sustainability

MDPI

Article

Development of a Chinese Chess Robotic System for the Elderly Using Convolutional Neural Networks

Pei-Jarn Chen, Szu-Yueh Yang, Chung-Sheng Wang, Muslikhin Muslikhin and Ming-Shyan Wang *

Department of Electrical Engineering, Southern Taiwan University of Science and Technology, Tainan 710, Taiwan; cpj@stust.edu.tw (P.-J.C.); Da720201@stust.edu.tw (S.-Y.Y.); ma620112@stust.edu.tw (C.-S.W.); muslikhin@uny.ac.id (M.M.)

* Correspondence: mswang@stust.edu.tw

Received: 23 March 2020; Accepted: 6 May 2020; Published: 13 May 2020

Abstract: According to the data from Alzheimer's Disease International (ADI) in 2018, it is estimated that 10 million new dementia patients will be added worldwide, and the global dementia population is estimated to be 50 million. Due to a decline in the birth rate and the development and great progress of medical technology, the proportion of elderly people has risen annually in Taiwan. In fact, Taiwan has become one of the fastest-growing aged countries in the world. Consequently, problems related to aging societies will emerge. Dementia is one of most prevailing aging-related diseases, with a great influence on daily life and a great economic burden. Dementia is not a single disease, but a combination of symptoms. There is currently no medicine that can cure dementia. Finding preventive measures for dementia has become a public concern. Older people should actively increase brain-protective factors and reduce risk factors in their lives to reduce the risk of dementia and even prevent the occurrence of dementia. Studies have shown that engaging in mental or creative activities that stimulate brain function has a relative risk reduction of nearly 50%. Elderly people should develop the habit of life-long learning to strengthen effective neural bonds between brain cells and preserve brain cognitive functions. Playing chess is one of the suggested activities. This paper aimed to develop a Chinese robotic chess system for the elderly. It mainly uses a camera to capture the contour of the Chinese chessman, recognizes the character and location of the chessman, and then transmits this information to the robotic arm, which will grab and place the chessman in the appropriate position on the chessboard. The camera image is transmitted to MATLAB for image recognition. The character of the chessman is recognized by convolutional neural networks (CNNs). Forward and inverse kinematics are used to manipulate the robotic arm. Even if the chessmen are arbitrarily placed, the experiment showed that their coordinates can be found through the camera as long as they are located within the working scope of the camera and the robotic arm. For black chessmen, no matter how many degrees they are rotated, they can be recognized correctly, while the red ones can be recognized 100% of the time within 90° of rotation and 98.7% with more than a 90° rotation.

Keywords: Chinese robotic chess system; convolutional neural network; dementia; forward and inverse kinematics

1. Introduction

The data from Alzheimer's Disease International (ADI) in 2018 estimate that 10 million new dementia patients will be added worldwide. This means that an average of one person will suffer from dementia every 3 s. The global dementia population is estimated to be 50 million, and by 2050 the

number will reach 152 million. In addition, the cost of care for dementia was estimated to be USD 1 trillion in 2018 and will double to USD 2 trillion by 2030 [1].

Due to a decline in the birth rate and the great development and progress of medical technology, the proportion of elderly people has risen annually in Taiwan. In fact, Taiwan has recently become one of the fastest-growing aged countries. Consequently, problems related to aging societies will emerge. Dementia is one of the prevailing age-related diseases with the greatest influence on daily life and the greatest economic burden. Based on the results of an epidemiological survey of dementia commissioned by the Taiwan Alzheimer's Disease Association in 2011 and the demographic data of the Ministry of the Interior at the end of December 2018, there were 3,433,517 elderly people aged 65 or older (14.56%), of which 626,026 had mild cognitive impairment (MCI), accounting for 18.23%, and 269,725 had dementia, accounting for 7.86% (of which 109,706 had extremely mild dementia) [1]. That is to say, there was about one case of dementia for every 12 people over 65 years of age, and there one case of dementia for every five people over 80 years of age.

Dementia is not a single disease, but a combination of symptoms. Its symptoms are not only memory loss, but also the degradation of other cognitive functions, including language ability, sense of space, computing ability, and the functions of judgment, abstract thinking ability, and attention. At the same time, symptoms such as disturbing behavior, personality changes, delusions, or hallucinations may occur. The severity of these symptoms is sufficient to affect patients' interpersonal relationships and workability. There is currently no medicine that can cure dementia, so how to prevent dementia has become a topic of public concern. As the research on dementia continues to progress, we have become more aware of the factors that help prevent or delay dementia. The public should actively increase brain-protective factors and reduce risk factors in their lives to reduce the risk of dementia and even prevent its occurrence. Studies have shown that engaging in mental or creative activities that stimulate brain functions can reduce the risk of developing dementia, with a relative risk reduction of nearly 50% [1]. Elderly people should develop the habit of life-long learning to strengthen effective neural bonds between brain cells and preserve brain cognitive functions. Playing chess is one of the suggested activities.

Chinese chess is played with flat discs, and the total chessmen are divided into two parts, generally a red and a black side, each with seven different Chinese characters and 16 chessmen, as shown in Figure 1. It is a very popular two-player board game in Taiwan, China, and some other Asian countries. It is especially popular with retired seniors. When the elderly play chess, a robotic chess system including a simple and low-cost camera and a small robotic arm can be used to implement an automatic chess-placing system to help the elderly place the chessmen and reduce some chores. People may even play chess with a chess robot that includes the robotic chess system and software for playing.

In recent years, vision sensors and image processing technologies have been continuously developed. Their main applications consist of automatic manufacturing, product inspection, welding automation, packaging, and logistics. The vision system can obtain accurate object measurement through a camera, image processing, and system calibration, such that it can increase the ability of the robotic arm to detect the scene, track, and adapt to scene changes. In order to improve the accuracy and intelligent control, researchers applied visual recognition to robotics. The robotic arm uses the camera's focus to visually identify the center of gravity and direction of the workpiece [2] to carry out automatic picking and placing tasks and improve production efficiency [3].

Since the picking and placing task in the eye-to-hand configuration has many conveniences, its construction tends to be rigid, to reduce the possibility of displacement. Althloothi et al. [4] proposed multiple kernel learning (MKL) using an RGB-D camera to recognize human activities. With the same camera, Jalal et al. [5] used a shape and motion feature approach to detect human activities with sharp visual results. Song et al. [6] introduced robust features for depth video, and this work was achieved to approximate the object-centered feature. Unlike the previous studies, Ge et al. [7] retained the traditional Hough transform image processing technique to recognize the classifications of strawberry environments, although applied to the eye-to-hand configuration. In a

similar environmental application, Chen et al. [8] blended the geometry with the epipolar constraint to achieve 3-D recovery in the concise eye-to-hand manipulator for localization and recognition.

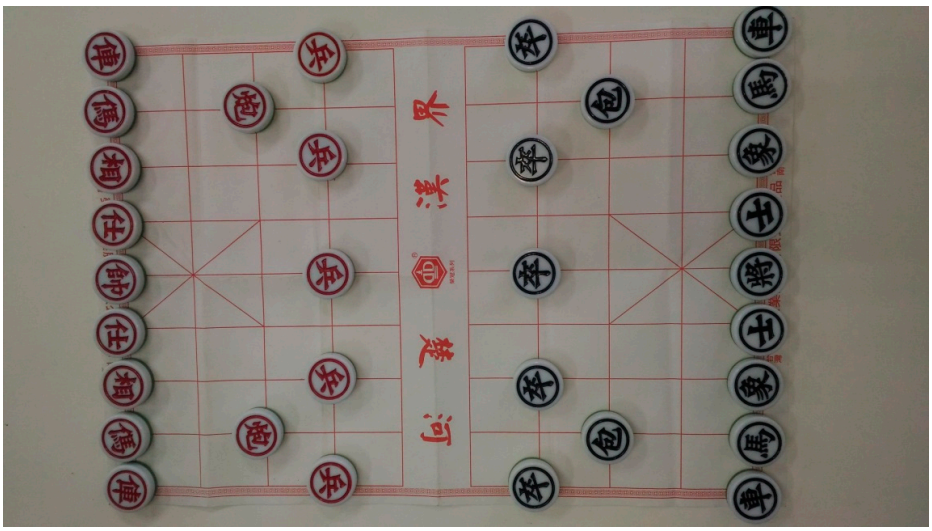


Figure 1. Chinese chess.

Before we applied convolutional neural networks (CNNs) in the Chinese chess game, previous studies focused massively on recognition and detection for applications such as healthcare, education, e-commerce, surveillance systems, and many others [9–15]. Regarding these strategies, Manwatkar et al. [16] introduced the process of converting images to text using document image analysis (DIA). A more sophisticated method was carried out by Lara et al. [17] and Jalal et al. [18] using human activity recognition (HAR). Although the method is not specifically used for text, its performance is quite good, with a recognition rate of 97.16%. In another paper [19,20], the hidden Markov model (HMM) was used to detect shapes and motion features. Thus, several methods for detecting moving targets have been introduced. However, in this paper, the chessman's target is not moving. Its position is very random, determined by the Chinese symbol.

General optical character recognition (OCR) begins by recognizing printed numbers and letters, and then develops to recognize the printed texts. Chinese chess has various font types and various characters. Wen [21] proposed an input image and database feature comparison method that consists of the noise filter, object extraction, normalization, feature calculation of the distance between the contour of the character and the center of the chessman, and maximum energy slop algorithm, for the Chinese chessmen. Seniman et al. [22] presented the backpropagation algorithm of a feed-forward neural network as well as direction feature extraction method by iterating and calculating the directions surrounding each pixel in the image to obtain the features and recognize Chinese chess characters. The proposed method had the ability to resist noise, brightness changes and rotation, and was tested by five different fonts. The image preprocessing and advanced Hough transformation [23] was used to segment the image and calculate the location of the center of the chessman and the circle edge of the chessman, respectively. Fang [24] designed a machine vision system for Chinese chess-playing robots with two color cameras taking two images from different angles simultaneously. A hierarchical Hough transform algorithm was used to detect lines and circles in the binarized image and the backpropagation neural network and ring intersection points were adopted to recognize the Chinese characters. In addition, experimental results verified that it can work well with higher reliability.

CNN is used to recognize Chinese chessmen in this paper; several previous studies have applied it to human activity recognition, face recognition, and text recognition [25–27]. Most of the recognition and segmentation work has involved hybrid methods, such as object recognition, human tracking, activity recognition, and human gait [28–30]. Meanwhile, [30] used depth image for face recognition, [26] adopted a similar technique but applied to time attendance systems, and [31] proposed

high multiplexing system performance to support face recognition over Wi-Fi. Action recognition, proposed by [32–35], is more complicated and can be solved using an RGB-D camera with intrinsic features. However, the whole process of identification used a fixed orientation. It is very diverse from Chinese chess, where the position of a round piece can change its orientation when picked by a gripper so that it will become an obstacle during recognizing. For this reason, in recognizing targets, approaches such as a human way of thinking are needed.

Previously, the artificial neural network (ANN) is a model developed based on imitating the structure and operation of the brain and becomes the basis of the convolution network. This method can be used to simulate complex models and prediction problems. The traditional neural network consists of three parts: the input layer, the hidden layer, and the output layer. The hidden layer has many neurons, and each neuron in each layer is connected to all neurons in the next layer. A network with multiple hidden layers is called a multilayer perceptron (MLP) [36].

For computer vision, a convolutional neural network (CNN) [37] is mainly used for image classification and object recognition. The main difference between an MLP and CNN is that only the last layer of a CNN is fully connected, while in an MLP, each neuron is connected to each neuron of the next layer, resulting in a large increase in the number of parameters. For large images, it generates complex vectors. In addition, it ignores spatial information and flattens the image as input. J. Jin et al. [38] used CNNs to recognize traffic signs and used hinge loss stochastic gradient descent to train CNNs which were evaluated on the German traffic sign recognition benchmark. Chen et al. [39] presented a hybrid deep convolutional neural networks (HDNNs) to recognize vehicles in satellite images by dividing the maps of the last convolutional layer and the max-pooling layer of DNN into multiple blocks of variable receptive field sizes or max-pooling field sizes to enable the HDNN to extract variable-scale features. In addition to images, CNNs are also used for speech recognition. O. Abdel-Hamid et al. [40] used a limited-weight-sharing scheme to simulate speech features in CNNs. Compared with DNNs, the bit error rate of the proposed method is reduced by 6–10%.

In this paper, the chess piece is photographed by a camera and the picture is input to a convolutional neural network (CNN) for chess recognition. At the same time, the coordinates of the chessman are obtained by image processing and sent to a robot system to grab the target chessman using the forward and inverse dynamics. In this paper, the CNN will be used to recognize the characters on the chessmen and distinguish the front or backside of the chessmen, even when they are randomly placed. The robot arm will be controlled to accurately grasp the chessmen and place them on exact positions of the chessboard. The remainder of this paper is organized as follows: the Chinese chess robotic system is introduced in Section 2 and the convolutional neural network is indicated in Section 3. Experimental results are analyzed in Section 4 along with the conclusions in Section 5 of the paper.

2. Chinese Chess Robotic System

The chessmen are photographed through the camera and the image is processed. Then, the coordinate transformation for the chessmen is setup by PC. Finally, the chessmen are randomly picked up by the robot arm through the gripper and placed on the proper positions on the chessboard, as shown in Figure 2. The robotic arm uses five-degree-of-freedom (5DOF) Microbot's TeachMover II, whose variables of the kinematics model are shown in Figure 3 [41], where the distance between each joint is respectively represented by constants H, L and LL with values of 195.0, 177.8, and 96.5 mm. Table 1 lists the relation between the motor step and the actual joint rotation.

In order to define the coordinate system of the robotic arm, it is necessary to first establish a coordinate on each link and use the Denavit–Hartenberg (DH) rule to determine the DH transformation matrix of each link. As long as this transformation matrix is used to achieve the transformation of the two coordinate systems, thus the equations of forward and inverse kinematics are derived.

Table 2 shows the D–H parameters of the robot, where α_i, a_i, d_i, and θ_i respectively, represent link twist, link length, link distance, and link angle.

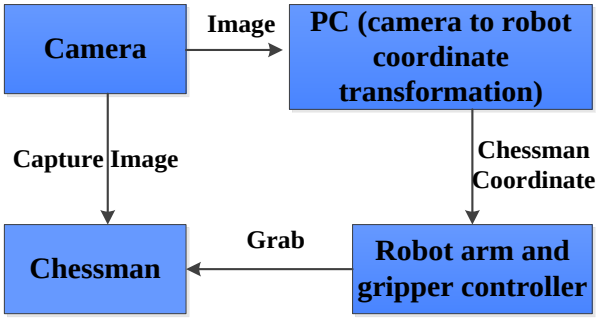


Figure 2. The system flow architecture.

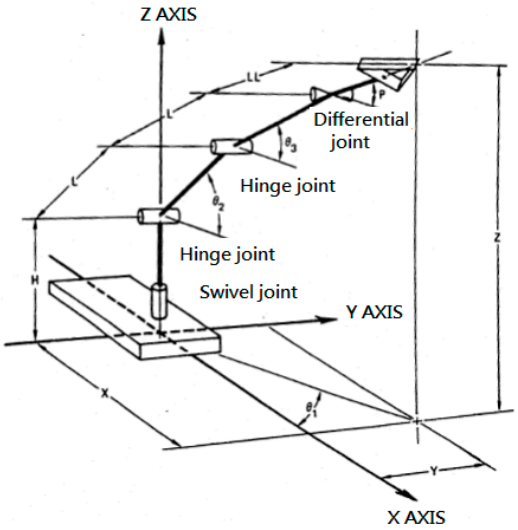


Figure 3. Schematics of Robotic arm.

Table 1. Relation between the motor step and the actual joint rotation.

Motor	Joint	Steps Per Degree
1	Base	19.64
2	Shoulder	19.64
3	Elbow	11.55
4	Right wrist	4.27
5	Left wrist	4.27

Table 2. Denavit–Hartenberg (D–H) parameters of TeachMover II.

Link	Joint Name	α_i	a_i	d_i	θ_i
1	Base	0	$\frac{\pi}{2}$	h	θ_1
2	Shoulder	L	0	0	θ_2
3	Elbow	L	0	0	θ_3
4	Pitch	0	$\frac{\pi}{2}$	0	$\frac{\pi}{2} + \theta_4$
5	Roll	0	0	LL	θ_5

Inverse kinematics estimates the motion angle of each joint axis if the position of the end-effector of the robotic arm is given. The angle of each joint can be also derived from the geometric point of view. As shown in Figure 4, when the point P at the end-effector of the arm with the known coordinates is projected onto the XY plane, we can find the angle θ_1. Referring the picture and geometric figure of the robotic arm in Figure 5, we may obtain the angles of θ_2, θ_3 and θ_4 as follows,

$$\theta_2 = \pi - \theta_a - \theta_b - \theta_c \quad (1)$$

$$\theta_3 = \theta_a + \theta_c \tag{2}$$

$$\theta_4 = \frac{\pi}{2} - \theta_3 \tag{3}$$

where d is the distance between points O and E, p is the distance between points B and D, and

$$\theta_a = \tan^{-1}(q/d) \tag{4}$$

$$\theta_b = \cos^{-1}\left(\frac{L1^2 + L2^2 - p^2}{2 \times L1 \times L2}\right) \tag{5}$$

$$\theta_c = \cos^{-1}\left(\frac{L^2 + p^2 - L^2}{2pL}\right) \tag{6}$$

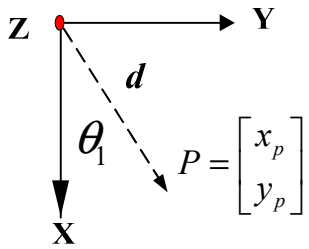


Figure 4. Calculating θ_1.

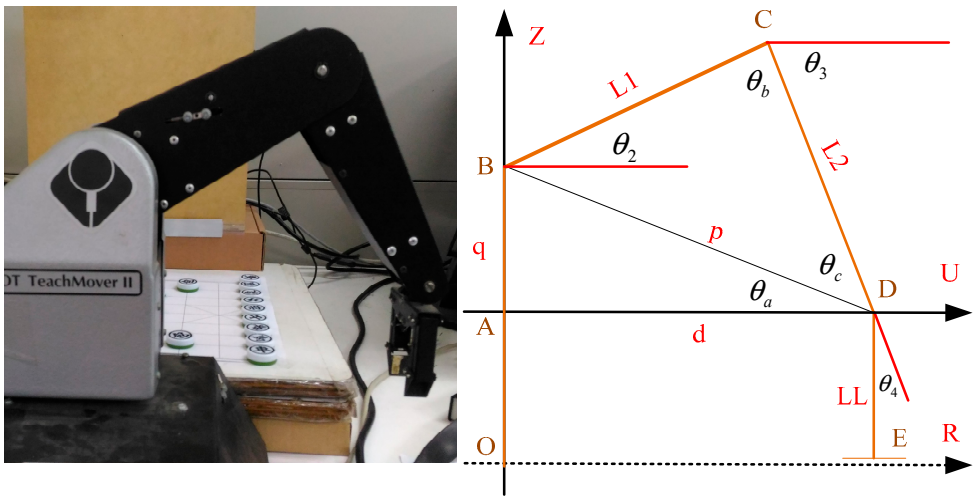


Figure 5. Geometric figure of the robotic arm.

3. Convolutional Neural Network

A convolutional neural network (CNN) consists of one or more convolutional layers, and then one or more fully connected layers (FCs) which are similar to the neural network structure. The structural design of the CNN uses the two-dimensional structure of the image as inputs to achieve local connection, weighting, and then pooling, which equips CNN translation-invariant features. Compared to neural networks with similar layers, CNNs have fewer parameters and connections and are therefore easier to train. CNNs consist of many convolutional and pooling layers, and finally a fully connected layer. A convolution layer adopts an image as its input and is formed by a plurality of different, generally 3×3, filters (called convolution kernels) to conduct convoluting operation and then produce different features.

The convolutional principle uses a small-sized window to slide from left to right and top to bottom to obtain the local features in the image as the inputs of the next layer. This sliding window is called a convolution kernel or filter. The matrix formed by sliding and calculating on the image is

called a convolution feature or feature map. The feature map is the output to the next layer through a rectified linear unit (ReLU) for activation function. It is a type of downsampling, because the size of the data will be reduced, so the number of parameters and calculations are reduced, which speeds up the system operation, reduces the possibility of overfitting, and has the effect of anti-interference. After sampling, the outputs are inputted to the fully connected layer [23,24]. The fully connected layer is a general neural network for classification. The connection layer is also the easiest way to learn a non-linear combination of the features from the previously convolutional layer and pooling layer. We flatten the feature map in the fully connected layer and update the weights in the neural network through backpropagation.

The softmax function is used in the output of the fully connected layer. The softmax function can convert an N-dimensional vector containing any real number into another N-dimensional real vector so that the range of each element in the vector is between 0 and 1, and the sum of all elements is 1. The equation of softmax function is described as

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, j = 1, \ldots, K. \tag{7}$$

Since the output of the Softmax function is between 0 and 1, it can be regarded as the probability of one type of class prediction. The loss function is an important part of the artificial neural network. It is used to measure the inconsistency between the predicted value and the actual label. Its output is a non-negative value. The robustness of the model increases as the value of the loss function decreases. This paper uses a cross-entropy algorithm to calculate the loss function, shown in Equation (8),

$$loss = -\sum_{i=1}^{N}\sum_{j=1}^{K} t_{ij} \ln \sigma(z)_{ij} \tag{8}$$

where N is the number of samples, K is the number of classifications, and t_{ij} is the actual label. This paper uses the stochastic gradient descent method to update the network parameters (weights) in each iteration to minimize the loss function through the negative gradient direction of the loss function. The equation for updating parameters is as follows:

$$z_{\ell+1} = z_\ell - \alpha \nabla loss(z_\ell) \tag{9}$$

where l is the number of iterations, $\nabla loss(w)$ is the gradient of the loss function, and α is the learning rate. The expression for calculating the loss function gradient is as follows:

$$\frac{\partial loss}{\partial z_i} = \frac{\partial loss}{\partial \sigma(z)_j}\frac{\partial \sigma(z)_j}{\partial z_i} = -\frac{t_i}{\sigma(z)_i} \times \sigma(z)_i(1 - \sigma(z)_i) = -t_i + t_i\sigma(z)_i \tag{10}$$

where j means all outputs and i is one of them.

The CNN architecture used in this paper is shown in Figure 6, including three convolutional layers, three pooling layers, and one connection layer.

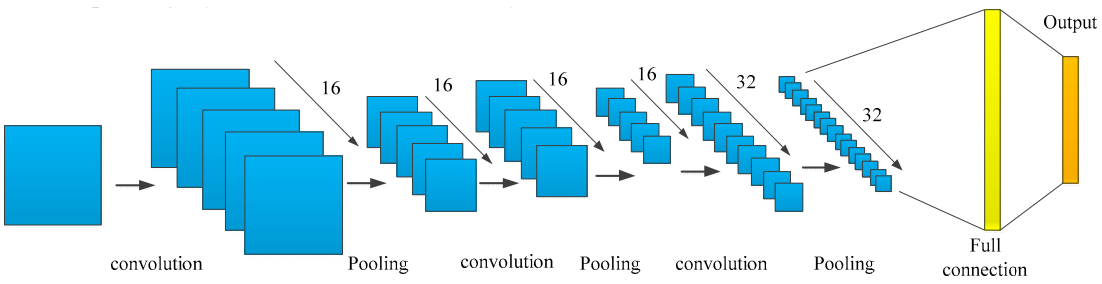


Figure 6. CNN architecture used in this paper.

4. Experimental Results

This proposed system includes a robotic arm and a camera, where the camera communicates PC via USB, and PC sends signals to the robotic arm controller via RS232 to complete the action. The camera is set up directly above the chessboard. All chessmen are randomly placed on the chessboard. The camera captures the image in this range, and then, from left to right and from bottom to top, the image is cut out sub-images of multiple chessmen. The sub-image is input to the CNN for recognition, and the recognition is repeated until the recognition of multiple chessmen is completed. The coordinates where the chessman is currently located and should be placed are transmitted to the robotic arm. The arm then picks up the recognized chessman and then places it in the correct position on the board. The system repeats the above procedure until all the chessmen are placed. The system block diagram is shown in Figure 7 and the experimental environment includes Logitech C310 camera and PC with CPU of Intel Core i5-3570 3.4 GHz shown in Figure 8.

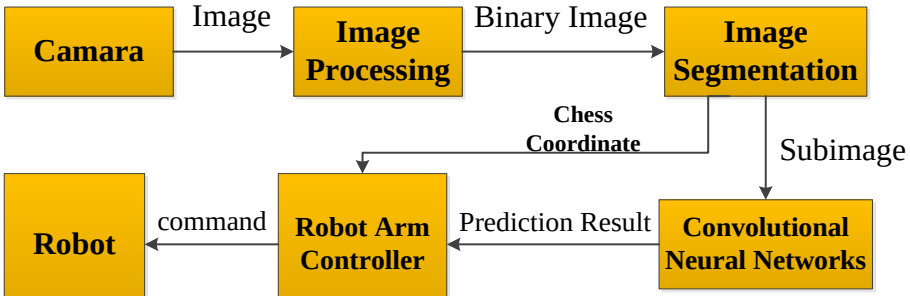


Figure 7. System block diagram.

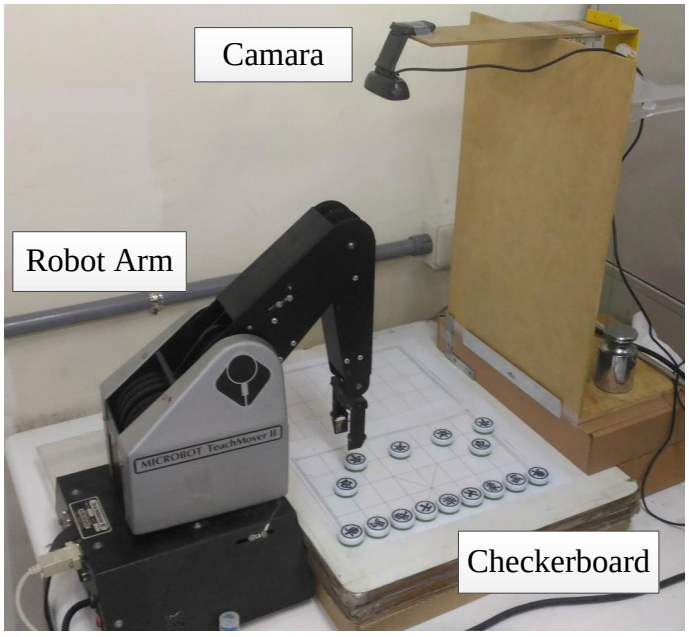


Figure 8. Experimental environment.

This paper uses MATLAB to integrate the program of the camera and the robotic arm, as shown in Figure 9. Under the GUI operation interface of MATLAB, the system performs basic actions such as picking up and placing chessmen. The upper left image is the original image, the lower left image is the binarized image, the upper right image is the chessman image, the lower right table shows the prediction result and its coordinates, and the system control block provides the keys for all operational functions.

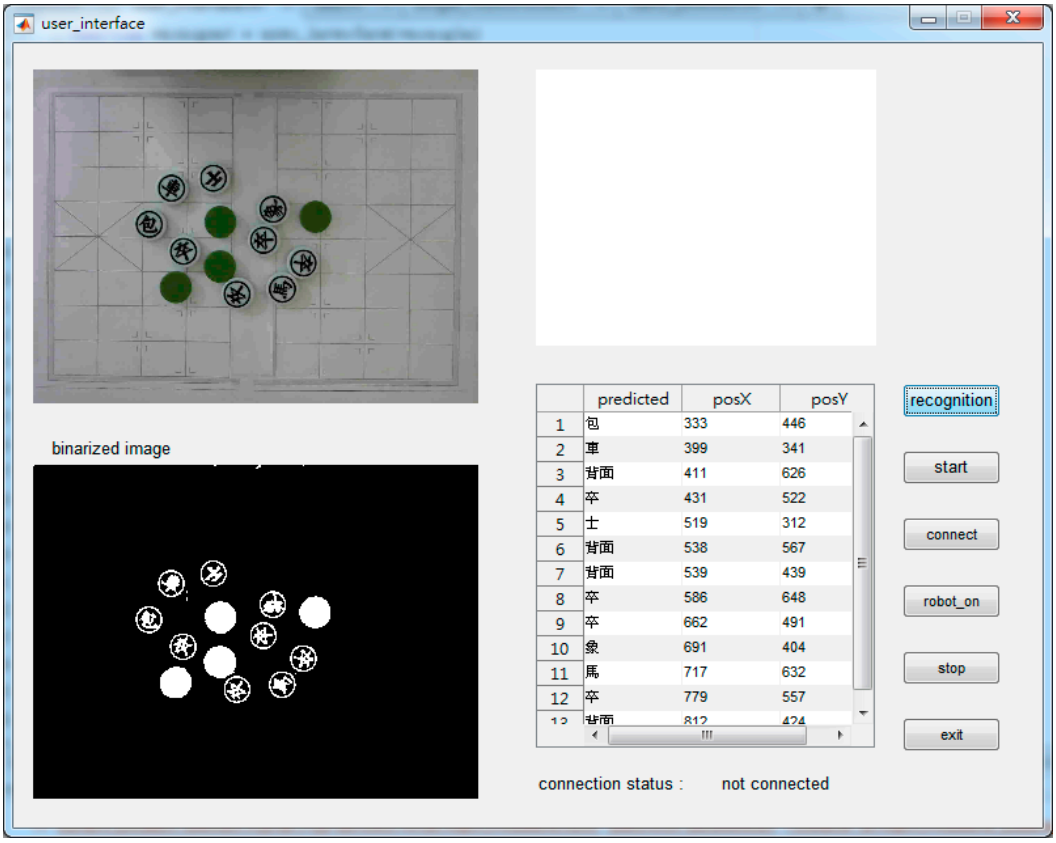


Figure 9. User interface.

When grasping an object, the exact coordinates of the object are required. But when using the camera, the imaging will be more or less distorted due to the problem of the camera itself [42]. Therefore, the camera must be calibrated to obtain the intrinsic parameters. The distortion is corrected through these parameters to obtain a correct image. The correction method uses a black and white 9×7 checkerboard diagram. Its grid size is 28×28 (mm^2). After placing the checkerboard image in front of the camera and allowing the camera to take the complete checkerboard image, we change the direction of the checkerboard image facing the camera for adjustment. Logitech's network camera C310 with resolution of 1280×960 pixels is used in this paper. The extrinsic parameters of the camera can be calculated by the cameraCalibrator function based the image captured by the camera [43]. The matrix of intrinsic parameters is:

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1416.7011 & 0 & 682.3046 \\ 0 & 1417.6963 & 494.6859 \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

where f_x and f_y are the focal lengths in the X and Y directions of the image plane, c_x and c_y are the reference points which are ideally the center of the image.

CNN learns various chessmen's features to recognize them. To obtain training data, a large number of images, which will be binarized, are obtained by rotating and translating the chessmen, as shown in Figure 10. Using these images as training data to train CNN, the trained CNN will have high accuracy in recognition and can accurately determine the characters of chessmen. The training process is shown in Figure 11. The upper part depicts the change in accuracy during training and the lower figure shows the change in loss during training. The horizontal axis is the number of iterations. In the ninth training period (Epoch 9), the accuracy does not change much. The recognition time of a single piece is about 0.35 s, and it takes about 11 s to recognize all chessmen. During the

recognition process, if affected by reflection, the chessman character will be incomplete, as shown in Figures 12 and 13. Table 3 shows the recognition tests of chessmen rotated by 0°, 45°, 90°, 105°, 120°, and 180°. For black chessmen, no matter how many degrees the chessmen are rotated, they can be recognized correctly, while the red ones can be recognized 100% of the time within 90° of rotation, and some chessmen are unable to reach a 100% recognition rate at more than a 90° rotation. The recognition of the red chessmen is obviously worse than that of the black ones, mainly because the characters of the black chessmen are quite different, but the red chessmen have the same radical, and the strokes are more likely to affect the recognition result. For the arbitrary placement test, considering the confusion matrices of red and black chessmen as shown in Tables 4 and 5, the accuracy of black chessmen is 100%, and the accuracy of red chessmen is 98.7%. In the case, the three chessmen of 俥, 傌, and 炮 are confused with each other, affecting the recognition.

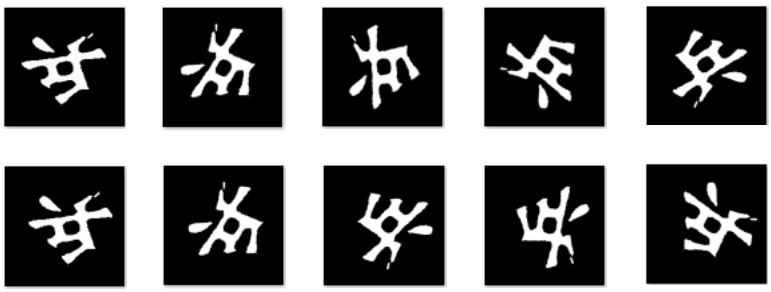


Figure 10. Images of a rotated and translated chessman.

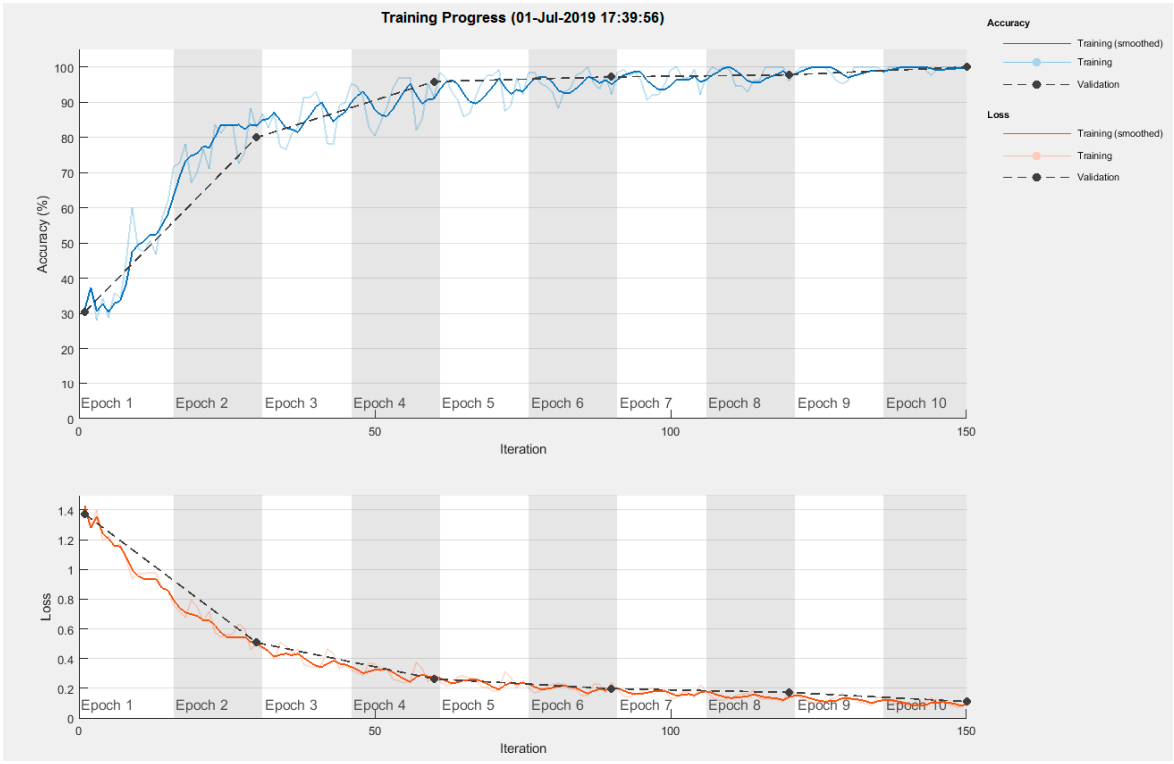
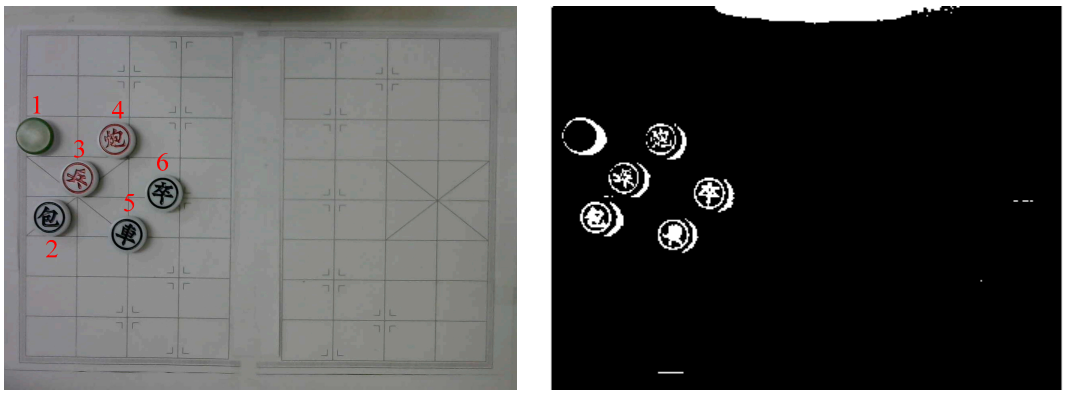


Figure 11. Training process.

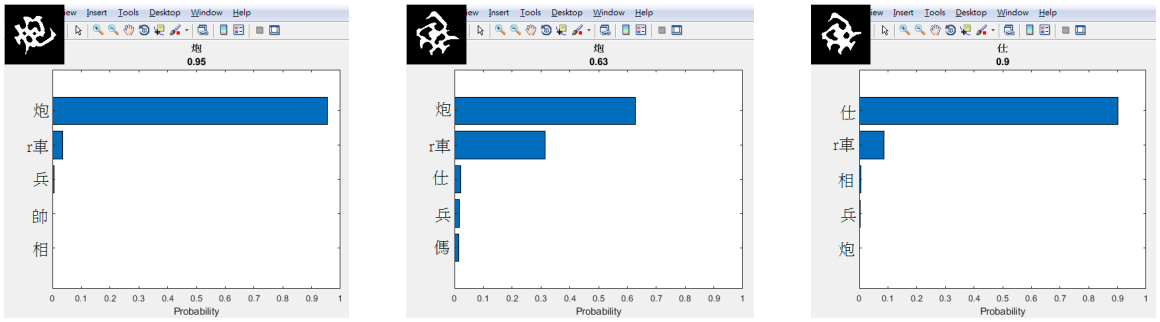


(**a**) (**b**)

	predicted	posX	posY
*1	士	100	331
2	包	111	530
*3	炮	183	430
*4	帥	278	332
5	車	307	572
6	卒	396	467

(**c**)

Figure 12. Chessman recognition. (**a**) Original captured image, (**b**) binarized image, (**c**) the table that shows the prediction result and its coordinates.



(**a**) (**b**) (**c**)

Figure 13. Recognition errors due to incomplete character of a chessman. (**a**) Recognition errors for 炮, 車, 兵, 帥, 相 characters, (**b**) recognition errors for 炮, 車, 仕, 兵, 傌 characters, (**c**) recognition errors for 炮, 仕, 車, 相, 兵 characters.

When the camera captures the image, the coordinates of the chessman will be different from the actual ones due to the height of the chessman. Therefore, the real coordinates of the chessman must be corrected to obtain an accurate grab, as shown in Figure 14. Point D is the camera position, point C is the actual position of the chessman, and point B is the position of the chessman estimated by the camera. The errors before and after the correction of the coordinates of the chessman are shown in Figure 15 and Table 6. Since the height h of the chessman is known, the actual coordinates of the chessman can be obtained through the trigonometric function after finding the camera position O and its height H,

$$C = B - d, d = \frac{p \times h}{H} \tag{12}$$

Table 3. Test results of numbers of recognition errors for different rotations.

Chessman	Rotating Degree					
	0°	**45°**	**90°**	**105°**	**120°**	**180°**
將	0	0	0	0	0	0
士	0	0	0	0	0	0
象	0	0	0	0	0	0
車	0	0	0	0	0	0
馬	0	0	0	0	0	0
包	0	0	0	0	0	0
卒	0	0	0	0	0	0
帥	0	0	0	0	0	0
仕	0	0	0	0	0	0
相	0	0	0	0	0	0
俥	0	0	0	1	1	2
傌	0	0	0	0	0	1
炮	0	0	0	1	2	2
兵	0	0	0	0	0	0

Table 4. Confuse matrix of black chessmen.

		Actual Class						
		將	**士**	**象**	**車**	**馬**	**包**	**卒**
Predicted class	將	100	0	0	0	0	0	0
	士	0	100	0	0	0	0	0
	象	0	0	100	0	0	0	0
	車	0	0	0	100	0	0	0
	馬	0	0	0	0	100	0	0
	包	0	0	0	0	0	100	0
	卒	0	0	0	0	0	0	100

Table 5. Confuse matrix of red chessmen.

		Actual Class						
		帥	**仕**	**相**	**俥**	**傌**	**炮**	**兵**
Predicted class	帥	100	0	0	0	0	0	0
	仕	0	100	0	0	0	0	0
	相	0	0	100	0	0	0	0
	俥	0	0	0	97	0	4	0
	傌	0	0	0	0	98	0	0
	炮	0	0	0	3	2	96	0
	兵	0	0	0	0	0	0	100

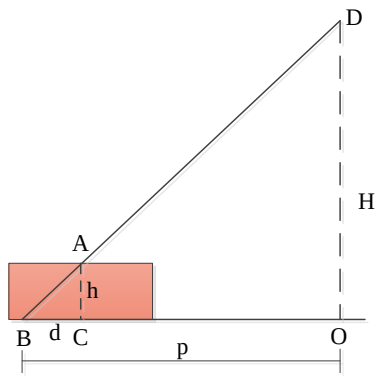


Figure 14. Geometric representation for real and camera coordinates.

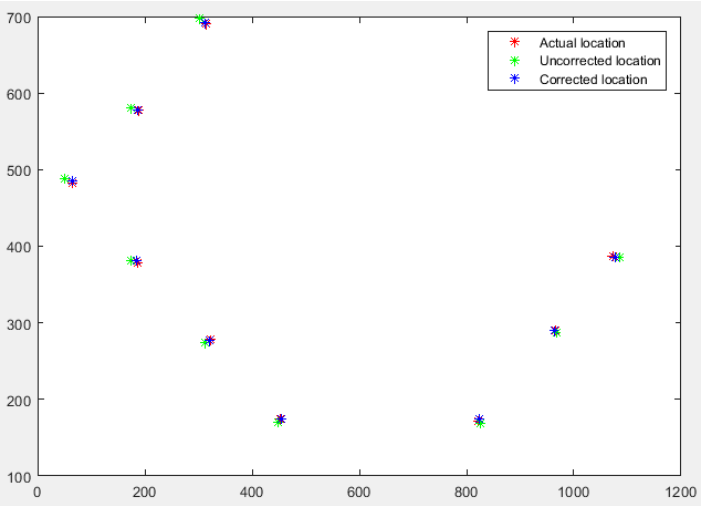


Figure 15. Locations of chessmen of actual, uncorrected, and corrected.

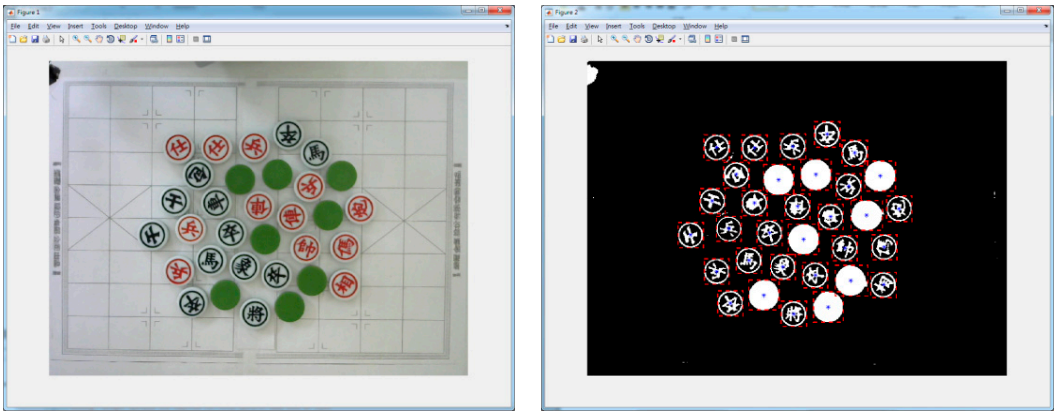
Table 6. Coordinates before and after correction and errors between them and real ones.

Real Coordinate	Coordinate before Correction	Error	Coordinate after Correction	Error
(64,483)	(50,488)	14.86	(64.9,486)	3.13
(188,578)	(175,581)	13.34	(186.2,578)	1.8
(186,379)	(174,381)	12.16	(185.2,381.3)	2.44
(314,691)	(303,698)	13.04	(311.5,692.5)	2.92
(322,278)	(313,274)	9.85	(321.3,277.5)	0.86
(440,780)	(436,791)	11.70	(441.6,783.5)	3.85
(452,174)	(448,170)	5.66	(454.4,174.8)	2.53
(822,171)	(826,168)	5	(823.3,173.8)	3.09
(826,786)	(831,794)	9.43	(828.2,786.4)	2.24
(952,686)	(958,692)	8.49	(952.5,686.6)	0.78
(965,290)	(969,287)	5	(963.3,290.3)	1.73
(1074,387)	(1085,385)	11.18	(1076.8,386.2)	2.91
(1082,590)	(1090,593)	8.54	(1081.7,589.7)	0.42
(1220,488)	(1230,493)	11.18	(1218.7,491.9)	4.11

Before recognizing, the original image must be cut into images of chessmen. After the original image is binarized, connected-component labeling (CCL) [44] is used to find the position of each chessman, as shown in Figure 16. These chessmen are cut out and recognized using CNN, as shown in Figure 17. The connected-component labeling algorithm scans the input binarized image and calculates its eight connectivity pixels when it encounters a value of 1. The labeling rules are [44]:

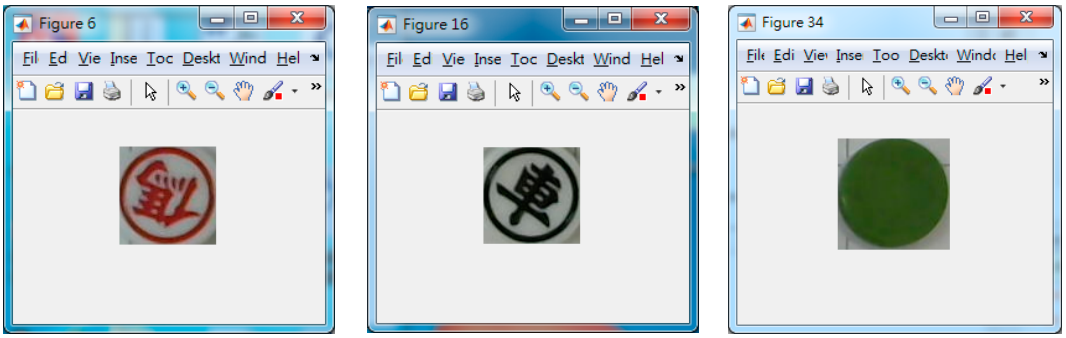
(a) If the values in all four directions are 0, then a new label is created at that position;
(b) If the labels in the four directions are the same, then the position label is the label of its field;
(c) If the labels in the four directions have two different labels, choose one of them, and record the two different labels.

After doing the image segmentation and CNN recognition, the character and coordinates of each chessman are known. Then, its coordinates are transferred to the robot arm for grabbing. Since the chessmen are randomly placed on the chessboard, the order of grabbing begins from the chessmen placed on both sides of the chessboard until all chessmen are placed, as shown in Figure 18. If there are back-side chessmen, the system can identify the situation and notify the robotic arm to turn over those first, and then perform image recognition, as shown in Figure 19. Figure 20 depicts the complete process of chess placement, beginning with the interface, locating the first chessman at the side of the chessboard, turning over the back-side chessman, finally finishing the placement.



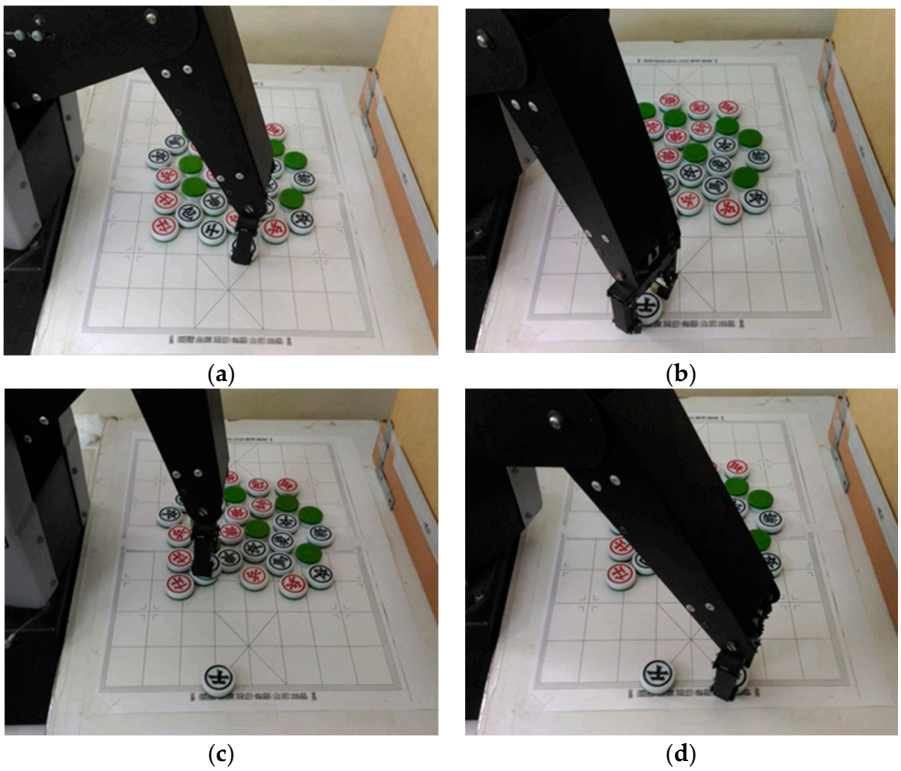
(**a**) (**b**)

Figure 16. Original image and its binarized one. (**a**) Original captured image, (**b**) binarized image.



(a) **(b)** **(c)**

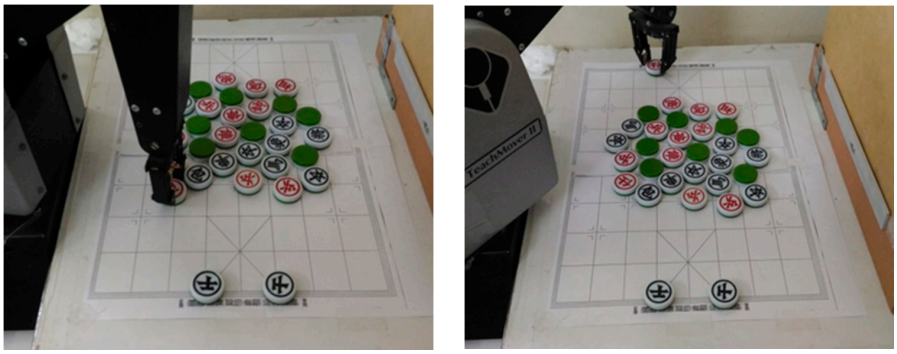
Figure 17. Cut-out images of each chessman. (**a**) The red colored chessman 傌, (**b**) the black colored chessman 車, (**c**) the back-side chessman.



(**a**) (**b**)

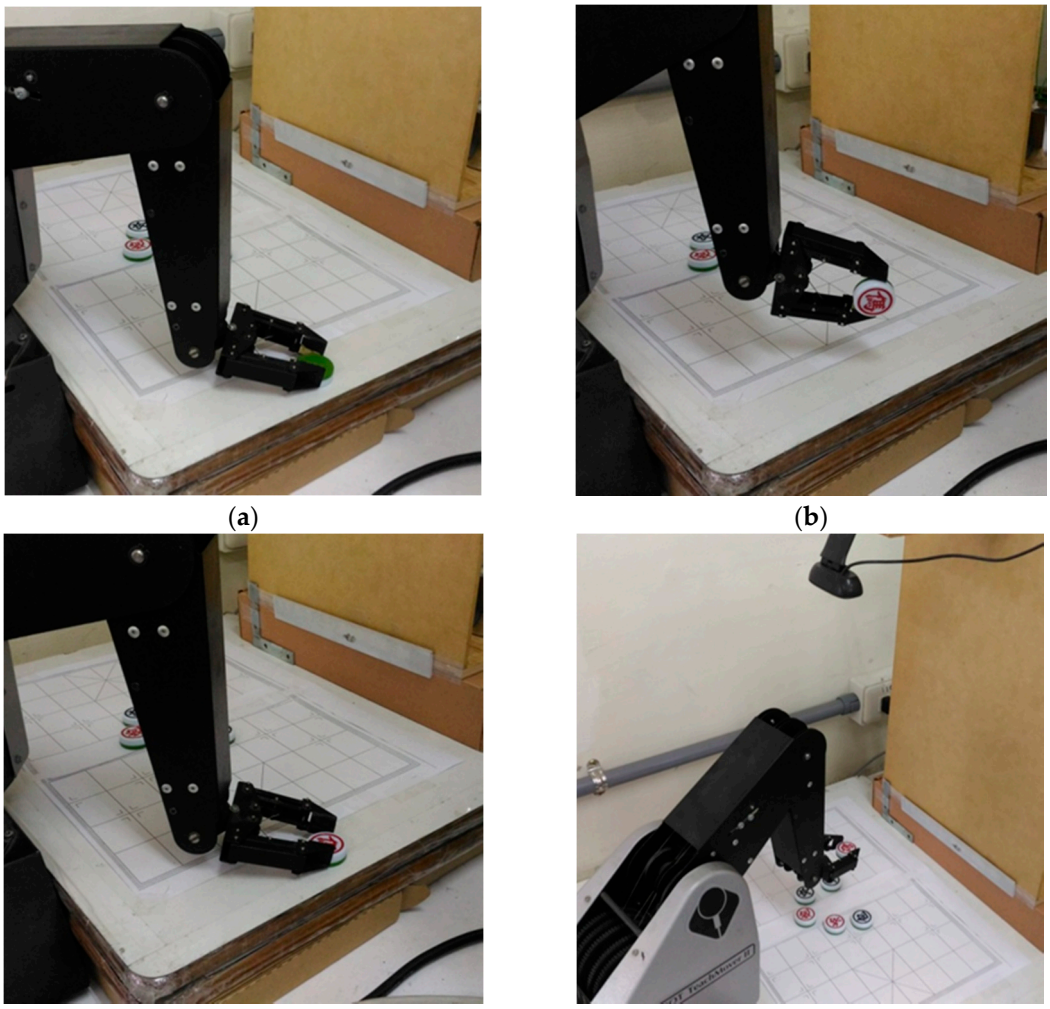
(**c**) (**d**)

Figure 18. *Cont.*



(**e**) (**f**)

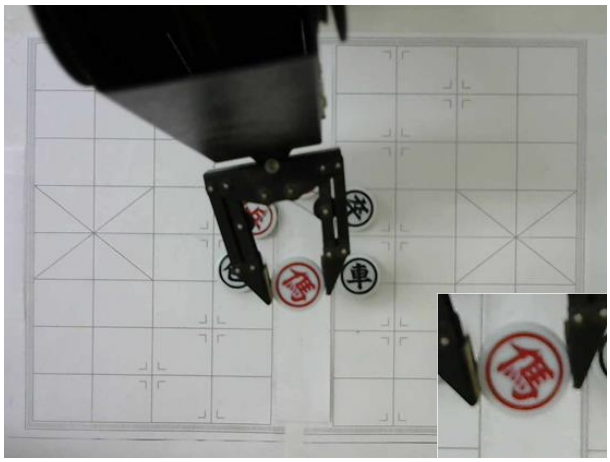
Figure 18. The process of robotic arm grabbing. (**a**) The robotic arm picking a random black chessman, (**b**) the robotic arm placing the chessman to its correct location, (**c**) the robotic arm picking another random black chessman, (**d**) the robotic arm placing the chessman to its correct location, (**e**) the robotic arm picking the first random red chessman, (**f**) the robotic arm placing the chessman to its correct location.



(**a**) (**b**)

(**c**) (**d**)

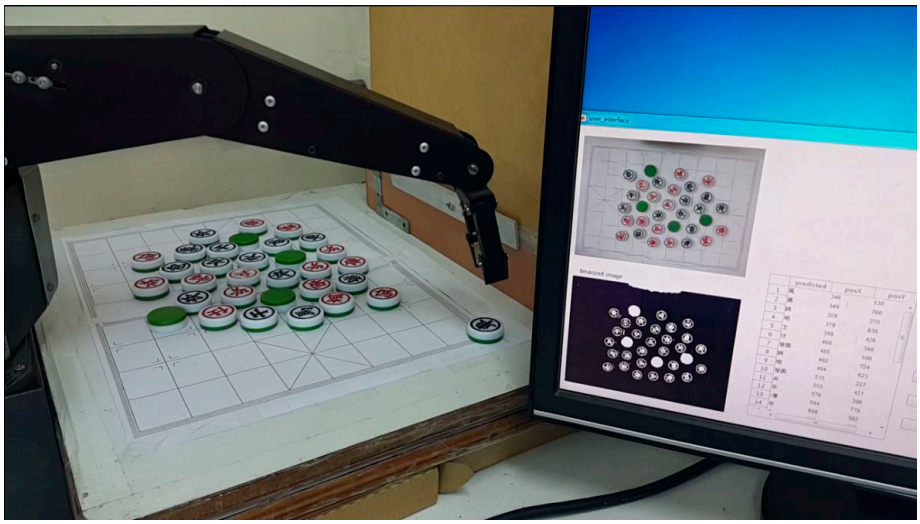
Figure 19. *Cont*.



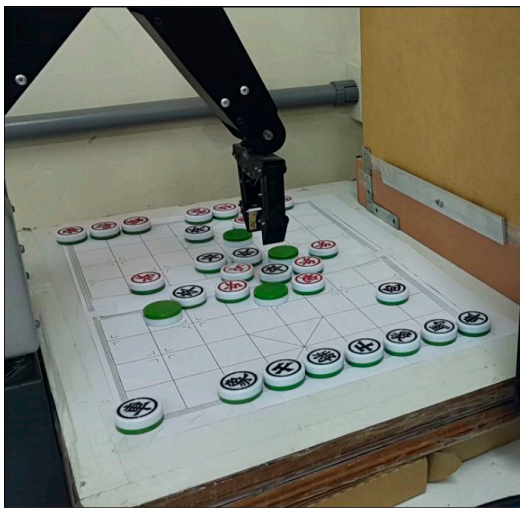
(**e**)

(**f**)

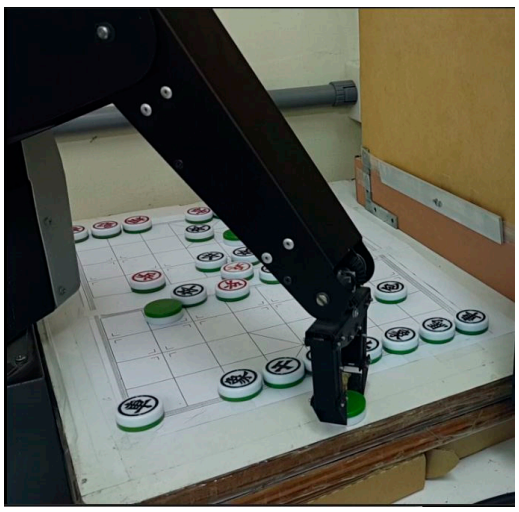
Figure 19. Turning over the back-side chessman and recognizing. (**a**) First, turning over a back-side chessman, (**b**) turning over the back-side chessman, (**c**) next, putting the chessman back, (**d**) then, putting it in front of the camera to be recognized, (**e**) lastly, the chessman is captured by the camera, (**f**) the chessman 傌 is recognized by the camera and the cut-out image is captured.



(**a**)



(**b**)



(**c**)

Figure 20. *Cont*.

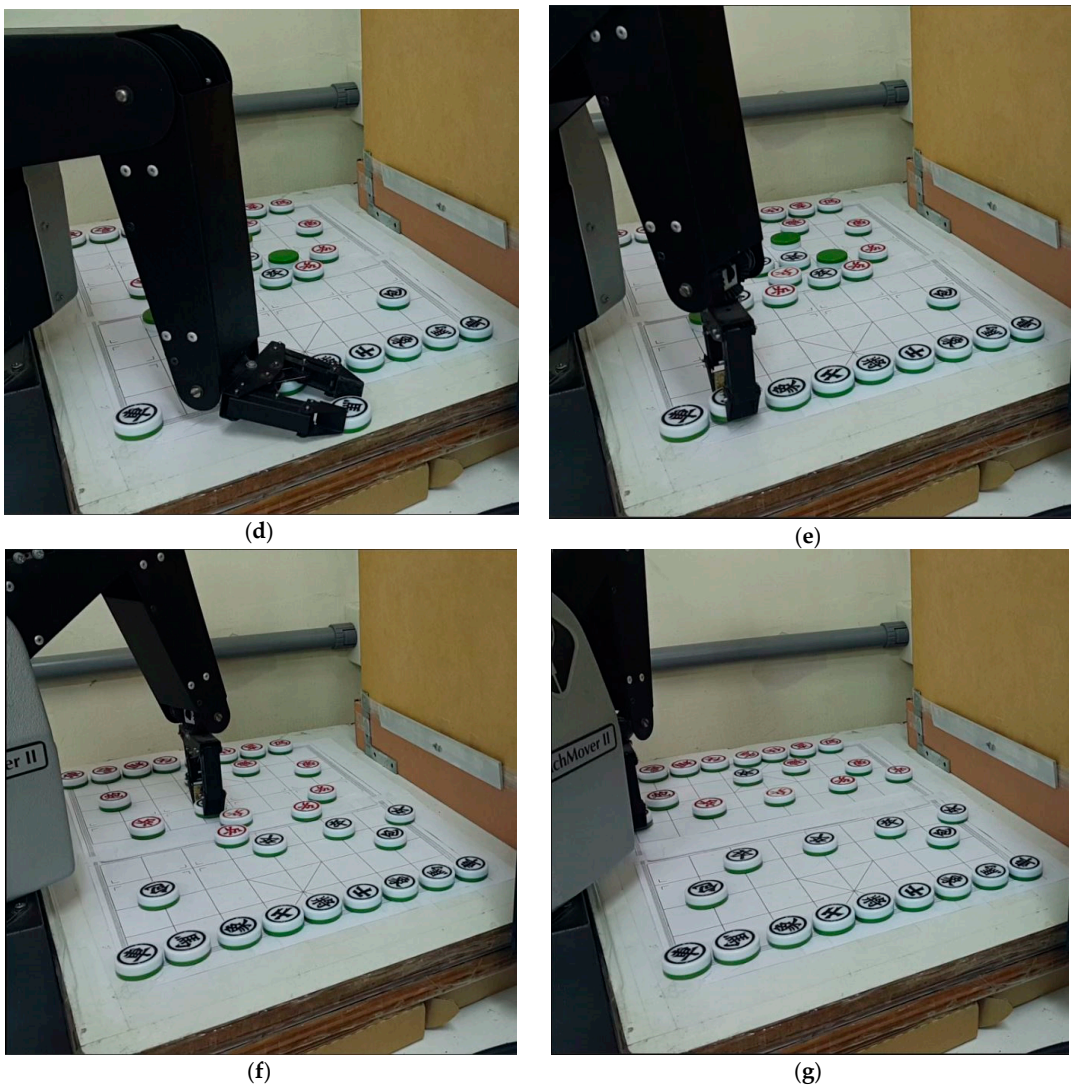


Figure 20. Process of chessmen placement. (**a**) The random black chessman picked up and placed in the correct location, (**b**) a random back-side chessman is recognized and picked up, (**c**) the random back-side chessman is placed on the side and turned over, (**d**) the random back-side chessman is turned over and will be placed in front of the camera to be recognized, (**e**) the random back-side chessman is placed in its correct location, (**f**) another random red chessman is recognized and picked up, (**g**) the random red chessman is placed in its correct location.

In summary, correct recognition will have the correct picking and placing of the chessmen. The failure cases come from three points, the first: the chessmen 俥 and 傌 have the same radical of Chinese characters; the second: there may be an erroneous recognition of chessman 炮 with the same radical of 俥 and 傌; and the third: the strokes of these three chessmen are more likely to affect the recognition result. As a result, another auxiliary way may be included to eliminate these cases.

5. Conclusions

This paper proposes a system for chessman recognition and automatic placement. First, through the techniques of image processing and convolutional neural network technology, the character of arbitrarily placed chessman is recognized and its position is found. If there are back-side chessmen, the system will turn over those first and then perform image recognition. After obtaining the coordinates of the chessman and through coordinate transformation, the coordinates are transmitted to the robot arm to grab the chessman and place it at the correct location on the chessboard. Comparing the

proposed method with several methods/approaches and improving the performance will be our future work. In the future, both the hardware and the functions of image vision and convolutional neural network technology can be improved to increase the recognition rate and speed and enhance the ability of the robot in playing chess. Then, if the software for playing chess is added, it will not only be a simple chessman placement system but also provide the function of playing chess with people. As a result, the proposed system can further enhance the elders' favor and develop their habit of playing chess to strengthen effective neural bonds between brain cells and reserve brain cognitive functions.

Author Contributions: P.-J.C. and M.-S.W. conceived and designed the experiments; S.-Y.Y., C.-S.W. and M.M. performed the experiments; M.-S.W. and C.-S.W. analyzed the data; M.-S.W. and P.-J.C. contributed materials and analytical tools; M.-S.W. wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Higher Education Sprout project from the Ministry of Education, Taiwan and contract No. of MOST 108-2221-E-218-029- from the Ministry of Science and Technology, Taiwan.

Acknowledgments: The authors would like to thank all the reviewers for their constructive reviews.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. 認識失智症 (Recognize Dementia). Available online: http://www.tada2002.org.tw/About/IsntDementia (accessed on 20 March 2020).
2. Ukida, H.; Terama, Y.; Ohnishi, H. Object tracking system by adaptive pan-tilt-zoom cameras and arm robot. In Proceedings of the 2012 SICE Annual Conference, Akita, Japan, 20–23 August 2012.
3. Lin, C.-J.; Shaw, J.; Tsou, P.-C.; Liu, C.-C. Vision servo based Delta robot to pick-and-place moving parts. In Proceedings of the 2016 IEEE International Conference on Industrial Technology (ICIT), Taipei, Taiwan, 14–17 March 2016.
4. Althloothi, S.; Mahoor, M.H.; Zhang, X.; Voyles, R.M. Human Activity Recognition Using Multi-features and Multiple Kernel Learning. *Pattern Recognit.* **2014**, *47*, 1800–1812. [CrossRef]
5. Jalal, A.; Kamal, S.; Kim, D. Shape and motion features approach for activity tracking and recognition from kinect video camera. In Proceedings of the 2015 IEEE 29th International Conference on Advanced Information Networking, Gwangiu, Korea, 24–27 March 2015.
6. Song, Y.; Tang, J.; Liu, F.; Yan, S. Body Surface Context: A New Robust Feature for Action Recognition from Depth Videos. *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *24*, 952–964. [CrossRef]
7. Ge, Y.; Xiong, Y.; Tenorio, G.L.; From, P.J. Fruit Localization and Environment Perception for Strawberry Harvesting Robots. *IEEE Access* **2019**, *7*, 147642–147652. [CrossRef]
8. Chen, Z.; Zhou, D.; Liao, H.; Zhang, X. Precision Alignment of Optical Fibers Based on Telecentric Stereo Microvision. *IEEE/ASME Trans. Mechatron.* **2016**, *21*, 1924–1934. [CrossRef]
9. Jalal, A.; Kamal, S.; Kim, D. A Depth Video Sensor-based Life-logging Human Activity Recognition System for Elderly Care in Smart Indoor Environments. *Sensors* **2014**, *14*, 11735–11759. [CrossRef]
10. Jalal, A.; Uddin, M.Z.; Kim, J.T.; Kim, T.S. Daily human activity recognition using depth silhouettes and R transformation for smart home. In Proceedings of the Smart Homes Health Telematics, Montreal, QC, Canada, 20–22 June 2011.
11. Jalal, A.; Zeb, M.A. Security enhancement for e-learning portal. *Int. J. Comput. Sci. Netw. Secur.* **2008**, *8*, 41–45.
12. Landan, M.I. E-commerce security issues. In Proceedings of the IEEE Conference on Future Internet of Things and Cloud, Barcelona, Spain, 27–29 August 2014.
13. Jalal, A.; Shazad, A. Multiple facial feature detection using vertex-modeling structure. In Proceedings of the IEEE Conference on Interactive Computer Aided Learning, Villach, Austria, 26–28 September 2007.
14. Jalal, A.; Uddin, I. Security architecture for third generation (3G) using GMHS cellular network. In Proceedings of the IEEE Conference on Emerging Technologies, Patras, Greece, 25–28 January 2007.
15. Jalal, A.; Sarif, N.; Kim, J.T.; Kim, T.S. Human Activity Recognition via Recognized Body Parts of Human Depth Silhouettes for Residents Monitoring Services at Smart Homes. *Indoor Built Environ.* **2013**, *22*, 271–279. [CrossRef]

16. Manwatkar, P.M.; Yadav, S.H. Text recognition from images. In Proceedings of the 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, India, 1–6 March 2015.
17. Lara, O.D.; Labrador, M.A. A Survey on Human Activity Recognition using Wearable Sensors. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 1192–1209. [CrossRef]
18. Jalal, A.; Kim, J.T.; Kim, T.S. Development of a life logging system via depth imaging based human activity recognition for smart homes. In Proceedings of the 8th International Symposium on Sustainable Healthy Buildings, Seoul, Korea, 19 September 2012.
19. Jalal, A.; Kamal, S.; Kim, D. Human Depth Sensors-Based Activity Recognition Using Spatiotemporal Features and Hidden Markov Model for Smart Environments. *J. Comput. Netw. Commun.* **2016**, *2016*, 1–11. [CrossRef]
20. Yang, A.Y.; Iyengar, S.; Sastry, S.; Bajcsy, R.; Kuryloski, P.; Jafari, R. Distributed segmentation and classification of human actions using a wearable motion sensor network. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Anchorage, AK, USA, 23–28 June 2008.
21. Wen, Y.C. Chinese-Chess Image Recognition by using Feature Comparison Techniques. *Appl. Math. Inf. Sci.* **2014**, *8*, 2443–2453.
22. Arisandi, D.; Rahmat, R.F.; Nababan, E.B. Chinese chess character recognition using direction feature extraction and backpropagation. In Proceedings of the 2016 International Conference on Data and Software Engineering (ICoDSE), Denpasar, Indonesia, 26–27 October 2016.
23. Gui, W.; Jun, T. Chinese chess recognition algorithm based on computer vision. In Proceedings of the 26th Chinese Control and Decision Conference, Changsha, China, 31 May–2 June 2014.
24. Rosenblatt, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychol. Rev.* **1958**, *65*, 386–408. [CrossRef] [PubMed]
25. Kamal, S.; Jalal, A.; Kim, D. Depth Images-based Human Detection, Tracking and Activity Recognition Using Spatiotemporal Features and Modified HMM. *J. Electr. Eng. Technol.* **2016**, *11*, 1921–1926. [CrossRef]
26. Siswanto, A.R.S.; Nugroho, A.S.; Galinium, M. Implementation of face recognition algorithm for biometrics based time attendance system. In Proceedings of the 2014 International Conference on ICT for Smart Society (ICISS), Bandung, Indonesia, 24–25 September 2014.
27. Tripathy, A.K.; Carvalho, R.; Pawaskar, K.; Yadav, S.; Yadav, V. Mobile based healthcare management using artificial intelligence. In Proceedings of the 2015 International Conference on Technologies for Sustainable Development (ICTSD), Mumbai, India, 4–6 February 2015.
28. Khurana, P.; Sharma, A.; Singh, S.N.; Singh, P.K. A survey on object recognition and segmentation techniques. In Proceedings of the IEEE International Conference on computing for sustainable Global Development, New Delhi, India, 16–18 March 2016.
29. Jalal, A.; Kamal, S.; Kim, D. Depth silhouettes context: A new robust feature for human tracking and activity recognition based on embedded HMMs. In Proceedings of the 12th IEEE International Conference on Ubiquitous Robots and Ambient Intelligence, Goyang, Korea, 25–28 October 2015.
30. Jalal, A.; Kim, Y.H.; Kim, Y.J.; Kamal, S.; Kim, D. Robust Human Activity Recognition from Depth Video using Spatiotemporal Multi-fused Features. *Pattern Recognit.* **2017**, *61*, 295–308. [CrossRef]
31. Kamal, S.; Meza, C.A.A.; Lee, K. Family of Nyquist-I Pulses to Enhance Orthogonal Frequency Division Multiplexing System Performance. *IETE Tech. Rev.* **2016**, *33*, 187–198. [CrossRef]
32. Farooq, A.; Jalal, A.; Kamal, S. Dense RGB-D Map-Based Human Tracking and Activity Recognition using Skin Joints Features and Self-Organizing Map. *KSII Trans. Internet Inf. Syst.* **2015**, *9*, 1856–1869.
33. Kamal, S.; Jalal, A. A Hybrid Feature Extraction Approach for Human Detection, Tracking and Activity Recognition Using Depth Sensors. *Arab. J. Sci. Eng.* **2016**, *41*, 1043–1051. [CrossRef]
34. Yacoob, N.I.; Tahir, N.M. Feature selection for gait recognition. In Proceedings of the IEEE Symposium on Humanities, Science and Engineering Research, Kuala Lumpur, Malaysia, 24–27 June 2012.
35. Martin, A.G.; Martinez, J.M. People Detection in Surveillance: Classification and Evaluation. *IET Comput. Vis.* **2015**, *9*, 779–788. [CrossRef]
36. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
37. Jin, J.; Fu, K.; Zhang, C. Traffic Sign Recognition with Hinge Loss Trained Convolutional Neural Networks. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 1991–2000. [CrossRef]

38. Chen, X.; Xiang, S.; Liu, C.; Pan, C. Vehicle Detection in Satellite Images by Hybrid Deep Convolutional Neural Networks. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 1797–1801. [CrossRef]
39. Abdel-Hamid, O.; Mohamed, A.; Jiang, H.; Deng, L.; Penn, G.; Yu, D. Convolutional Neural Networks for Speech Recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2014**, *55*, 1533–1545. [CrossRef]
40. Más de 20 Años Creando Soluciones Especializadas en TI. Available online: http://www.teachmover.com/ (accessed on 20 March 2020).
41. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15), Boston, MA, USA, 1–9 June 2015.
42. Du, Y.C.; Muslikhin, M.; Hsieh, T.H.; Wang, M.S. Stereo Vision-Based Object Recognition and Manipulation by Regions with Convolutional Neural Network. *Electronics* **2020**, *9*, 210. [CrossRef]
43. What Is Camera Calibration? Available online: http://www.mathworks.com/help/vision/ug/camera-calibration.html (accessed on 20 March 2020).
44. Connected-Component Labeling. Available online: https://en.wikipedia.org/wiki/Connected-component_labeling (accessed on 20 March 2020).

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).