applied sciences

Article

Robotic-Based Touch Panel Test System Using Pattern Recognition Methods

Chia-Chi Lu [1] and Jih-Gau Juang [2,*]

[1] Wistron Corporation, 158 Xingshan Road, Taipei 11469, Taiwan; luchingrose@gmail.com
[2] Department of Communications, Navigation and Control Engineering, National Taiwan Ocean University, 2 Pei-Ning Road, Keelung 20224, Taiwan
* Correspondence: jgjuang@ntou.edu.tw

Received: 10 October 2020; Accepted: 19 November 2020; Published: 24 November 2020

Abstract: In this study, pattern recognition methods are applied to a five-degrees-of-freedom robot arm that can key in words on a touch screen for an automatic smartphone test. The proposed system can recognize Chinese characters and Mandarin phonetic symbols. The mechanical arm is able to perform corresponding movements and edit words on the screen. Pattern matching is based on the Red-Green-Blue (RGB) color space and is transformed to binary images for higher correct rate and geometric matching. A web camera is utilized to capture patterns on the tested smartphone screen. The proposed control scheme uses a support vector machine with a histogram of oriented gradient classifier to recognize Chinese Mandarin phonetic symbols and provide correct coordinates during the control process. The control scheme also calculates joint angles of the robot arm during the movement using the Denavit–Hartenberg parameters (D-H) model and fuzzy logic system. Fuzzy theory is applied to use the position error between the robot arm and target location then resend the command to adjust the arm's position. From the experiments, the proposed control scheme can control the robot to press desired buttons on the tested smartphone. For Chinese Mandarin phonetic symbols, recognition accuracy of the test system can reach 90 percent.

Keywords: support vector machine; histogram of oriented gradient; fuzzy control; image process; geometric matching

1. Introduction

In the past decade, intelligent control has brought a huge change to our daily lives. Image processing and robot arms play important roles in many intelligent control designs. There are a lot of applications of image processing and robot arms, such as medical surgery, automatic manufacturing, home care, food processing, etc. According to target positions, robot movement needs to be designed for high precision and stability in the control process. As robotic technologies improve in a variety of ways, many difficult or time-consuming tasks can be performed by robots with high precision and recognition capability [1–4]. Nowadays, global industry is moving to Industry 4.0, which holds the promise of increased flexibility in manufacturing, better quality, and improved productivity [5]. This study is mainly focused on improving the efficiency of smartphone testing procedures so that the proposed system can replace the labor force in routine smartphone tests. Intelligent robots have become a key part in many applications [6–10]; they can provide health care, home service, entertainment, industrial automation, etc. Intelligent robots are used in many research fields, such as for path planning, obstacle avoidance, simultaneous localization and mapping, visual servoing control and image processing [11–13]. Expert knowledge has been integrated into the robot system design that makes interactions between human and robot harmonious. The objective of this study is to replace manual smartphone testing with automatic robot arm adapted to the Chinese character environment. The proposed control scheme can automatically test a smartphone by the use of a robotic system. With support vector machine (SVM) [14], a histogram of oriented gradients (HOG) [15], neural networks, and fuzzy theory, complex computation [16] can be reduced in the control process.

Boser, B.E. et al. [17] improved SVMs and proposed a method to generate nonlinear classifiers using the kernel trick to the maximum margin hyperplane. With HOG describing local object appearance and shape within an image by the distribution of intensity gradients or edge directions, the SVM was able to classify data more successfully. The method utilized for coordinate conversion so that the robot arm can be driven to target positions is the Denavit–Hartenberg parameters (D-H) model [18]. In our study, the target positions are on the smartphone, which are the desired buttons to be pressed. Image processing and pattern recognition techniques are needed to locate the target button position. The webcam captures an image using Red-Green-Blue (RGB) color, which is easily affected by lightness. Hue, saturation and lightness (HSL) color space is usually applied to reduce this effect. The histogram of oriented gradient (HOG) can recognize features of the picture. With HOG describing local object appearance and shape within an image by the distribution of intensity gradients or edge directions, the SVM will be able to classify data more successfully. The SVM-HOG can recognize symbol characters on the screen of the smartphone and ignore the effect of lightness and camera angle. Thus, the HSL transformation process can be omitted, which reduces the calculation burden. In image processing, we do not have to convert the image to HSL color space or another form of binary image. This is an advantage of SVM-HOG in system processes. In optical character recognition, the accuracy of Asian characters documents is usually lower than that for English documents [19–21]. While using optical character recognition (OCR) [22,23] on Chinese characters and Chinese Mandarin phonetic symbol recognition, it is hard to use OCR to compare symbol characters and pictures because the system is based on an English environment. Besides, the performance of OCR is low on symbol recognition. This study replaces OCR character recognition by picture comparison similar to car detection and traffic sign detection. In [24], a supervised learning model called SVM-HOG is used for a car detection methodology in an outdoor environment. In [25], the authors compare k-Nearest Neighbor (k-NN) and SVM; SVM-HOG is better than k-NN-HOG. With SVM-HOG, we can recognize symbol characters on the screen of the smartphones and ignore the effects of lightness and camera angle. In addition, we need machine vision to check whether the character typed by the robot arm is correct or incorrect. Geometric matching [26] can help match the picture of a character in a specific area.

Denavit [27] established four parameters and transformed the matrix of a robot's coordinates. The robot can be controlled to track an assigned trajectory. J.C. Wang [28] designed a system of fast inverse kinematics by using a least-squares approximation that can obtain the joint degree faster. We can transform target coordinates to the robot arm's motor faster by using this method. When the robot arm grabs an object, the magnitude of joint torque can be obtained as in [29]. In [30], fuzzy sliding mode control was applied to a two-link robot arm. The control parameters are tuned by fuzzy-neural techniques. The robot arm consists of forward and inverse kinematics algorithm. In general, forward kinematics is set up by a D-H model [4,31]. Differential kinematics has been derived to control and change the posture of a robot arm. With the D-H model, we can obtain the kinematics of robot arms easily. In inverse kinematics, the complexity is usually more difficult than in forward kinematics. Therefore, many studies presented inverse kinematics by different systems. The robot system in [32] is designed to push the buttons of an elevator using the inverse kinematics technique. In [33], a system solved the inverse kinematics problem that controls the robot arm to lock a screw with the hybrid fuzzy control. These methods are applied to the proposed control design for a smartphone test system.

2. System Description

The robot arm is composed of five MX-28T Dynamixel servo motors. USB2Dynamixel is a device that can directly control Dynamixel from a personal computer or notebook. The visual sensor provides image data to obtain the distance between the robot arm and the target positions of a smartphone by the use of an image processing technique. The robot arm is a five-degrees-of-freedom articulated robot;

the end of the arm is a stylus to touch the smartphone screen. The length of the robot arm is 46.1 cm, as shown in Figure 1. A webcam is used for capturing object image, recognizing the characters on the smartphone screen and identifying buttons on the touch panel.

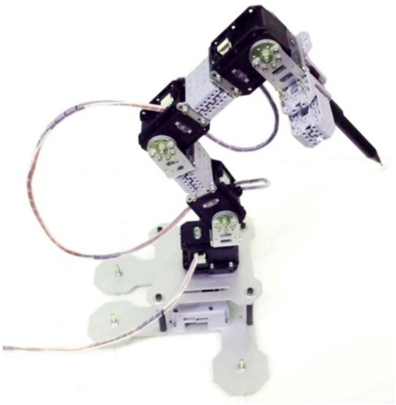


Figure 1. Five-degrees-of-freedom robot arm.

Kinematics [34] mainly discusses the conversion of Cartesian coordinates (x, y and z) (x, y and z) and the mechanical arm angle ($\theta 1$, $\theta 2$, $\theta 3$, $\theta 4$ and $\theta 5$) ($\theta 1$, $\theta 2$, $\theta 3$, $\theta 4$ and $\theta 5$) of the joints. Forward kinematics is a mapping process that concerts the joint space to the operational space with the coordinate of the robot end-effector. Inverse kinematics converts the operational space with the coordinate of the robot end-effector to the joint space and is an inverse of forward kinematics. The presented kinematic model of the arm has five degrees of freedom. Figure 2 shows the model of robotic arm.

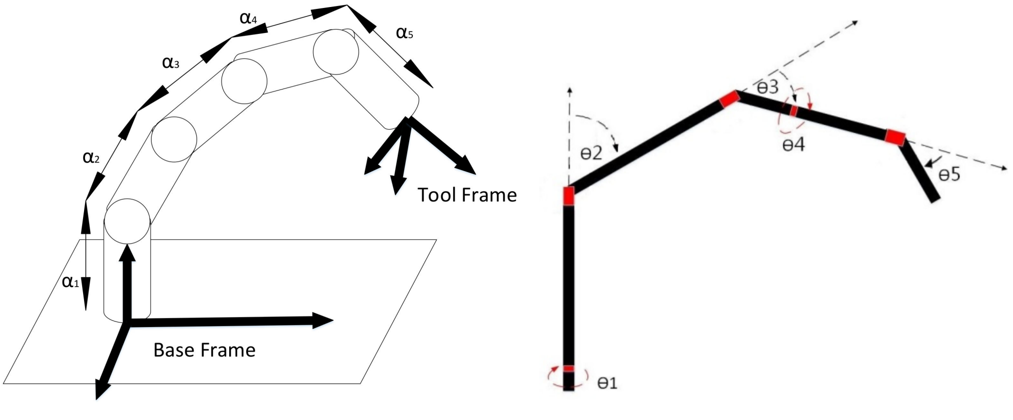


Figure 2. Kinematic model of a robotic arm with five degrees of freedom.

Four parameters are given that describe the kinematic relations of all joints and links [1]. They are:

1. Two neighboring joint relations: the link length a_{i} and the link twist angle α_{i}.
2. Two neighboring link relations: the link offset d_{i} and the joint angle θ_{i}.

These parameters can be obtained by the D-H model which is shown in Figure 2. Table 1 shows the relationship between each joint. The forward kinematics can obtain the Cartesian position and orientation of the end-effector when the angles of joints are given. The D-H parameters of Table 1 are used to obtain homogeneous transformation matrices; it can transform the motion from one coordinate frame reference to the other [1,32].

Table 1. Denavit–Hartenberg (D-H) parameters.

Joint	a_i	α_i	d_i	θ_i
1	a_1 (72 mm)	−90°	d_1 (81 mm) (Base frame)	θ_1
2	a_2 (80 mm)	0°	0	θ_2
3	a_3 (80 mm)	0°	0	θ_3
4	a_4 (45 mm)	−90°	0	θ_4
5	0	0°	d_5 (151 mm) (Stylus)	θ_5

3. Image Processing

The purpose of this study is to control the robot arm to touch the target position. Although optical character recognition (OCR) can recognize Chinese characters, it does not work with Chinese Mandarin phonetic symbols. Because each symbol is too simple and similar, OCR easily fails to recognize them. In our testing experiments, the recognition rate of OCR is only around 40% to 50%. Here, we chose the SVM classifier with the HOG feature descript to recognize Chinese Mandarin phonetic symbols. The coordinate calculation of symbols by using SVM-HOG is processed on MATLAB; then, coordinate information is sent to LabView which is used on robot arm control. The recognition rate increases to about 70 percent by doing so. Still, we need pattern matching to see if the symbols or characters typed by the robot arm are correct or incorrect. Geometric matching provides pattern matching at different angles; even those that are covered by other patterns. Geometric matching uses National Instruments Vision Assistant (NIVA) to achieve the specified function. The NIVA allows one to easily configure and benchmark a sequence of visual assistant steps. For NIVA environment setup, used for the purpose of acquiring images, the system must have NI image acquisition (IMAQ) hardware with NI-IMAQ 3.0 or later installed for IEEE 1394 Cameras 1.5 or later. The NIVA is installed on the Microsoft Windows operating system. A webcam provides images and then the words and symbols on the smartphone screen are recognized by the proposed process as shown in Figure 3. The process has three main steps: image preprocessing, image recognition and dictionary check. Dictionary check is also executed through the NIVA on MATLAB. By using the proposed method, the recognition accuracy increases to 90%.

The angle of image captured by the camera is important. We cut an image into small squares to let each symbol appear in a different square. Because of the viewing angle, symbols might not be completed if cut in the traditional way. The traditional way means a picture is cut into several frames of the same size, and each frame has no overlap with other frames. Thus, we make each square's half part the same as the previous square. In other words, if one square is composed of 50 × 50 pixels, we cut the image at every 25 pixels. This method ensures that every symbol appears completely in at least one square, as shown in Figures 4 and 5. Then, we can use HOG to find features of each symbol in the squares.

The HOG [15,35] is applied to object detection. It is a feature descriptor and is used in image processing. It calculates occurrences of gradient orientation in localized portions of an image. It is similar to shape contexts and edge orientation histograms. The difference is that it calculates dense grids of uniformly spaced cells and uses overlapping local contrast normalization for improving accuracy. In image processing, the first step of computation in feature detectors is to ensure normalized color and gamma values. Since the ensuing descriptor normalization essentially achieves the same result, this step can be skipped in HOG computation. Thus, this image processing produces less impact on performance. The first step is the computation of the gradient values. The most-used technique is the 1-D centered, point discrete derivative mask in one or both of the horizontal and vertical directions. This technique needs to filter the color or intensity data of the image with the following filter kernels:

$$[-1,\ 0,\ 1]\ \mathit{and}\ [-1,\ 0,\ 1]^T \tag{1}$$

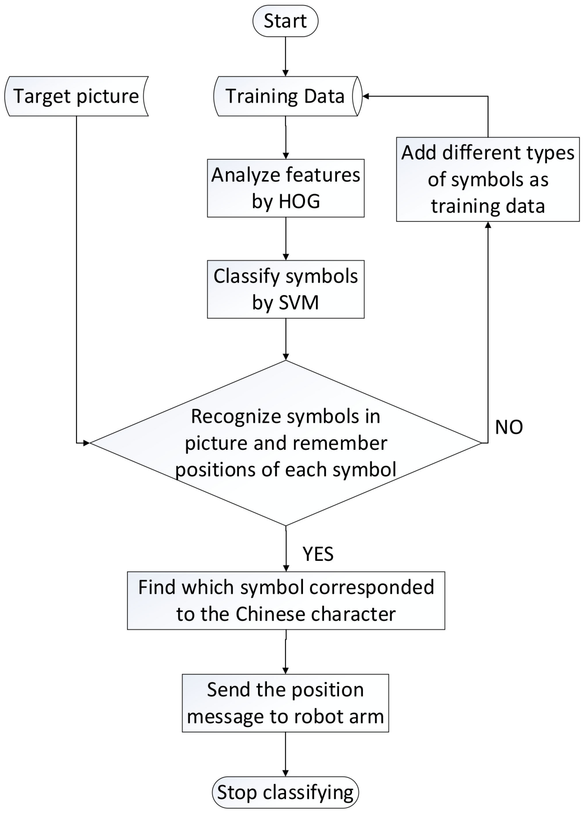

Start
Target picture
Training Data
Analyze features by HOG
Add different types of symbols as training data
Classify symbols by SVM
Recognize symbols in picture and remember positions of each symbol
NO
YES
Find which symbol corresponded to the Chinese character
Send the position message to robot arm
Stop classifying

Figure 3. Procedure of image recognition technique.

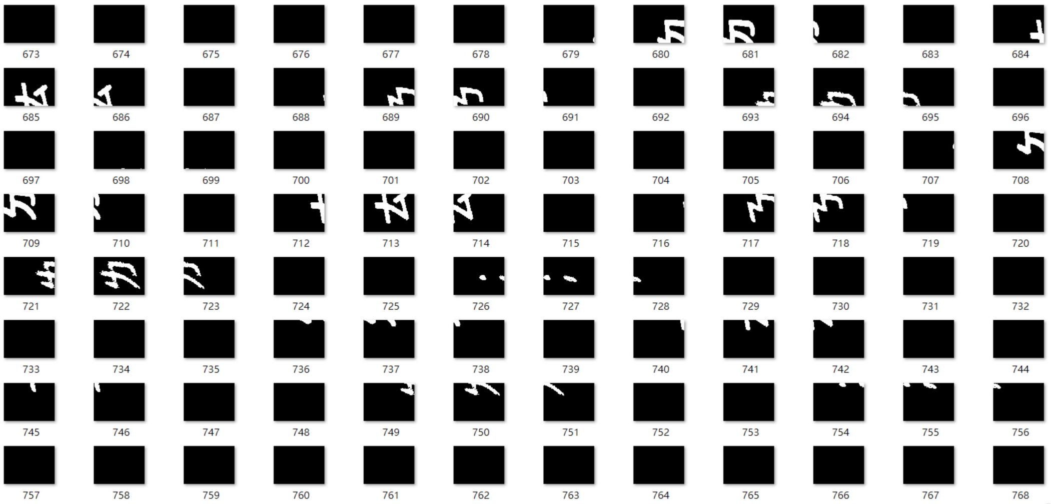


Figure 4. Part A of image preprocessing.

In machine learning, an SVM [14,36] is a supervised learning model with an associated learning algorithm for classification and regression analysis. A set of training examples is marked into two categories. The SVM can assign new examples to one category or the other and make it a non-probabilistic binary linear classifier. The SVM can represent the examples as points in space, and the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples can be mapped into the same space and their belonging can be predicted based on which side of the gap they fall on. For unlabeled data, where supervised learning cannot be applied, unsupervised learning is required. Unsupervised learning can obtain natural clustering of the data to groups and then map new data to these formed groups. Support vector clustering provides improvement to the SVM and is often used in industrial applications whether or not the data are unlabeled or partially labeled. It can be used as a preprocessing for a classification pass, as shown in Figure 6 [36,37]. According to the SVM algorithm, we set a training dataset of n points of the form, where $\boldsymbol{y_i}$ is either 1 or −1 and each $\vec{x}_i$ is a p-dimensional real vector.

$$\left(\vec{x}_i,\ y_i\right),\ \ldots,\ \left(\vec{x}_n,\ y_n\right) \tag{2}$$

Then, we can write any hyperplane satisfying the set of points $\vec{x}$, where $\vec{w}$ is the normal vector to the hyperplane and $\boldsymbol{b}$ is the offset.

$$\vec{w}\cdot\vec{x} - \boldsymbol{b} = \mathbf{0} \tag{3}$$

For each symbol, we prepared seven patterns to help the SVM classify correctly, as shown in Figures 7 and 8. We tried to choose patterns from different angles and in different lights so that the system can adapt to more situations. We first chose different colors of the picture as our pattern but the correct rate was not satisfactory. The correct rate is determined by using seven target pictures as training patterns and another three pictures as testing patterns. We only checked the clearest and most complete squares to determine the correct rate, such as square 1 in Figures 7 and 8. As long as we add more patterns, the correct rate of SVM-HOG will increase generally. However, each pattern for one symbol should not be too similar. When they are too similar, this means the overlap area for different patterns for the same symbol is above 80%. This would cause overfitting, which means the correct rate will be very low when there is a new target picture. Figure 9 shows the database which includes 37 different symbols and 5 other shapes of the buttons on the screen.

Figure 5. Part B of image preprocessing.

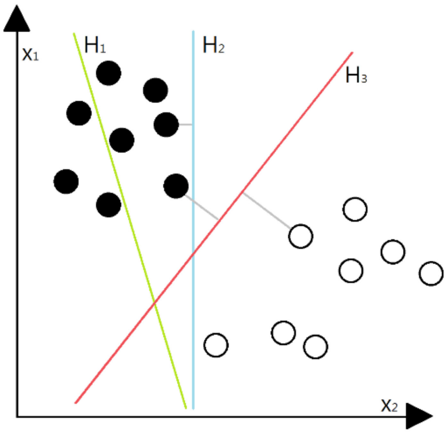


Figure 6. Classification of the support vector machine (SVM).

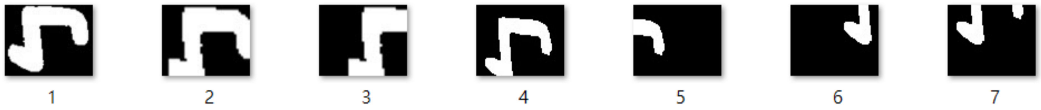


Figure 7. Symbol "ㄅ" in different types.

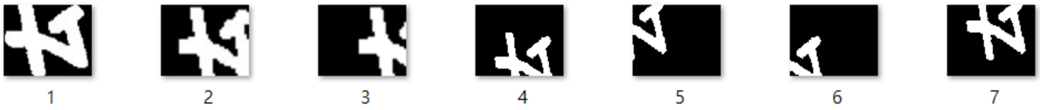


Figure 8. Symbol "ㄊ" in different types.

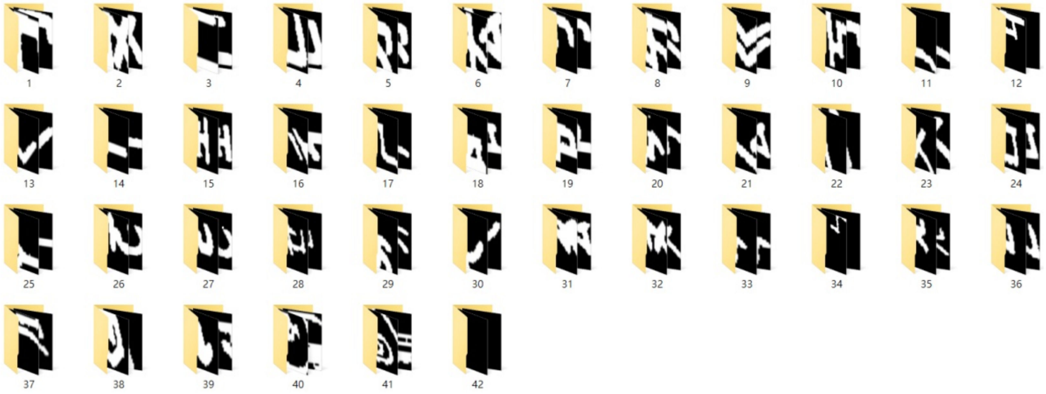


Figure 9. Database of symbols.

Each square includes one piece of coordinate information. When the SVM finishes classification, there will be several pieces of coordinate information for each classification, as shown in Tables 2 and 3. We can see that the number of information in each classification is different because the correct rate of SVM-HOG is not 100 percent. If we calculate and compare the distance of each set of information, it will be difficult and costly. Therefore, we propose an idea which calculates the most common value. In Table 2, the most common values are 820 and 1480, and in Table 3, the most common values are 700 and 1440. This method is positive because there are about five correct squares in each classification, as the red line shown in Figure 10 indicates. These five squares include the most common values of the *x*-coordinate and *y*-coordinate which lead to the correct square of the symbol. Instead of calculating all information, we only calculate one set of information which has the most common value. Using this method, the information of other squares will be abandoned immediately.

Table 2. Coordinate information of symbol "ㄋ".

Square Number	***x*-axis**	***y*-axis**
1	160	1160
2	490	1360
3	820	1440
4	820	1480
5	850	1480

The recognition technique mainly includes two parts in this study, which are SVM-HOG and geometric matching. The Personal Computer (PC) screen only shows the commands of different characters (numbers or words) from the control system, and we can see that there is a strong contrast between the characters and the background. Smartphones also show the characters but we need to consider if the characters are in poor contrast with the background. Examples are shown in Figure 11.

Table 3. Coordinate information of symbol “丨”.

Square Number	*x*-axis	*y*-axis
1	160	1280
2	190	1640
3	220	1880
4	400	1480
5	520	1680
6	700	1440
7	700	1440
8	730	1440

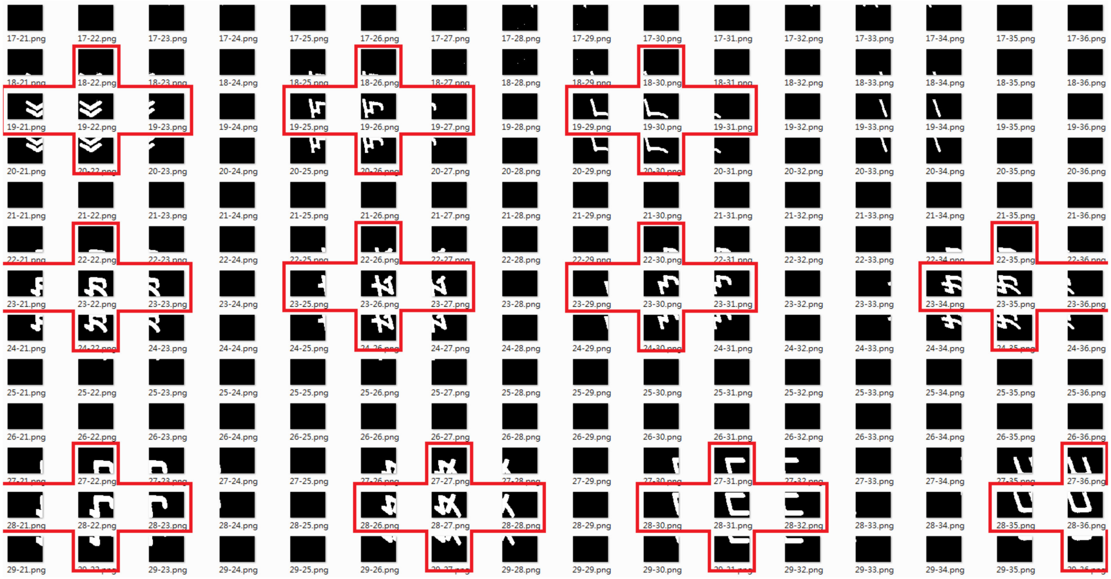
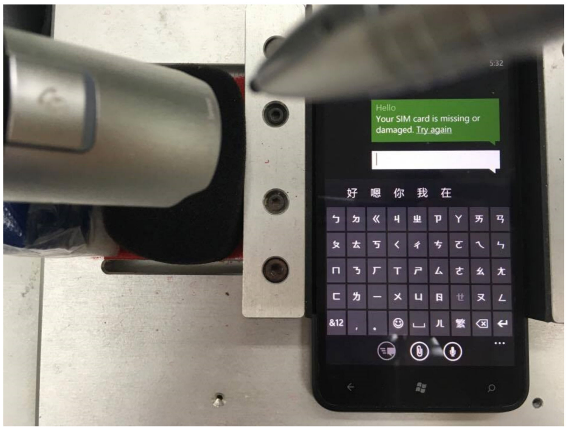

17-21.png 17-22.png 17-23.png 17-24.png 17-25.png 17-26.png 17-27.png 17-28.png 17-29.png 17-30.png 17-31.png 17-32.png 17-33.png 17-34.png 17-35.png 17-36.png
18-22.png 18-24.png 18-26.png 18-28.png 18-30.png 18-32.png 18-33.png 18-34.png 18-35.png 18-36.png
19-21.png 19-22.png 19-23.png 19-24.png 19-25.png 19-26.png 19-27.png 19-28.png 19-29.png 19-30.png 19-31.png 19-32.png 19-33.png 19-34.png 19-35.png 19-36.png
20-21.png 20-23.png 20-24.png 20-25.png 20-27.png 20-28.png 20-29.png 20-31.png 20-32.png 20-33.png 20-34.png 20-35.png 20-36.png
21-21.png 21-22.png 21-23.png 21-24.png 21-25.png 21-26.png 21-27.png 21-28.png 21-29.png 21-30.png 21-31.png 21-32.png 21-33.png 21-34.png 21-35.png 21-36.png
22-22.png 22-24.png 22-26.png 22-28.png 22-30.png 22-32.png 22-33.png 22-35.png
23-21.png 23-22.png 23-23.png 23-24.png 23-25.png 23-26.png 23-27.png 23-28.png 23-29.png 23-30.png 23-31.png 23-32.png 23-33.png 23-34.png 23-35.png 23-36.png
24-21.png 24-23.png 24-24.png 24-25.png 24-27.png 24-28.png 24-29.png 24-31.png 24-32.png 24-33.png 24-34.png 24-36.png
25-21.png 25-22.png 25-23.png 25-24.png 25-25.png 25-26.png 25-27.png 25-28.png 25-29.png 25-30.png 25-31.png 25-32.png 25-33.png 25-34.png 25-35.png 25-36.png
26-21.png 26-22.png 26-23.png 26-24.png 26-25.png 26-26.png 26-27.png 26-28.png 26-29.png 26-30.png 26-31.png 26-32.png 26-33.png 26-34.png 26-35.png 26-36.png
27-22.png 27-24.png 27-25.png 27-27.png 27-29.png 27-31.png 27-33.png 27-34.png 27-36.png
28-21.png 28-22.png 28-23.png 28-24.png 28-25.png 28-26.png 28-27.png 28-28.png 28-29.png 28-30.png 28-31.png 28-32.png 28-33.png 28-34.png 28-35.png 28-36.png
29-21.png 29-23.png 29-24.png 29-25.png 29-26.png 29-28.png 29-29.png 29-30.png 29-32.png 29-33.png 29-34.png 29-35.png

Figure 10. Right squares in target picture.


5:32

Hello
Your SIM card is missing or damaged. Try again

Figure 11. The Mandarin phonetic symbols and Chinese characters image for recognition.

We use geometric matching [21] to locate known references in the image. The image will not be affected by location, orientation, lightness and temperature changes. A desired sample model of the object can be created; then, the similarity of each image can be calculated based on this sample. This model is called a template and should be an ideal representation of the pattern or object. Whether the object is present or not is based on the similarity measurement. The similarity measurement is based on the Euclidean distance. The cross-correlation function is computed from the similarity measurement. If we want to use it for recognizing symbols on the virtual keyboard, we have to regulate the target picture and use YUV (luma component, blue projection and red projection) color space. Figure 12 shows the metric based on the standard Euclidean distance between two sectors [1].

$$I(x,y) = \sqrt{\sum_{x=0}^{L-1}\sum_{y=0}^{K-1}(f(x,y) - p(x,y))^2} \tag{4}$$

$$I(x,y)^2 = \sum_{x=0}^{L-1}\sum_{y=0}^{K-1}\left(f(x,y)^2 - 2\times f(x,y)\times p(x,y) + p(x,y)^2\right) \tag{5}$$

I(*x*,*y*) is the common measure employed when comparing the similarity of two images (e.g., the template *p*(*x*,*y*) and the test image *f*(*x*,*y*)). The normalized cross-correlation (NCC) is used for finding incidences of a pattern or object within an image. NCC uses product concept and is scaled from 0 to 1. When *R* equals 1, that means *p*(*x*,*y*) equals *f*(*x*,*y*).

$$R(x,y) = \frac{\sum\limits_{x=0}^{L-1}\sum\limits_{y=0}^{K-1}(f(x,y)\times p(x,y))}{\sqrt{\sum\limits_{x=0}^{L-1}\sum\limits_{y=0}^{K-1}(f(x,y))^2} \times \sqrt{\sum\limits_{x=0}^{L-1}\sum\limits_{y=0}^{K-1}(p(x,y))^2}} \tag{6}$$

The geometric component feature [26] is a combination of several primitive features that consist of at least two primitive features, such as blobs, corners and edges. At location *x*, the geometric feature vector can be calculated based on the reference point:

$$x_i = x_{i-1} + \sigma_{i-1}d_i\begin{bmatrix} \cos(\theta_{i-1} + \varnothing_i) \\ \sin(\theta_{i-1} + \varnothing_i) \end{bmatrix} \tag{7}$$

$$\theta_i = \theta_{i-1} + \Delta\theta_i \tag{8}$$

$$\sigma_i = \sigma_{i-1}\Delta\sigma_i \tag{9}$$

Here, *x* means the location of features, *θ* is the orientation and *σ* is the intrinsic scale. We determine the correct rate by using 10 target pictures from a camera with different lightness and different angles. The correct rate is 75 percent and geometric matching can successfully find coordinate information on the computer screen.

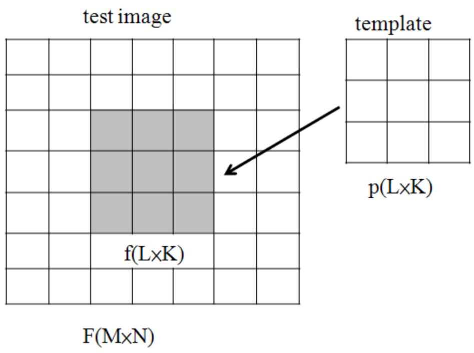
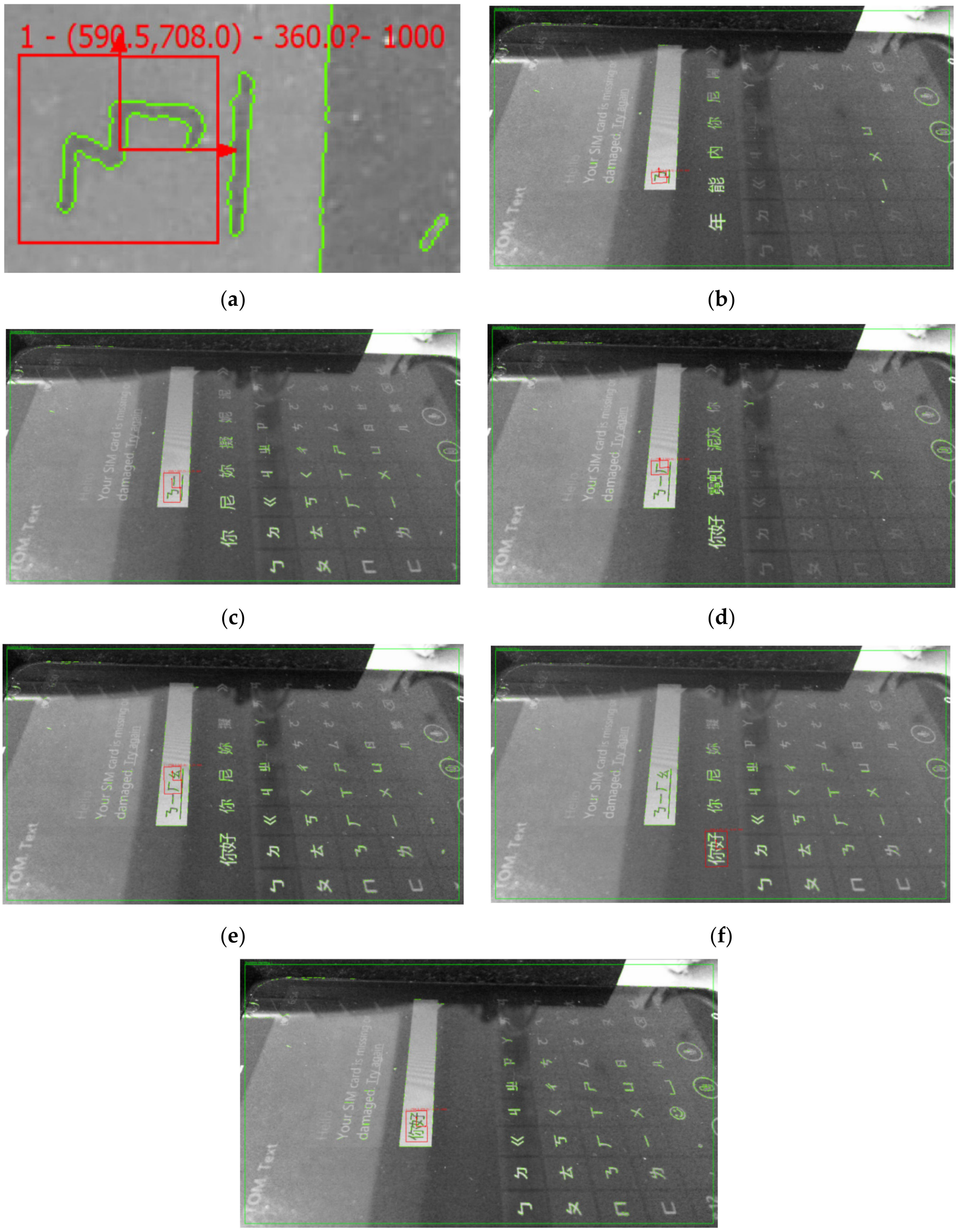


Figure 12. Match pattern.

In addition, the proposed robot system needs to know whether it has pressed the correct button or not. Character recognition and image processes for a poor-contrast smartphone screen were added, which can confirm whether the pressed button is the desired button. The results of geometric matching are shown in Figure 13. We make a decision by checking the command and geometric matching result. If the camera has been moved, this method can also recognize patterns and confirm the pressed button. When the image rotates, geometric matching can still find patterns and its coordinate information. We also set the system to search for patterns which are composed of a white background and black lines so that symbols on the virtual keyboard will not be considered.



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

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

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

(**g**)

Figure 13. Recognition result. (**a**) Coordinate information of "ㄋ". (**b**) Press symbol "ㄋ". (**c**) Press symbol "丨". (**d**) Press symbol "ㄏ". (**e**) Press symbol "ㄠ". (**f**) Press characters "你好". (**g**) Confirm characters "你好".

4. Control Scheme

In most image processing applications, the camera that is used to capture the image has to be stationary. If the camera is moved, related coordinates must be calculated again. The test process will be disturbed in the factory and is very inefficient. In previous work, we used an improved back propagation (BP) neural network with the Levenberg–Marquardt Hidden-Layer Partition (LM-HLP) method [38]. The LM-HLP method is excellent for training the relation of characters and coordinates. Block cyclic data distribution combined the characteristic of block distribution and cyclic distribution. Block distribution distributes source element arrays by the block size that we set up. Cyclic distribution distributes source element arrays to each processor applied by the serial number of the processor. The distribution example is shown in Figure 14. The scheme, in this case, of LM-HLP is shown in Figure 15. Here, a_i is the data to be identified and p_i is the group number. The LM-HLP neural network uses block cyclic distribution on the parallel processing of the hidden layer. Input data are distributed in a block cyclic fashion to each processor for calculation. Each processor is a group, and the elements of each group are the same. This allows processors to calculate at same time, which reduces the learning time of LM-HLP. The flowchart is shown in Figure 16. Once training is completed, the system can handle most situations of error recognition.

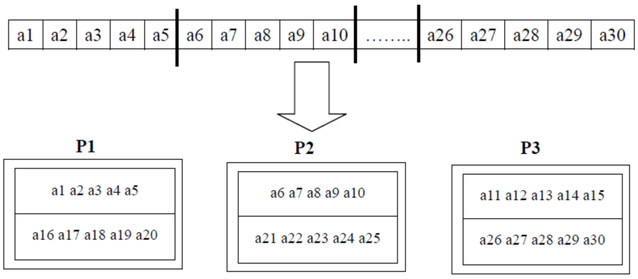


Figure 14. Example of block cyclic distribution.

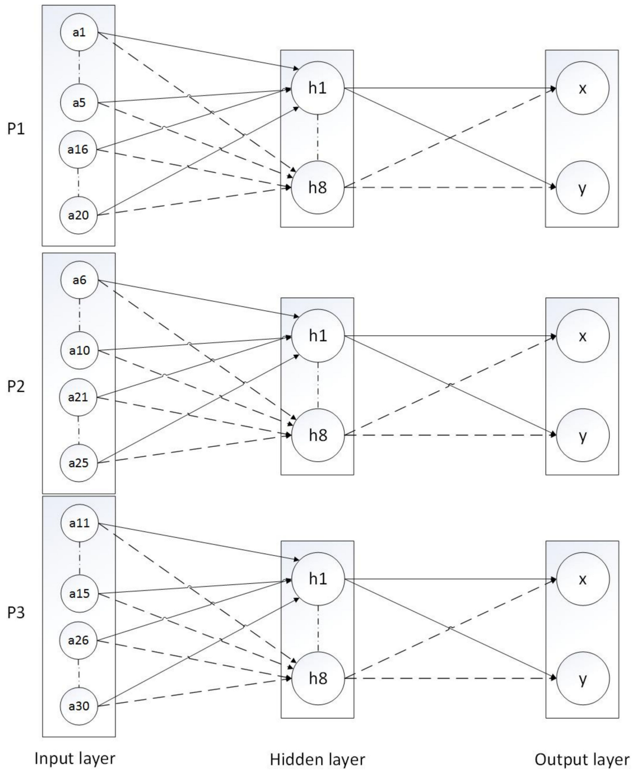


Figure 15. Scheme of Levenberg–Marquardt Hidden-Layer Partition (LM-HLP) neural network.

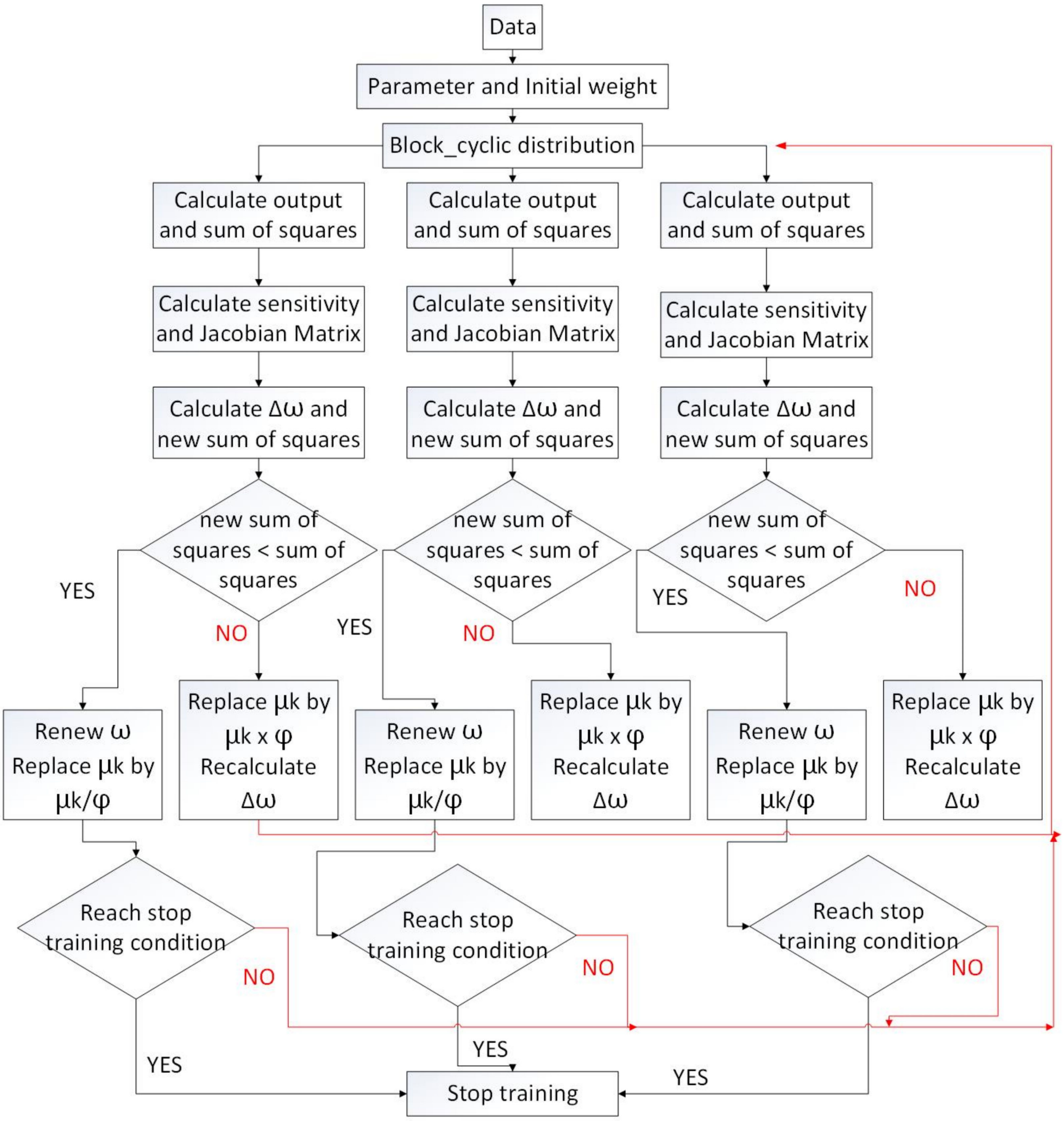

Data
Parameter and Initial weight
Block_cyclic distribution
Calculate output and sum of squares
Calculate sensitivity and Jacobian Matrix
Calculate Δω and new sum of squares
new sum of squares < sum of squares
YES
NO
Renew ω Replace μk by μk/φ
Replace μk by μk x φ Recalculate Δω
Reach stop training condition
Stop training

Figure 16. Flowchart of LM-HLP.

However, the camera must be stationary. Our solution is to make SVM-HOG keep calculating. As mentioned in previous section, and shown in Figures 17 and 18, SVM-HOG does not require that the image be transformed from RGB color space to YUV color space, which is required with LM-HLP. The results of these two methods are almost the same. Further, SVM-HOG can recognize at least one square correctly and at most can recognize five squares correctly. This ensures that unexpected accidents will not disturb the process. We let the system execute SVM-HOG every 30 s, which keeps the robot arm from continuously pressing the wrong button. In addition to the camera being moved, the smartphone itself being moved has also been considered. As long as the smartphone is still in field of view of the camera, the proposed method is very useful. Figures 19 and 20 show the camera being moved, which causes the target picture to change. Figures 21 and 22 compare the changes between the original and latest coordinate information. Compare with [38]: its scheme is executed every 30 s and can also complete this task, but the proposed method only uses 25 s and is custom fit to a Chinese and Mandarin phonetic symbols environment.

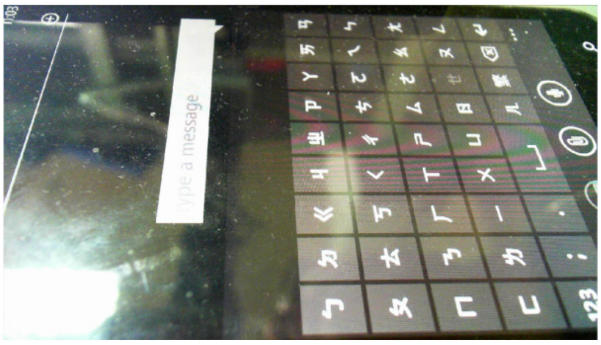


Figure 17. RGB image.

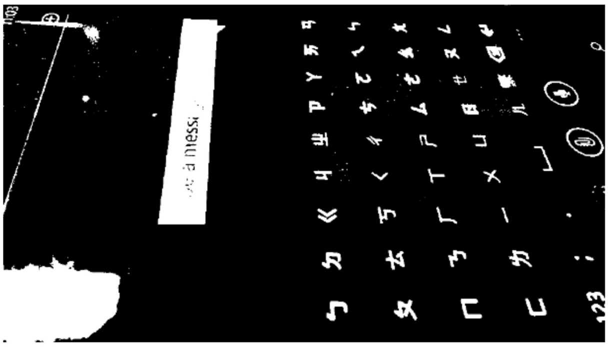


Figure 18. Binary image.

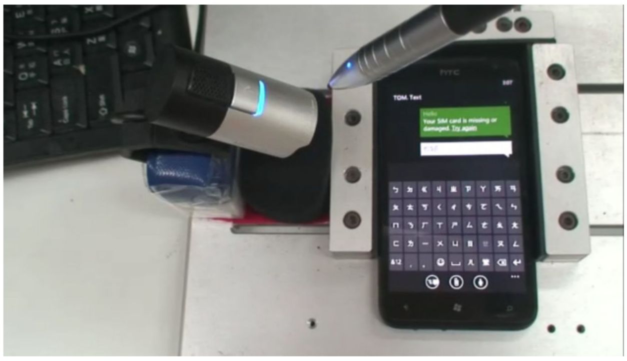


Figure 19. Before moving the camera.

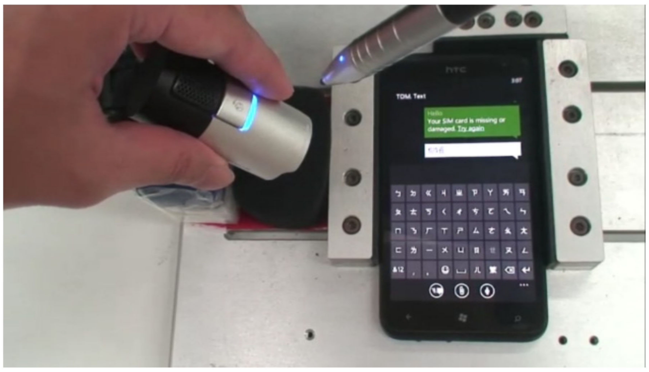
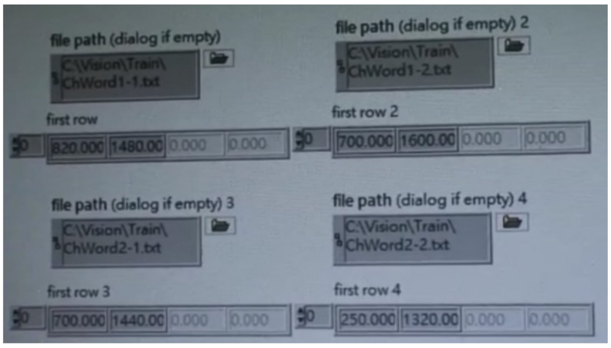


Figure 20. After moving the camera.



file path (dialog if empty)
C:\Vision\Train\ChWord1-1.txt

first row
0 820.000 1480.00 0.000 0.000

file path (dialog if empty) 2
C:\Vision\Train\ChWord1-2.txt

first row 2
0 700.000 1600.00 0.000 0.000

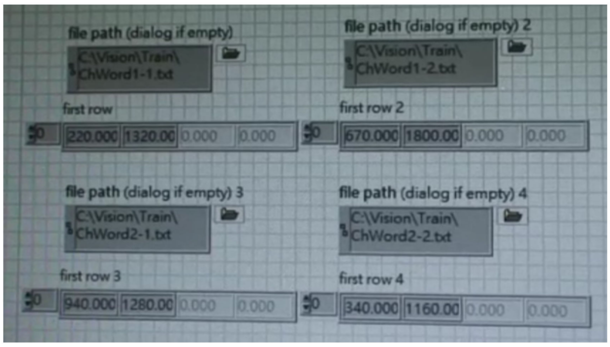
file path (dialog if empty) 3
C:\Vision\Train\ChWord2-1.txt

first row 3
0 700.000 1440.00 0.000 0.000

file path (dialog if empty) 4
C:\Vision\Train\ChWord2-2.txt

first row 4
0 250.000 1320.00 0.000 0.000

Figure 21. The original coordinate information.



file path (dialog if empty)
C:\Vision\Train\ChWord1-1.txt

first row
0 220.000 1320.00 0.000 0.000

file path (dialog if empty) 2
C:\Vision\Train\ChWord1-2.txt

first row 2
0 670.000 1800.00 0.000 0.000

file path (dialog if empty) 3
C:\Vision\Train\ChWord2-1.txt

first row 3
0 940.000 1280.00 0.000 0.000

file path (dialog if empty) 4
C:\Vision\Train\ChWord2-2.txt

first row 4
0 340.000 1160.00 0.000 0.000

Figure 22. The latest coordinate information.

Fuzzy logic controllers have been utilized to control nonlinear systems successfully. Figure 23 shows the position control scheme with the D-H model and a fuzzy controller. The target coordinates obtained from the camera are normalized and sent into the D-H model to obtain five angles for each joint of the robot arm. Figure 24 shows the control sequence.

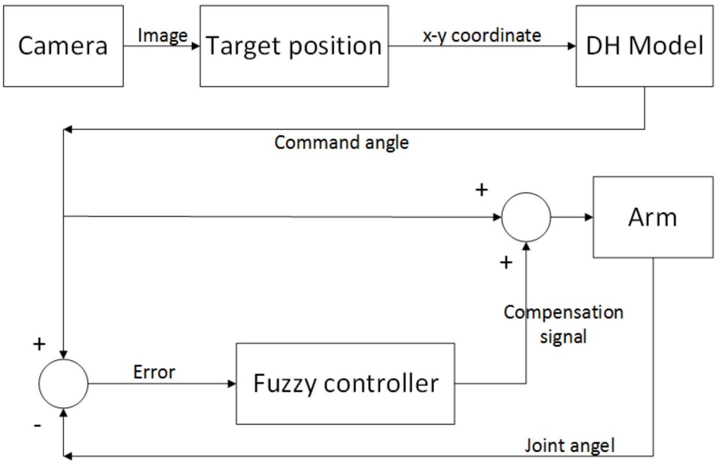


Figure 23. Position control scheme with LM-HLP and fuzzy controller.

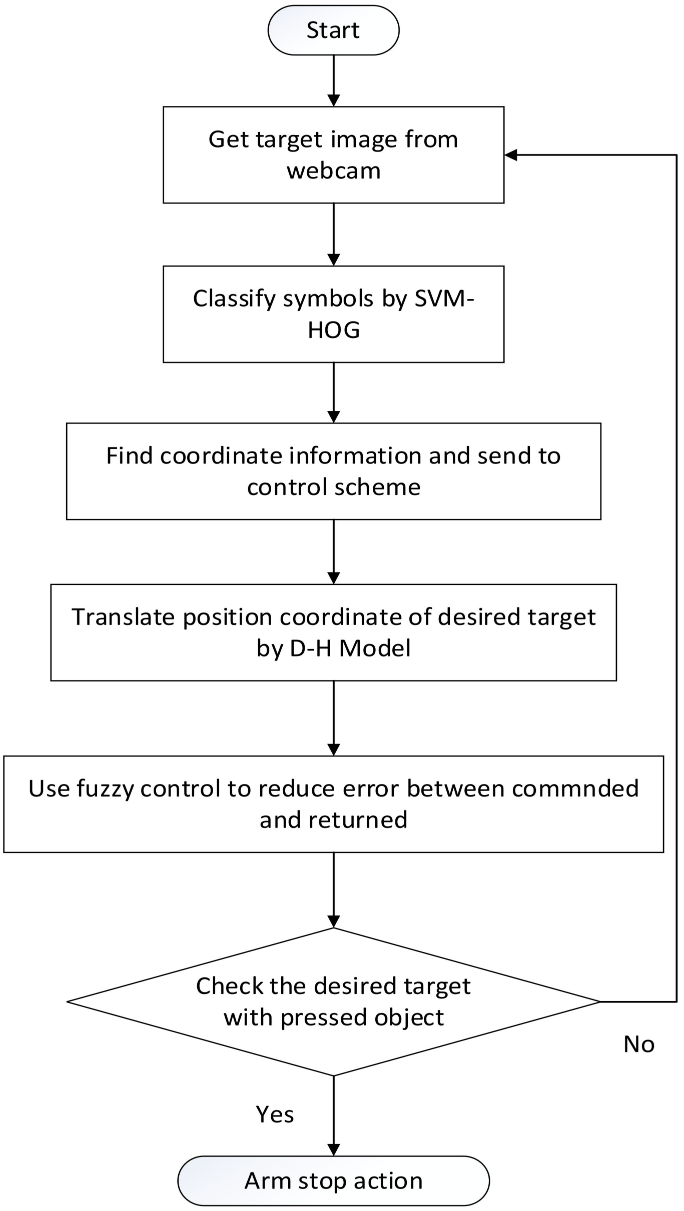

Start
Get target image from webcam
Classify symbols by SVM-HOG
Find coordinate information and send to control scheme
Translate position coordinate of desired target by D-H Model
Use fuzzy control to reduce error between commnded and returned
Check the desired target with pressed object
No
Yes
Arm stop action

Figure 24. Flowchart of the control sequence.

Fuzzy theory is used to reduce error between commanded and returned degrees of five servo motors. In a fuzzy system, the complex mathematical model of the robot arm is not required. Each layer has only one input and one output. Fuzzy rules are given as follows:

Rule 1: If *Error* is negative big, then M is NB.
Rule 2: If *Error* is negative small, then M is NS.
Rule 3: If *Error* is zero, then M is Z.
Rule 4: If *Error* is positive small, then M is PS.
Rule 5: If *Error* is positive big, then M is PB.

where *Error* is $C - R$, M is θ, C is command degree and R is returned degree.

Fuzzy sets are NB, NS, Z, PS and PB, which represent turning negative big, negative small, zero, positive small and positive big, respectively. Figure 25 shows the membership functions of the degree *Error*. Figure 26 shows the membership functions of the angle error. Figure 27 shows the fuzzy control block diagram.

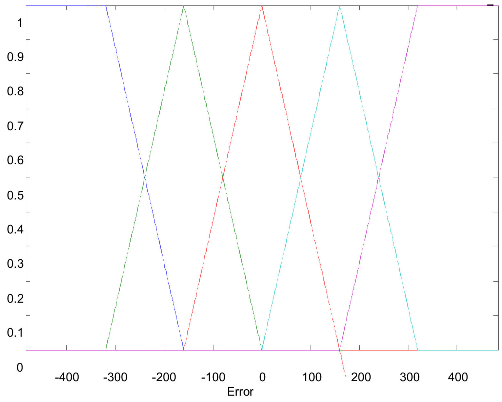


Figure 25. Membership functions of the degree error.

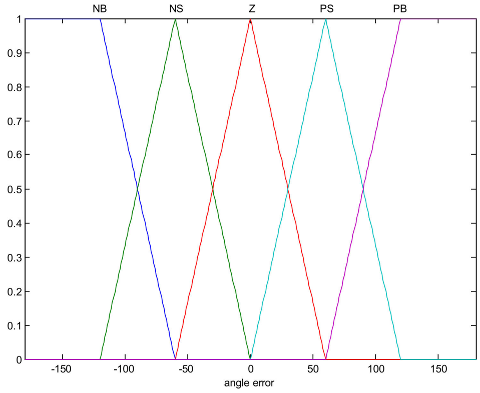


Figure 26. Membership functions of the angle error.

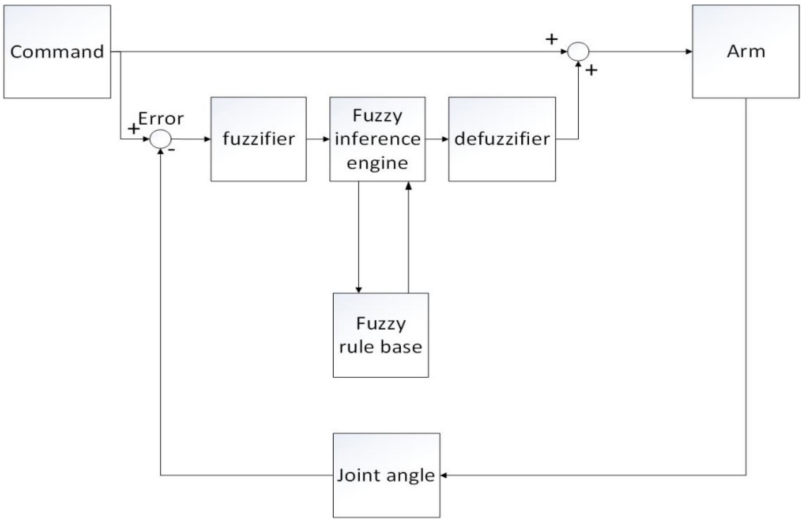


Figure 27. Fuzzy control block diagram.

5. Experiments

In our experiments, we designed three kinds of motions for the robot arm; these are Automatically update coordinate information, Delete wrong symbols and Select the right Chinese characters. Table 4 shows the recognition performance of different symbols and characters recognized by the proposed system. The average recognition rate is 90.3%. Symbols that are difficult to recognize are “ㄧ” and “ㄏ”, which is because of false recognition and unrecognized symbols. Performance can be improved by adding a dictionary process. The recognition accuracy can reach up to 98%. This experiment presents automatically updated coordinate information, as shown in Figure 28. First, type four symbols “ㄋ”,

"ㄧ", "ㄏ", "ㄠ" and characters "你好". We used geometric matching to confirm whether it is right or wrong at every interval. Second, the camera is move to change the target picture. After the system calculates and updates the new coordinate information, we let the robot arm type four symbols, "ㄋ", "ㄧ", "ㄏ" and "ㄠ", and the characters "你好" again with the new coordinate information. There are 15 steps in Figure 28.

The next experiment presents the robot arm deleting wrong symbols, as shown in Figure 29. We used geometric matching to confirm whether it is right or wrong at every interval. If the system fails to find a pattern, the robot arm will press the delete button and then press the right button again. For example, when the symbol "ㄨ" appears on the screen but not the symbol "ㄏ", then the system will fail to find the pattern "ㄏ". There are eight steps in Figure 29.

The last experiment presents a mechanism for the robot arm to select the right Chinese characters, as shown in Figure 30. The purpose of this mechanism is to make system select the correct characters without the five tones in Traditional Chinese. With geometric matching, the system does not need to press all of steps as in the first experiment. There are five steps in Figure 30.

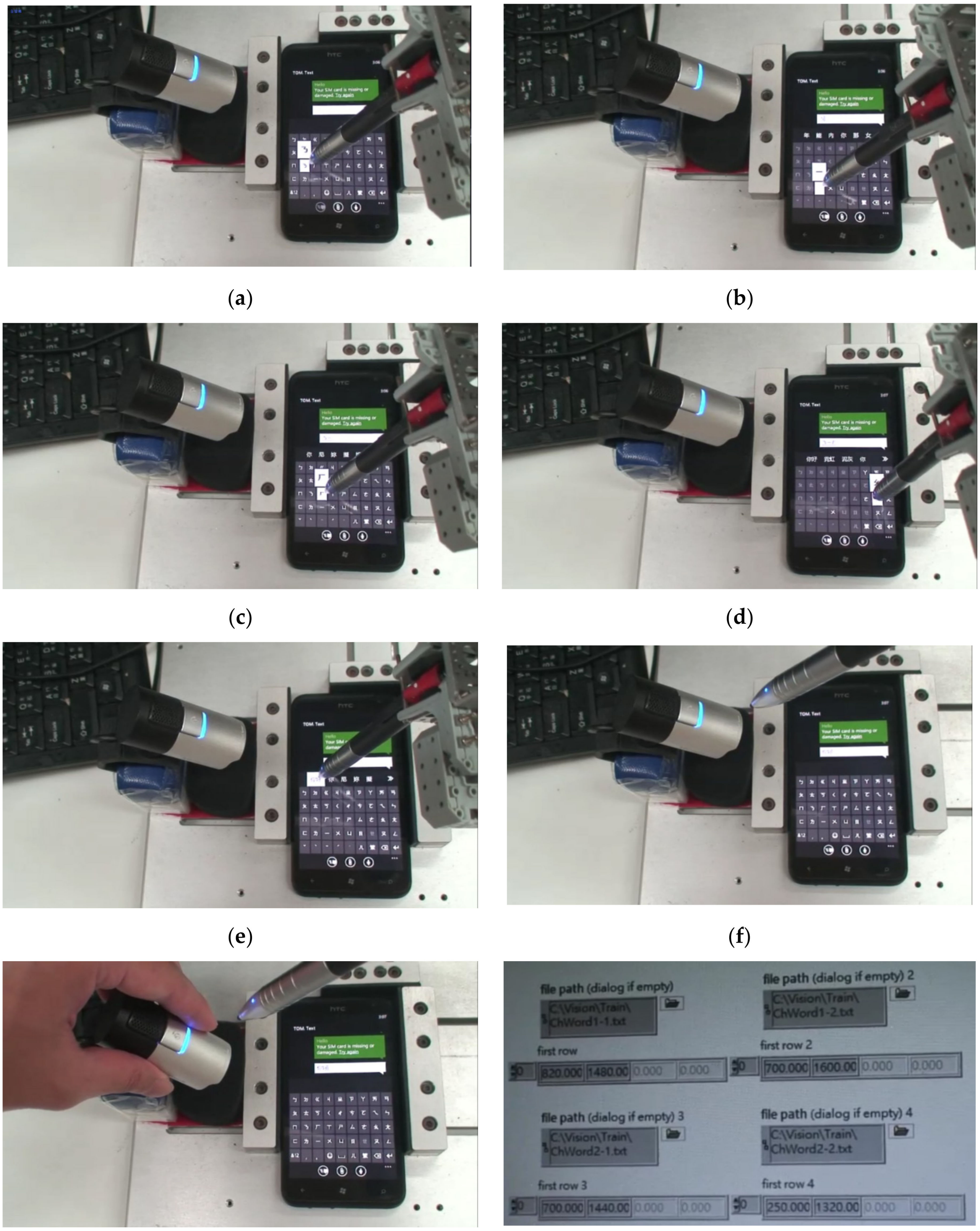
An intelligent control scheme based on image processing, SVM-HOG, the D-H model and fuzzy control was applied to a robot arm for position control. Using these methods, the control system can compute an accurate position of the coordinate information that results in a success rate of symbol recognition of up to 98% in real time. Characters can be checked by the use of geometric matching. Table 5 shows the success rates of experiments after 10 times. The second mechanism, Delete wrong symbols, has the lowest success rate. Because different wrong symbols will cause different results, there are some symbols very similar to each other and they are difficult to identify. The third mechanism, Select the right Chinese characters, has the highest success rate because the pattern of Chinese characters is very clear to recognize.

Table 4. Recognition performance of symbols and characters on a smartphone screen (testing times: 100).

Characters	Recognized %	Problems	Dictionary %
你	100	-	100
好	100	-	100
我	100	-	100
是	100	-	100
Symbols	**Recognized %**	**Problems**	**Dictionary %**
ㄋ	100	-	100
ㄋㄧ	73	ㄧ recognized as ㄏ or other lines	98
ㄏ	89	ㄏ recognized as ㄧ or ㄕ	100
ㄏㄠ	100	-	100
ㄨ	93	ㄨ not recognized	100
ㄨㄛ	91	ㄛ recognized as ㄜ	99
ㄕ	86	ㄕ recognized as ㄏ	98

Table 5. The experimental results (testing times: 10).

Command	Success Rate
Automatically update coordinate information	98%
Delete wrong symbols	88%
Select the right Chinese characters	100%



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

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

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

(**g**) (**h**)

file path (dialog if empty)
C:\Vision\Train\ChWord1-1.txt

file path (dialog if empty) 2
C:\Vision\Train\ChWord1-2.txt

first row
0 820.000 1480.00 0.000 0.000

first row 2
0 700.000 1600.00 0.000 0.000

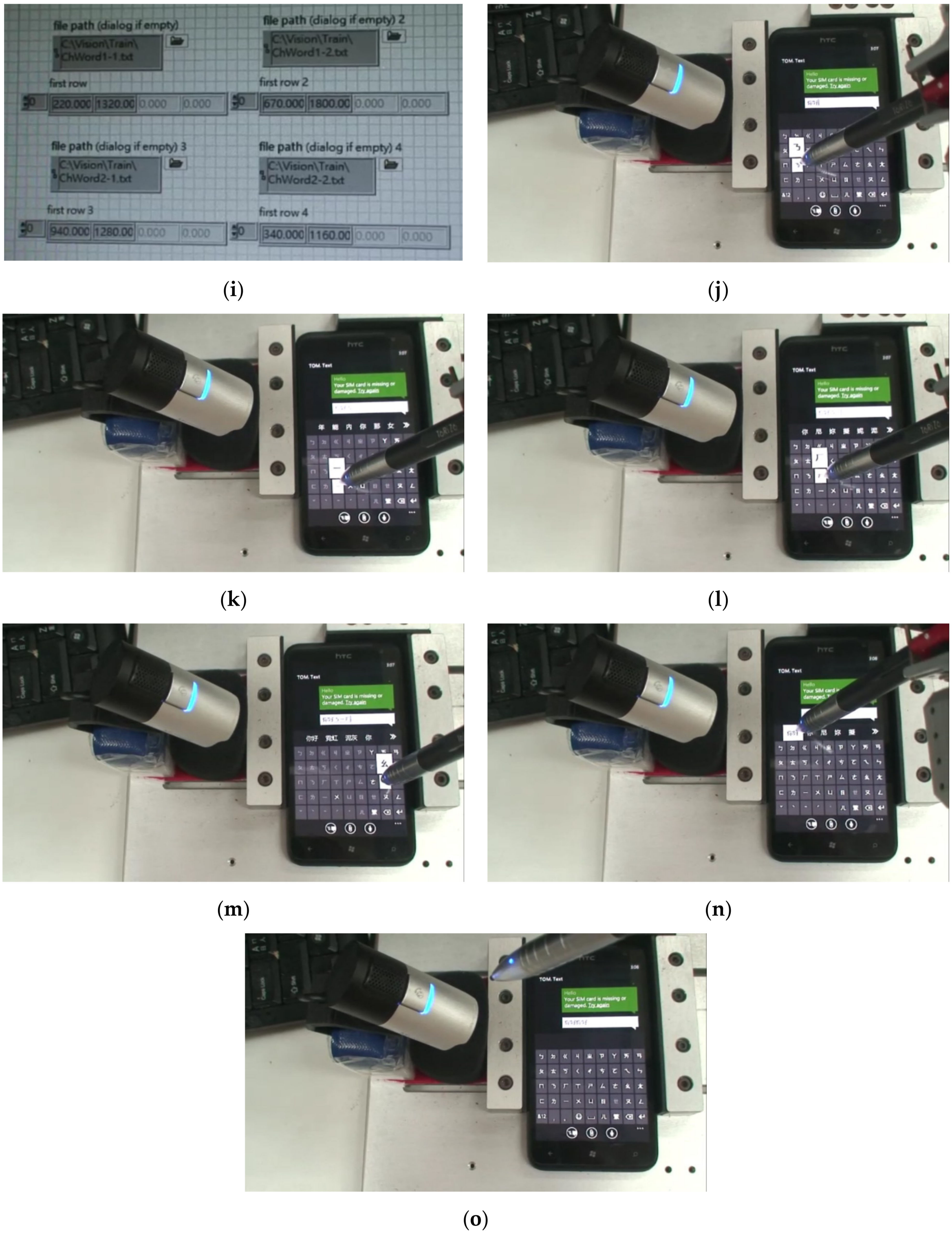
file path (dialog if empty) 3
C:\Vision\Train\ChWord2-1.txt

file path (dialog if empty) 4
C:\Vision\Train\ChWord2-2.txt

first row 3
0 700.000 1440.00 0.000 0.000

first row 4
0 250.000 1320.00 0.000 0.000

Figure 28. *Cont.*



file path (dialog if empty)
C:\Vision\Train\ChWord1-1.txt
first row
0 220.000 1320.00 0.000 0.000

file path (dialog if empty) 2
C:\Vision\Train\ChWord1-2.txt
first row 2
0 670.000 1800.00 0.000 0.000

file path (dialog if empty) 3
C:\Vision\Train\ChWord2-1.txt
first row 3
0 940.000 1280.00 0.000 0.000

file path (dialog if empty) 4
C:\Vision\Train\ChWord2-2.txt
first row 4
0 340.000 1160.00 0.000 0.000

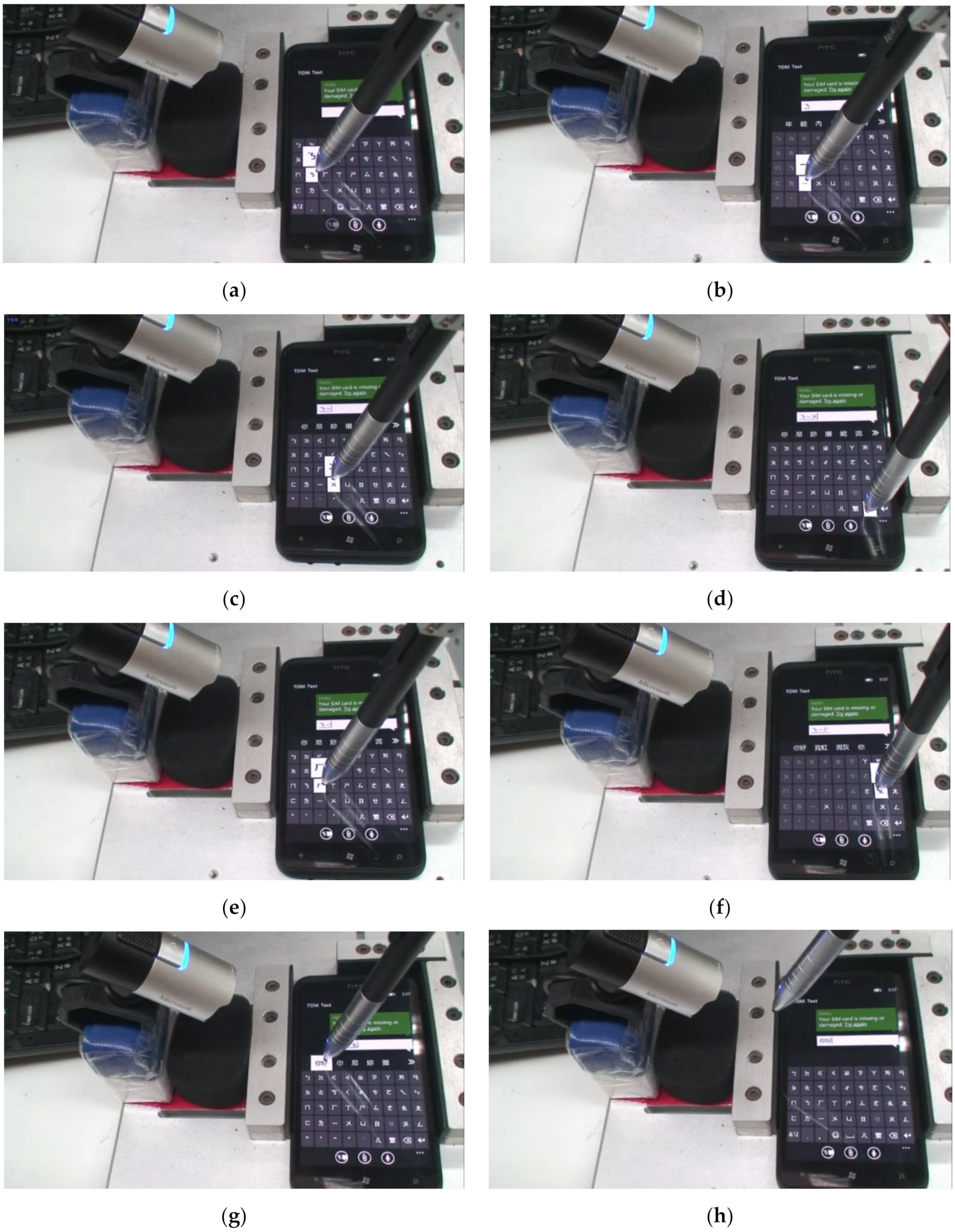
(i) (j)

(k) (l)

(m) (n)

(o)

Figure 28. Automatically updating coordinate information. (**a**) Press symbol "ㄋ". (**b**) Press symbol "ㄧ". (**c**) Press symbol "ㄏ". (**d**) Press symbol "ㄠ". (**e**) Press characters "你好". (**f**) Prepare to move camera. (**g**) Move camera. (**h**) The original coordinate information. (**i**) The latest coordinate information. (**j**) Press symbol "ㄋ" again. (**k**) Press symbol "ㄧ" again. (**l**) Press symbol "ㄏ" again. (**m**) Press symbol "ㄠ" again. (**n**) Press characters "你好" again. (**o**) Confirm characters.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**) (**g**) (**h**)

Figure 29. Delete wrong symbols. (**a**) Press symbol "ㄋ". (**b**) Press symbol "ㄧ". (**c**) Press wrong symbol "ㄨ". (**d**) Delete wrong symbol "ㄨ". (**e**) Press symbol "ㄏ". (**f**) Press symbol "ㄠ". (**g**) Press characters "你好". (**h**) Confirm characters "你好".

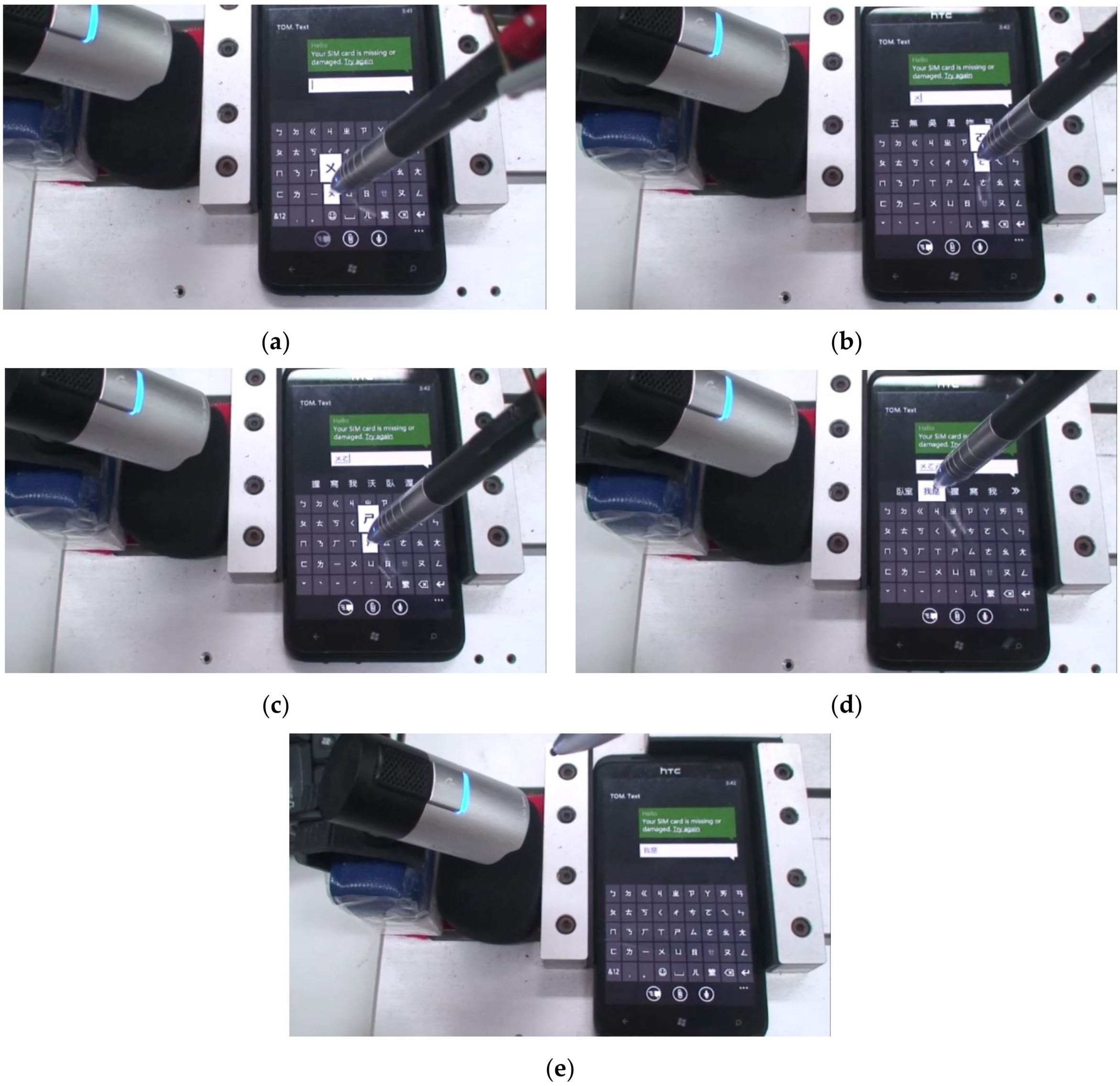


Figure 30. Selection of the right Chinese characters. (**a**) Press symbol “ㄨ”. (**b**) Press symbol “ㄛ”. (**c**) Press symbol “ㄕ”. (**d**) Press characters “我是” but not “室”. (**e**) Confirm characters.

6. Conclusions

This study proposed a control scheme that applied pattern recognition, fuzzy control, and SVM-HOG to a robotic smartphone testing system that can replace manual testing and reduce manpower. The inverse kinematics, SVM-HOG, fuzzy theory, and transformation between webcam coordinates and robot arm coordinates are used for catching the digits or letters outside of the target center. The solution for the inverse kinematics is obtained by use of the D-H model. The images are transformed from the RGB color space to the binary color space. Generally speaking, binary color space is not necessary for SVM-HOG. SVM-HOG is also capable of analyzing and classifying RGB color space images, but for a higher correct rate and geometric matching, transforming RGB color space to binary color space is less burdensome than transforming to YUV color space. A frame of image is cut into small squares, and its half part is the same as the previous one. The SVM classifies symbols with 42 classifications after the HOG analyzes features of the squares. Vision Assistant allows one to configure and benchmark a sequence of visual inspection steps and apply a visual inspection system for automated inspection. A geometric matching recognition program has been employed by the use of NIVA, and patterns can be corrected by a dictionary process. In addition, characters can be checked by image-processing techniques and the program is able to recognize patterns that are incomplete. In the control scheme, the robot system can compute a position in real time using the D-H model and fuzzy controller. By converting coordinates every time through the D-H model, the degrees of robot joints can be updated by SVM-HOG. Recognition accuracy is 90.3% for images taken from the webcam and 98% after using a dictionary process. From the experiments, the proposed control scheme can allow the robot arm to perform different mechanisms of the smartphone test in a Traditional Chinese language environment successfully. Most importantly, the robot arm is able to deal with the situation of the camera or smartphone being slightly moved. The limitation is that the movement of the camera or the smartphone cannot be greater than 1 cm, and in such a case, the system setup needs to be recalibrated.

Author Contributions: C.-C.L.: data curation, methodology, resources, software, writing—original draft. J.-G.J.: conceptualization, formal analysis, funding acquisition, supervision, writing—review & editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding. The APC was funded by the National Taiwan Ocean University.

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

References

1. Juang, J.-G.; Cheng, I.-H. Application of Character Recognition to Robot Control on Smartphone Test System. *Adv. Mech. Eng.* **2017**, *9*, 1–17. [CrossRef]
2. Juang, J.G.; Yu, C.L.; Lin, C.M.; Yeh, R.G.; Rudas, I.J. Real-Time Image Recognition and Path Tracking to Wheeled Mobile Robot for Taking an Elevator. *Acta Polytech. Hung.* **2013**, *10*, 5–23.
3. Juang, J.-G.; Tsai, Y.-J.; Fan, Y.-W. Visual Recognition and Its Application to Robot Arm Control. *Appl. Sci.* **2015**, *5*, 851–880. [CrossRef]
4. Chen, L.; Guo, H.; Wang, H.; Chen, Y.-L.; Wu, X. The Visual Location of Workpiece based on Hermite Interpolation and Mapping for Robot Arms. In Proceedings of the 2015 5th International Conference on Information Science and Technology (ICIST), Changsha, China, 24–26 April 2015; pp. 171–176.
5. Zhong, R.; Xu, X.; Klotz, E.; Newman, S.T. Intelligent Manufacturing in the Context of Industry 4.0: A Review. *Engineering* **2017**, *3*, 616–630. [CrossRef]
6. Chi, Z.; Xu, Q. Precision Control of Piezoelectric Actuator using Fuzzy Feedback Control with Inverse Hysteresis Compensation. In Proceedings of the 10th IEEE International Conference on Nano/Micro Engineered and Molecular Systems, Xi'an, China, 7–11 April 2015; pp. 219–224.
7. Gang-Feng, L.; Ying-Yong, Z.; Chang-Le, L.; Ji-Hong, Y. Design and Optimization of Spraying Robot Arm for Hull Blocks. In Proceedings of the 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai, China, 6–9 December 2015; pp. 2615–2620.
8. Rivai, M.; Rendyansyah; Purwanto, D. Implementation of Fuzzy Logic Control in Robot Arm for Searching Location of Gas Leak. In Proceedings of the 2015 International Seminar on Intelligent Technology and Its Applications (ISITIA), Surabaya, Indonesia, 20–21 May 2015; pp. 69–74. [CrossRef]
9. Pamungkas, D.S.; Ward, K. Immersive Teleoperation of a Robot Arm using Electro-tactile Feedback. In Proceedings of the 2015 6th International Conference on Automation, Robotics and Applications (ICARA), Queenstown, New Zealand, 17–19 February 2015; pp. 300–305.
10. Suligoj, F.; Jerbic, B.; Svaco, M.; Sekoranja, B.; Mihalinec, D.; Vidakovic, J. Medical Applicability of a Low-cost Industrial Robot Arm Guided with an Optical Tracking System. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 3785–3790.
11. Shih, C.-H.; Juang, J.-G. Moving Object Tracking and Its Application to an Indoor Dual-Robot Patrol. *Appl. Sci.* **2016**, *6*, 349. [CrossRef]
12. Juang, J.G.; Wang, J.A. Fuzzy Control Simultaneous Localization and Mapping Strategy Based on Iterative Closest Point and k-Dimensional Tree Algorithms. *Sens. Mater.* **2015**, *27*, 733–741.
13. Huang, G.S.; Tung, C.K.; Lin, H.C.; Hsiao, S.H. Inverse Kinematics Analysis Trajectory Planning for a Robot Arm. In Proceedings of the 2011 IEEE International Conference on Robotics & Automation, Kaohsiung, Taiwan, 15–18 May 2011; pp. 965–970.

14. Ardianto, S.; Chen, C.-J.; Hang, H.-M. Real-time Traffic Sign Recognition using Color Segmentation and SVM. In Proceedings of the 2017 International Conference on Systems, Signals and Image Processing (IWSSIP), Poznan, Poland, 22–24 May 2017; pp. 1–5.
15. Kumar, P.; Happy, S.L.; Routray, A. A Real-time Robust Facial Expression Recognition System using HOG Features. In Proceedings of the 2016 International Conference on Computing, Analytics and Security Trends (CAST), Pune, India, 19–21 December 2016; pp. 289–293.
16. Juang, J.-G.; Yang, C.-Y. Document Delivery Robot Based On Image Processing and Fuzzy Control. *Trans. Can. Soc. Mech. Eng.* **2016**, *40*, 677–692. [CrossRef]
17. Boser, B.E.; Guyon, I.M.; Vapnik, V.N. *A Training Algorithm for Optimal Margin Classifiers*; Association for Computing Machinery (ACM): New York, NY, USA, 1992; pp. 144–152.
18. Wu, L.; Crawford, R.W.; Roberts, J.M. An Analytic Approach to Converting POE Parameters into D–H Parameters for Serial-Link Robots. *IEEE Robot. Autom. Lett.* **2017**, *2*, 2174–2179. [CrossRef]
19. Park, J.; Lee, E.; Kim, Y.; Kang, I.; Koo, H.I.; Cho, N.I. Multi-Lingual Optical Character Recognition System Using the Reinforcement Learning of Character Segmenter. *IEEE Access* **2020**, *8*, 174437–174448. [CrossRef]
20. Kaur, S.; Bawa, S.; Kumar, R. A Survey of Mono- and Multi-lingual Character Recognition using Deep and Shallow Architectures: Indic and non-indic scripts. *Artif. Intell. Rev.* **2019**, *53*, 1813–1872. [CrossRef]
21. Yin, Y.; Zhang, W.; Hong, S.; Yang, J.; Xiong, J.; Gui, G. Deep Learning-Aided OCR Techniques for Chinese Uppercase Characters in the Application of Internet of Things. *IEEE Access* **2019**, *7*, 47043–47049. [CrossRef]
22. Wang, X.; Lai, W. Edge Detection for Chinese Text Image based on Novel Differential Operator. In Proceedings of the 2010 International Conference on Computer and Information Application, Tianjin, China, 3–5 December 2010; pp. 44–47.
23. Liu, M.; Suo, Y.; Ding, Y. Research on Optimization Segmentation Algorithm for Chinese/English Mixed Character Image in OCR. In Proceedings of the 2014 Fourth International Conference on Instrumentation and Measurement, Computer, Communication and Control, Harbin, China, 18–20 September 2014; pp. 764–769.
24. Guzmán, S.; Gómez, A.; Diez, G.; Fernández, D. Car Detection Methodology in Outdoor Environment Based on Histogram of Oriented Gradient (HOG) and Support Vector Machine (SVM). In Proceedings of the 6th Latin-American Conference on Networked and Electronic Media (LACNEM 2015), Medellin, Colombia, 23–25 September 2015; p. 4.
25. Sugiharto, A.; Harjoko, A. Traffic Sign Detection based on HOG and PHOG using Binary SVM and k-NN. In Proceedings of the 2016 3rd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), Semarang, Indonesia, 19–20 October 2016; pp. 317–321.
26. Sharma, G.; Sood, S.; Gaba, G.S.; Gupta, N. Image Recognition System using Geometric Matching and Contour Detection. *Int. J. Comput. Appl.* **2012**, *51*, 48–53. [CrossRef]
27. Denavit, J.; Hartenberg, R.S. A Kinematic Notation for Lower-pair Mechanisms Based on Matrices. *Trans. ASME J. Appl. Mech.* **1955**, *23*, 215–221.
28. Manigpan, S.; Kiattisin, S.; Leelasantitham, A. A simulation of 6R Industrial Articulated Robot Arm using Backpropagation Neural Network. In Proceedings of the ICCAS 2010, Gyeonggi-do, Korea, 27–30 October 2010; pp. 823–826.
29. Koga, M.; Kosuge, K.; Furuta, K.; Nosaki, K. Coordinated Motion Control of Robot Arms based on the Virtual Internal Model. *IEEE Trans. Robot. Autom.* **1992**, *8*, 77–85. [CrossRef]
30. Efe, M. Önder Fractional Fuzzy Adaptive Sliding-Mode Control of a 2-DOF Direct-Drive Robot Arm. *IEEE Trans. Syst. Man Cybern. Part B* **2008**, *38*, 1561–1570. [CrossRef] [PubMed]
31. Bin, W.A.; Daud, W.; Faizura, W.; Adly, M.A.; Elamvazuthi, I.; Begam, M. Kinematic Modeling of Humanoid Arm. In Proceedings of the 2010 International Conference on Intelligent and Advanced Systems, Manila, Philippines, 15–17 June 2010; pp. 1–4.
32. Wang, W.J.; Huang, C.H.; Lai, I.H.; Chen, H.C. A Robot Arm for Pushing Elevator Buttons. In Proceedings of the SICE Annual Conference, Taipei, Taiwan, 18–21 August 2010; pp. 1844–1848.
33. Chang, W.C. Hybrid Fuzzy Control of an Eye-to-Hand Robotic Manipulator for Autonomous Assembly Tasks. In Proceedings of the SICE Annual Conference, Taipei, Taiwan, 18–21 August 2010; pp. 408–414.
34. Juang, J. Collision avoidance using potential fields. *Ind. Robot. Int. J.* **1998**, *25*, 408–415. [CrossRef]
35. Histogram of Oriented Gradients. Available online: https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients (accessed on 11 January 2016).

36. Support Vector Machine. Available online: http://en.wikipedia.org/wiki/Support_vector_machine (accessed on 10 November 2009).
37. Ben-Hur, A. Support vector clustering. *J. Mach. Learn. Res.* **2001**, *2*, 125–137. [CrossRef]
38. Hsu, W.-T.; Lu, C.-C.; Yao, L.; Zhong, S.; Kikuta, H.; Juang, J.-G.; Anpo, M. Application of the LM-HLP Neural Network to Automatic Smartphone Test System. *Adv. Mech. Sci. Technol. Ind. Revol.* **2017**, *4*, 217–225. [CrossRef]

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.