Methods for location and recognition of chess pieces based on machine vision

Yongfeng Yang ,Yaoqi Wang ,Xiaopeng Wang
School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou, China
3220465944@qq.com, 3342271810@qq.com, Wangxiaopeng@mail.lzjtu.cn
Corresponding Author: Yongfeng Yang Email:3220465944@qq.com

Abstract—**The positioning and recognition of chess pieces is the key to the chess robot's recognition of games. Aiming at the positioning and identification of chess pieces a method of centroid positioning of the maximum connected components and a method of convolutional neural network identification were proposed. First, the pieces were pre-segmented based on the HSV color space, and then the segmentation results were eroded and dilated to obtain the maximum connected components of the pieces, and the centroid coordinates of the connected components were calculated as the positioning results of the pieces. Finally the chess pieces were identified using the trained convolutional neural network. The network used a variety of techniques to reduce the number of parameters under the premise of ensuring a high recognition rate so that it could be deployed on resource-constrained devices. The results show that the average positioning error of the method on a chessboard with a size of 28cm and 28cm is 0.48mm, the average positioning time is 11ms and the recognition accuracy of chess pieces is 98.7%.**

Keywords—image processing; convolutional neural network; chinese chess robot; location of chess pieces; chess pieces recognition

I. INTRODUCTION

As an entertainment robot, the chess robot can observe and recognise chess games like a human, think independently about chess strategies and move a piece using its robot arm, which has some research value. The positioning and recognition of chess pieces is the key to achieving the function of a chess robot. Early chess robots used hardware circuits to locate and identify pieces, for example, using RFID to locate pieces [1]. However, this hardware-based approach is complex in terms of equipment and difficult to apply universally. Image processing and machine learning technologies have gradually replaced the previous hardware circuit model as a hot topic of research due to their high intelligence and simple equipment requirements.

Piece positioning is the key to accurate piece capture by robots. Since chess pieces are round or have circles printed on them, Zhu Chenxi,Gao Junwei,Fang guodong,Kong Deshuai used the Hough circle detection algorithm to detect circles to locate chess pieces, but the Hough circle detection algorithm is sensitive to light [2]. Chen Guoliang,Li Conghao,Ge Kaikai used differential images to get the coordinates of the moving pieces, but more steps increased the algorithm running time [3].

Pawn recognition is the key to the robot's ability to perceive the game and make a move strategy. The characters on the pieces are complex and have different rotation angles, which makes the piece recognition difficult. Feng Yuanhua,Wang Sihua,Liu Ning,Wang Gao used the pattern matching method to recognise chess pieces [4]. Han Xie,Zhao Rong,Sun Fusheng used a convolutional neural network to identify chess pieces, but the structure of this network is complex and the running time is too long [5].

To address the above problems that the positioning of chess pieces is inaccurate and inefficient, a localization method based on the centroids of the maximum connected components of chess pieces is proposed. To address the problems of low accuracy and efficiency of traditional chess character recognition, a convolutional neural network-based chess character recognition algorithm is proposed. The proposed convolutional neural network can be deployed on resource-constrained devices by reducing the number of layers of convolutional and pooling layers, decreasing the size of convolutional kernels, and using a stack of small convolutional kernels instead of large convolutional kernels, resulting in a smaller number of network parameters. Experiments have shown that convolutional neural networks are more accurate in recognising chess pieces than traditional methods.

II. GENERAL FLOW OF THE ALGORITHM

The chess piece positioning and recognition algorithm: the chess board image is first preprocessed and the chess pieces are located, then the chess pieces are segmented according to the positioning results and sent into a pre-trained convolutional neural network for recognition. The algorithm consists of three steps: image acquisition, chess pieces positioning and chess pieces recognition. The overall algorithm flow chart is shown as Fig. 1.

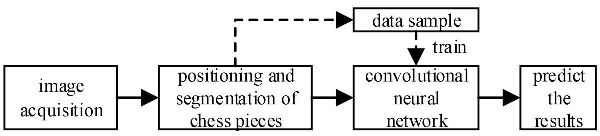

Fig. 1. The total algorithm flow chart.

2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC) | 978-1-6654-7968-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/IMCEC55388.2022.10020012

III. PIECE POSITIONING AND SEGMENTATION

The image is acquired and processed as follows: 1)the image is thresholded according to the colour features of the red and black chess pieces in HSV space; 2) the image is denoised, greyed out and binarized; 3) morphological operations are performed on the image, first by erosion operations to remove the effect of noise, followed by expansion operations to join the various parts of the Chinese characters together to find the maximum connected components of the chess pieces; 4) the centroids of the connected components are calculated and the pixel coordinates of the chess pieces are obtained, followed by the conversion of the pixel coordinates to the physical coordinates of the pieces.

A. Piece Pre-segmentation and Denoising

This paper proposes the use of chess piece colour features to preprocess and segment chess pieces. Chess pieces have two colours, red and black. H, S and V in the HSV colour space denote hue, saturation and brightness respectively, which are the same as the human intuitive colour perception. The three HSV components of the chess pieces are analysed and the pieces are found to have the following characteristics: the H range of the red pieces is [0,10] and the V range of the black pieces is [0,70], so that the red pieces are thresholded by the H component and the black pieces are thresholded by the V component. Image denoising is performed by median filtering. Take the segmentation of the red pieces using the H component as an example. The mask is defined according to formula (1), with values within the H-component range set to 255 and values outside the range set to 0. In formula (1), M represents the mask and $I_{\mathrm{H}}(x,y)$ represents the H component value of the image at a point (x,y). The mask is then compared with the original Fig. 2(a) by bit and the result is shown as Fig. 2(b).

$$M=\begin{cases}255, I_{\mathrm{H}}(x,y)\in[0,10]\\ 0, I_{\mathrm{H}}(x,y)\notin[0,10]\end{cases} \tag{1}$$

B. Finding the Piece Connected Domain

As shown in Fig. 2(c), the segmented pieces are greyed out and binarised, followed by a morphological opening operation, where the pieces are first eroded to remove noise, and then a dilation operation is performed to join the individual components of the Chinese characters together into a connected domain, yielding the result as Fig. 2(d).

C. Calculate the Coordinates of the Chess Piece Center

This paper uses the centroid formula for the connected domain to find the centroid coordinates of the pieces. The pixel intensity of a digital image at pixel point (x,y) is $array(x,y)$, and its geometric moment m_{ji} of $j+i$ order is calculated as formula (2).

$$m_{ji}=\sum_{x,y}(array(x,y)\cdot x^{j}\cdot y^{i}) \tag{2}$$

The mass centre coordinate of a connected domain is (x,y) and it is related to its geometric moments as follows

$$x=m_{10}/m_{00} \tag{3}$$

$$y=m_{01}/m_{00} \tag{4}$$

The pixel intensity values within the connected domain can be considered as a constant, so that the calculated mass centre coordinates are also the form centre coordinates. The Green's formula is used to calculate the centroid coordinate of the connected domain. As shown in Fig. 2(e), the green dots in the diagram represent the computed centroids of the connected domains of the pieces, so that the pieces can be located using the centroids.

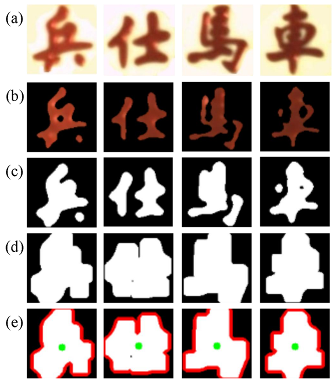

Fig. 2. Chess positioning flow.(a) Original character;(b) character segmentation image;(c) binary image;(d) connected component image;(e) connected component centroid image.

The centroid coordinates of the pawns can be converted into physical coordinates of the pawns in the plane of the board according to a certain proportional relationship. If the ratio of the actual length to the pixels length of the board is k , the relationship between the physical coordinates (X,Y) of the pieces and the pixel coordinates (x,y) of the pieces is shown in (5) and (6). After the pawns are positioned, the image of the pawns is captured with the centroid as the center and input to the convolutional neural network for recognition.

$$X=x\times\mathrm{k} \tag{5}$$

$$Y=y\times\mathrm{k} \tag{6}$$

IV. PIECE CHARACTER RECOGNITION

Convolutional neural networks are used for their excellent performance in a variety of image recognition tasks. Convolutional neural networks are first trained on a large amount of chess piece data and the trained model can be used to predict similar images.

The history of the development of convolutional neural networks has seen the emergence of excellent network structures such as AlexNet and VGGNet [6~7].The Alexnet model consists of five convolutional layers, three pooling layers and three fully connected layers; the relu nonlinear activation function is added to enhance the nonlinear representation of the model. The AlexNet network uses large convolutional kernels to extract information from a neighbourhood of the input image, while VGGNet uses a stack of small convolutional kernels to achieve the same effect as the large ones. For example, a superposition of two 3 × 3 convolutional kernels has the same perceptual field as a 5 × 5 convolutional kernel. The use of multiple small convolutional kernels superposition instead of large convolutional kernels reduces the number of network parameters and enhances the nonlinear mapping of the network, thus improving the expressive power of the network [7]. Based on the network structure and design ideas of AlexNet and VGGNet, this paper proposes a convolutional neural network for classifying chess characters.

The network constructed in this paper consists of five convolutional layers C1, C2, C3, C4 and C5, two maximum pooling layers P1 and P2, and three fully-connected layers F1, F2 and F3. The network is divided into three parts according to function: the input layer, the intermediate layer and the fully connected layer. The middle layer consists of alternating convolutional and pooling layers. Based on the idea of using small convolutional kernels superposition instead of large convolutional kernels, the VGGNet network uses a stack of three 3×3 convolutional kernels, and the convolutional effect is the same as 7 × 7 convolutional kernels. The network structure of this paper is shown in Fig. 3. The information on the parameters of the convolutional, pooling and fully connected layers is shown in Table 1. Taking C1 as an example, the kernel size 5 × 5 × 1 × 8 indicates the convolution kernel size is 5 × 5, the convolution kernel depth is 1, and the number of output channel is 8. A 96 × 96 pixels image with a channel number of 1 becomes a 92 × 92 pixels image with a channel number of 8 after C1 processing. The kernel size of P1 indicates a pooling window size of 2×2.

(1) The input layer is a binary image with a size 96 × 96. The RGB three-channel image is pre-processed with filtering, greyscaling and binarisation, followed by cropping to generate a fixed size binary image and the results will be input to the network, in order to reduce the network parameters and decrease the model size.

(2) The intermediate layer is composed of alternating convolutional and pooling layers. The Relu function is computationally efficient and solves the gradient disappearance problem to a certain extent, allowing the network to converge quickly. The pooling layer uses maximum pooling.

(3) The fully connected layer is a 3-layer classification network: the first layer F1 is the column vector obtained by expanding all the feature maps obtained from pooling, the second layer F2 is the Relu activation function mapping layer, and the last layer F3 outputs the final result, using the Softmax activation function to output 11 categories. The Softmax activation function expression is shown in (7), which converts the model output into category probabilities. The category with the highest probability among the 11 categories is finally selected as the final output. The loss function used is shown in (8). The Adam optimiser is used to make the loss function decreasing. In (7) and (8) x_i is the output of the ith node, C is the total number of categories(C =11), i and j are the category indexs ($i = 1, 2, \cdots, \mathrm{C}$), n is the image index, N is the number of images and y_i is the label corresponding to the ith category [5].

$$F_i(x) = \frac{\exp(x_i)}{\sum_{j=1}^{\mathrm{C}} \exp(x_j)} \tag{7}$$

$$CE(x) = -\frac{1}{\mathrm{N}} \sum_{n=1}^{\mathrm{N}} \sum_{i=1}^{\mathrm{C}} y_i \log F_i(x) \tag{8}$$

V. EXPERIMENTAL RESULTS AND ANALYSIS

The system uses an MV-EM gigabit network industrial camera with a two megapixel industrial lens for image acquisition. The chess pieces are 24 mm in diameter and the board size is 28 cm × 28 cm. The computer CPU is Intel i5-10500 at 3.1 GHz and the algorithms are verified by writing programs on the software development platform Visual Studio 2013 and PyCharm.

A. Piece Positioning Experiment

Five images of the chessboard were captured using this system, each image containing 32 pieces, for a total of 160 pieces for the positioning experiment. The actual coordinates were obtained by manually calibrating the centre of the circle and the actual coordinates were compared with the calculated coordinates obtained from the positioning results to obtain the positioning error. (X', Y') represents the actual coordinates and (X, Y) represents the calculated coordinates of the images. The positioning error R is calculated as shown in (9).

$$R = \sqrt{(X' - X)^2 + (Y' - Y)^2} \tag{9}$$

The experimental results show that the average positioning error of the chess pieces is 0.48 mm and the highest positioning accuracy reaches 0.08 mm, which is accurate. The average positioning time of the pieces is 11ms, which meets the requirements of fast positioning of chess robots.

B. Piece Recognition Experiment

(1) Piece dataset

CNN must be trained with a large amount of data in order to achieve good classification results. In this paper, in order to reduce the structural complexity and the number of parameters of CNN, 96×96 fixed size binary images of chess pieces are generated as the dataset. Data augmentation is performed for each binary image rotated 360 degrees to simulate the different rotation angles of pieces in a real application environment. Then each image is panned by 1, 2, 3 and 4 pixels in the left, right, top and bottom directions respectively, resulting in 991440 images. 80% of the whole dataset was input into the CNN as the training set with 24 iterations, and the remaining 20% was used as the validation set to evaluate the model. Taking the king, the advisor, the rook and the pawn as examples, the generated partial dataset is shown in Fig. 4.

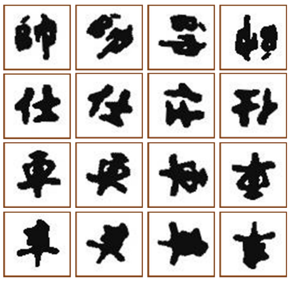

Fig. 4. Examples of chess data.

(2) Piece recognition results

In this paper, 30 pictures of chess games were input into the convolutional neural network for recognition experiments, each picture contained 32 pieces, and the results were classified by character as shown in Table 2. The number of correctly identified pieces was 948, with a recognition rate of 98.7%, which can meet the requirements of chess robot applications. The incorrect recognition occurred for the characters pawn and rook, which was analysed to be related to the fact that the outer contours of these two characters were too close to each other and the images were not clear enough.

TABLE II. RECOGNITION RESULT

Piece	Sum	Correct number	Error number	Recognition rate/%
R-king	30	30	0	100
B-king	30	30	0	100
R-bishop	60	60	0	100
B-bishop	60	60	0	100
Cannon	120	120	0	100
R-advisor	60	60	0	100
B-advisor	60	60	0	100
Horse	120	120	0	100
Rook	120	115	5	95.8
R-pawn	150	150	0	100
B-pawn	150	143	7	95.3
Total	960	948	12	98.7

(3) Comparison of positioning and recognition results

The experimental results of this paper were compared with the convolutional neural network-based positioning and recognition method of the [5], and the results obtained are shown in Table 3. From Table 3, it can be seen that the two performance indicators of this paper, the average localization error and the maximum localization error, are better than those of the [5]. The positioning time of this paper is 11ms, which is much smaller than the 212ms of [5], but the recognition rate of this paper is not much different from that of [5]. However, the network structure used in this paper is simpler, and the number of parameters and floating point operations (FLOPs) is much smaller than that in [5], which allows it to be deployed in resource-constrained embedded devices. The test results of this paper are therefore better than those of the [5] and our method has good application prospects.

TABLE III. COMPARISON OF TEST RESULT

Performance	This paper	Paper[5]
Average positioning error /mm	0.48	0.51
Maximum positioning error /mm	0.71	1.15
Positioning time/ms	11	212
Recognition rate/%	98.7	98.59
Average recognition time /ms	252	-
Order of magnitude for parameters	10^5	10^7
Order of magnitude for FLOPs	10^7	10^9

VI. CONCLUSIONS

This paper addresses the problem of inaccurate and inefficient chess piece positioning and recognition of Chinese chess robots, and proposes a positioning method based on the shape centre of the maximum connected domain of chess pieces and a recognition method based on convolutional neural network. The experimental results show that the method can achieve fast and accurate localization of chess pieces, with an average localization error of 0.48mm and a localization time of 11ms on a 28×28cm chess board; the convolutional neural network can accurately recognize chess pieces with an accuracy rate of 98.7%, and the number of network parameters is small by using various techniques. There is room for further improvement of the algorithm in terms of positioning and recognition time, which will be the target of the next research step.

REFERENCES

[1] Zhou L L. Research and Design of Robot System for Chinese Chess[D]. Shenyang:Northeastern University,2017.

[2] Zhu C X, Gao J W, Fang G D, et al. A plane control system of man - machine chess based on machine vision[J]. Manufacturing Automation,2021,43(05):93-98+137.

[3] Chen G L, Li C H, Ge K K. Research on fast capture and location of chess pieces in man-machine Chinese chess[J]. Journal of Huazhong University of Science and Technology(Natural Science Edition),2017,45(10):122-127.

[4] Feng Y H, Wang S H, Liu N, et al. Application of machine vision technology in design of chess playing intelligent robot[J]. Computer Engineering and Design, 2009,30(14):4.

[5] Han X, Zhao R, Sun F S. Methods for Location and Recognition of Chess Pieces Based on Convolutional Neural Network[J]. Laser & Optoelectronics Progress,2019,56(08): 081007.

[6] Krizhevsky A, Sutskever I, Hinton G. ImageNet Classification with Deep Convolutional Neural Networks[J]. Communications of the ACM, 2017,60(6):84-90.

[7] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.

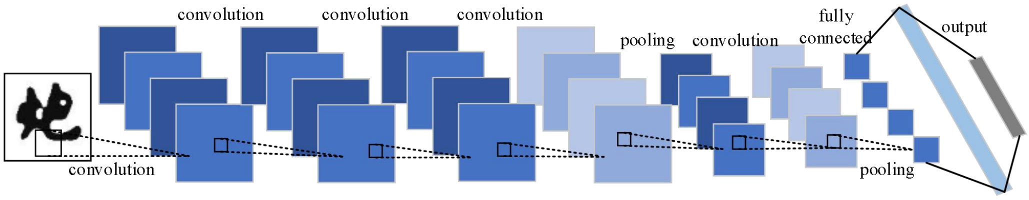

Fig. 3. Network structure.

TABLE I. NETWORK PARAMETERS

Layer name	Kernel size	Activation	Stride	Input size	Output size
C1	$5\times5\times1\times8$	Relu	1	96,96,1	92,92,8
C2	$3\times3\times8\times8$	Relu	1	92,92,8	90,90,8
C3	$3\times3\times8\times8$	Relu	1	90,90,8	88,88,8
C4	$3\times3\times8\times16$	Relu	1	88,88,8	86,86,16
P1	2×2	—	2	86,86,16	43,43,16
C5	$3\times3\times16\times16$	Relu	1	43,43,16	41,41,16
P2	2×2	—	2	41,41,16	20,20,16
F1	—	—	—	20,20,16	6400
F2	—	Relu	—	6400	128
F3	—	Softmax	—	128	11