IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 9, 2019, accepted January 26, 2019, date of publication February 21, 2019, date of current version March 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2900424

Appearance-Based Gaze Estimator for Natural Interaction Control of Surgical Robots

PENG LI[1], (Member, IEEE), XUEBIN HOU[1], XINGGUANG DUAN[2], HIUMAN YIP[3], (Student Member, IEEE), GUOLI SONG[4], AND YUNHUI LIU[3], (Fellow, IEEE)

[1]School of Mechanical Engineering and Automation, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China
[2]School of Mechatronical Engineering, Beijing Institute of Technology, Beijing, China
[3]Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong
[4]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China

Corresponding author: Guoli Song (songgl@sia.cn)

This work was supported in part by the National Natural Science Foundation of China through the Shenzhen Robot Research Program under Grant U1613221, in part by the Shenzhen Fundamental Research under Grant JCYJ20170307150346964, in part by the Guangdong Provincial Science and Technology Funds under Grant 2017A020211001, and in part by the National Key Research and Development Program of China under Grant 2017YFB1302800.

ABSTRACT Robots are playing an increasingly important role in modern surgery. However, conventional human–computer interaction methods, such as joystick control and sound control, have some shortcomings, and medical personnel are required to specifically practice operating the robot. We propose a human–computer interaction model based on eye movement with which medical staff can conveniently use their eye movements to control the robot. Our algorithm requires only an RGB camera to perform tasks without requiring expensive eye-tracking devices. Two kinds of eye control modes are designed in this paper. The first type is the pick and place movement, with which the user uses eye gaze to specify the point where the robotic arm is required to move. The second type is user command movement, with which the user can use eye gaze to select the direction in which the user desires the robot to move. The experimental results demonstrate the feasibility and convenience of these two modes of movement.

INDEX TERMS Deep learning, surgical robot, gaze estimation, convolutional neural network.

I. INTRODUCTION

There are a variety of human-computer interactions between medical personnel and surgical robots. Usually, medical staff can interact with the robot by hand, foot, voice and other means. However, operating the robot using a joystick is not a good method because, in a surgical environment, the hands of medical staff may be occupied by other medical devices, which makes difficult for them to use a joystick to operate surgical robots. Yip *et al.* [1] attempted to control the robot arm with an eye-tracking device. This was a good attempt. This human-computer interaction mode allows the medical staff to easily operate the robotic arm while still allowing other tasks to continue. However, the shortcoming of this solution is that the eye-tracking device is too expensive and the tracking accuracy is not high. A bigger problem is that some eye-tracking devices have higher requirements for the wearer's posture. These shortcomings greatly limit the use of eye-tracking devices in controlling medical robots.

Eye gaze direction prediction has always been a popular topic in the field of computer vision [2]. The recent eye gaze direction prediction algorithm mainly utilizes large-scale real data and synthetic data and combines machine learning algorithms to complete the gaze direction prediction [3]–[6]. Using a monocular RGB camera to accomplish the eye gaze direction task in an unconstrained state is promising since monocular RGB cameras are very common on mobile devices [7], [8]. Combining the eye gaze direction prediction algorithm with the medical robot arm control is a very interesting research topic. In recent years, deep learning has achieved great success in some computer vision tasks, proving the great prospect of convolutional neural network models in image processing. In this paper, we use a convolutional neural network model to accomplish the eye gaze direction prediction task. In order to avoid errors that are easily caused by single-frame images [9], we used

The associate editor coordinating the review of this manuscript and approving it for publication was Giancarlo Fortino.

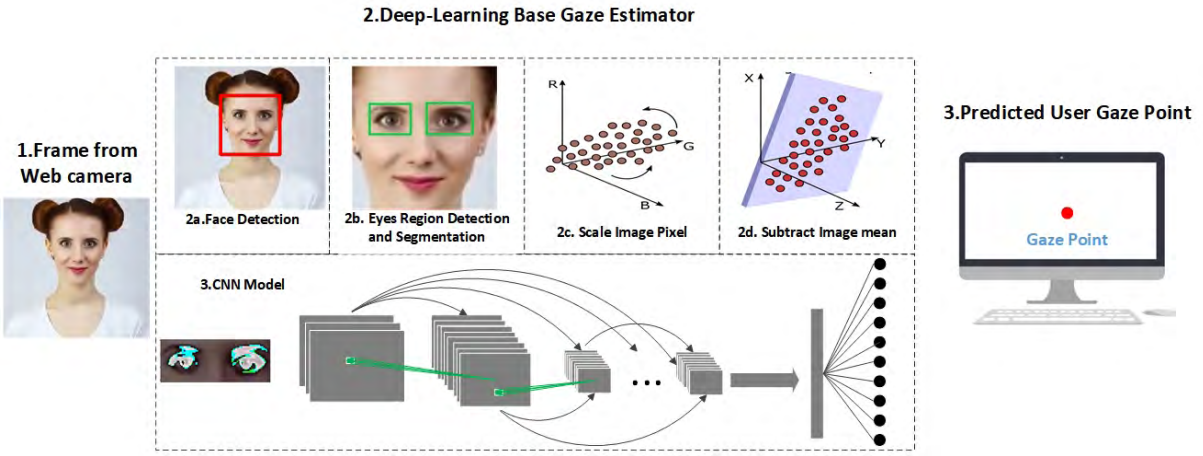

FIGURE 1. Pipeline of our eye-tracking algorithm. We present DenseGaze, a novel approach for eye gaze prediction using a convolutional neural network. DenseGaze is an end-to-end network that can accurately predict the users gaze point based on images captured by a web camera mounted on a monitor.

multi-frame images for gaze direction prediction. The input to the model is the image of the user's eye, and the output is the predicted value of the user's gaze direction. The pipeline of our algorithm is shown in Fig. 1. First, the user's image is obtained by using a common RGB camera, and the user's eye image is obtained by the face detection and the human eye detection algorithms. After the user's eye image is obtained, the pixel values of the image are normalized. Through the process described above, we obtain the standard input of the gaze estimation model. Our gaze estimation model consists of multiple layers of convolutional layers, pooling layers, and activation functions. Through an end-to-end method, the user's eye image is input into the prediction model to obtain the user's gaze direction prediction value. The process of the gaze direction prediction model based on a convolution neural network method is simple, and the prediction results are accurate.

Gaze estimation is a popular topic in the field of computer vision. Recently, more attention has been focused on training the gaze prediction model using synthetic data. These synthesized data are mixed with real data and used in a training model, which can greatly improve the performance of the model [3]–[6]. Most mobile devices today are equipped with a monocular camera, which makes the use of monocular cameras for unconstrained gaze direction prediction algorithms more common [7], [8]. An interesting topic for study is whether we can use gaze estimation information to control surgical robots. Gaze direction prediction can be roughly divided into two methods; the first method is a model-based prediction method, and the second method is an appearance-based method. Model-based prediction methods mainly use prior knowledge of the human face and eye to build a model and use this model for gaze direction prediction. The appearance-based method can predict the gaze direction based on the user's eye image or an image of the entire face. A fixed head pose was required in the early stages of appearance-based methods. Later, works focused on pose-free gaze estimation either from depth images [10] or monocular RGB images [11]–[13]. Although learning-based methods have a promising future to achieve pose and person independence, it requires large amounts of labeled training data [3], [6], [14], [15]. As a result, increasing the size of the gaze estimation datasets has been important in recent years [16]–[18]. Most previous works predict gaze direction with a single eye image as input to the predictor. Some research has used two images, one of each eye [19], or a single image covering the eye region [16]. Reference [14] found that gaze estimation methods that use individual eye images as well as a face image, and a face grid as input can improve performance. Eye gaze control has shown to be a promising modality in robot-assisted surgery and can be used to control endoscopes [20] and surgical robots [21], [22].

In the past few years, deep learning has performed very well in many areas such as image classification, speech recognition, and natural language processing. Among all kinds of deep neural networks, convolutional neural networks (CNN) have been widely used because of their excellent performance in image processing and natural language processing. Due to the emergence of large-scale label data and the great improvement of computational power, the research of convolutional neural networks has developed rapidly and has achieved the best results in many tasks. Convolutional neural networks are a deep learning framework inspired by biological vision neural systems. Lecun *et al.* [23] presented LeNet-5, which laid the foundation for the current CNN structure, and used this model for handwriting recognition. Consisting of a multilayer artificial neural network, LeNet-5 extracts useful features of the original image, enabling it to identify valid patterns directly from the pixels. However, due to limited computational power and lack of training data at the time, the CNN model was not used in more complex scenarios such as large-scale image or video classification [24]–[29].

In 2012, Krizhevsky *et al.* [30] proposed a classic convolutional neural network structure (AlexNet) and applied it to the task of image classification, which has achieved a significant lead over other methods. Since AlexNet appeared and achieved excellent results, researchers have conducted considerable research to improve convolutional neural networks, such as ZFNet, VGNet, GoogleNet, ResNet and DenseNet. With the improvement of the CNN network structure, a clear trend is that the network is becoming increasingly deeper; that is, the number of layers in the neural network is increasing. For example, ResNet, which won the ILSVRC 2015 competition, has 20 times the number of layers as the AlexNet neural network and 8 times the number of layers of VGGNet. As the number of neural network layers increases, the nonlinear increase of the network allows it to better fit the objective function and obtain a better feature representation. However, an increase in network depth also makes the training more difficult and more likely to overfit the training data [31]–[35].

Although CNN has achieved excellent performance in many fields, there are still many problems for researchers to explore and solve. For example, in recent years, the depth of the CNN network has become increasingly deeper, which makes the demand for large-scale labeled training data and high-performance computers higher and labeling the training data requires considerable manpower. Therefore, there have been increasing studies on neural networks for unsupervised learning. At the same time, the deep network has higher requirements on the performance of the computing machine, and it is necessary to invest in researching high-performance parallel training algorithms. Deep networks limit the application scope of deep learning algorithms; for example, it is difficult to apply to some devices that lack computing resources, such as smartphones and tablets [36]–[38].

The contributions of this paper can be summarized as follows: (1) an eye gaze direction prediction algorithm is designed to assist the control of the medical robot arm. The control mode is divided into two main types. The first model moves in the specified direction. The model predicts the user's gaze direction based on the eye image and then moves the arm in this direction. The second model moves to a specified location, where the user focuses on the point, and the medical robot arm automatically moves to the corresponding position. Fig. 2 shows the schematic diagram of the two control modes. (2) We designed an automatic annotation program to obtain a large number of eye image training data with user gaze direction labels. With this program, a large amount of training data can be obtained in a relatively short period of time. (3) The model used to predict the direction of the user's gaze is a very deep convolutional neural network. The structure of the network adopts the design idea of DenseNet [24], which reuses a large number of feature information. This allows for better prediction performance, while the model requires fewer computing resources than the traditional convolution neural network structure (AlexNet, VggNet). This feature makes it possible to transplant the model to some computing resource-constrained devices, such as FPGA, mobile phones and other devices. At the same time, the amount of data required to train the multilayer convolutional neural network is greatly reduced, and the human resources and time resources for collecting the labeled data are reduced. (4) We apply the idea of transfer learning to the training of convolutional neural networks. We first use the relevant data to pretrain the model and then transfer the pretrained model to our task. The advantage is that the training model requires fewer data, and the convergence rate of the model is faster.

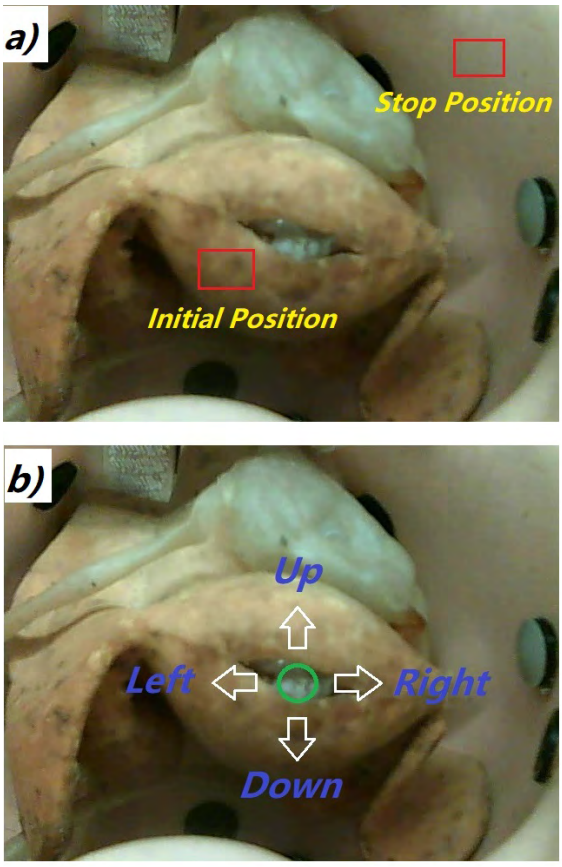

FIGURE 2. Functionality of the proposed eye gaze control. (a) The user can specify the start position and destination using our eyes gazing estimation algorithm, and the surgical robot can automatically move from the start position to the destination. (b) The user can specify the robot's moving direction using our eye gaze algorithm.

II. METHODS

A. TRADITIONAL CNN AND DENSENET

In general, convolutional neural networks are a kind of feed-forward neural network and are used in machine learning tasks of various backgrounds. Krizhevsky *et al.* [30] used the multilayer convolutional neural network to complete the image classification task in ImageNet data. Since the advent of AlexNet, there have been many models of different convolutional neural network structures, such as the residual network [25] and the inception network [31]. To better introduce our work, we need to briefly introduce convolutional neural networks. Convolution is a common method in the field of

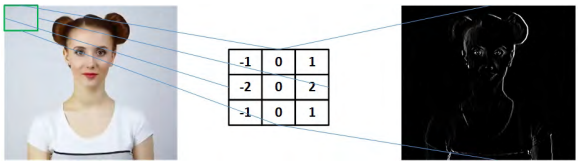

FIGURE 3. Convolution operations using the Sobel operator are commonly used for edge detection. The left side is the original image, the middle is the Sobel operator, and the right side is the output image after the convolution operation using the Sobel operator.

image processing. Figure 3 shows the process of convolution using the sobel operator. The mathematical definition of a continuous space convolution operation is shown in Equation (1), where the symbol $\otimes$ represents the convolution operation.

$$[g \otimes h](t) = \int_{-\infty}^{+\infty} g(\tau)h(t-\tau)d\tau \tag{1}$$

The mathematical expression of the 1d convolution operation in discrete space is shown in Equation (2).

$$[g \otimes h](n) = \sum_{t=-\infty}^{+\infty} g(t)h(n-t) \tag{2}$$

where n and t are integers. However, in the CNN model, the mathematical expression of the discrete two-dimensional convolution operation is usually expressed by Equation (3).

$$[A \otimes B][j_1, j_2] = \sum_{t=-\infty}^{+\infty} g(t)h(n-t) \tag{3}$$

Pointwise multiplication is a method commonly used in convolution operations to simplify calculations, and its mathematical expression is shown in Equation (4).

$$[g * h](_{i,j}) = G(_{i,j})\cdot H(_{i,j}) \tag{4}$$

The CNN model is usually composed of a multilayer convolutional neural network, and the output of the layer l^{th} convolutional network is defined as x_l. In the traditional convolutional neural network model, the l^{th} layer output x_l is usually calculated by a nonlinear transformation function H_l, and the input of this function is the output of the $l-1^{st}$ layer x_{l-1}.

$$X_l = H_l(X_{l-1}) \tag{5}$$

The nonlinear function H is usually composed of a convolution function, a nonlinear function and a dropout function [39]. Models with very multilayered convolutional neural networks are often difficult to train. To make the CNN model easier to train, ResNets [25] use a residual block that sums the input and output of a layer. The mathematical expression of this relationship is shown in Equation (6), where X_l represents the output of the l^{th} network.

$$X_l = H_l(X_{l-1}) + X_{l-1} \tag{6}$$

The residual block structure enables the CNN model to reuse features so that gradients are better transmitted in each layer of the neural network. In this case, the H function consists of multiple repetitions of batch normalization (BN), ReLU, and convolution functions.

DenseNet [24] expands on the idea of ResNet. ResNet sums the input and output of each layer, and DenseNet uses all the outputs before each layer as input to this layer of network functions. The mathematical expression of the output X_l of each layer of the neural network is shown in Equation (7), where [. . .] represents the concatenation operation.

$$X_l = H_l([X_{l-1}, X_{l-2}, \ldots, X_0]) \tag{7}$$

In the DenseNet, the nonlinear function H consists of BN, ReLU, convolution, and dropout functions. This network structure allows each layer of the neural network to receive gradient monitoring information directly while reusing features. The output of each layer network has K feature maps, so parameter K is also called the growth rate. K is a hyperparameter, usually set to a very small value, such as 12. Based on the above, the feature maps output by each layer of the DenseNet will grow linearly. For example, when calculating in l layers, we need to consider that layer $[x_{l-1}, x_{l-2}, ..., x_0]$ will have $l \times k$ feature maps. The dimensionality reduction function is needed to reduce the dimensionality of the input and reduce the number of calculations. DenseNet uses 1×1 convolution to reduce the number of feature maps, and 2×2 pooling operations to reduce the dimensions. The main component of DenseNet, dense block, is shown in Fig. 4.

B. EYE-TRACKING DATA AND USER SURFACE

The eye-tracking algorithm is based on a web camera mounted on a laparoscopic monitor, as shown in Fig. 5. The web camera captures the user's face images when the user is gazing at the laparoscopic monitor, and these images can be used as the CNN model input. The flow chart of the eye-tracking algorithm is shown in Fig. 6.

The monitor is divided into 36 blocks; the user can specify the block, or the so-called region of interest (ROI), by gazing at the point in the laparoscopic monitor that the user interested . Once the ROI is selected, this region will be enlarged, and the user can select the ROI of the enlarged pictures until the ROI is small enough. Once the starting and ending regions have been specified, the surgical robot can be automatically driven from the starting region to ending region by using the method proposed in [1]. Fig. 7 illustrates the general concepts and operating flow of this approach. This control approach is named pick and place.

The surgical robot can also be controlled to move in 9 directions where the moving directions are specified by users using eye gaze information. The user's gaze directions are divided into 9 different directions instead of 36 directions. This control approach is named user command.

C. EYE GAZE DIRECTION CLASSIFICATION

We use the face-detector module in OpenCV to detect and extract faces from the images captured by a web camera mounted on a monitor. After the face region is detected,

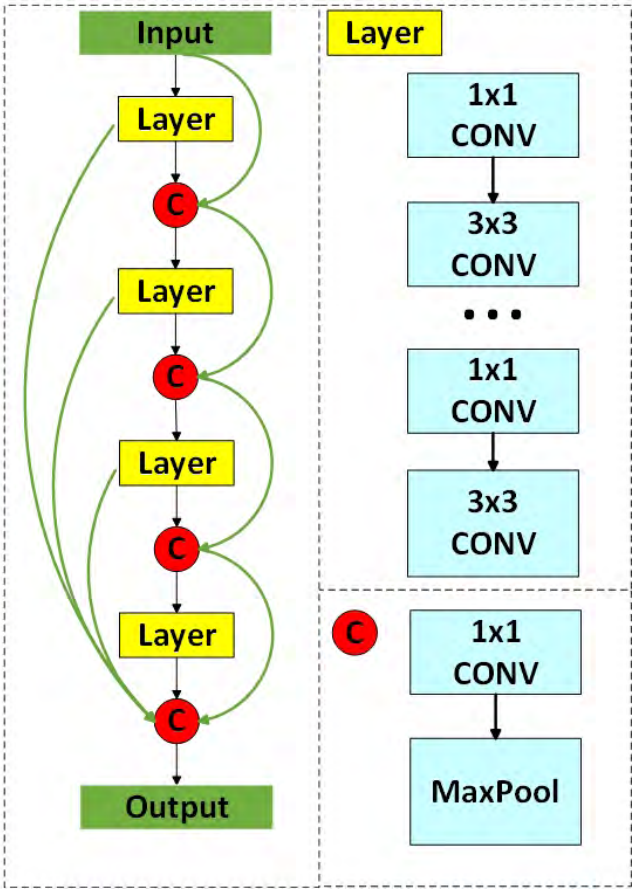

FIGURE 4. The first layer network accepts input and output k-layer feature maps and stacks the k-layer feature maps with the original input as a new input. The second layer of the network accepts the new input, outputs k feature maps, and then stacks it with the input to form a new input. The above steps are repeated 4 times. The output of the block is the concatenation of the output of each layer, and thus contains $4 \times k$ feature maps.

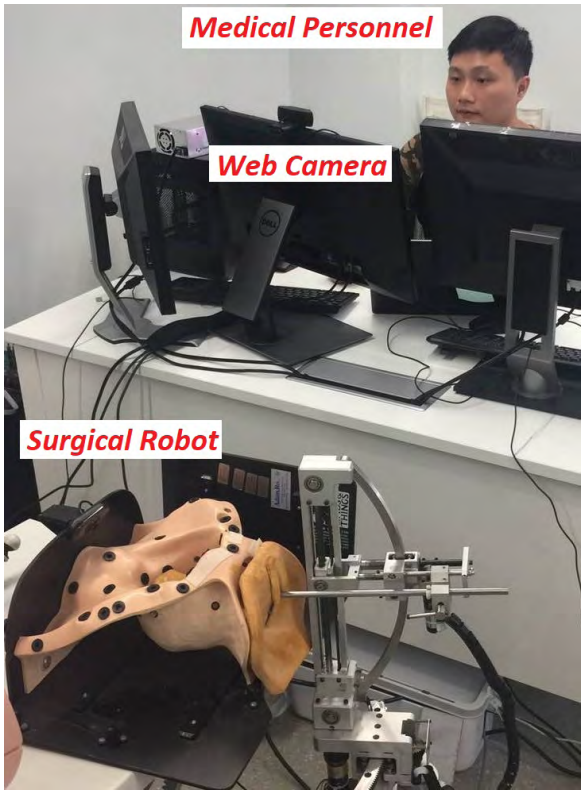

FIGURE 5. Medical personnel can control the robot with the help of a web camera mounted on a monitor.

an eye detection [40] algorithm is used to segment the eye region, after this step the size of the image becomes 128×128. The CNN architecture is normally composed of four parts, namely, the convolutional layers, the pooling layers, the non-linearity layers and the fully connected layers. The architecture of our eye-tracking CNN model is adopted from DenseNet [24]. DenseNet is much more computationally efficient as a result of feature reuse. First, we need to obtain the users eye-gazing images from our web camera mounted on the laparoscopic monitor. After obtaining the user's face image, double eyes detector cascaded using the Viola-Jones algorithm [40] is applied to detect the eyes and crop the eye region from the face images. Then, we normalize the eye images to ensure that the images pixel value has zero means. After all the above processes have been completed, this image is sent to our CNN model to estimate the user's eye gazing direction. A feature map is yielded by sliding a learnable filter over the RGB image and computing the dot product of the filter and the image. This feature map, which is also called the activation map, is an abstract of the image. The convolution not only preserves the spatial relation between pixels by learning features from the eye images, but it also extracts features from the image. After obtaining a features map, we replace all the negative pixel values with zero by applying a rectified linear unit (ReLU), which is a nonlinear activation function to reduce the exploding gradient or vanishing gradient problem. Maxpooling is applied to reduce the dimensionality of each feature map without losing the most important image information. This process is also called spatial pooling or downsampling. As shown in Fig. 8, the eye region images are sent to the CNN model while the output of the model is the user's eye gaze direction. The gaze direction of users is divided into 36 blocks as shown in Fig. 9.

D. EYE GAZING DATA SET

Deep learning algorithms require a considerable amount of labeled training data. A program is designed which can acquire and label eye images at the same time. Therefore, it is not trivial for us to introduce the method we used to produce and label the eye gaze dataset. With the method our paper presented, one can generate their own gaze estimate data with little time and effort. First, we divide the users monitor into an $m \times n$ block (in our paper, we divided the monitor into 6×6 blocks, as shown in Fig. 10), the user gazing in any position of the same block is considered as a gaze at the center of the block. By making this assumption, we convert the gaze position regression problem to a gaze direction prediction problem, which means that we will use a deep

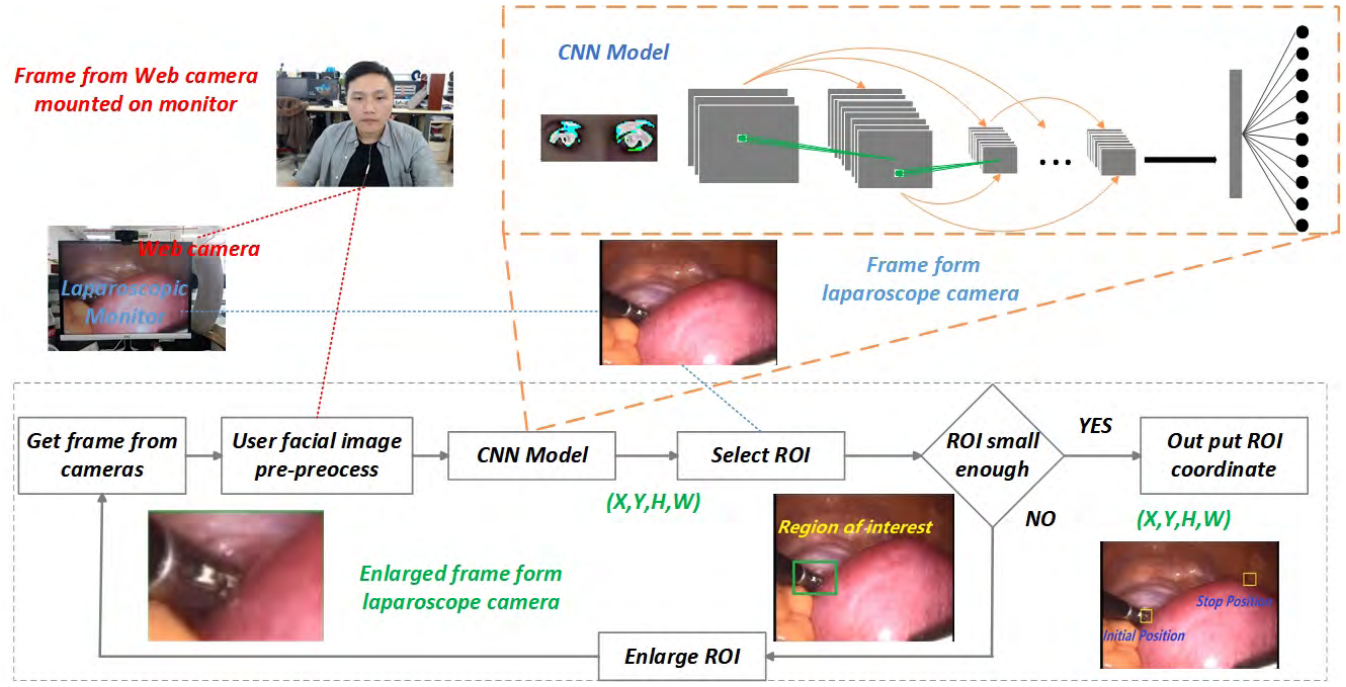

FIGURE 6. Overview of the proposed eye-tracking algorithm. The user can select the region of interest (ROI) by gazing at the monitor. The monitor displays the images captured by the laparoscope camera. When the doctor is looking at the laparoscopic image, the webcam mounted on the monitor captures the doctor's facial image. The facial image is sent to the CNN model. The model predicts which position the doctor is looking in the laparoscopic image.

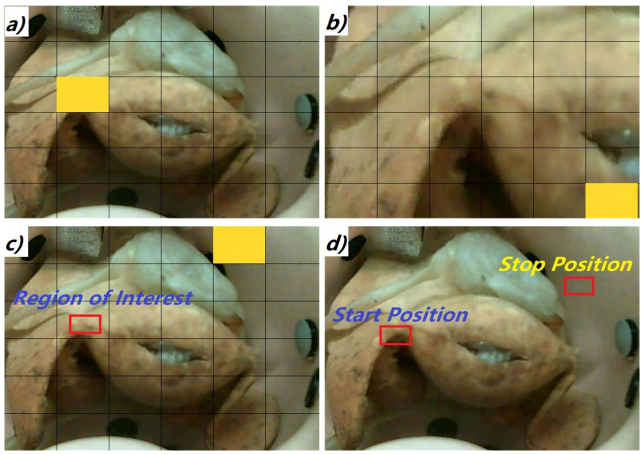

FIGURE 7. The user can select the region of interest (ROI) by gazing at the monitor. (a) The region marked with a green line for the region user gazing at. (b) Once the region has been selected, this region will be enlarged to fill with the whole monitor. (c) The user can select the ROI of the enlarged pictures again until the ROI is small enough. (d) Using the above process, the user can select the start position and destination of the surgical robot.

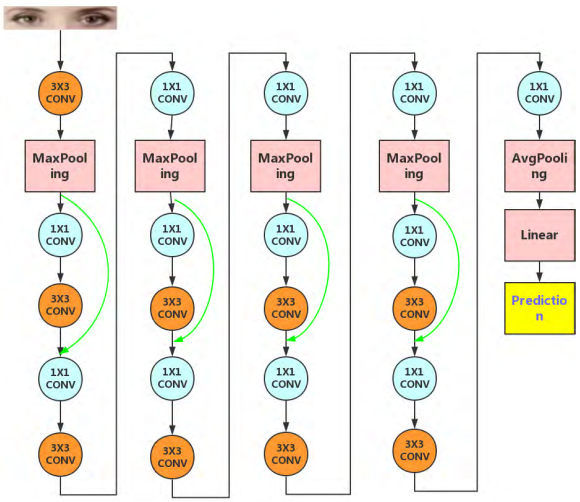

FIGURE 8. The CNN model we used to estimate the user's gaze position. Our CNN model has 21 layers, which can improve the performance of the estimate, and the trainable parameters' memory consumption is approximately 10KB (32 bit). Note that each "CONV" layer shown in the table corresponds to the sequence Convolution-BatchNormalize-ReLU.

learning algorithm to find a mapping relation from the users eye image to the users eyes gaze direction. Although this kind of approximation may cause some inaccuracy, which means our algorithm cannot predict the exact gazing position that the user is looking at but only the approximate region, this kind of assumption yield a considerable advantage. To predict the approximate region of the users gaze is much easier than that of predicting the exact position. We can train a neural network more easily using less labeled data. A neural network that can predict the exact point that a user is looking at requires a considerable amount of data that covers every position the user might look at on the screen, which is a considerable number if the network needs to be fully trained. However, if we divide the screen into 36 small blocks, that means we need much less training data to cover each small block, which

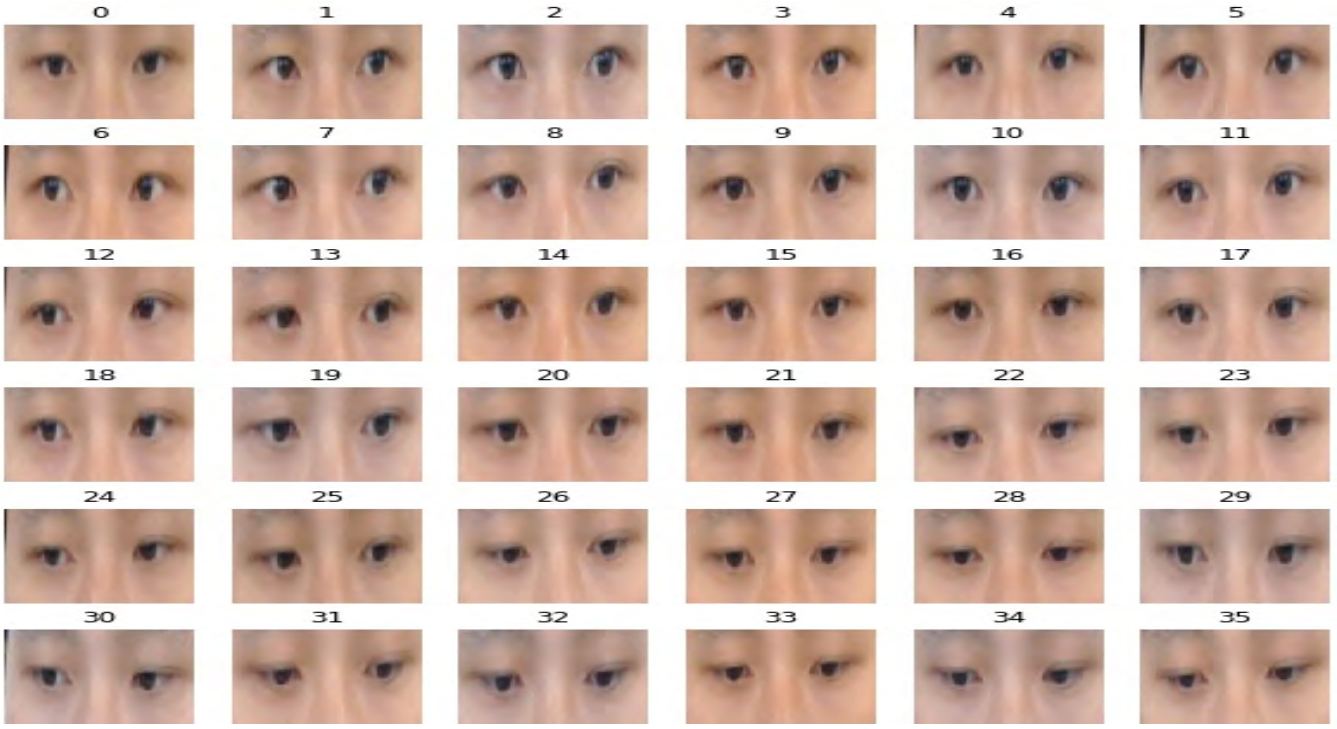

FIGURE 9. We divide the user's gaze direction into 36 blocks. The index above the picture shows the block index corresponding to the screen. For example, the left up eye picture that has index number 0 indicates that the user is gazing at the left top up of the screen.

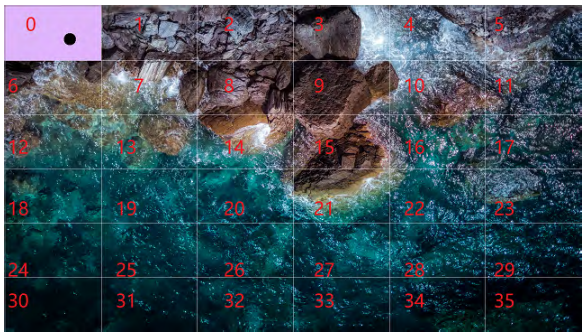

FIGURE 10. The computer screen is divided into 36 small blocks; if the user is looking at any position of the same block, it will be considered the same.

is much easier to realize. Following is the procedure for generating the eye gaze estimate neural network training dataset. When acquiring data, the user is requested to look at the small block of the screen that is highlighted. The positions of the highlighted block are specified. When the user is looking at the specified block, a web camera mounted on the screen automatically acquires the users image. Simultaneously, a small black circle randomly appears in the highlighted block to ensure that the users gaze dataset cover most of the locations on the screen.

E. GAZE ESTIMATE MODEL BASED ON LONG SHORT-TERM MEMORY NETWORK

In the real environment, people tend to have eye drift or blink when they look at the same point for a long time. The gaze direction prediction using a single image is susceptible to interference. Since the blinking movement is a natural human movement, we propose a gaze direction prediction model using multiframe images as input to avoid this interference. Fig. 11 shows the images captured by the user during blinking. Using these images to predict the direction of the gaze can lead to erroneous results. Specifically, the application background of our gaze estimation algorithm is in the operating room. Compared with other scenarios, the requirements for safety in the operating room environment will be higher. Therefore, we chose to use multiframe pictures in a certain period of time to predict the direction of the user's gaze during this period, rather than relying solely on a single picture. To utilize the timing information of multiframe pictures, we first extract the features of each frame using the CNN model trained in the previous section (Section II-C). Then, these image features are sequentially input into a bi-LSTM to encode the temporal features. At the same time, we introduced an attention mechanism to score the importance of each frame of the input image. The multiframe image features are then weighted averaged.

To avoid the user's blind eye causing gaze estimation errors when using the eyeball to control the surgical robot arm, thereby affecting the surgery, we use multiframe image information for prediction. The monocular camera used in the experiment can capture 30 images per second. To avoid excessive movement of the robot arm, the robot arm controller is set to accept no more than two control commands in one second. We use 200 milliseconds as a period of time,

FIGURE 11. The gaze estimate model based on a single frame image is susceptible to blinking. (a) User blinking process. (b) User blinking process. (c) User blinking process.

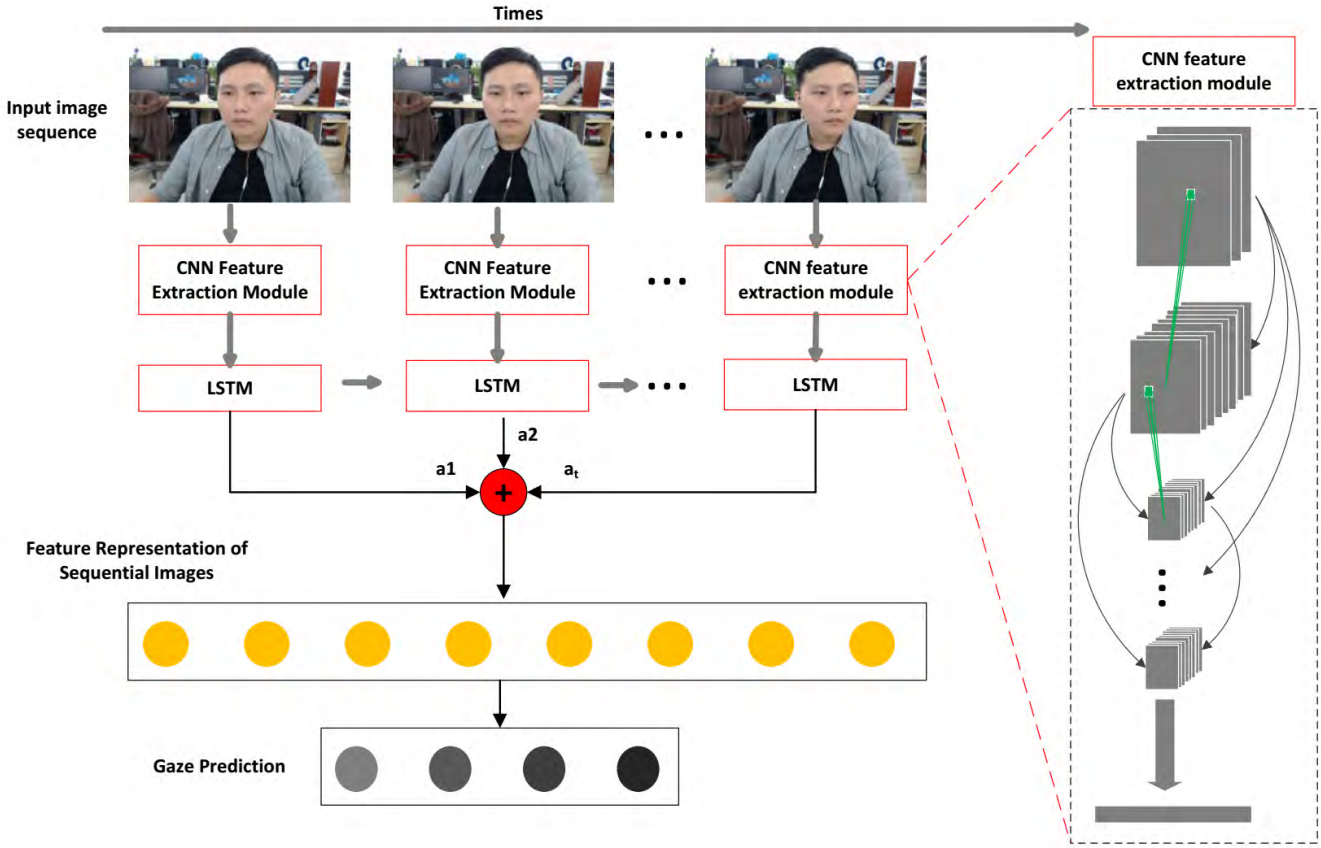

FIGURE 12. Gaze estimation model based on a long short-term memory network.

and use the 6 frames taken by the camera in this period as the input of the gaze estimation model. After face detection, eye detection and image normalization, each frame image is sent to a CNN-LSTM model for gaze direction prediction. We use the CNN model trained in Section II-C as the image feature extractor. Each frame of the image is first input into the CNN model for image feature extraction, and then these image features are sequentially fed into the long short-term memory network (LSTM) to obtain the prediction results. The processing flow of the gaze estimation model based on a multiframe image sequence is shown in Fig. 12. Each frame of the image captured by the camera is used for face detection, eye segmentation, data normalization, and then input into the CNN model for feature extraction. After completing the above process, the extracted image features are stored, waiting for the next frame image and repeating the process of feature extraction. When the stored image features exceed 6 frames, these image features are input into the LSTM network in time order to extract temporal features such as eye trajectory information. Then, the importance of each frame's hidden vectors is scored based on an attention mechanism. After determining the weight value of each frame image, the feature sum of the input image is calculated as the expression of the eye movement feature in this period. Finally, the feature is fed into the classifier to obtain the gaze direction predicted by the model during this time period.

III. RESULTS

In this section, we discuss the relationship between the network structure and the accuracy of the eye gaze estimate. The user's gaze direction is divided into 36 blocks. Using the methods described above, we produced approximately

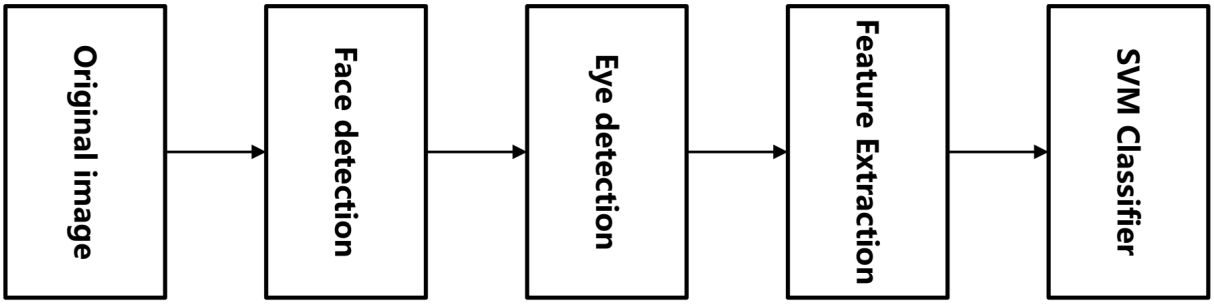

FIGURE 13. The processing flow of the gaze estimate algorithm.

20 thousand datasets with gaze direction labels that contain a user's eye information. In the following, various classifiers are used to classify these datasets and perform performance comparisons.

A. HOG FEATURE COMBINED WITH SVM CLASSIFIER BASELINE

The processing flow of the gaze estimation algorithm is shown in Fig. 13. After the original image of the user is captured by the camera, the image of the user's eye region is obtained through a process such as a face region segmentation, eye region segmentation, and image normalization. After obtaining the user's eye region image, we need to extract the features of the image, that is, extract the important details or features of the image, and remove some unnecessary details. There are mainly two kinds of image feature extraction algorithms. The first algorithm is based on the artificially designed image feature descriptor, and the second algorithm is based on the convolutional neural network. To test the effect of the artificially designed image feature extraction algorithm on the gaze direction prediction task, we use the histogram of oriented gradient (HOG) as the feature descriptor to describe the user's eye image.

The histogram of oriented gradient (HOG) feature is a feature descriptor used for object detection in computer vision and image processing. The HOG feature composes features by calculating and counting the gradient direction histograms of the local regions of the image. A support vector machine (SVM) is a supervised learning model commonly used in the field of machine learning, which is often used in classification or regression tasks. We chose the method commonly used in the field of computer vision, that is, using the HOG feature descriptor to describe the eye image, and then using the SVM to classify the feature descriptors of these images as a benchmark. As shown in Fig. 5, after capturing the image of the user with the camera mounted on the endoscope display, an original image with a resolution of 640×480 is obtained. Then, face detection is carried out on the original image to determine the region where the face exists in the image. After obtaining the position of the face region, human eye region detection is performed to obtain the position information of the human eye region in the image. In the process of face detection and eye detection, if there is no detection of human eyes or face, the current image will be abandoned, and the user's image will be retrieved. After the process of face detection and eye detection, an RGB image of the user's eyes with 128×128 image length and width is captured from the original image. After the user's eye image is obtained, HOG is used as the image feature descriptor to extract the eye image. After inputting a user's eye image with a resolution of $128 \times 128 \times 3$, hog features with a dimension of 1×34020 are obtained. After collecting a large number of eye images using the method in Section II-D, the user's gaze directions of these images are divided into 36 blocks. The role of the image feature descriptor is to convert the original eye image into an image feature representation. After converting the eye image dataset into a HOG feature as described above, all eye image features are divided into a test set and a training set according to a ratio of 2:8. The data of the training set is used to train the SVM classifier, and the data of the test set is used to evaluate the effect of the SVM classifier. After the SVM classifier is trained, the data of the test set are used to validate the prediction effect of eye gaze direction.

As seen in Fig. 14, the accuracy of the HOG combined with the SVM method on the 36 classification tasks is 17%, which is far less than methods based on deep learning. As seen from the classification results, the HOG feature descriptors perform poorly in describing the eye, resulting in very low accuracy of classification tasks. We visualize the HOG feature descriptor. From Fig. 15, we see that when the user changes gaze direction, only a small part of the corresponding eye image has changed; that is, the eyeball has changed its position. The HOG feature descriptors and did not identify this point, resulting in relatively poor classification results. To explore why the accuracy of the deep learning-based method is far greater than that of the HOG combined with SVM solution, we visualize the feature map of the first layer of convolution. From Fig. 16 we see that the convolution kernel has learned the position of the eyeball information, which is very important for the gaze estimation task, so the eyeball region is cut out by a convolution operation. By comparing the handcrafted feature descriptors (HOG) and the feature descriptors that the machine has learned from different tasks,

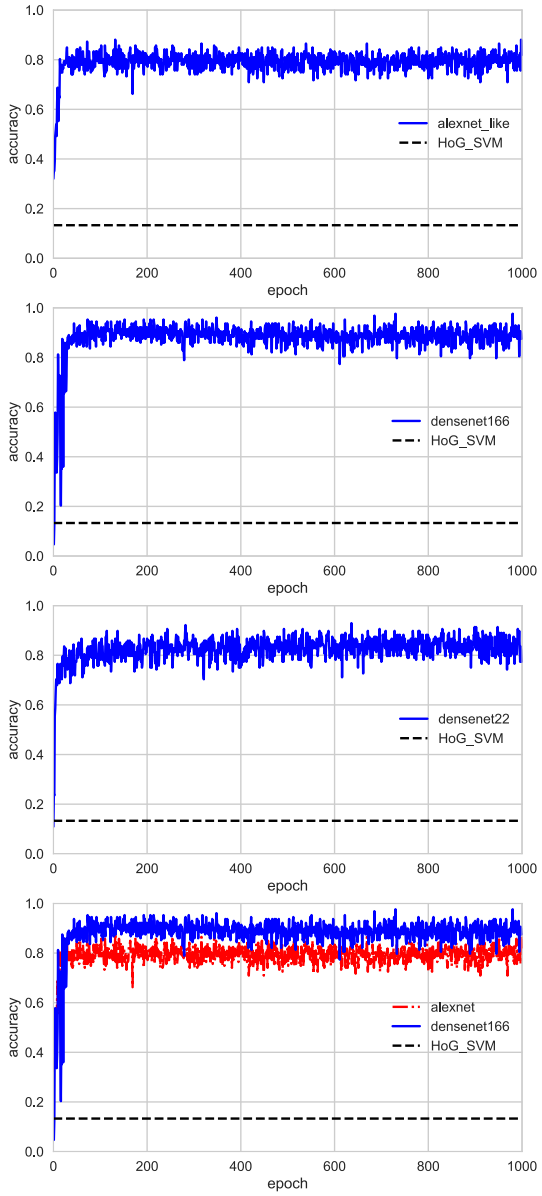

FIGURE 14. The accuracy of deep learning-based methods compared with the HOG+SVM baseline.

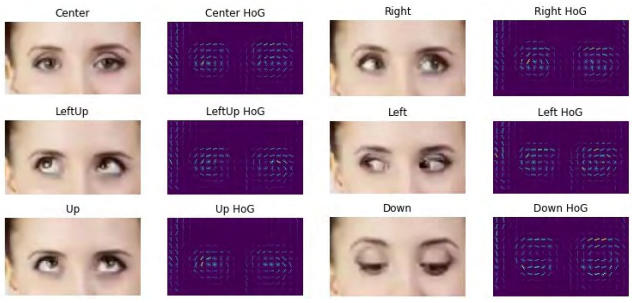

FIGURE 15. The visualization of HOG features shows that HOG features do not well describe the movement of the eyes.

we find that the latter has a stronger description of the image features and focuses on different regions based on different classification backgrounds.

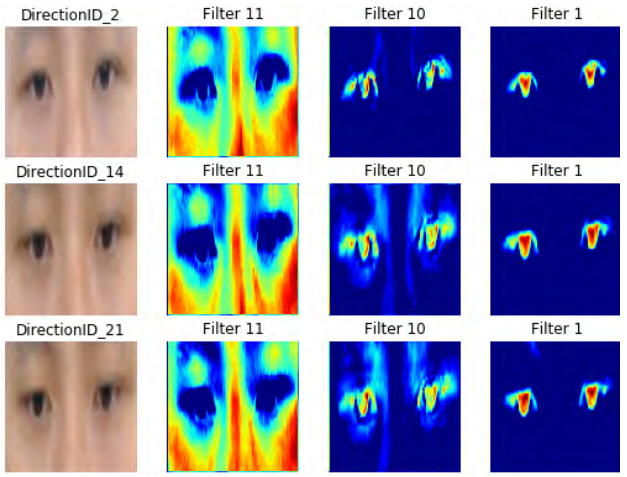

FIGURE 16. The convolution filter focuses on the eye location of the users when describing eye images. When the location changes slightly, the feature can also describe this change very well.

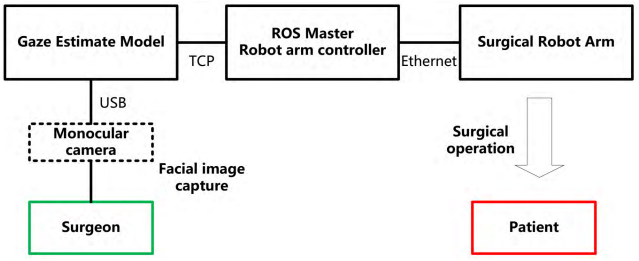

FIGURE 17. A block diagram of the relationship between the gaze estimate system and the robot arm control system.

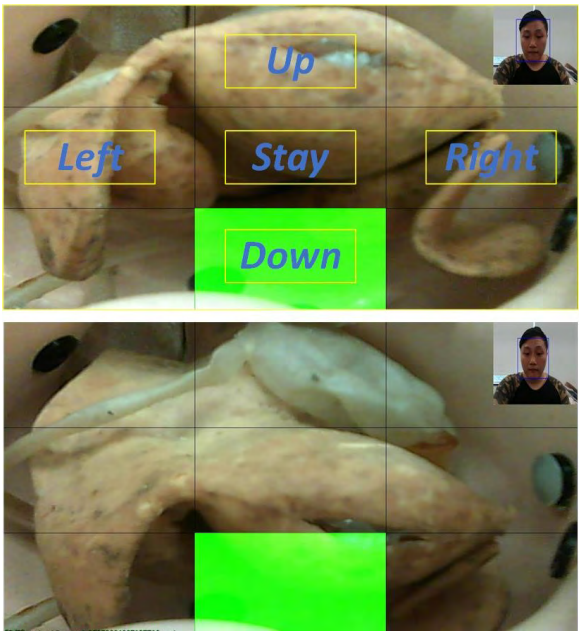

FIGURE 18. The user can control the robot with eye movements. When the user gazes at corresponding region on the screen, the robot moves accordingly.

B. CONTROLLING THE ROBOT WITH EYE MOVEMENT

A hysterectomy is a commonly used procedure in gynecology. Patients often need to remove the uterus because of

FIGURE 19. User command mode, users use eye movement to control robot arm movement. (a) Select the robot arm to move to the left. (b) Robot arm moves to the left. (c) User image at selection. (d) Select the robot arm to move to the right. (e) Robot arm moves to the right. (f) User image at selection. (g) Select the robot arm to move up. (h) Robot arm moves up. (i) User image at selection. (j) Select the robot arm to move down. (k) Robot arm moves down. (l) User image at selection.

uterine lesions. In the case of laparoscopic hysterectomy, an assistant is required to use a passive positioning device to move the patient's uterus to help the surgeon achieve a better operating angle or viewing angle. However, prolonged surgery can make assistants tired, leading to operational errors. We designed a hysterectomy robot for assisting doctors in surgery. To test the effect of eye movement control for the robot arm, we used the hysterectomy robot to perform experiments on a human pelvic model. The purpose of the eye gaze direction prediction is to provide a control signal for the movement of the robot arm as a means of human-computer interaction. Fig. 5 illustrates how a doctor can use the movement of the eye to control the movement of the arm. Doctors need to keep an eye on the endoscope screen during the operation. At this time, the monocular camera located above the monitor captures the real-time image of the doctor's face. After face detection, eye detection and image normalization, the eye image is sent into the prediction model to determine where the doctor is looking at the monitor. After obtaining the gaze direction information of the doctor, the gaze information is converted into a movement instruction of the robot arm through a special mapping relationship and then sent to the robot arm controller through the TCP/IP protocol. The relationship between the gaze direction

TABLE 1. Comparison of convolutional neural networks with different depths.

Model Name	Parameter Size(32 Bit)	Inference Time(ms/frame)	Network depth(layer)	Test Accuracy(%)
AlexNet_base	16.26 (MB)	19.3	7	77
Dense_22	10(KB)	15.6	22	82
Dense_46	752(KB)	73.8	46	85
Dense_86	2.77(MB)	139	86	87
Dense_166	6.5(MB)	237	166	90

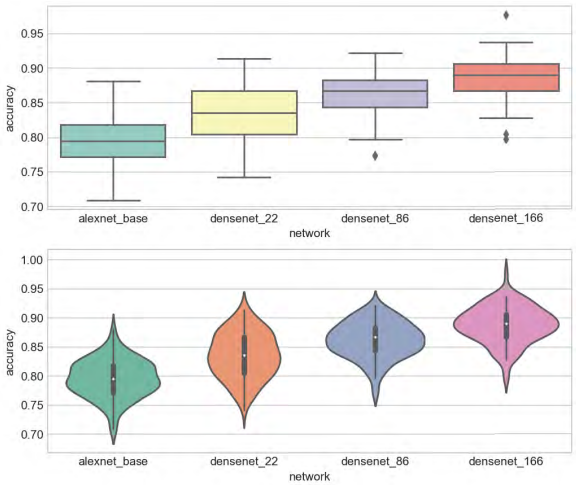

FIGURE 20. The CNN model with different convolutional network layers compared with the classification accuracy.

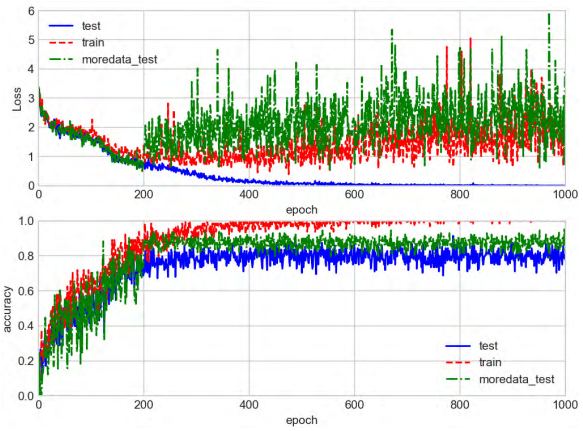

FIGURE 21. Overfitting of the neural network. The green dotted line indicates the model that was trained using more training data. It can be seen that this model performs better on the test set.

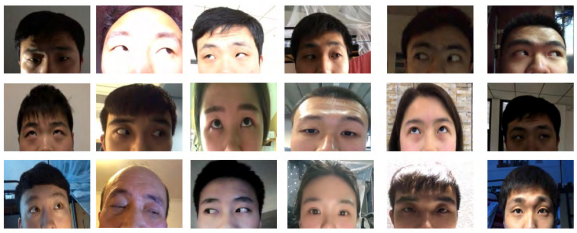

FIGURE 22. Sample image from the data set provided by [41].

prediction system and the robot arm control system is shown in Fig. 17.

To test the feasibility of our eye gaze direction prediction algorithm in a real environment, we performed the following experiments with the surgical robot. Our experimental environment is shown in Fig. 5. The experimental results show that the user can use the movement of the eye to control the movement of the robot. We first divide the screen into nine parts, and the user can move the robot arm in the corresponding direction while looking at different parts of the screen. The corresponding moving direction of each part of the screen is shown in Fig. 18. First, the camera captures the user's face image and then captures the user's eye area as the inputs of the CNN model. The model is responsible for predicting which part of the screen the user is looking at based on the user's eye picture. After obtaining the user's gaze direction information, we convert this direction information into a robot arm movement command and send this command to the robot arm controller. After completing the above steps, the robot arm moves in the specified direction according to the corresponding command. The experimental results show that our algorithm can control the robot arm in real time according to the user's eye movement, which can greatly reduce the burden on the user to operate the robot arm.

C. RESEARCH ON THE RELATIONSHIP BETWEEN NEURAL NETWORK DEPTH AND PREDICTION ACCURACY

From the above experiment, it can be seen that the deep learning-based methods perform much better than the traditional solution (HOG+SVM) in the gaze direction prediction task. The core idea of the deep learning methods is to allow the convolution kernel to learn the most suitable convolution descriptor by using a stochastic gradient descent update. To explore the effect of the depths of convolutional neural network layers on the performance of the CNN model, we compared the models with different convolutional neural network layers. These model are evaluated from the aspects of accuracy, computational time and parameter size. We compare the CNN models with different convolutional neural network layers, in which the depths of convolutional neural networks of the AlexNet-based model is 7 layers, the depths of the DenseNet_22 model is 22 layers, and the depths of the DenseNet_86 model network is 86 layers. As indicated in Fig. 20, when the network layers become deeper, the classification accuracy of the model becomes higher. When the number of network layers increased from 7 to 166, the classification accuracy rate also rose from 77% to 90%.

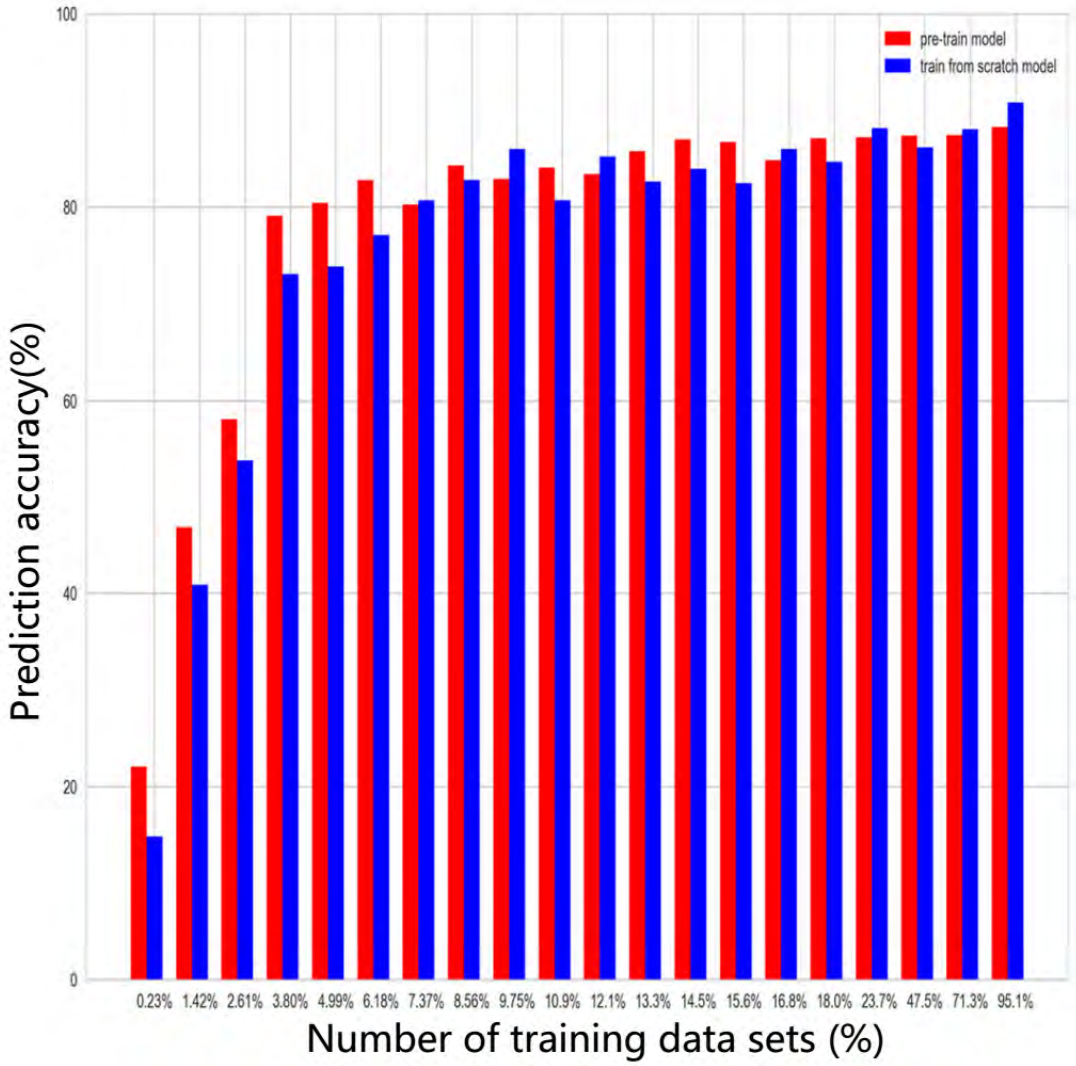

FIGURE 23. The influence of the number of training datasets on the effect of transfer learning.

Although more neural network layers can achieve higher classification accuracy, the real-time performance of network classification is lower; that is, the time to predict a map will become longer. Moreover, deeper networks require more training data and obtaining more labeled training data usually requires considerable manpower and time costs. At the same time, because of the vanishing gradients, it becomes more difficult to train deeper neural networks. Table 1 compares the convolutional neural networks with different depths in terms of classification accuracy, forward inference time and parameter memory.

The AlexNet-based model in Table 1 represents the recurrence of the model proposed by [41] and the classification model retrained under the training set of this task. It has a 7 layer convolution neural network. The storage space occupied by the trainable parameters of these networks is approximately 16 MB (32-bit). Such large network parameters limit the application of convolutional neural network algorithms to devices that are in short supply of computing resources such as mobile phones, FPGAs and embedded devices. The network using a dense connection (shown in Fig. 4) structure has a smaller number of convolution kernels because each neural network can be directly connected to the input or gradient. Therefore, compared to the DenseNet_166 and AlexNet_base models, it can be seen that even if the DenseNet_166 model has 166 neural network layers, the required trainable parameters are only 40% of the trainable parameters of the AlexNet base with 7 neural network layers, and the classification accuracy is 12% higher than that of the AlexNet-based model.

At the same time, we find that the DenseNet_22 model with a 22 neural network layers has a storage space of only 10KB, and it can predict 64 pictures per second, and the prediction accuracy is approximately 82% (36 categories). A model that only occupies 10KB of storage space can be easily ported to some mobile computing devices such as

embedded devices or FPGAs, greatly improving the range of application of the algorithm.

IV. DISCUSSION

As indicated in Fig. 20, the deeper the neural network expands, the higher the classification accuracy reaches. However, a deeper neural network requires more training data at the same time. Otherwise, it is easy to overfit; that is, the model has a high classification accuracy on the training set, but it has poor performance on the test set, as shown in Fig. 21. It can be seen that when there is more training data, the overfitting phenomenon is alleviated. However, the tagging of training data in supervised learning is time-consuming and laborious, so we have solved this problem in two aspects. First, we designed a program to automate annotation data, enabling faster and more convenient access to a large number of tagged training datasets. We request that when a certain area of the screen is highlighted, the user stare at the area. When the user looks at the area, the camera captures these pictures. Since the position of the highlighted region is known, the user's gaze direction data is captured. Using this method, we obtained approximately 50,000 labeled training data within 3 hours.

Second, using the method of transfer learning, the model is pretrained with other open source datasets, and the pretrained model parameters are migrated to the new model as the initialization parameter values. Transfer learning has proven to be an effective method for addressing classification problems where the features of one task can be generalized to another and the dataset is sparse. Due to the sparsity of training data and a shortage of computational resources for our problem, transfer learning and fine-tuning minimal parameters seemed to be a good way to approach this task. Considering that most of the data or tasks are relevant, we can learn the model parameters (understood as model learned knowledge) through transfer learning, which can speed up the convergence of the model and achieve better performance with fewer data. The open source training dataset in [41] is presented in their paper, which divides the eye gaze direction into 10 directions. This dataset contains approximately 100 thousand pictures. The dataset provided by the paper is compared with our dataset as shown in Fig. 22. We can see that our classification task is more complex than that of [41], but these two datasets have a high ratio of similarities. Thus, the knowledge that is learned from the dataset in [41] is also instructive for our classification tasks.

To verify that transfer learning reduces the amount of data required for model training, we used only a small part of the entire dataset to train the model. At the same time, the accuracy difference between training from the scratch model and the transfer learning model is compared under the same data quantity.

As indicated in Fig. 24, when the training set is relatively small, the training strategy using transfer learning can greatly enhance the performance of the model. However, it should be noted that when the training data is relatively small,

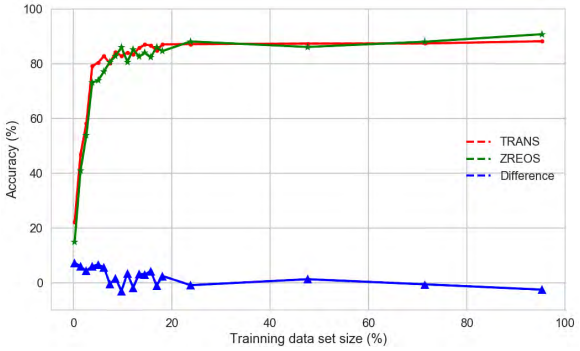

FIGURE 24. The red line represents the transfer learning model, and the green line represents the train from scratch model. The blue line represents the difference between the classification accuracy of the two models.

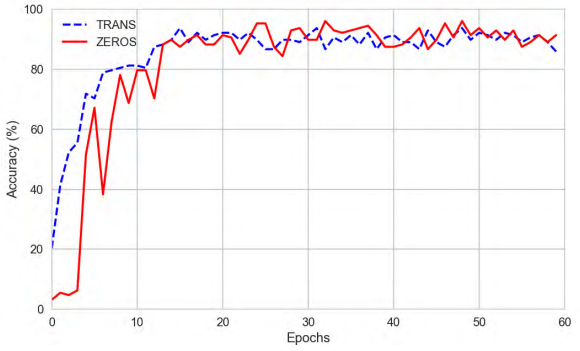

FIGURE 25. The blue line represents the transfer learning model and the red line represents the train from scratch model. The lateral axis represents the number of iterations of the training, and the vertical axis represents the classification accuracy of the model.

the accuracy rate is very low whether it is the transfer learning model or the train from scratch model, but the performance of the transfer learning model is much better. When the training dataset is relatively large, the advantages of the transfer learning model are less obvious, and the performance of the two models is not much different. From Fig. 24, it can be found that when the training set is relatively small, it is beneficial to use the pretraining model for initialization, and this can greatly improve the performance of the model. However, when the amount of training data is adequate, the advantage of this is not obvious.

As indicated in Fig. 25, when the amount of training data is sufficient, the transfer learning model does not improve the accuracy of the classification dramatically, but it accelerates the convergence of the model and reduces the time for the training.

V. CONCLUSION

In this paper, an eye-tracking algorithm that can be used to control surgical robots is designed. Users can specify the moving direction or position of the surgical robot using an eye gazing method. Since a considerable amount of labeled data is needed for training neural networks, an automatic gaze

picture capturing and labeling program is developed, which can generate a considerable amount of labeled data in a short time. Our gaze estimate algorithm is based on a very deep convolutional neural network. The architecture of this convolutional neural network is adopted from DenseNet, which is highly computationally efficient as a result of feature reuse. Our CNN model has very few trainable parameters, which makes it not only feasible to deploy on a field-programmable gate array (FPGA) and other hardware with limited memory but also reduces the need for a large amount of training data.

ACKNOWLEDGMENT

(Peng Li and Xuebin Hou contributed equally to this work.)

REFERENCES

[1] H. M. Yip, D. Navarro-Alarcon, and Y.-H. Liu, "An image-based uterus positioning interface using adaline networks for robot-assisted hysterectomy," in *Proc. IEEE Int. Conf. Real-Time Comput. Robot.*, Jul. 2017, pp. 182–187.

[2] A. Kar and P. Corcoran, "A review and analysis of eye-gaze estimation systems, algorithms and performance evaluation methods in consumer platforms," *IEEE Access*, vol. 5, pp. 16495–16519, 2017.

[3] K. A. F. Mora and J.-M. Odobez, "Person independent 3D gaze estimation from remote RGB-D cameras," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2013, pp. 2787–2791.

[4] T. Schneider, B. Schauerte, and R. Stiefelhagen, "Manifold alignment for person independent appearance-based gaze estimation," in *Proc. IEEE Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2014, pp. 1167–1172.

[5] Y. Sugano, Y. Matsushita, and Y. Sato, "Learning-by-synthesis for appearance-based 3D gaze estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1821–1828.

[6] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling, "Learning an appearance-based gaze estimator from one million synthesised images," in *Proc. 9th Biennial ACM Symp. Eye Tracking Res. Appl.*, 2016, pp. 131–138.

[7] E. Wood and A. Bulling, "Eyetab: Model-based gaze estimation on unmodified tablet computers," in *Proc. ACM Symp. Eye Tracking Res. Appl.*, 2014, pp. 207–210.

[8] Y. Zhang, A. Bulling, and H. Gellersen, "SideWays: A gaze interface for spontaneous interaction with situated displays," in *Proc. ACM SIGCHI Conf. Hum. Factors Comput. Syst.*, 2013, pp. 851–860.

[9] P. Li, X. Hou, L. Wei, G. Song, and X. Duan, "Efficient and low-cost deep-learning based gaze estimator for surgical robot control," in *Proc. IEEE Int. Conf. Real-Time Comput. Robot.*, Aug. 2018, pp. 58–63.

[10] K. A. Funes-Mora and J.-M. Odobez, "Gaze estimation in the 3D space using RGB-D sensors," *Int. J. Comput. Vis.*, vol. 118, no. 2, pp. 194–216, 2016.

[11] F. Lu, T. Okabe, Y. Sugano, and Y. Sato, "Learning gaze biases with head motion for head pose-free gaze estimation," *Image Vis. Comput.*, vol. 32, no. 3, pp. 169–179, 2014.

[12] Y. Sugano, Y. Matsushita, Y. Sato, and H. Koike, "Appearance-based gaze estimation with online calibration from mouse operations," *IEEE Trans. Human-Mach. Syst.*, vol. 45, no. 6, pp. 750–760, Dec. 2015.

[13] K. Tamura, R. Choi, and Y. Aoki, "Unconstrained and calibration-free gaze estimation in a room-scale area using a monocular camera," *IEEE Access*, vol. 6, pp. 10896–10908, 2018.

[14] K. Krafka *et al.*, "Eye tracking for everyone," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2176–2184.

[15] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4511–4520.

[16] Q. He *et al.*, "ainen, "OMEG: Oulu multi-pose eye gaze dataset," in *Proc. Scand. Conf. Image Anal.* Cham, Switzerland: Springer, 2015, pp. 418–427.

[17] K. A. F. Mora, F. Monay, and J.-M. Odobez, "EYEDIAP: A database for the development and evaluation of gaze estimation algorithms from RGB and RGB-D cameras," in *Proc. ACM Symp. Eye Tracking Res. Appl.*, 2014, pp. 255–258.

[18] B. A. Smith, Q. Yin, S. K. Feiner, and S. K. Nayar, "Gaze locking: Passive eye contact detection for human-object interaction," in *Proc. ACM Symp. User Interface Softw. Technol.*, 2013, pp. 271–280.

[19] Q. Huang, A. Veeraraghavan, and A. Sabharwal, "TabletGaze: Dataset and analysis for unconstrained appearance-based gaze estimation in mobile tablets," *Mach. Vis. Appl.*, vol. 28, nos. 5–6, pp. 445–461, 2017.

[20] G. P. Mylonas, A. Darzi, and G. Z. Yang, "Gaze-contingent control for minimally invasive robotic surgery," *Comput. Aided Surg.*, vol. 11, no. 5, pp. 256–266, 2006.

[21] D. P. Noonan, G. P. Mylonas, J. Shang, C. J. Payne, A. Darzi, and G.-Z. Yang, "Gaze contingent control for an articulated mechatronic laparoscope," in *Proc. IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatronics (BioRob)*, Sep. 2010, pp. 759–764.

[22] K. Fujii, A. Salerno, K. Sriskandarajah, K.-W. Kwok, K. Shetty, and G.-Z. Yang, "Gaze contingent Cartesian control of a robotic arm for laparoscopic surgery," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 3582–3589.

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[24] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4700–4708.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[26] R. K. Srivastava, K. Greff, and J. Schmidhuber. (2015). "Highway networks." [Online]. Available: https://arxiv.org/abs/1505.00387

[27] G. Larsson, M. Maire, and G. Shakhnarovich. (2016). "FractalNet: Ultra-deep neural networks without residuals." [Online]. Available: https://arxiv.org/abs/1605.07648

[28] S. Yang and D. Ramanan, "Multi-scale recognition with DAG-CNNs," in *Proc. IEEE Int. Conf. Comput. Vis.*, May 2015, pp. 1215–1223.

[29] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 447–456.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[31] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Sep. 2015, pp. 1–9.

[32] S. Zagoruyko and N. Komodakis. (2016). "Wide residual networks." [Online]. Available: https://arxiv.org/abs/1605.07146

[33] Z. Liao and G. Carneiro. (2015). "Competitive multi-scale convolution." [Online]. Available: https://arxiv.org/abs/1511.05635

[34] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 4278–4284.

[35] D. A. Gudovskiy and L. Rigazio. (2017). "ShiftCNN: Generalized low-precision architecture for inference of convolutional neural networks." [Online]. Available: https://arxiv.org/abs/1706.02393

[36] P. Goyal *et al.* (2017). "Accurate, large minibatch SGD: Training ImageNet in 1 hour." [Online]. Available: https://arxiv.org/abs/1706.02677

[37] P. Li, J. Xie, Q. Wang, and W. Zuo, "Is second-order information helpful for large-scale visual recognition?" in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Jun. 2017, pp. 2070–2078.

[38] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 646–661.

[39] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[40] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, 2001, p. 511.

[41] C. Zhang, R. Yao, and J. Cai, "Efficient eye typing with 9-direction gaze estimation," *Multimedia Tools Appl.*, vol. 77, no. 15, pp. 19679–19696, 2018.

PENG LI (M'18) received the B.Eng. degree in mechanical engineering from North Eastern University, Shenyang, China, in 2004, and the Ph.D. degree in mechatronics from the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, in 2010. From 2010 to 2015, he was a Postdoctoral Researcher and then a Research Associate with the Department of Mechanical and Automation Engineering. He is currently an Assistant Professor in mechanical and automation engineering, Harbin Institute of Technology (Shenzhen). His research interests include developing surgical robots, medical devices, and novel mechanisms.

XUEBIN HOU received the B.Eng. degree in automation engineering from North Eastern University, Qinhuangdao, in 2016. He is currently pursuing the degree in mechanical and automation engineering with the Harbin Institute of Technology (Shenzhen), Shenzhen, China. His research interest includes developing algorithms for surgical robots.

XINGGUANG DUAN received the B.Eng. degree in mechanical engineering from the Hebei University of Technology, Tianjin, China, in 1988, and the Ph.D. degree in mechatronics engineering from the Beijing Institute of Technology, Beijing, China, in 2009, where he is currently a Full Professor and a Doctoral Supervisor. His research interests include the fields of medical robotics, mobile robots, and bionic mechanism and control.

HIUMAN YIP (S'14) received the B.Eng., M.Phil., and Ph.D. degrees in mechanical and automation engineering from The Chinese University of Hong Kong, Hong Kong, in 2010, 2013, and 2016, respectively. Her research interest includes surgical robot control.

GUOLI SONG received the Ph.D. degree from the University of Chinese Academy of Sciences, in 2016. Since 2016, he has been a Research Assistant with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, where he became an Associate Professor, in 2017. His current research interests include the investigation and control of surgical robot and medical image registration.

YUNHUI LIU (S'90–M'92–SM'98–F'09) received the B.Eng. degree in applied dynamics from the Beijing Institute of Technology, Beijing, China, in 1985, the M.Eng. degree in mechanical engineering from Osaka University, Osaka, Japan, in 1989, and the Ph.D. degree in mathematical engineering and information physics from the University of Tokyo, Tokyo, Japan, in 1992.

He was with the Electrotechnical Laboratory, Ministry of International Trade and Industry, Ibaraki, Japan, from 1992 to 1995. Since 1995, he has been with The Chinese University of Hong Kong (CUHK), Hong Kong, where he is currently a Professor with the Department of Mechanical and Automation Engineering and the Director of the CUHK T Stone Robotics Institute. He is also visiting the State Key Laboratory of Robotics Technology and System, Harbin Institute of Technology, Harbin, China, and is the Director of the Joint Centre for Intelligent Sensing and Systems, National University of Defense Technology, Hunan, China, and CUHK. He has published more than 200 papers in refereed journals and refereed conference proceedings. His research interests include visual servoing, medical robotics, multifingered robot hands, mobile robots, and machine intelligence.

Dr. Liu has received numerous research awards from international journals and international conferences in robotics and automation and government agencies. He served as an Associate Editor for the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION and as the General Chair for the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. He is the Editor-in-Chief of *Robotics and Biomimetics* and an Editor of *Advanced Robotics*. He was listed in the Highly Cited Authors (Engineering) by Thomson Reuters, in 2013.