# An industrial intelligent grasping system based on convolutional neural network

*Jiang Daqi, Wang Hong, Zhou Bin and Wei Chunfeng*
School of Mechanical Engineering and Automation, Northeastern University, Shenyang, China

**Abstract**

**Purpose** – This paper aims to save time spent on manufacturing the data set and make the intelligent grasping system easy to deploy into a practical industrial environment. Due to the accuracy and robustness of the convolutional neural network, the success rate of the gripping operation reached a high level.

**Design/Methodology/Approach** – The proposed system comprises two diverse kinds of convolutional neuron network (CNN) algorithms used in different stages and a binocular eye-in-hand system on the end effector, which detects the position and orientation of workpiece. Both algorithms are trained by the data sets containing images and annotations, which are generated automatically by the proposed method.

**Findings** – The approach can be successfully applied to standard position-controlled robots common in the industry. The algorithm performs excellently in terms of elapsed time. Procession of a $256 \times 256$ image spends less than 0.1 s without relying on high-performance GPUs. The approach is validated in a series of grasping experiments. This method frees workers from monotonous work and improves factory productivity.

**Originality/Value** – The authors propose a novel neural network whose performance is tested to be excellent. Moreover, experimental results demonstrate that the proposed second level is extraordinary robust subject to environmental variations. The data sets are generated automatically which saves time spent on manufacturing the data set and makes the intelligent grasping system easy to deploy into a practical industrial environment. Due to the accuracy and robustness of the convolutional neural network, the success rate of the gripping operation reached a high level.

**Keywords** Deep learning, Convolutional neural network, Vision positioning, Eye-in-hand, Grasp detection, Image processing

**Paper type** Research paper

## 1. Introduction

With the development of digital manufacturing technology, the component assembly production based on industrial robots becomes increasingly efficient (Yang *et al.*, 2016). For assembly production, industrial robots free workers from monotonous, duplication work. Nevertheless, for most industrial robots, if there is a small-scale change, the industry must redesign work process and programming, which will greatly affect the economic benefits of the plant. Therefore, a smart system with forceful adaptability is particularly essential (Qiao *et al.*, 2014).

The object detection algorithm is the core of the intelligent system (Hua *et al.*, 2019). The traditional object detection algorithms are designed to reduce the amount of calculation as much as possible on the premise of manually extracting rich feature points, thereby improving the calculation efficiency and the recognition speed. However, although manual feature extraction is easy to understand and straightforward and intuitive, it cannot cope with the identification of a large number of categories. When the target recognizer changes, it needs to perform complex feature design and extraction again. The target detection algorithm based on deep learning uses neural networks to extract the bottom- and high-level features of the image, which not only can extract more abundant and expressive features but also do not require manual participation in feature extraction and can also achieve end-to-end training and prediction.

However, despite the promising advantages revealed above, crucial issues might arise. Almost every deep learning algorithms require a considerable number of training data sets to achieve acceptable performance. This has become one of the most important reasons holding up the deployment of deep learning algorithms in industrial environments. Insufficient training samples are a colossal challenge for the learning of intelligent agents. This is due to the fact that the current deep neural network has not yet reached the powerful knowledge transfer and logical reasoning abilities as humans. This problem is the famous Moravec paradox (Vadim, 2013), which has received unprecedented attention in the new wave of artificial intelligence.

Humans glance at an image and instantaneously remember the characteristics of the objects in the image (Enrique *et al.*, 2018). For a long time, albeit the background environment changed significantly, humans could still immediately identify the object when similar objects appeared. This ability to quickly identify surrounding objects allows humans to quickly

The current issue and full text archive of this journal is available on Emerald Insight at: **https://www.emerald.com/insight/0144-5154.htm**

Assembly Automation
42/2 (2022) 236–247

[DOI 10.1108/AA-03-2021-0036]

The authors gratefully acknowledge the financial support from the National Key R&D Program of China (2021YFF0306405).

Received 29 March 2021
Revised 17 May 2021
1 July 2021
16 August 2021
Accepted 20 December 2021

understand their surroundings during childhood. Accurate algorithms for object detection would allow robots system convenient to recognize the specified object.

In this paper, these problems have been considered and a novel intelligent system for grasping specific workpieces under a new industrial environment is proposed. Specifically, a system with two disparate types of visual equipment was designed. In this system, the first-level visual equipment placed in an uplifted position captures a global image including the whole area where workpieces may appear. The collected images are solely used to calculate the category and location information and do not need to judge the size and contour. After acquiring this information, the first-level visual equipment will be kicked off. Based on the position information, the robot arm will reach a specified height above the workpiece with a relatively lower spatial precision and hold on. Then the second-level vision system will start to work. The vision system is a binocular eye-in-hand visual equipment on the end effector that detects the multifarious information ranging from the position, distance to contour of the workpiece. The robotic arm will work out a reasonable trajectory and the specific orientation of the grasping based on this information. For these two sets of vision equipment, two diverse kinds of convolutional neural network algorithms are used in different stages. The former algorithm aims at object detection which detects the category and location information. Inversely, the later algorithm is based on semantic segmentation neural network. In this paper, a binocular eye-in-hand visual equipment on the end effector is adopted.

The deep learning algorithms reinforce the robustness of the visual system, endowing the system a capability to adapt to significantly various illumination or viewpoint (Wang, 2021). The robustness was confirmed by the experiments where the manipulator completed the mission of grasping workpiece with extremely high success rates under various environmental circumstances. Furthermore, the method does not need to manufacture data sets. The data sets containing images and annotations are generated automatically by the proposed method. Hence, merely a few basic photos of the background and workpiece are sufficient for mission completed, and the rest of the procedure containing generating training data sets and training convolutional neural network could all be automatically executed. A test on the automobile component assembly is implemented; the results denote that the intelligent grasping system has high efficiency and excellent applicability.

In the remainder of this paper, Section 2 summarizes related prior work. Section 3 presents the detailed descriptions of the proposed method. Then, Section 4 presents the experimental process and results. Finally, Section 5 concludes the paper.

## 2. Related work

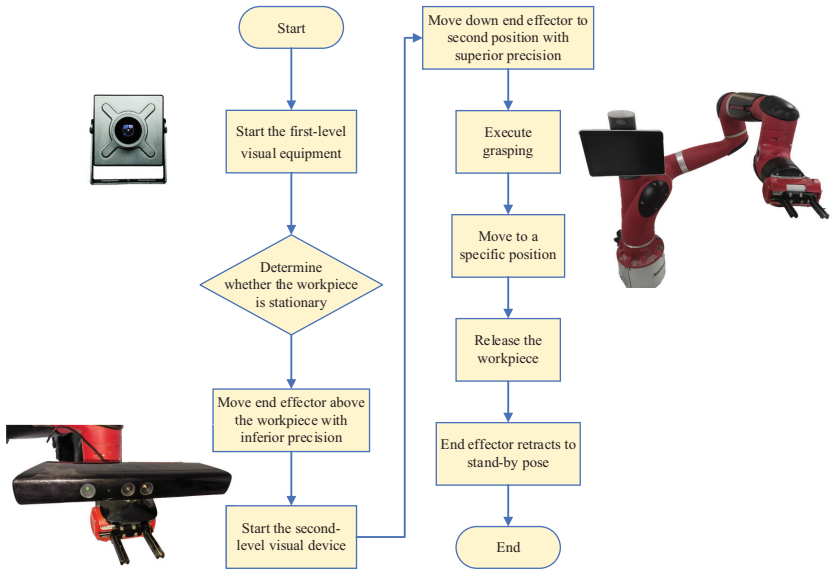
In conventional image processing methods, manual feature extraction is the dominating means. Nevertheless, although manual feature extraction is easy to understand and simple and intuitive, it cannot cope with the identification of a large number of categories (Chen *et al.*, 2019; Troniak *et al.*, 2013; Oron *et al.*, 2018; Ouyang *et al.*, 2012). In recent years, owing to its excellent performance, deep learning has been favored by major research institutions (Wang, 2021). The most widely used deep learning model in the field of image processing is convolutional neural networks. Convolutional neural networks need to learn enormous parameters. Early on, there was not enough training data, and sufficient computing power and overfitting was frequent to occur. Nowadays, due to the dynamically rapid development of high-performance GPUs and the release of large-scale data sets such as ImageNet, convolutional neural networks have produced far more precision than conventional algorithms (Krizhevsky *et al.*, 2012).

In 2006, Hinton *et al.* proposed a layer-by-layer training method to effectively alleviate the training complexity and successfully solve the problem of difficult training of neural networks (Hinton and Salakhutdinov, 2006). At the same time, with the development of GPU technology, neural networks regained the attention of the academic community and industry. In 2012, AlexNet proposed by Alex Krizhevsky *et al.* won the champion in the ImageNet image recognition competition, and its error rate was reduced by about 10% compared to the second place (Krizhevsky *et al.*, 2012). The AlexNet network uses the rectified linear unit (ReLU) (Glorot *et al.*, 2011) function to replace the traditional Sigmoid function as a new activation function, alleviating the problem of gradient dispersion and using the Dropout regularization technology to improve the robustness of the algorithm and prevent overfitting (Srivastava *et al.*, 2014). In 2014, Simonyan *et al.* proposed the VGG-Net structure which was plain and effective (Simonyan and Zisserman, 2014). The first few layers used 3*3 convolution kernels instead of large convolution kernels, making the receptive fields of each layer the same and increase the depth of the network to obtain more nonlinear expressions and reduce the size of the feature map through maximum pooling. The last three layers are two fully connected layers and a softmax layer. In the then ImageNet competition, VGG-Net achieved excellent results. In 2015, Ronneberger *et al.* designed the U-Net, which is U-shaped neural network architecture (Ronneberger *et al.*, 2015). The U-Net uses skip connections to calculate multiscale information. In 2017, Huang *et al.* proposed a novel network structure, DenseNet (dense convolutional network), that optimized the network structure (Huang *et al.*, 2017). The number of feature maps output by each convolutional layer was very limited, ensuring fewer parameters can converge apace. Group Normalization proposed by Wu and He (2018) in 2018 replaced Batch Normalization, making the normalization operation no longer affected by the batch size, reducing the dependence on high-performance GPUs.

## 3. Proposed system description

The procedure of intelligent grasping system proposed in this paper is illustrated in Figure 1. In this system, the first-level visual equipment placed in uplifted position captures a global image including the whole area where workpieces may appear. The collected images are solely used to calculate the category and location information based on the object detection convolutional neural network. When it is detected that the position of the object has moved less than 12 mm in a continuous 1 s, the system determines that the workpiece keeps motionless. This scheme is to take into account environmental fluctuations and improve the

**Figure 1** Illustration of the proposed procedure of intelligent grasping system




robustness of the system. After determining that the object is stationary, the robotic arm will work out a reasonable trajectory and move to the specified height above the corresponding position with a relatively lower spatial precision. Then the first-level visual equipment will be kicked off and the second-level visual system will come into operation. The second-level visual system is a binocular eye-in-hand visual equipment on the end effector whose assignment is to detect the multifarious information ranging from the position, distance to contour of the workpiece. Based on this information, the calculation unit will calculate the relative position of the workpiece and the end effector. The robotic arm will make fine adjustments accordingly and recalculate the relative position. This process is repeated until the center of workpiece is located directly below the center of the two lenses within the allowable error band. The end effector will execute grasping and move to a preset designated location. Eventually, the end effector releases the object and then comes back to the idle position.

The coordinate detection and semantic segmentation convolutional neural network are respectively deployed on two levels of visual equipment. On account of all task of the first-level vision equipment is rough positioning, the size and contour information of the workpiece and deployed a monocular camera is ignored. The spatial tolerance of first-level system is within ±8 mm. The precision ensures that the mechanical arm can move to the vicinity of the workpiece. The target of the second-level vision system is to accurately detect the pose of the workpiece. Hence, a binocular eye-in-hand vision system where each camera detects the contour of workpiece in its own scene is deployed. The precise position of the workpiece relative to the end effector is then calculated based on the difference between the contours of the two cameras. What follows in the paper is the detailed description of the two diverse kinds of convolutional neural network algorithms and the method of automatically generating data sets containing images and annotations.

Image processing domain has distinguishing tasks including image classification, object detection and scene understanding (also called semantic segmentation). As shown in Figure 2, convolutional neuron network (CNN) performs wonderfully by designing multifarious network structures and output formats in each task. The method presented in this paper is based on object detection used in the first-level visual equipment and semantic segmentation used in the binocular eye-in-hand visual equipment.

### 3.1 Generation of training images data sets

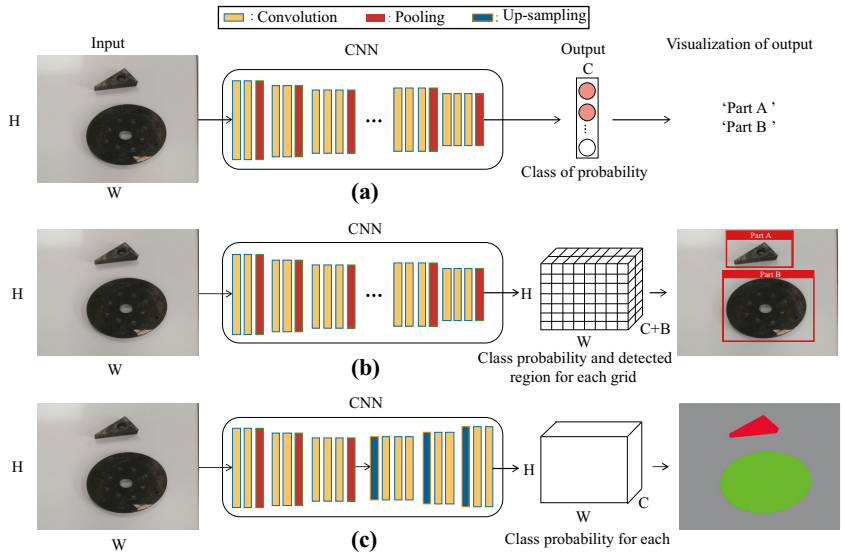
As the two-level algorithms are respectively based on object detection and semantic segmentation convolutional neural network, the data sets contain two series. The annotations of the first series are the position information of the workpiece. Contrastively, the annotations of the second series are the contour information.

In the first series of data sets, the pictures captured by the first-level visual equipment are selected as the background images. Then, take an image for every type of workpiece and extract the region of interest that only includes workpiece and excludes background parts. The region of interest is regarded as object images. Next, resize the object images with a calculated coefficient and use them to cover the specified position of background images. The coefficient takes into account the effects of background, object size and lens distortion. The effects of background and object size can be estimated based on the size ratio of the object to the background. The effects of lens distortion can be calculated by the radial distortion formula.

Let $\boldsymbol{\eta} = [\eta_x, \eta_y]$ be the resizing coefficient where $\eta_x$ is the $x$-axis scale and $\eta_y$ is $y$-axis. Let $w_{obj}$, $h_{obj}$ be the width and height of the workpiece. Similarly, let $w_{back}$, $h_{back}$ be the width and height of the background, respectively. The image resolutions of the workpiece and the background are $m_{obj} \times n_{obj}$ and $m_{back} \times n_{back}$, respectively. $\boldsymbol{\eta}$ can be calculated by the following equation:

$$\eta_x = \frac{w_{obj}}{w_{back}} \cdot \frac{h_{back}}{h_{obj}} \cdot \mu_x \tag{1}$$

**Figure 2** Application of CNN in different image recognition task




**Notes:** (a) Image classification; (b) object detection; (c) semantic segmentation

$$\eta_y = \frac{h_{obj}}{h_{back}} \cdot \frac{n_{back}}{n_{obj}} \cdot \mu_y \tag{2}$$

where $\mu_x$ and $\mu_y$ are, respectively, influence coefficient due to radial distortion in horizontal and vertical directions. The calculation process of $\mu_x$ and $\mu_y$ is derived as follows.

Let $(x, y)$ be the normalized pixel image coordinates in image without distortion and $(\tilde{x}, \tilde{y})$ the corresponding normalized coordinates in real observed image. We have (Zhang, 2000):

$$\tilde{x} = x + x\left[k_1\left(x^2 + y^2\right) + k_2\left(x^2 + y^2\right)^2\right] \tag{3}$$

$$\tilde{y} = y + y\left[k_1\left(x^2 + y^2\right) + k_2\left(x^2 + y^2\right)^2\right] \tag{4}$$

where $k_1$ and $k_2$ are the radial distortion coefficients of the lens. Print a black and white checkerboard pattern and paste it on a planar board. Then, place it in front of the camera and take ten images from different orientations by moving the board. Each image contains $10 \times 8$ corners on the board; hence, there are 800 sets of corner coordinates in total. Each set comprise distortion-free coordinates $(x, y)$ and distorted coordinates $(\tilde{x}, \tilde{y})$. Substitute these coordinates into the formula, and put all equations together and acquire altogether 1600 equations. Convert them into a matrix form, which is $\mathbf{Dk} = \mathbf{d}$, where $\mathbf{k} = [k_1, k_2]^T$. We have the linear least-squares solution:

$$\mathbf{k} = \left(\mathbf{D}^T\mathbf{D}\right)^{-1}\mathbf{D}^T\mathbf{d} \tag{5}$$

After $k_1$ and $k_2$ is solved, the functional relationship between distortion-free coordinates $(x, y)$ and distorted coordinates $(\tilde{x}, \tilde{y})$ is determined. Use $\tilde{x} = f(x, y)$ and $\tilde{y} = g(x, y)$ to denote the functional relationship. To derive the expression of the resizing coefficient $\boldsymbol{\eta}$, assume that there are three points on the plane: $(x, y)$, $(x + \Delta x, y)$, $(x, y + \Delta y)$ where $\Delta x$ and $\Delta y$ are negligible change. We have the following:

$$\Delta\tilde{x} = f(x + \Delta x, y) - f(x, y) = \frac{\partial f(x, y)}{\partial x} \cdot \Delta x + o(\Delta x) \tag{6}$$

$$\Delta\tilde{y} = g(x, y + \Delta y) - g(x, y) = \frac{\partial g(x, y)}{\partial y} \cdot \Delta y + o(\Delta y) \tag{7}$$

From the above equality and approximately regard $\eta_x$ and $\eta_y$ as $\frac{\Delta\tilde{x}}{\Delta x}$ and $\frac{\Delta\tilde{y}}{\Delta y}$, we have the following:

$$\eta_x = \frac{w_{obj}}{w_{back}} \cdot \frac{h_{back}}{h_{obj}} \cdot \left(1 + 2k_1x^2 + 4k_2x^2r^2 + k_1r^2 + k_2r^4\right) \tag{8}$$

$$\eta_y = \frac{h_{obj}}{h_{back}} \cdot \frac{n_{back}}{n_{obj}} \cdot \left(1 + 2k_1y^2 + 4k_2y^2r^2 + k_1r^2 + k_2r^4\right) \tag{9}$$

where $r$ equal $\sqrt{x^2 + y^2}$ is the distance from the specified point to the center of the radial distortion. Use $(x_i, y_j)$ to represent the coordinate position where the object images cover the background images. $x_i$ varies from 0 to $w_{back}$ and $y_j$ varies from 0 to $h_{back}$. The distribution of $(x_i, y_j)$ follows the equations:

$$x_i = \frac{i}{n_x} w_{back} \tag{10}$$

$$y_j = \frac{j}{n_y} h_{back} \tag{11}$$

where integers $n_x$ and $n_y$ are the number of points in the horizontal and vertical directions, respectively, and $i$ and $j$ are

indices in the horizontal and vertical directions, respectively:$i$ = 1, 2, 3, . . ., $n_x j$ = 1, 2, 3, . . ., $n_y$.

Through the above process, we obtain $m^* n$ images that simulated the workpiece placed in different positions, and then the object image is rotated $\theta_k$°, and the above process is repeated. The distribution of $\theta_k$ follows the equations:

$$\theta_k = \frac{k}{n_\theta} \cdot 360^\circ \tag{12}$$

where integer $n_\theta$ is the number of rotations per object image. The distribution of $k$ follows the equations:$k$ = 1, 2, 3, . . ., $n_\theta$

To reduce the computational cost spending in the training process and prevent overfitting, a portion of the generated images is randomly discarded and only images with a ratio of $p$ are reserved. Each image matches its corresponding coordinates ($x_i$, $y_j$) as the label. Obviously, a data set with $n_x n_y n_\theta p$ images is obtained.

For the second series of data sets, the pictures captured by the second-level visual equipment are selected as the background images. Obviously, the proportion of the object image in the background image becomes larger. The previous steps are akin. After extracting the region of interest, the object images are resized and covered to the background image on a specific point. The difference is the annotation method. As the second-level algorithm is based on semantic segmentation convolutional neural network, the annotation of each image is a matrix in the same size as the image. Initially, the annotation matrix is whole padded with 0. Next, replace the region where the object exists with label value, such as 1, 2, 3, . . ., $n$. Since the position of the object images is known, the process of generating annotations is automatic without manual operation. The schematic diagram of the data sets is shown in Figure 3.

### 3.2 First-level convolutional neuron network configurations

Use the first series training image data sets generated by the above method to train the first-level convolutional neural network. Set $n_x$ = 20, $n_y$ = 20, $n_\theta$ = 12, $p$ = 0.5 and obtain a total of 2,400 images. Experimental evidence demonstrates that the quantity is adequate for the convolutional neural network to provide precise coordinate detection. Take 12 background images containing including a variety of illumination effects and angles and generate several data sets for experiment.

On account of the task of the first-level convolutional neural network is detecting the position of the workpiece, a novel neural network whose inputs are 256 × 256 images resized from the original images and outputs are coordinate position of the workpiece is proposed. The neural network configuration is inspired by the VGGNet (Simonyan and Zisserman, 2014), using a very small convolution kernel and a 2 × 2 pooling window with stride 2. All inputs were scaled from unsigned integers ranging from 0 to 255 to a float-32 varying from 0 to 1. The architecture of the neural network is shown in Figure 4, composed of an input layer, three convolution layers, each followed by a maximum pooling layer, a flattened layer and the output layer. As the network only contains convolutional layers and flattened layers, without fully connected layers that are frequently arranged after the convolutional layers, it was named convolution-flattened neural (CFN) network. The convolutional layers use a 3 × 3 receptive field with stride 1. For the first convolution layer, eight filters are used, 16 filters are used for the second convolution layer, and 32 filters are used for the third convolution layer. The pooling layers use the 4 × 4 maximum-pooling window with stride 4 for downsampling. The output layer uses a sigmoid activation function, and all other layers use the following leaky rectified linear activation (Redmon *et al.*, 2016):

$$\phi(x) = \begin{cases} x, & (x > 0) \\ 0.1x & (x \le 0) \end{cases} \tag{13}$$

The output is a binary array ($m$, $n$) containing two float-32 ranging from 0 to 1. $m$ and $n$ are respectively multiplied by the width and height of the background image $w_{back}$, $h_{back}$ to obtain the actual coordinates of the workpiece ($x$, $y$). Mean squared error is chosen as the loss function:

**Figure 3** Schematic diagram of the data sets

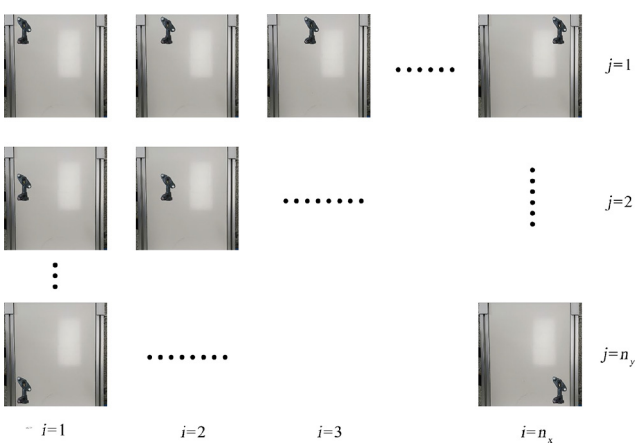
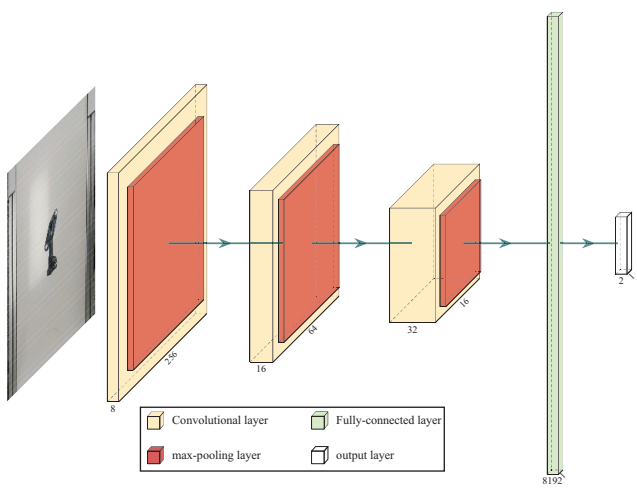



**Figure 4** Structure of the first-level convolutional neural network




**Note:** The height, width and channel of the layers are denoted for each group of convolution layers and fully connected layers

$$L = \frac{1}{N}\sum_{i=1}^{N}\left(Y_i - f_i(X)\right)^2 \tag{14}$$

where $Y_i$ is the true value in the label and $f_i(X)$ is the predicted value. Update parameter approach is Adam. Resize the generated imaged to 256 × 256 and then input them into the neural network. The total number of trainable parameters is 105,499. To train the training set of 2,400 images, the PC is equipped with 32.0 GB RAM, Core i7-9700F CPU 3.0 GHz and the NVIDIA GeForce RTX 2060. A batch size of 32 and 3,000 epochs are specified. The total time for generating the data sets and training the CFN network is around 90 min.

### 3.3 Second-level convolutional neuron network configurations

The second series image data sets are used to train the second-level convolutional neural network. In this case, set $n_x$ = 10, $n_y$ = 10, $n_\theta$ = 12, $p$ = 0.5 and obtain a total of 600 images. The second-level neural network is mainly responsible for semantic segmentation so that the size and contour information of the workpiece can be detected. A superexcellent architecture that merely comprises convolutional and upsampling layers is built up. The neural network architecture is illustrated in Figure 5. It is composed of the convolution part (left side) and the upsampling part (right side). The convolution part consists of four convolution layers whose receptive field is 3 × 3 with stride 1. For the first convolution layer, eight filters are used each followed by a 2 × 2 maximum-pooling window with stride 2. At each convolution layer, the quantity of feature channels is doubled. The upsampling part comprises an upsampling of the feature map, which is followed by a 3 × 3 convolution that the quantity of feature channels is halved, a concatenation with the feature map from the corresponding convolution part and a convolution layer whose receptive field is 3 × 3 with stride 1. The final layer is a 1 × 1 convolution layer which is used to map 16 feature channels and calculate the probability that the pixel is a part of the workpieces. In total, the neural network has eight convolutional layers. The output layer uses a softmax activation function, and all other layers use the ReLU activation function. The softmax function is defined as:

$$p_n(x) = \frac{e^{a_n(x)}}{\sum_{k=1}^{N} e^{a_k(x)}} \tag{15}$$

where $a_n(x)$ is the activation value in the $k$th feature channel at the pixel in position $x$. $N$ denotes the number of classes. $p_n(x)$ denotes the output value of the softmax function inclined to select the maximum input. E.g. $p_n(x)$ tends to 1 for the $n$ corresponding to the maximum activation $a_n(x)$, and $p_n(x)$ tends to 0 for all other $n$. On the issue of the loss function, due to the disparity between the background and the object in the sample, selecting the cross-entropy function directly as the loss function will make the prediction result tend to be background which results in the end effector unable to grasp the workpieces. To solve these disadvantages disclosed above, a balanced cross-entropy loss function is presented. In conventional cross-entropy function, the loss function is defined as:

$$L = -\sum_{i=1}^{N} \alpha y^{(i)} \log \hat{y}^{(i)} + (1-\alpha)\left(1 - y^{(i)}\right)\log\left(1 - \hat{y}^{(i)}\right) \tag{16}$$

where $y^{(i)}$ is the label of the sample whose value is 0 or 1. $\hat{y}^{(i)}$ is the probability that the sample is predicted to be positive. Import a weighting factor $\alpha \in [0, 1]$ to balance the disparity (Lin *et al.*, 2017). In practice, we define the loss function as:

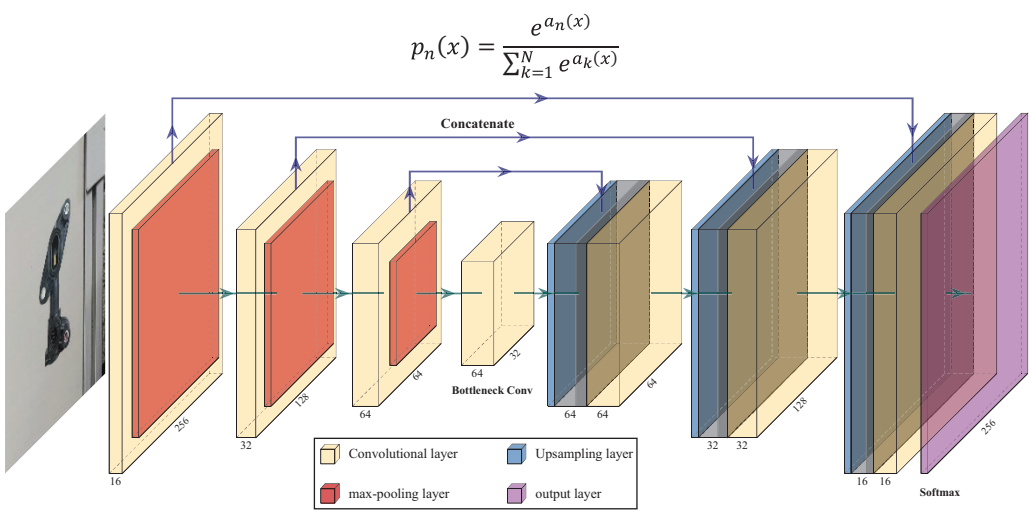



**Figure 5** Architecture of the second-level convolutional neural network

**Notes:** Each box corresponds to a multichannel feature map. The number of channels is denoted on side edge of the box. The x-y size is provided at the lower edge of the box. The arrows denote the concatenating operations

$$L = -\sum_{i=1}^{N} \alpha y^{(i)} \log \hat{y}^{(i)} + (1-\alpha)\left(1-y^{(i)}\right)\log\left(1-\hat{y}^{(i)}\right) \tag{17}$$

where $\alpha$ denotes the weight coefficients for different classes. This method is adopted in the experiments as it dramatically improves accuracy over the conventional cross-entropy function and finds the following value to work first-rate in the experiments.

$$\alpha = \frac{w_{back} h_{back}}{w_{obj} h_{obj} + w_{back} h_{back}} \tag{18}$$

The neural network is trained with Adam update parameter approach. A batch size of 32 and 5000 epochs are specified. The total time for generating the data sets and training the CFN network is around 120 min. The IOU index of the final training result exceeds 0.95. The calculation method of IOU will be detailed in the next section.

### 3.4 Binocular eye-in-hand vision system

In the second-level visual equipment, a binocular eye-in-hand system is deployed to obtain precise depth and pose information of the workpieces (Engel *et al.*, 2015). Two cameras are mounted in parallel on the end effector where the suction cup and gripper are deployed. During the work of the second-level vision system, the two cameras independently detect the contour of the workpiece and obtain the relative coordinates of the workpiece in different coordinate systems. Obviously, the images acquired by the two cameras will be slightly different, which is called disparity. By means of disparity, the distance between the workpiece and the end effector can be calculated. Meanwhile, via the contour of the workpiece, the angles including yaw, pitch and roll can be worked out. Through these parameters, the position of the end effector can be continuously adjusted until the end effector is at a specific height directly above the workpiece.

For the general situation of a relative mounting position of the left and right cameras with no special requirements, the image-forming principle of the binocular camera is shown in Figure 6. Note that the optical axes of the two cameras are not parallel in this case and there is no special relationship between the coordinate axes of the camera coordinates of the left and right cameras. Let $o_l x_l y_l z_l$ and $o_r x_r y_r z_r$ be the left and right camera coordinates, respectively, and $oxyz$ the world coordinate system. Assume that the coordinate system of the left camera and the world coordinate system completely coincide. We have the following:

$$s_l \begin{pmatrix} X_l \\ Y_l \\ 1 \end{pmatrix} = \begin{pmatrix} f_l & 0 & 0 \\ 0 & f_l & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{19}$$

$$s_r \begin{pmatrix} X_r \\ Y_r \\ 1 \end{pmatrix} = \begin{pmatrix} f_r & 0 & 0 \\ 0 & f_r & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_r \\ y_r \\ z_r \end{pmatrix} \tag{20}$$

where $X_l$, $Y_l$ and $X_r$, $Y_r$ are the coordinates in the left and right image coordinate system, respectively; $f$ is the focal length and $s$ is the scale factor. $M_{lr}$ denotes the transformation relationship between left and right camera coordinate systems. We have the following:

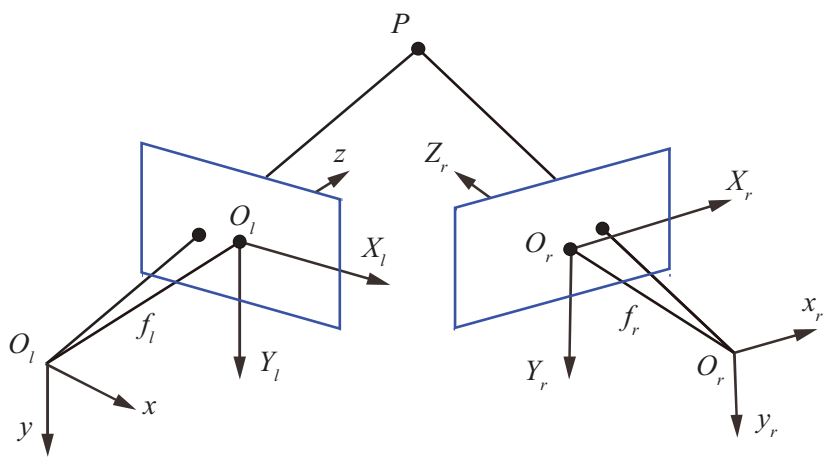
$$\begin{pmatrix} x_r \\ y_r \\ z_r \end{pmatrix} = M_{lr} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = [R\ |t] \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{21}$$

where $M_{lr}$ equals $[R, t]$. $R$ and $t$ are the rotation and translation matrices from one coordinate system to another. From the above equality, we have the following:

$$\rho_s \begin{pmatrix} X_r \\ Y_r \\ Z_r \end{pmatrix} = \begin{pmatrix} f_r r_1 & f_r r_2 & f_r r_3 & f_r t_x \\ f_r r_4 & f_r r_5 & f_r r_6 & f_r t_y \\ r_7 & r_8 & r_9 & t_z \end{pmatrix} \begin{pmatrix} \frac{z X_l}{f_l} \\ \frac{z Y_l}{f_l} \\ z \\ 1 \end{pmatrix} \tag{22}$$

Solving this equation, we can obtain the following:

**Figure 6** Image-forming principle of the binocular camera



$$\begin{cases} x = \dfrac{zX_l}{f_l} \\ y = \dfrac{zY_l}{f_l} \\ z = \dfrac{f_l(f_r t_x - X_r t_z)}{X_r(r_7 + r_8 Y_l + f_r r_9) - f_r(r_1 X_l + r_2 Y_l + f_l r_3)} \\ = \dfrac{f_l(f_r t_x - X_r t_z)}{Y_r(r_7 X_l + r_8 Y_l + f_l r_9) - f_r(r_4 X_l + r_5 Y_l + f_l r_3)} \end{cases} \tag{23}$$

In this paper, two identical cameras as binocular vision system is deployed. Through calibration, the optical axis of the right camera is parallel to the left camera. Assuming $R$ is the unit matrix, $t_y = t_z = 0$, then the above equation can be simplified to:

$$\begin{cases} x = B\dfrac{X_1}{d} \\ y = B\dfrac{Y_1}{d} \\ z = B\dfrac{f}{d} \end{cases} \tag{24}$$

where $B$ is the baseline width and $d$ is the disparity. $B = t_x$, $f_l = f_r = f$, $d = X_l - X_r$. According to the above equation, the distance between the object and the lens is proportional to the disparity. As long as two projection points of a point in the ideal binocular image are found, its coordinates in the world coordinate system can be worked out.

In addition to the above, the centroid of the workpiece can be calculated according to the contour. According to the centroid position and distance of the workpiece, plan the trajectory of the end effector, and let the end effector be closer to the specified height directly above the workpiece. Next, the binocular vision system reacquires the contour information of the workpiece and re-executes the above process. After several iterations, when the centroid is within a certain threshold in the center of the visual field, the end effector is considered to have reached directly above the object and then executes grasping.

## 4. Experiments and analysis

A series of experiments including static experiment of test algorithm and experiment of grasping workpieces were conducted to test and verify the effectiveness and precision of the proposed convolutional neural networks and system. Considering that two levels of convolutional neural networks are respectively responsible for detecting the position and contour of the workpieces and the inputs are disparate, different experiments are conducted to verify these two levels of algorithms. The input of the first-level neural network was taken from a higher fixed position; hence, the main factors affecting the grasping performance are contaminations and illumination. Contrastively, the images input to the second-level neural network were taken from a lower dynamic position; thus, the experiments verify the stability of the proposed system in various viewing angles and distances.

**Figure 7** Four data sets for the illumination and contamination test

### 4.1 Effects of interferent and illumination

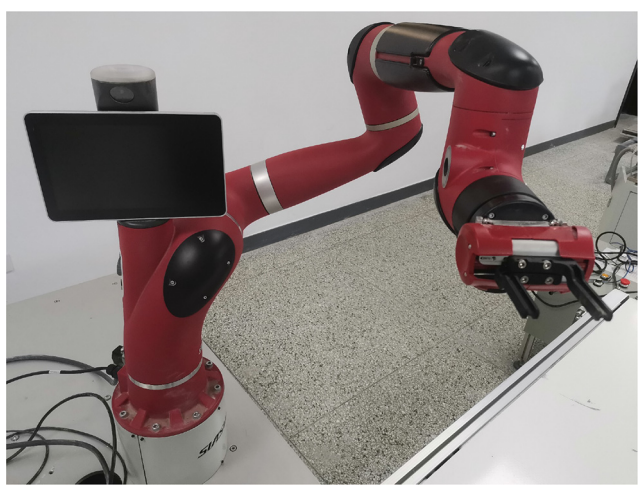
In the experiment probing into the influence of interferent and illumination on the first-level neural network, a total of four circumstances are set up, which conclude strong light, weak light, stochastic light, stochastic light with contamination. Under four circumstances, all the data sets are generated in the method in subsection 3.1, and the neural network architectures are same as shown in Figure 4. Yet, the four models are trained with diverse training data sets. As shown in Figure 7, data set A is generated by a sole background image with strong light, data set B by a single background image with weak light, data set C by eight background images in various illumination conditions and data set D by eight background images in various illumination and interferent conditions. Obviously, the amount of data in data sets C and D is eight times that of data sets A and B; hence, the training time will be dramatically increased (from 127 s to 718 s).

As shown in Figure 8, a Rethink Sawyer is deployed to conduct the experiment. The workpiece is randomly placed, and the true position is simultaneously recorded. The first-level visual equipment captures a global image including the whole area where workpieces may appear. The neural network uses the forward propagation of the trained model to output the predicted position of the workpiece, and the robotic arm will work out a reasonable trajectory and move to the specified height above the corresponding position. After this process is completed, on the condition that the workpiece appears within the visual field of the eye-hand binocular cameras mounted on the robotic arm, the second-level vision device can take over the next workflow, and therefore, the first level algorithm is considered competent. The differences among predicted and true positions and success ratio are indicators to measure the performance of the first-level neural network.

#### *4.1.1 Result of success ratio*

The experiments are conducted to demonstrate that the models trained by the four data sets perform extremely differently. The success ratios of models trained by four data sets under diverse environments are shown in Table 1. Distinctly, it was found that data set B generates the worst performance. It is due to the fact that

**Figure 8** Photo of the robot arm deployed to conduct the experiment



the images under weak illumination cannot provide enough features. Data set D outperforms other data sets with a higher success ratio than others under every condition. This is due to data set D, which contains various interferent and illumination features. Data set C performs superiorly under random light conditions and yet performs inferiorly in the presence of interferent. That is why data set C does not contain images of interferent; however, data set D considers the effect of interferent. Data set A performs well under strong light circumstance and conversely performs poorly under weak light circumstance. Hence, while subjected to stochastic illumination circumstance, the algorithm generates a moderate level of success ratio.

*4.1.2 Result of positioning accuracy*

In the previous section, the performance of models trained by four data sets is discussed in detail. Data set B generates the worst performance. Therefore, in the next experiment, data set B is excluded and the performance of models trained by the other three data sets is compared. The workpiece is randomly placed in a specific area, and the true position in the world coordinate system is synchronously noted. The neural network uses the forward propagation of the trained model to work out the predicted position of the workpiece. The difference between the true and predicted coordinates in the $x$ and $y$ directions are taken as indicators to measure the performance of the first-level neural network. As shown in Figure 9, the performance of the models trained by the three data sets (data set B excluded) is plotted in the statistical graph. It can be found from the statistical chart that the average error of the model corresponding to the data set A is minimal and the error of the other two models is slightly larger than it. However, the model trained by data set A performs poorly under weak light or disturbing circumstances. From the above results, model selection depends on the environment where the grasping system is deployed. In addition, the errors in the $x$ and $y$ directions are slightly different. It may be caused by the resizing of the image before input to the neural network.

### 4.2 Effects of distance and visual angle

This part of the experiment mainly explores the effect of distance and visual angle on the performance of the second-level neural network. In the proposed system, the first-level neural network works out the position of the workpiece, and the robotic arm will work out a reasonable trajectory and move to the specified height above the corresponding position. However, in the actual manufacture process, due to the positioning error and interference factors of the robot arm, the angle of view and distance may alter, which requires the algorithm to adapt to these variances. The degree of visual angle can be described by three angles including yaw, pitch and roll. A total of three groups are disposed. The first group of images was taken with a rotation of ±10° and ±20° in the yaw direction. The second group of images was taken with a rotation of ±10° and ±20° in the pitch direction. The third group of images was taken with a rotation of ±30° and ±60° in the roll direction. Each group of images is divided into long distance, standard distance and short distance. The sample images are shown in Figure 10. Input a total of 195 images into

**Table 1** Success rates of four models

| Condition | Model 1(%) | 2(%) | 3(%) | 4(%) |
|---|---|---|---|---|
| **Low light** | 0 | 0 | 100 | 100 |
| **Strong light** | 90 | 0 | 100 | 100 |
| **Mixed light** | 50 | 0 | 100 | 100 |
| **Mixed + cntm** | 90 | 0 | 80 | 90 |
| **Average** | 57 | 0 | 95 | 98 |

**Figure 9** Measurements of the positioning errors in the x and y directions for the three data sets

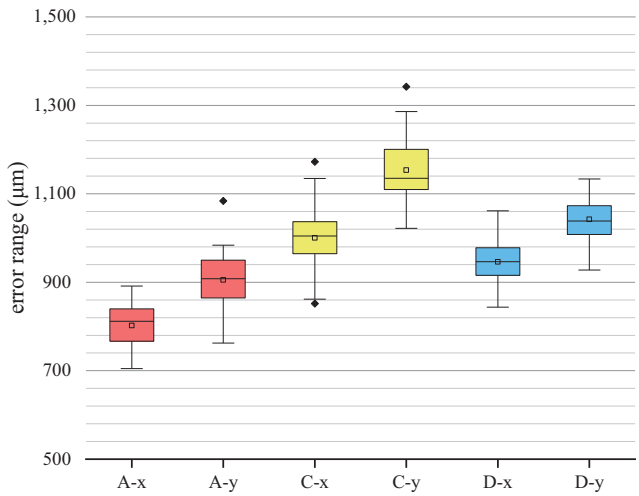
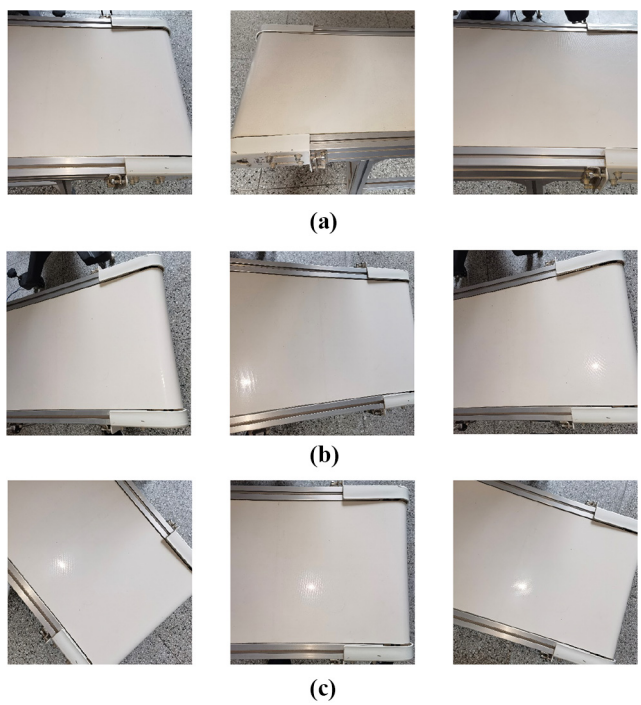



**Figure 10** Sample images of various viewing angles




**Notes:** (a) Yaw; (b) pitch; (c) roll

the second-level neural network and analyze the performance of their output.

To evaluate each output accurately, a pixel-wise evaluation metrics for testing the performance of the output are adopted. In the semantic segmentation task, based on predictions and true values, each pixel can be divided into four categories containing TP, TN, FP and FN, which are elaborated in Table 2. The evaluation metrics cover recall, precision and F1 score (also called Dice similarity coefficient). The evaluation metrics are defined as follows:

$$Recall = \frac{TP}{TP + FN} \tag{25}$$

$$Precision = \frac{TP}{TP + FP} \tag{26}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{27}$$

In this paper, the F1 score is adopted to evaluate the outputs. The result is as shown in Figure 11. For the rolling effect, the proposed system is hardly affected by the rotation of the rolling direction. At 30° and 60° of rotation, the F1 score only drops by 2% and 4%, respectively, which is due to the fact that the generated data sets contain target objects at multifarious angles. In terms of the effect of yaw rotation, the F1 score decreases by 8% at ±10° and 14% at ±20°. In terms of the pitch effect, the F1 score decreases by 7% at ±10° and 13% at ±20°. The distance between the camera and the target object has little effect on the result. Without considering rotation, the F1 score drops by 4% at long distances and 5% at short distance. Experimental results demonstrate that the proposed second level is extraordinarily robust subject to environmental variations.

### 4.3 Grasping experiment

#### *4.3.1 Experiment setting and procedures*

As shown in Figure 6, a binocular hand-eye vision system is mounted on the end effector and used as the second-level vision system for detecting the pose of the workpieces. Two identical cameras are mounted symmetrically at a distance of 15 cm. To obtain the background image, move the end effector to the grasping position and then move vertically upward by 30 cm to obtain the left and right camera images as the background image of the second-level neural network. The first-level visual equipment is placed in an uplifted position and then captures a global image including the whole area where workpieces may appear. The image serves as the background image of the first-level neural network. Using these background images, the training set is generated with the method proposed in subsection 3.1.

The workflow is as shown in Figure 2. First, the workpiece is randomly placed, and the first-level visual equipment captures a global image. Then, the neural network uses the forward propagation of the trained model to output the predicted position of the workpiece, and the robotic arm will work out a reasonable trajectory and move to the specified height above the corresponding position. After this process is completed, on condition that the workpiece appears within the visual field of the eye-hand binocular cameras, the first-level vision device will be kicked off, and the second-level vision device will take over the next workflow. Conversely, on condition that the workpiece does not appear in the visual field, the manipulator will retract, restart the first-level visual equipment and repeat the above process until it succeeds. Next, the second-level vision equipment obtains the contour of the workpiece and then calculates the centroid position and distance of the workpiece. The algorithm plans the trajectory of the end effector and lets the end effector to execute grasping.

In the experiment, to test the robustness of the proposed system to environmental undulation, the illumination changes randomly, and the interferers are added at random. The grasping success ratio, iterative times and elapsed time were recorded and analyzed in the next section.

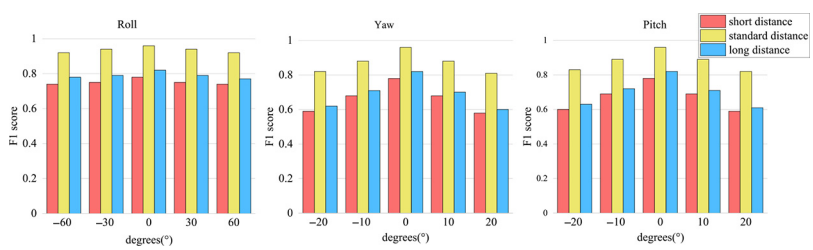
#### *4.3.2 Results of experiment*

The experiment results show that the success ratio of the proposed system is extremely high. As long as the first-level visual detection is successful, the success rate of subsequent process is nearly 100%. Only when deliberately place extremely disturbing analogues in the target area, the first-level visual detection system may output incorrect coordinates, resulting in increased elapsed time. The average positioning error of first-level is visual detection ±1.9 mm in the $x$ direction and ±1.5 mm in $y$ direction, which is never greater than the radius of the visual field of the second-level visual equipment. Hence, the second-level visual equipment is always able to smoothly take over the next workflow. The

**Table 2** Confusion matrix of segmented mask prediction

| Ground truth | Prediction | Category |
|---|---|---|
| **Positive** | Positive | True positive(TP) |
| **Positive** | Negative | False negative(FN) |
| **Negative** | Positive | False positive(FP) |
| **Negative** | Negative | True negative(TN) |

**Figure 11** Performance under different circumstances measured by F1 scores



average F1 score of semantic segmentation is 0.92, which ensures the precision of the pose estimation and the success ratio of grasping. In all 400 successful experiments, the second level of visual detection medially takes 3.5 iterations to complete the task with a maximum of eight iterations. The first-level visual detection is averagely completed in 1.2 s. The entire grasping workpiece task takes about 32 s. Attributed to the abundance of the features of the generated data sets, the effect of illumination on time and success rate is negligible.

## 5. Conclusion

In this paper, an industrial intelligent grasping system based on convolutional neural network and binocular eye-in-hand visual system is presented. This grasping system is able to adapt to various positioning errors including angle and distance under various environmental circumstances. Notably, the neural network is trained automatically by the proposed method without manual labeling, which saves time spent on manufacturing the data set and makes the intelligent grasping system easy to deploy into a practical industrial environment. From the above experiments and analysis, the system has two properties that help to achieve high grasping effectiveness.

1 The proposed system uses a two-level vision device and equips with different neural networks accordingly. The main task of the first-level vision equipment is rough positioning and determining whether the object is still. Diversely, the second-level vision system aims to accurately detect the pose of the workpiece. The two systems coordinate to achieve preferable grasping effectiveness.
2 The generated data set contains sundry images under various illumination and interferent conditions. Attributed to the abundance of the features of the generated data sets, the effect of illumination on time and success rate is negligible. Moreover, experimental results demonstrate that the proposed second level is extraordinarily robust to environmental variations.

In the future, the proposed system is expected to be deployed in more factories and to improve the productivity as well as the quality of the automatic manufactural lines.

## References

Chen, F., Ye, X., Yin, S., Ye, Q., Huang, S. and Tang, Q. (2019), "Automated vision positioning system for dicing semiconductor chips", *The International Journal of Advanced Manufacturing Technology*, Vol. 100 Nos 9/12, pp. 2669-2678.

Engel, J., Stückler, J. and Cremers, D. (2015), "Large-scale direct SLAM with stereo cameras", *International Conference on Intelligent Robots and Systems*, pp. 1935-1942.

Enrique, C., Carrasco, M. and Ríos, S. (2018), "Evaluation of an eye-pointer interaction device for human-computer interaction", *Heliyon*, Vol. 4 No. 3, p. e00574.

Glorot, X., Bordes, A. and Bengio, Y. (2011), "Deep sparse rectifier neural networks", *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, Vol. 15, pp. 315-323.

Hinton, G. and Salakhutdinov, R.R. (2006), "Reducing the dimensionality of data with neural networks", *Science*, Vol. 313 No. 5786, pp. 504-507.

Hua, C., Wang, H., Wang, H., Lu, S., Liu, C. and Khalid, S.M. (2019), "A novel method of building functional brain network using deep learning algorithm with application in proficiency detection", *International Journal of Neural Systems*, Vol. 29 No. 1, p. 1850015.

Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q. (2017), "Densely connected convolutional networks", *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700-4708.

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012), "ImageNet classification with deep convolutional neural networks", *Advances in Neural Information Processing Systems*, Vol. 25, pp. 1097-1105.

Lin, T.Y., Goyal, P., Girshick, R., He, K. and Dollár, P. (2017), "Focal loss for dense object detection", *Proceedings of the IEEE international conference on computer vision*, pp. 2980-2988.

Oron, S., Dekel, T., Xue, T., Freeman, W.T. and Avidan, S. (2018), "BestBuddies similarity-robust template matching using mutual nearest neighbors", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 40 No. 8, pp. 1799-1813.

Ouyang, W., Tombari, F., Mattoccia, S., Di Stefano, L. and Cham, W.K. (2012), "Performance evaluation of full search equivalent pattern matching algorithms", *IEEE Transactions on Pattern Analysis & Machine Intelligence*, Vol. 34 No. 1, pp. 127-143.

Qiao, H., Wang, M., Su, J., Jia, S. and Li, R. (2014), "The concept of "attractive region in environment" and its application in high-precision tasks with low-precision systems", *IEEE/ASME Transactions on Mechatronics*, Vol. 20 No. 5, pp. 2311-2327.

Ronneberger, O., Fischer, P. and Brox, T. (2015), "U-net: Convolutional networks for biomedical image segmentation", *In International Conference on Medical image computing and computer-assisted intervention*, Springer, pp. 234-241.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016), "You only look once: unified, real-time object detection", *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779-788.

Simonyan, K. and Zisserman, A. (2014), "Very deep convolutional networks for large-scale image recognition", arXiv preprint, arXiv: 1409.1556.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014), "Dropout: a simple way to prevent neural networks from over fitting", *The Journal of Machine Learning Research*, Vol. 15 No. 1, pp. 1929-1958.

Troniak, D., Sattar, J., Gupta, A., Little, J.J., Chan, W., Calisgan, E., Croft, E. and Van der Loos, M. (2013), "Charlie rides the elevator-integrating vision, navigation and manipulation towards multi-floor robot locomotion", *2013 international conference on computer and robot vision, IEEE*, pp. 1-8.

Vadim, S. (2013), "Moravec's paradox: consideration in the context of two brain hemisphere functions", *Activitas Nervosa Superior*, Vol. 55 No. 3, pp. 108-111.

Wang, S. (2021), "Efficient deep learning", *Nature Computational Science*, Vol. 1 No. 3, pp. 181-182.

Wu, Y. and He, K. (2018), "Group normalization", *Proceedings of the European conference on computer vision*, pp. 3-19.

Yang, C., Jiang, Y., Li, Z., He, W. and Su, C.Y. (2016), "Neural control of bimanual robots with guaranteed global stability and motion precision", *IEEE Transactions on Industrial Informatics*, Vol. 13 No. 3, pp. 1162-1171.

Zhang, Z. (2000), "A flexible new technique for camera calibration", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22 No. 11, pp. 1330-1334.

## Corresponding author

**Wang Hong** can be contacted at: hongwang@mail.neu.edu.cn