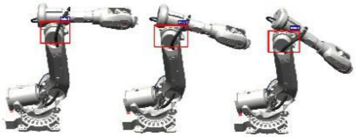
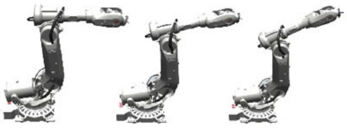
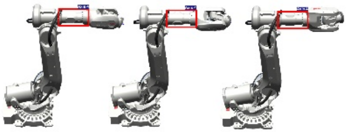
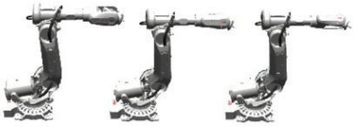
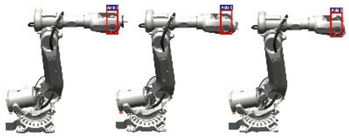
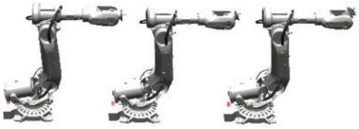
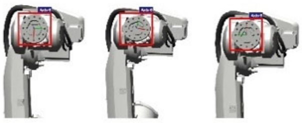
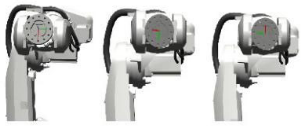


Figure 6. Hand gesture teleoperation system.

Hand gestures are implemented to control the robotic arm based on the teleoperation system. Thirteen types of hand gestures were designed for different movements of the robotic arm. The correlation between gestures and commands are listed in Table 2. The table also presents the action and core code of each command. Independent Continuous Movement (IndCMove) is used to change an axis to the independent mode and allow the axis to move continuously at a specific speed. An independent axis moves separately from other axes in the robot system. “MecUnit” denotes the mechanical unit. “Speed” is the axis speed in degrees/s. “\Ramp” is the deceleration from maximum performance when necessary (1%–100%, 100% = maximum performance). When IndCMove is executed, the specified axis starts to move at the programmed speed. The movement direction is specified as the sign of the speed argument. If “\Ramp” is programmed, then a reduction in acceleration/deceleration occurs. “Break” is programmed to stop the movement of the robotic arm.

Table 2. Correlation between gestures and commands of each action.

Gesture	Command	Action	Code
a	Axis 1: rotate clockwise		IndCMove MecUnit,1,Speed (positive)[\Ramp]
b	Axis 1: rotate counterclockwise		IndCMove MecUnit,1,Speed (negative)[\Ramp]
c	Axis 2: rotate clockwise		IndCMove MecUnit,2,Speed (positive)[\Ramp]
d	Axis 2: rotate counterclockwise		IndCMove MecUnit,2,Speed (negative)[\Ramp]
e	Axis 3: rotate clockwise		IndCMove MecUnit,3,Speed (positive)[\Ramp]
f	Axis 3: rotate counterclockwise		IndCMove MecUnit,3,Speed (negative)[\Ramp]
g	Axis 4: rotate clockwise		IndCMove MecUnit,4,Speed (positive)[\Ramp]
h	Axis 4: rotate counterclockwise		IndCMove MecUnit,4,Speed (negative)[\Ramp]
i	Axis 5: rotate clockwise		IndCMove MecUnit,5,Speed (positive)[\Ramp]
j	Axis 5: rotate counterclockwise		IndCMove MecUnit,5,Speed (negative)[\Ramp]
k	Axis 6: rotate clockwise		IndCMove MecUnit,6,Speed (positive)[\Ramp]
l	Axis 6: rotate counterclockwise		IndCMove MecUnit,6,Speed (negative)[\Ramp]
m	Stop		Break

4. Experiments

4.1. Experimental setup

The teleoperation experiment was carried out in the laboratory. An experiment based on the robotic arm is implemented to simulate the prefabrication assembly process. The robotic arm ABB IRB 6700-235/2.65 with six axes (six degrees of freedom) has a handling capacity and wrist torque of 235 kg and 1324 N·m, respectively. The schematic of the operating range is illustrated in Figure 7.

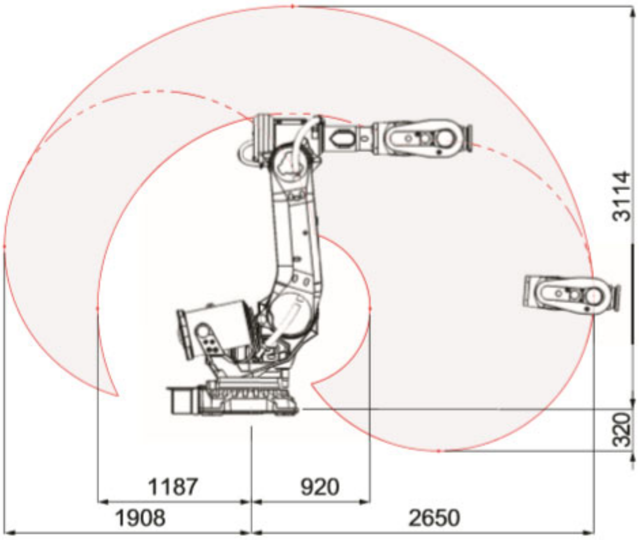


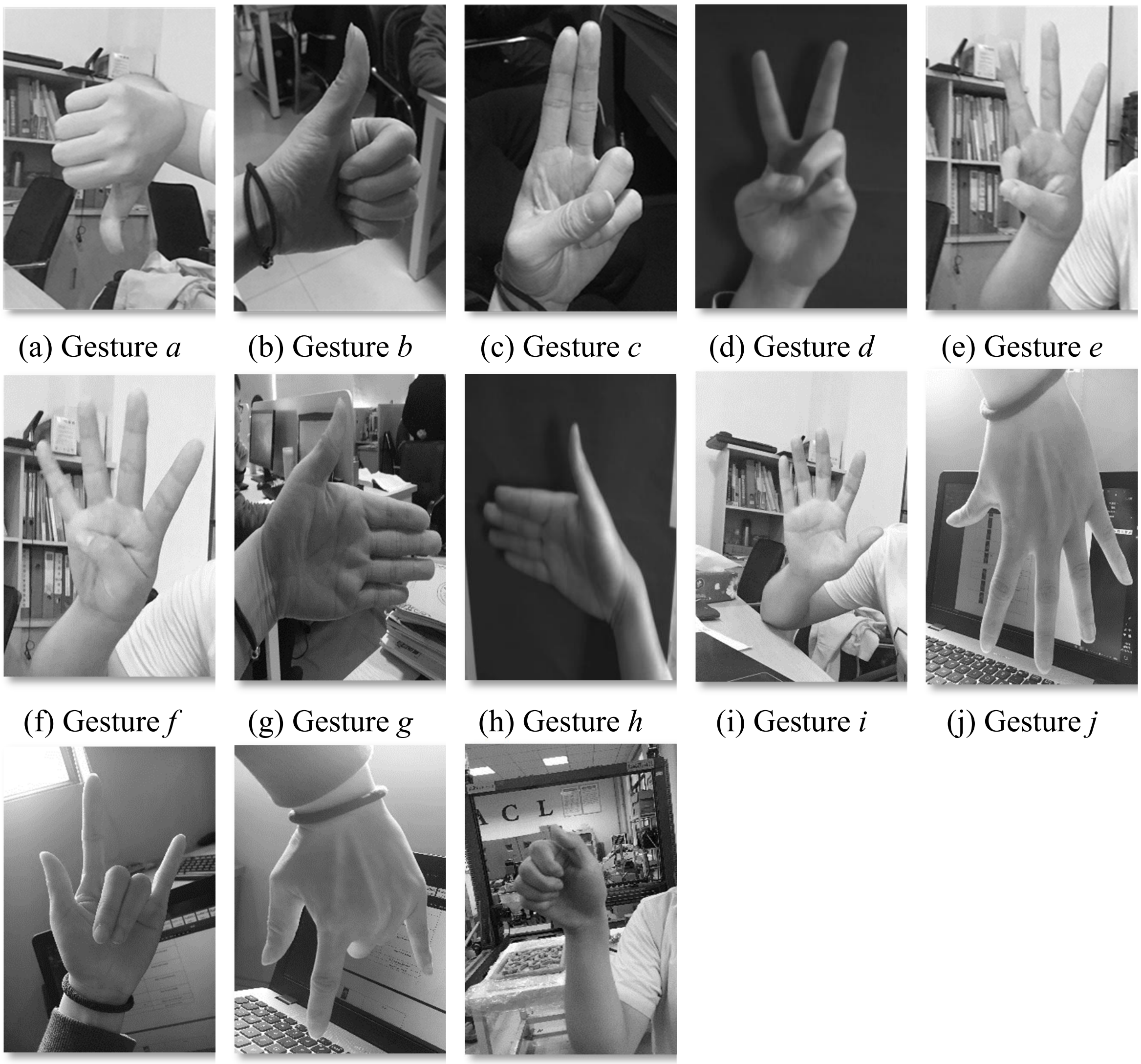
Figure 7. Schematic of the operating range of ABB IRB 6700-235/2.65.

The proposed method was implemented on Python with Intel Xeon CPU E3-1535M v6 and NVIDIA Quadro P5000. The proposed algorithm can be executed on any PC with Python installed. The program on the PC analyzes the hand gesture image on the basis of a transfer learning method. A piece of code is used to convert image recognition results into specific numbers. The number is then sent to the robotic arm using the interface protocol. The number is automatically compared with the preset rules, and the robotic arm moves according to the command.

4.2. Datasets and preprocessing

Hand gesture images were used in the training process Xception DCNN. The authors first designed 13 types of hand gestures that represent different movements of the robotic arm. Three students were then invited to perform the different gestures. The authors recorded gestures in several places to eliminate interference factors of the background. A place with good lighting is necessary during recording. The photographer attempted to maintain a distance of 50 cm from the operator while taking pictures to ensure the similar size of each hand gesture in the pictures.

A total of 2815 images were generated and then classified into 13 documents according to the gesture type. Several gesture samples are shown in Figure 8. Although the width of each image is randomly collected, Keras reshaped the input images into 299 × 299 pixels to ensure that cropping them to the same size is unnecessary. The dataset is divided into 60% training, 20% test, and 20% validation sets to improve accuracy. The number of pictures for each hand gesture in the training, test, and validation sets is presented in Table 3. The training set is used to fit the existing model, which is the model from ImageNet in this study. Multiple models are obtained using the classifier. The model with optimal performance is selected among the models after training and testing the training and validation sets with determined parameters. The test set is then used to evaluate the performance of the optimal model. Designing simple and easily recognized gestures, such as numbers, instead of complicated ones, such as the half-open palm, allows the robotic arm to operate precisely. Commands are designed on the basis of axial movements of the robotic arm.



(a) Gesture *a* (b) Gesture *b* (c) Gesture *c* (d) Gesture *d* (e) Gesture *e*

(f) Gesture *f* (g) Gesture *g* (h) Gesture *h* (i) Gesture *i* (j) Gesture *j*

(k) Gesture *k* (l) Gesture *l* (m) Gesture *m*

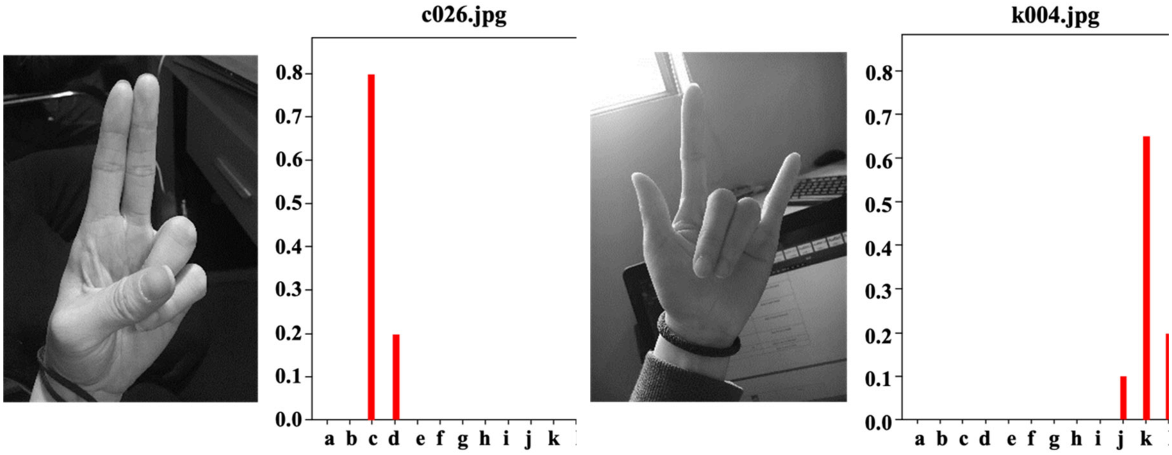
Figure 8. Designed hand gestures.

Table 3. Number of the pictures of each module for training, test, and validation sets.

	a	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*	*k*	*l*	*m*	**Total**
Training set	130	129	130	129	130	130	130	129	129	129	131	132	132	1,690
Test set	43	43	43	43	43	44	43	43	43	43	43	44	44	562
Validation set	43	43	43	43	43	44	43	43	43	43	44	44	44	563
Total	216	215	216	215	216	218	216	215	215	215	218	220	220	2,815

4.3. Results of hand gesture recognition

The output of the algorithm provides the probabilities for each class. We examine the probabilities and assume that the class corresponds to the highest probability for test purposes. Figure 9 shows some recognition results of hand gestures. Probabilities corresponding to the detection of each hand gesture type are presented in histograms. Table 4 lists the results of the two examples in Figure 9 in detail. The confusion matrix is presented in Table 5.



(a) (b)

Figure 9. Examples of recognition results of hand gestures.

Table 4. Results of the examples shown in Figure 9.

	a	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*	*k*	*l*	*m*	**Result**
a	0	0	**0.8**	0.2	0	0	0	0	0	0	0	0	0	*c*
b	0	0	0	0	0	0	0	0	0.05	0.1	**0.65**	0.2	0	*k*

Table 5. Confusion matrix.

Predicted Condition **True Condition**	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*	*k*	*l*	*m*
a	42	1	0	0	0	0	0	0	0	0	0	0	0
b	1	41	1	0	0	0	0	0	0	0	0	0	0

Table 5. *Cont.*

True Condition \ Predicted Condition	***a***	***b***	***c***	***d***	***e***	***f***	***g***	***h***	***i***	***j***	***k***	***l***	***m***
c	0	0	39	1	3	0	0	0	0	0	0	0	0
d	0	0	2	41	0	0	0	0	0	0	0	0	0
e	0	0	0	0	41	0	0	0	2	0	0	0	0
f	0	0	0	0	2	42	0	0	0	0	0	0	0
g	0	0	0	0	0	0	43	0	0	0	0	0	0
h	0	0	0	0	0	0	1	42	0	0	0	0	0
i	0	0	0	0	1	0	0	0	42	0	0	0	0
j	0	0	0	0	0	0	0	0	0	43	0	0	0
k	0	0	0	0	0	0	0	0	0	1	42	0	0
l	0	0	0	0	0	0	0	0	0	0	0	44	0
m	0	0	0	0	0	0	0	0	0	0	0	0	44

Precision and recall are used to evaluate the performance of the recognition algorithm of hand gestures. Precision is the proportion of examples classified as positive. Recall is the proportion of examples correctly classified and labeled as positive. Both metrics can measure the recognition capability as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{7}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{8}$$

Accuracy is the number of examples divided by the number of correctly classified examples. The error rate describes the proportion of misclassification. Both metrics are expressed as follows:

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

$$\text{Error} = \frac{\text{FP+FN}}{\text{TP+TN+FP+FN}} \tag{10}$$

F_β-score is the combination of precision and recall and expressed as follows:

$$\text{F}_\beta = \frac{(1+\beta^2)\cdot\text{Precision}\cdot\text{Recall}}{\beta^2\cdot\text{Precision}+\text{Recall}} \tag{11}$$

The value of β demonstrates different weights of recall and precision. If $\beta > 1$, then precision has more weight; if $\beta < 1$, then recall has more weight. We choose $\beta = 1$ in this study, and F_1-score is expressed as follows:

$$\text{F}_1 = \frac{2\cdot\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}} = \frac{2\text{TP}}{2\text{TP+FN+FP}} \tag{12}$$

The precision, recall, accuracy, error, and F_1-score of each class is listed in Table 6. The accuracy result shows satisfactory performance.

Table 6. Precision, recall, accuracy, error, and F_1-score of each class.

	Precision (%)	**Recall (%)**	**Accuracy (%)**	**Error (%)**	**F_1-score (%)**
Gesture *a*	97.67%	97.67%	99.64%	0.36%	97.67%
Gesture *b*	97.62%	95.35%	99.47%	0.53%	96.47%
Gesture *c*	92.86%	90.70%	98.75%	1.25%	91.76%
Gesture *d*	97.62%	95.35%	99.47%	0.53%	96.47%
Gesture *e*	87.23%	95.35%	98.58%	1.42%	91.11%
Gesture *f*	100.00%	95.45%	99.64%	0.36%	97.67%
Gesture *g*	97.73%	100.00%	99.82%	0.18%	98.85%
Gesture *h*	100.00%	97.67%	99.82%	0.18%	98.82%
Gesture *i*	95.45%	97.67%	99.47%	0.53%	96.55%
Gesture *j*	97.73%	100.00%	99.82%	0.18%	98.85%
Gesture *k*	100.00%	97.67%	99.82%	0.18%	98.82%
Gesture *l*	100.00%	100.00%	100.00%	0.00%	100.00%
Gesture *m*	100.00%	100.00%	100.00%	0.00%	100.00%
Average	97.22%	97.15%	99.56%	0.44%	97.16%

4.4. Teleoperation control of the robotic arm

This experiment aims to simulate the prefabrication assembly process using ABB IRB 6700 based on teleoperation control. The robotic arm must carry and transfer the dome from its original location to the target location to ensure that the prefabrication assembly process can be simulated. The end effector of the robotic arm was first changed to a hook to catch the dome, as shown in Figure 10. A student was assigned to be the chief operator to control the robotic arm and complete the experiment smoothly. The student first used hand gestures to adjust the location of the robotic arm for operating convenience. Considering the instability of the dome when the robotic arm lifted it, we used three ropes to secure the dome and asked two students to help lift the dome until it was stable. After the hook caught the dome, the operator continued to control the robotic arm with hand gestures and moved the prefabricated component to the designated place. During the assembly process, the angle of each axis was continuously adjusted until the robotic arm successfully reached the target location. The working range was tested many times to avoid unsafe factors during the prefabricated assembly process. The entire hand gesture controlling process continued until the structure was finished. The robot implemented a prefabrication assembly dome in this experiment. We tested the entire process many times, and Figure 11 shows one of the experiments we recorded. The overall process of the task is described as follows:

Step 1: The operator was required to stand approximately 3 m away from the robotic arm.

Step 2: After switching the controlling cabinet on, the operator performed Gestures *f* and *b* to test the moving flexibility of the robotic arm.

Step 3: The direction of the robotic arm was adjusted to an appropriate location using the defined hand gestures as preparation for the experiment.

Step 4: With the help of assistants, the operator performed corresponding gestures to control the robotic arm movements and connect the dome and the hook.

Step 5: The operator instructed the robot to return to its initial place at the end of the teleoperation control experiment.

A series of hand gestures was performed to complete the prefabrication assembly process. The general sequence of gestures is “6-13-2-13-3-13-6-13-5-13.” During the entire process, the robotic arm cannot reach the designated location. The pseudocode of the entire process is presented in Table 7. With the help of two assistants, one provides the gesture commands and one ensures the safety of the experimental operations. Due to the relatively large number of gesture types, the assistants needed to be trained in certain operations before the experiment. The entire experiment lasted about an hour to realize the robotic arm within its working range. The experiment realized the prefabricated assembly of the dome through a variety of moving paths, and initially verified the operation efficiency and execution accuracy of the experiment.

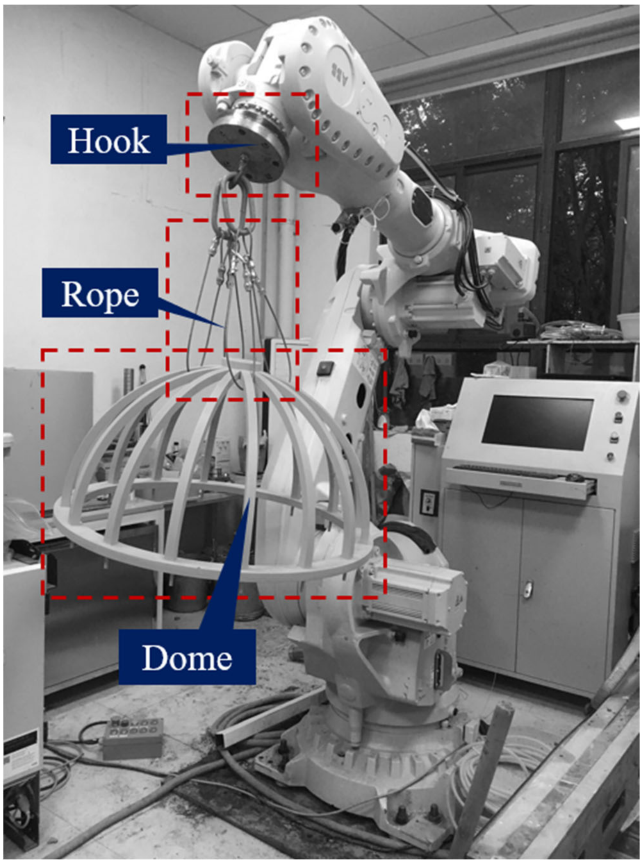


Figure 10. Equipment on the robotic arm.

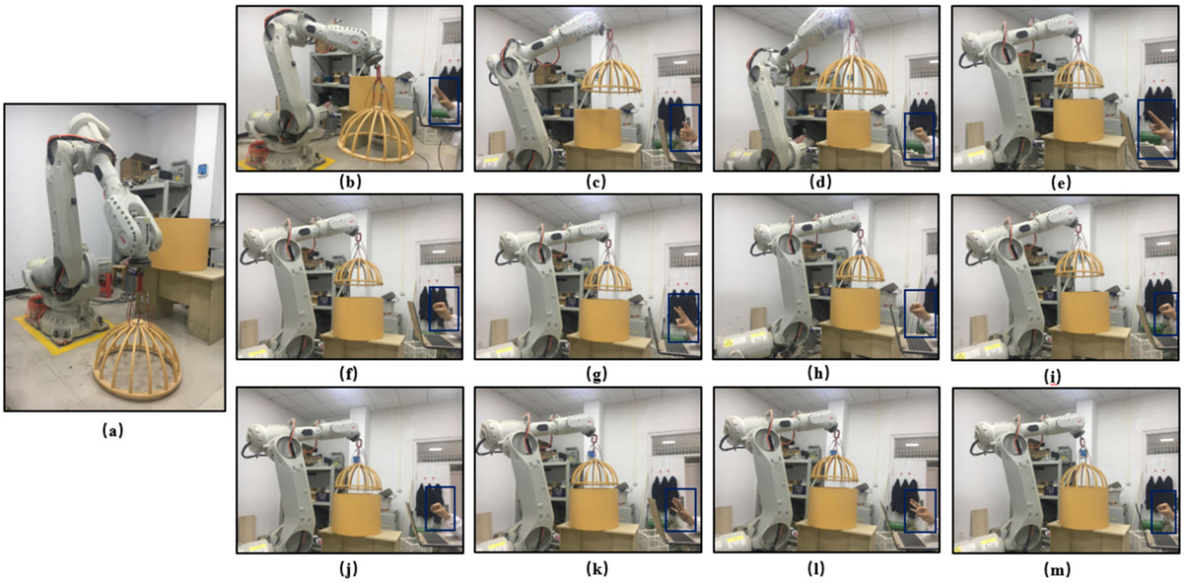


Figure 11. Prefabrication assembly process based on teleoperation control.

Table 7. Pseudocode of the prefabrication assembly process.

Prefabrication assembly process

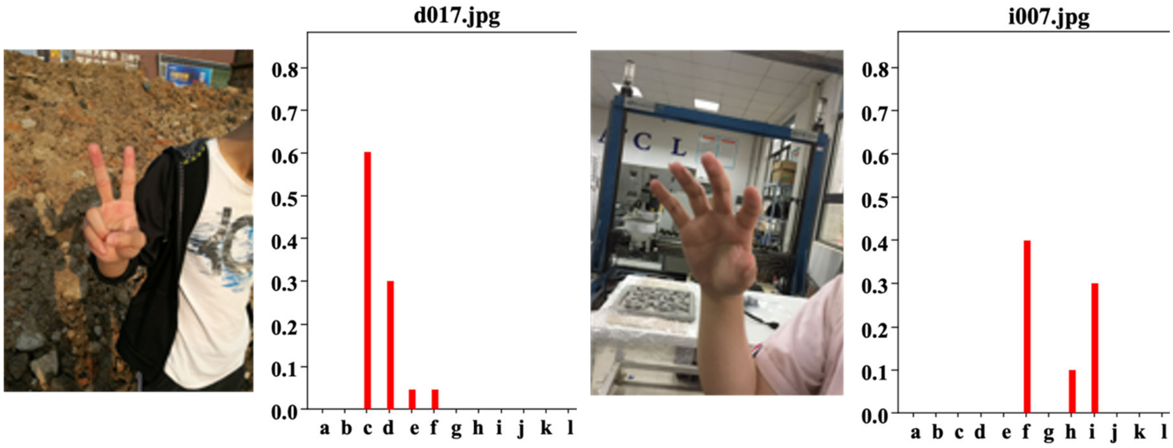
Input: number of robotic arm movement n_0=4, designated location T_0, T_1, ……, T_{n_0}, error threshold θ_r

Output: times_0, times_1, ……, times_{n_0}, prefabrication assembly product

```
for $i$=0; $i$<$n_0$; $i$++ do
    if $i$=0 then
        IndCMove $MecUnit$,3,-20\$Ramp$=50
    if $i$=1 then
        IndCMove $MecUnit$,1,-20\$Ramp$=50
    if $i$=2 then
        IndCMove $MecUnit$,2,20\$Ramp$=50
    if $i$=3 then
        IndCMove $MecUnit$,3,-20\$Ramp$=50
    if $i$=4 then
        IndCMove $MecUnit$,3,20\$Ramp$=50
    record location $S_i$
    $Error=\left\|\overrightarrow{T_iS_i}\right\|_2$
    while ($Error \geq \theta_r$) do
        for $times_i$=1; $times_i$<10; $times_i$++ do
            adjust the robotic arm
            record location $S_i$
            calculate $Error$
        end for
    end while
    pause (t)
end for
```

5. Discussion

Thirteen gestures were individually recognized in the test set. The accuracy of each gesture is 99.64%, 99.47%, 98.75%, 99.47%, 98.58%, 99.64%, 99.82%, 99.82%, 99.47%, 99.82%, 99.82%, 100.00%, 100.00% (Table 6). Gestures l and m were the most accurate, while Gesture c was the least accurate. Some gestures are confusing. Figure 12 shows a partial misprediction during recognition training. Probabilities corresponding to the detection of gestures are presented in Table 8.



(a) (b)

Figure 12. Mispredicted images.

Table 8. Probabilities corresponding to hand gesture detection in Figure 12.

	a	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*	*k*	*l*	*m*	**Prediction**	**Truth**
a	0	0	**0.6**	0.3	0.05	0.05	0	0	0	0	0	0	0	*c*	*d*
b	0	0	0.05	0	0	**0.4**	0.05	0.1	0.3	0.05	0	0	0.05	*f*	*i*

The final recognition accuracy in this study was 99.56%, which may not be the optimal result. Many factors, including some subjective and objective factors, affect accuracy. The results may improve when the quality of the dataset is enhanced or parameters are reset. Subjective factors typically refer to the quality and quantity of hand gesture images. In terms of the quality of hand gesture images, some gestures were not expressed properly during data collection and the recognition algorithm had difficulty in recognizing the real meaning of gestures. For example, index and middle fingers were excessively close together when they should have been separate or far apart when they should have been close together. The shooting direction also affects recognition results. As shown in Figure 13, the index finger nearly overlaps the middle finger. The clarity of the gesture may lead to different recognition results. Low accuracy may also be due to the unlabeled dataset we used and complex backgrounds of gestures that led to the inefficient recognition of the algorithm. In terms of the quantity of hand gesture images, 2815 images were applied to accomplish this experiment. The large number of images allows the network to discover additional details for each gesture, avoid mispredictions, and improve recognition results.

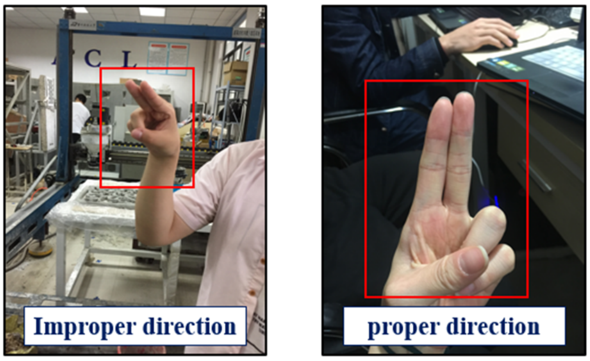


Figure 13. Gesture images taken from different directions.

Apart from subjective factors, objective factors will also affect recognition accuracy. The beginning of the algorithm is the transfer learning process shown in Figure 14. The transfer part demonstrated poor performance when the epoch was set to 5, and the curve in the graph first increased but decreased when it reached the peak. Although the direction of the curve may change when the epoch increases, the fine-tuning part will determine the final result, which is used to improve the recognition accuracy. However, nontrainable layers are uncertain; the accuracy increases with the increasing number of training layers and vice versa. The epoch at this time can ensure the recognition accuracy. Several attempts are necessary to obtain the appropriate value of parameters. After adjusting the number of epochs and nontrainable layers, the accuracy of the proposed method "Xception + transfer learning" (Xception + TL) reached 99.56%. Red and green lines denote the accuracy of validation and training sets, respectively.

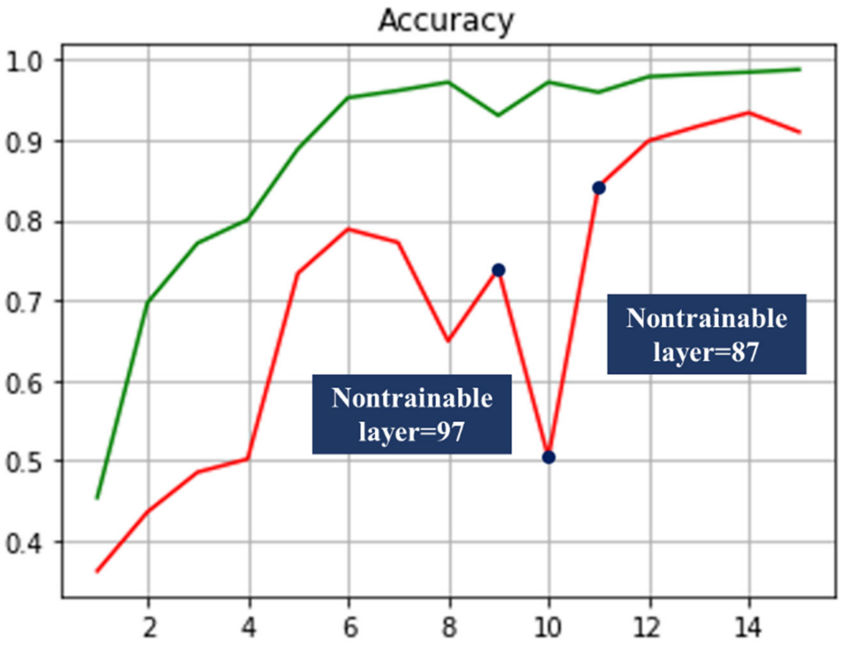


Figure 14. Transfer learning process.

During the experiment, there were cases where the operator could not control the movement accuracy of the robotic arm. This was mainly due to the fact that the robotic arm usually could not reach the specified position during the assembly process, and the commands had to be repeated at each step. When the robotic arm moved faster than expected at times, the operator had to execute the opposite command. Currently, the target point of the robotic arm is observed by the human eye. Due to the small size of the experimental setup, the prefabricated assembly movements are complicated. However, for lifting activities in construction scenarios, the execution accuracy is relatively less demanding than in laboratory scenarios. In order to improve the execution accuracy, it is necessary to obtain the feedback data through the sensors on the robotic arm and dynamically adjust the relevant operation posture to avoid human error. By improving the above limitations, the human-computer interaction-based robotic arm remote lifting technology will effectively improve the safety and construction accuracy at the construction site.

6. Conclusion

A human–machine interaction controlling system for teleoperation of robotic arms in prefabrication assembly is described in this study. DCNN with Xception architecture was used to recognize 13 different hand gesture types in the prefabrication assembly process and control the robotic arm ABB IRB 6700-235/2.65 in the laboratory setting. This system provides safety and convenience to operators in construction sites. An algorithm for controlling a robotic arm based on human–machine interaction and a prefabrication assembly method using a robotic arm were put forward in this study. The proposed system demonstrates satisfactory performance, and the developed algorithm can be used for teleoperation of robotic arms in prefabrication assembly to provide feasible support in prefabricated construction. The developed method can also be applied in hoisting works.

The limitations of this study are presented as follows. The initial design in this study was to implement a teleoperation control system for prefabrication assembly in construction sites. However, the complex environment of construction sites poses a challenge because the robotic arm is required to implement assembly tasks from a remote place. The operating room is set within the construction site at a proper distance from the robotic arm to ensure that the signal is strong. The signal intensity of the teleoperation system must be discussed in future investigations. Complex site environments may also affect the accuracy of gesture-based remote control. As an inevitable factor, shaking of the robotic arm in construction sites must be considered. The object being assembled is impossible to remain motionless during the operation of the robotic arm. The movement of the object to a certain direction can bring remarkable challenges to the prefabrication assembly. Research on the controlling system for teleoperation of robotic arms in prefabrication assembly continues despite the many difficulties and challenges. This study attempted to implement the teleoperation control of a robotic arm to supplement the prefabrication assembly in fabricated construction projects in complex and extreme environments, such as the moon, Mars, and polar regions. In addition, the method has some limitations, such as the operating range is restricted due to the constraints of the test site and equipment. In addition, the acquisition of photo material is prone to receive interference from strong sunlight.

Future investigations will include but not be limited to improving the dataset, information transmission, and the algorithm used in this study. A total of 2815 images were collected for training and testing the recognition algorithm. Although transfer learning requires a small dataset, the images are still insufficient. Therefore, additional images must be collected to improve accuracy. Meanwhile, the image quality is also an important factor that influences the result. The experiment implemented in this study was conducted in the laboratory, and communication between the robotic arm and the operator through a wire limits the operating distance to some extent. 5G can be used in information transmission during teleoperation. 5G communication can increase the transmission speed and shorten the reaction time when the robotic arm is commanded to move. Some hazardous environments have remarkably longer distances of teleoperation than the laboratory. The use of 5G technology will simplify the control of robotic arms and the use of robots will make the presence of humans in dangerous construction sites unnecessary. The “Xception + TF” method is implemented in this study. The robustness of the algorithm can be strengthened in future investigations. The algorithm can be improved to strengthen its ability to resist disturbance because hand gestures can sometimes be inaccurate.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (2021YFF0500300, 2023YFB3711300) and the Strategic Research and Consulting Project of the Chinese Academy of Engineering (2023-XZ-90, 2023-JB-09-10).

Conflicts of interests

We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Authors’ contribution

Cheng Zhou: Formal analysis and Writing; Rui Chen: Validation and Writing; Przemyslaw Sekula: Methodology; Bin Tang: Data analysis and Writing; Yu Qu: Validation.

References

[1] Barlow J, Childerhouse P, Gann D, Hong-Minh S, Naim M, *et al*. Choice and delivery in housebuilding: lessons from Japan for UK housebuilders. *Build. Res. Inf.* 2003, 31(2):134–145.

[2] Li Z, Shen GQ, Alshawi M. Measuring the impact of prefabrication on construction waste reduction: An empirical study in China. *Resour. Conserv. Recycl*. 2014, 91:27–39.

[3] Zhang W, Lee MW, Jaillon L, Poon CS. The hindrance to using prefabrication in Hong Kong's building industry. *J. Clean. Prod.* 2018, 204:70–81.

[4] Sacks R, Eastman CM, Lee G. Process model perspectives on management and engineering procedures in the precast/prestressed concrete industry. *J. Constr. Eng. Manag.* 2004, 130(2):206–215.

[5] Jaillon L, Poon CS, Chiang YH. Quantifying the waste reduction potential of using prefabrication in building construction in Hong Kong. *Waste Manag.* 2009, 29(1):309– 320.

[6] Mollet N, Chellali R. Virtual and augmented reality with head-tracking for efficient teleoperation of groups of robots. In *Proceedings of 2008 International Conference on Cyberworlds*, Hangzhou, China, September 22–24, 2008, pp. 102–108.

[7] Fang B, Sun F, Liu H, Guo D. A novel data glove using inertial and magnetic sensors for motion capture and robotic arm-hand teleoperation. *Ind. Robot*. 2017, 44(2):155–165.

[8] Fujii T, Lee JH, Okamoto S. Gesture recognition system for human-robot interaction and its application to robotic service task. In *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS 2014)*, Hong Kong, China, October 20–22, 2014, pp. 63–68.

[9] Yin X, Xie M. Finger identification and hand posture recognition for human–robot interaction. *Image Vis. Comput*. 2007, 25(8):1291–300.

[10] Wen R, Tay WL, Nguyen BP, Chng CB, Chui CK. Hand gesture guided robot-assisted surgery based on a direct augmented reality interface. *Comput. Methods Programs Biomed.* 2014, 116(2):68–80.

[11] Noda K, Yamaguchi Y, Nakadai K, Okuno HG, Ogata T. Audio-visual speech recognition using deep learning. *Appl. Intell.* 2015, 42:722–737.

[12] Budiharto W, Gunawan AA. Development of coffee maker service robot using speech and face recognition systems using POMDP. In *Proceedings of First International Workshop on Pattern Recognition*, Tokyo, Japan, May 11–13, 2016.

[13] Shan X, Yang EH, Zhou J, Chang VW. Human-building interaction under various indoor temperatures through neural-signal electroencephalogram (EEG) methods. *Build. Environ.* 2018, 129:46–53.

[14] Hong J, Song S, Kang H, Choi J, Hong T, *et al.* Influence of visual environments on struck-by hazards for construction equipment operators through virtual eye-tracking. *Autom. Constr*. 2024, 161:105341.

[15] Wang Y, Yang C, Wu X, Xu S, Li H. Kinect based dynamic hand gesture recognition algorithm research. In *Proceedings of 2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics*, Nanchang, China, August 26–27, 2012, pp. 274–279.

[16] Pedersoli F, Benini S, Adami N, Leonardi R. XKin: an open source framework for hand pose and gesture recognition using kinect. *Vis. Comput.* 2014, 30:1107–1122.

[17] Marin G, Dominio F, Zanuttigh P. Hand gesture recognition with leap motion and kinect devices. In *Proceedings of 2014 IEEE International Conference on Image Processing (ICIP)*, Paris, France, October 27–30, 2014, pp. 1565–1569.

[18] Atitallah AB, Said Y, Atitallah MA, Albekairi M, Kaaniche K, *et al.* An effective obstacle detection system using deep learning advantages to aid blind and visually impaired navigation. *Ain Shams Eng. J.* 2024, 15(2):102387.

[19] Fang Q, Li H, Luo X, Ding L, Rose TM, *et al.* A deep learning-based method for detecting non-certified work on construction sites. *Adv. Eng. Informatics.* 2018, 35:56– 68.

[20] Zhong B, Pan X, Love PE, Ding L, Fang W. Deep learning and network analysis: Classifying and visualizing accident narratives in construction. *Autom. Constr*. 2020, 113:103089.

[21] Bar Y, Diamant I, Wolf L, Lieberman S, Konen E, *et al*. Chest pathology detection using deep learning with non-medical training. In *Proceedings of 2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, New York, USA, April 16–19, 2015, pp. 294–297.

[22] Gopalakrishnan K, Khaitan SK, Choudhary A, Agrawal A. Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. *Constr. Build. Mater.* 2017, 157:322–330.

[23] Szegedy C, Liu W, Jia Y, Sermanet P, Reed SE, *et al.* Going Deeper with Convolution. IEEE Computer Society. In *Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 7–12, 2015, pp. 1–9.

[24] Lytle AM, Saidi KS, Bostelman RV, Stone WC, Scott NA. Adapting a teleoperated device for autonomous control using three-dimensional positioning sensors: experiences with the NIST RoboCrane. *Autom. Constr*. 2004, 13(1):101–118.

[25] Doggett W. Robotic assembly of truss structures for space systems and future research plans. In *Proceedings of IEEE Aerospace Conference*, Big Sky, MT, USA, March 9–16, 2002, pp. 3589–3598.

[26] Gambao E, Balaguer C, Gebhart F. Robot assembly system for computer-integrated construction. *Autom. Constr*. 2000, 9(5-6):479–487.

[27] Wing R, Atkin B. FutureHome-a prototype for factory housing. In *Proceedings of the 19th International Symposium on Robotics and Automation in Construction*, Gaithersburg, Maryland, September 23–25, 2002, pp. 173–179.

[28] Dörfler K, Sandy T, Giftthaler M, Gramazio F, Kohler M, *et al*. Mobile robotic brickwork: automation of a discrete robotic fabrication process using an autonomous mobile robot. In *Robotic Fabrication in Architecture, Art and Design 2016.* Cham: Springer, 2016, pp. 204–217.

[29] Neto P, Pires JN, Moreira AP. Accelerometer-based control of an industrial robotic arm. In *Proceedings of RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication*, Toyama, Japan, September 27–October 2, 2009, pp. 1192–1197.

[30] Wang Z, Chen D, Xiao P. Design of a voice control 6DoF grasping robotic arm based on ultrasonic sensor, computer vision and Alexa voice assistance. In *Proceedings of 2019 10th International Conference on Information Technology in Medicine and Education (ITME)*, Qingdao, China, August 23–25, 2019, pp. 649–654.

[31] Poulos M, Rangoussi M, Chrissikopoulos V, Evangelou A. Person identification based on parametric processing of the EEG. In *Proceedings of ICECS '99. 6th IEEE International Conference on Electronics, Circuits and Systems*, Pafos, Cyprus, September 5–8, 1999, pp. 283–286.

[32] Yazdani A, Roodaki A, Rezatofighi SH, Misaghian K, Setarehdan SK. Fisher linear discriminant based person identification using visual evoked potentials. In *Proceedings of 2008 9th International Conference on Signal Processing*, Beijing, China, October 26–29, 2008, pp. 1677–1680.

[33] Palaniappan R. Electroencephalogram signals from imagined activities: A novel biometric identifier for a small population. In *Proceedings of Intelligent Data Engineering and Automated Learning*, Burgos, Spain, September 20–23, 2006, pp. 604– 611.

[34] Okogbaa OG, Shell RL, Filipusic D. On the investigation of the neurophysiological correlates of knowledge worker mental fatigue using the EEG signal. *Appl. Ergon.* 1994, 25(6):355–365.

[35] Chen J, Song X, Lin Z. Revealing the "Invisible Gorilla" in construction: Estimating construction safety through mental workload assessment. *Autom. Constr*. 2016, 63:173– 183.

[36] Jeelani I, Han K, Albert A. Automating and scaling personalized safety training using eye-tracking data. *Autom. Constr*. 2018, 93:63–77.

[37] Bretzner L, Laptev I, Lindeberg T. Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. In *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, Washington, D.C., USA, May 20–21, 2002, pp. 423–428.

[38] Li Y. Hand gesture recognition using Kinect. In *Proceedings of 2012 IEEE International Conference on Computer Science and Automation Engineering*, Beijing, China, June 22–24, 2012, pp. 196–199.

[39] Ren Z, Yuan J, Meng J, Zhang Z. Robust part-based hand gesture recognition using kinect sensor. *IEEE Trans. Multimed.* 2013, 15(5):1110–1120.

[40] John V, Boyali A, Mita S, Imanishi M, Sanma N. Deep learning-based fast hand gesture recognition using representative frames. In *Proceedings of 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, Gold Coast, Australia, November 30–December 2, 2016, pp. 1–8.

[41] Wang W, Yang J, Xiao J, Li S, Zhou D. Face recognition based on deep learning. In *Proceedings of International Conference on Human Centered Computing (HCC 2014)*, Phnom Penh, Cambodia, November 27–29, 2014, pp. 812–820.

[42] Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, *et al*. Natural language processing (almost) from scratch. *J. Mach. Learn. Res*. 2011, 12:2493–2537.

[43] Kruger N, Janssen P, Kalkan S, Lappe M, Leonardis A, *et al*. Deep hierarchies in the primate visual cortex: What can we learn for computer vision? *IEEE Trans. Pattern Anal. Mach. Intell*. 2012, 35(8):1847–1871.

[44] Sanchez-Riera J, Hsiao YS, Lim T, Hua KL, Cheng WH. A robust tracking algorithm for 3d hand gesture with rapid hand motion through deep learning. In *Proceedings of IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, Sichuan, China, July 14–18, 2014, pp. 1–6.

[45] Oyedotun OK, Khashman A. Deep learning in vision-based static hand gesture recognition. *Neural Comput. Appl*. 2017, 28(12):3941–3951.

[46] Kolar Z, Chen H, Luo X. Transfer learning and deep convolutional neural networks for safety guardrail detection in 2D images. *Autom. Constr*. 2018, 89:58–70.