# Research on Functional Test of Mobile APP Based on Robot

Zhaolai Pan, Jianxin Chen, Ling Yao, and ZhuAn Chen
Key Lab of Broadband Wireless Communication and Sensor Network Technology
Nanjing University of Posts and Telecommunications, Ministry of Education
Nanjing, China
e-mail: chenjx@njupt.edu.cn, 1018010604@njupt.edu.cn

2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP) | 978-1-7281-6896-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICSIP49896.2020.9339300

***Abstract*—With the rapid popularization of smartphones, various applications (Application: APP) have come into being. Therefore, automated testing of mobile applications is particularly important. At present, the automated testing on the market is often completed through test frameworks. However, when using the test framework we need to consider the compatibility of the mobile platform system. What's worse, accessing multiple apps at the same time has a bad impact on accuracy. In view of these problems, a robot-based automated test system for mobile applications is proposed. Firstly, a wireless projection technology is utilized to project the screen of a mobile device onto a computer screen. Then, the optical character recognition (OCR) algorithm and the normalized correlation matching algorithm are used to position the coordinates of the elements. Finally, the robot performs a click operation under control. The experimental results show that the system can achieve cross-platform testing, including automatically testing Chinese software or English software on Android and iOS platforms, and also has high accuracy and robustness.**

***Keywords- mobile application; automated testing; optical character recognition algorithm; normalized correlation matching algorithm; robot.***

## I. INTRODUCTION

With the advent of the 5G era, more and more applications will emerge. The application needs to be automatically tested before it is put on the market. Among them, the functional automation test is an extremely important part of the test [1], [2]. Functional automation testing [3]-[7] is mainly to traverse application functions by writing test scripts, and execute them repeatedly. When an abnormality occurs in the application during the test, the abnormality is recorded, which is convenient for developers to maintain. At present, the functional automation test of the software mainly adopts the test framework. The Android system itself provides a test framework named Instrumentation for developers, as well as a secondary development interface. Robotium [7], [8] is developed based on this framework, but it is easier to use than the former. The Cafe automatic testing framework developed by Baidu is based on the Robotium testing framework, which creatively proposes cross-thread testing. Because Robotium and Cafe are based on the Instrumentation test framework, they can only be applied to application testing on the Android platform, not cross-platform testing. Of course, in addition to the automated testing framework developed by the company, there are many researchers who did research on functional automated testing. Arnaldo Marulitua Sinagaet et al. [9] applied the mainstream testing frameworks Appium, Robotium and UI Automator to four different Android programs for testing. Through the analysis of the test results, the advantages and disadvantages of each test tool are obtained, and recommendations for use are put forward according to the characteristics of each tool. Yangyang Zhu et al. [10] and others gave a detailed introduction to Robotium. Gaurang Shah et al. [11] mainly studied the Appium testing framework and pointed out that the main purpose of automated testing is to produce high-quality and complete software. Arnaldo Marulitua Sinaga, Zhu Yangyang, and Gaurang Shah's research on automated testing is based on the test framework. These studies have a common problem, that is, the mobile terminal needs to be connected to the computer through a data cable, and an APP related to the test framework must be installed. Ke Mao et al. [12] proposed a new functional automated test system that applied robots to automated testing and created the robot mobile device test generator AXIZ. Although Ke Mao avoided the need for automated testing frameworks for Arnaldo Marulitua Sinaga and Gaurang Shah, he only tested Google Calculator, which is still far from true functional automated testing. In response to the above problems, the advantages of our system are as follows:

1. We abandoned the automated test framework and adopted a robot-based-test method. The robot is combined with computer vision technology to perform automation test. This can simulate the real scene that users use.
2. We also abandoned the traditional camera as a tool to monitor the user interface, instead using advanced wireless projection technology. This saves costs.
3. Our system can be tested across mobile platforms. At the same time, the advanced OCR technology is adopted, which allows our system to accurately determine the position of the text element to be clicked from the complex background image.

The remaining papers mainly include the following sections: Section II mainly introduces the process of building the system. Section III mainly introduces the test results. Section IV made a conclusion.

## II. SYSTEM OVERVIEW

Wireless projection technology can project the screen of a mobile device onto a computer screen. The OCR algorithm can identify text elements in the interface, and the normalized correlation matching algorithm can identify icon elements in the interface. The robot can replace the manual operation of tapping the screen of the mobile device. This article combines algorithms with robots and proposes a user interface automated test system. The framework of the automated test system is shown in Fig. 1.

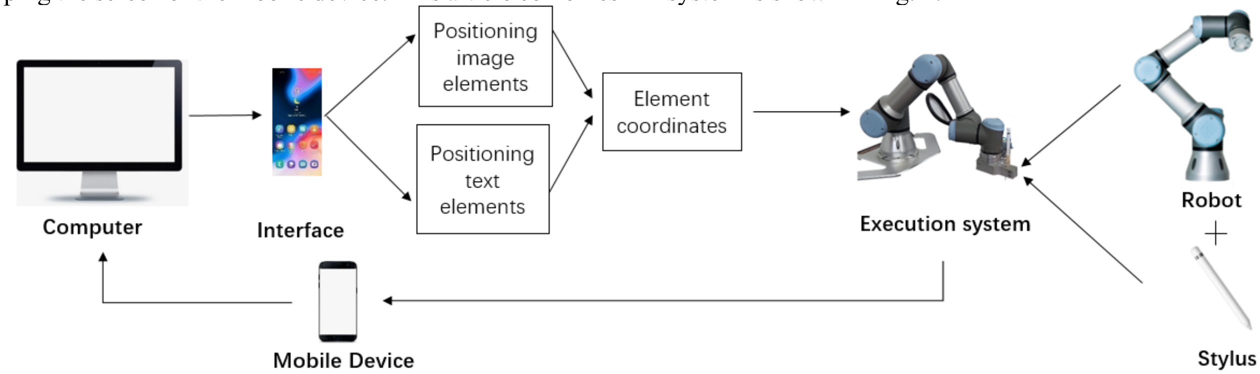



Figure 1. Automated test system framework.

### A. User Interface Element Detection

Detecting element information on the APP interface can be divided into the following three steps. Firstly, the screen of the mobile device onto the computer screen is projected. Then, the screen of the mobile device is detected from the computer screen. Finally, OCR and normalized correlation matching algorithm is used to locate the elements on the user interface.

(a) Wireless Projection

Generally speaking, in order to monitor the screen of a mobile device, a camera should be installed above it. However, the system proposed in this article abandoned the traditional camera. It is replaced by a more convenient wireless projection technology. When using Android devices to cast screens, the technology is based on the Mircast protocol. When using iOS devices to cast screens, the technology is based on the AirPlay protocol. The wireless projection display is shown in Fig. 2.

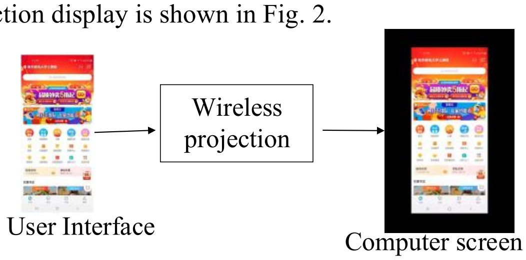



Figure 2. Wireless projection

(b) Mobile Device Interface Detected From Computer Screen

In order to accurately detect the screen of the mobile device, the computer screen needs to be processed. Firstly, Gaussian smoothing is performed on the computer screen obtained through automatic screenshots [12], so as to achieve the purpose of filtering noise. Secondly, perform edge detection [13], expansion and smoothing operations to eliminate any gaps between edges in the edge image. Finally, this paper uses the contour detection [14] algorithm to detect the entire mobile device interface. The detection process is shown in Fig. 3.

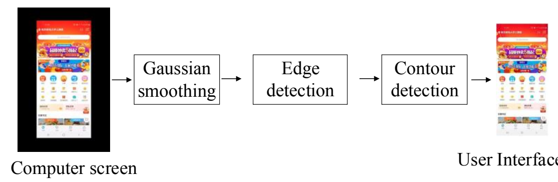



Figure 3. User interface detection

After processing a series of images, the mobile device interface in the image can be easily detected. Figure 4 shows the interface of the detected mobile device after a series of processing on the picture. Figure 4 (a) shows the unprocessed image. Figure 4 (b) shows the image processed by the algorithm. Figure 4 (c) shows the interface of the mobile device detected from the image. It is difficult to detect the mobile interface in the image from Figure 4 (a), but it is very simple from the processed Figure 4 (b).

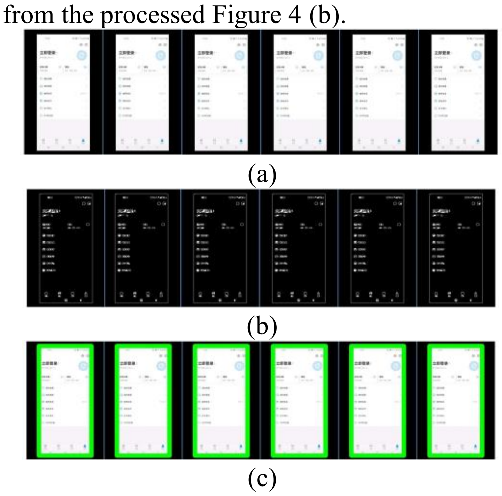



Figure 4. (a) The original image; (b) Processed image;(c) Image of detected interface.

(c) Locate The Elements of The Application User Interface

Through the above operations, this article has successfully obtained the user interface. Next, we will use algorithms to locate the elements on the interface. The positioning method proposed in this paper is based on computer vision. The user interface mainly includes text elements and image elements.

The text element uses OCR to locate and return coordinates. The image elements are located using a normalized correlation matching algorithm and the coordinates are returned. The block diagram of the element positioning process is shown in Figure 5.

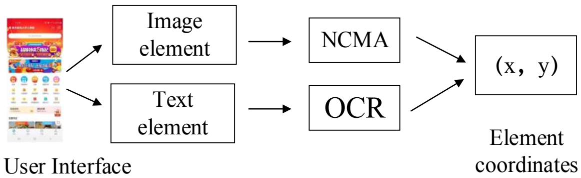



Figure 5. Block diagram of element positioning process

OCR [15], [16] is an algorithm for recognizing characters in pictures. Therefore, for the positioning of text elements, OCR will be used. The recognition process of this algorithm includes two parts: image processing and text recognition. Image processing facilitates feature extraction in the process of text recognition by preprocessing the original image. The algorithm steps are as follows:

- Perform grayscale processing on the original image $Pic_O$, and process the original color image into a grayscale image $Pic_g$.
- Binarize the grayscale image $Pic_g$ to obtain the binary image $Pic_B$.
- Use filtering algorithm to filter the binary image $Pic_B$, remove the noise in the image, and get the image $Pic_p$.
- Use the Hough transform algorithm to perform tilt correction on the image $Pic_p$, correct the image to the horizontal position, and get the image $Pic_{lv}$.
- Segment the text in the image $Pic_{lv}$ to get the line where the text is located, and then segment the characters to get the position of each character.
- Extract the features of each character, classify these features through a classifier, and finally recognize the text in the image.

Due to the complexity of the user interface picture background, it brings great difficulties to text recognition. In order to ensure the accuracy of text element recognition, this article will use the OCR for text recognition.

This paper uses the normalized correlation matching algorithm[17] to locate the icon of the image element as the template image $T$ .Then, the screen of the mobile device monitored in real time is used as the image $I$ to be matched. Calculate the normalized correlation coefficient $\rho(x,y)$ between the template image and the image to be matched:

$$\rho(x,y)=\frac{\sum_{x',y'}(T(x',y')*I(x+x',y+y'))}{\sqrt{\sum_{x',y'}T(x',y')^2*\sum_{x',y'}I(x+x',y+y')^2}} \quad (1)$$

Among them, $(x,y)$ represents the coordinates of the upper left corner of image $T$ imag $I$ , $\sum_{x',y'}T(x',y')^2$ represents the energy of image $T$ Value, $\sum_{x',y'}I(x+x',y+y')^2$ represents the energy value of image $I$ .By analyzing $\rho(x,y)$, a matching image $R$ with the same size as the image $T$ and the largest correlation can be obtained. Since the image to be matched is the screen of the mobile device monitored in real time, the image $R$ is not necessarily the last desired icon element. Therefore, this paper uses perceptual hash algorithm[18] to judge the similarity of image $T$ and image $R$. Use the perceptual hash algorithm to obtain the hash value $hash$ of the image $T$ and the image $R$ and calculate the Hamming distance $H_d$ between the two. When $H_d$ is less than the threshold 3, the image $R$ is the desired icon element, an d the matching is successful. Otherwise, the match fails.

## B. Execution System

The coordinates of the elements can be obtained from the coordinates above, and then sent to the robot to perform the click operation. It needs to be initialized before clicking. The purpose of initialization is to obtain the position information of the mobile device interface on the computer screen and the pixel size of the interface.

(a) Coordinate System Conversion

Before performing the click operation, this article needs to map the coordinate system of the robot to the coordinate system of the mobile device interface. Formula 2 can be used to calculate the ratio $rate$ between the pixel distance of the mobile device interface and the physical distance.

$$rate=\frac{Len_r}{Len_p} \quad (2)$$

Among them, $Len_r$ represents the actual physical length of the mobile device screen, and $Len_p$ represents the pixel length of the detected mobile device interface. Then select the upper left corner of the mobile device as the common origin of the robot coordinate system and the mobile device coordinate system.

(b) Click Action

After mapping the robot coordinate system to the coordinate system of the mobile device interface, formula 3 can calculate the coordinate point $(x,y)$ to which the robot will move.

$$(x,y)=(x_p\times rate, y_p\times rate) \quad (3)$$

Among them, $x_p$represents the pixel abscissa of the element in the interface, and $y_p$represents the ordinate of the element in the interface. Then, the robot moves to the coordinates $(x,y)$to perform a click operation.

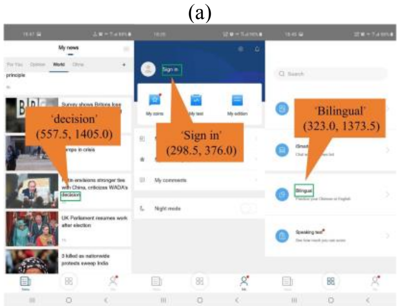
# III. Experimental Analysis

The control terminal of this system uses a computer with a processor of 3.50GHzIntel®Core™i7-3770K and an operating system of Windows 10 64-bit. This computer is used to monitor the mobile device interface and execute test scripts. The UR3 six-axis robot and Nissan's active capacitive pen are selected as the execution end. Choose Samsung's Galaxy A8s and Apple's iPhone 8Plus as mobile devices for testing. This article first tested the recognition accuracy of the API provided by Baidu. Then, this paper tested the Chinese application Eleme and the English software China Daily on the Android platform and iOS platform.

## A. *OCR Recognition Results*

Figure 6 shows the result of using the API to identify text elements. This article selects six different interfaces for recognition, and the returned result is the recognized text element and the corresponding pixel coordinates. The origin of the coordinate system is at the upper left corner of the interface, the x-axis to the right of the origin, and the y-axis below.


(a)

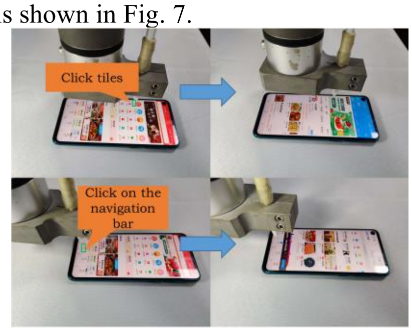



(b)

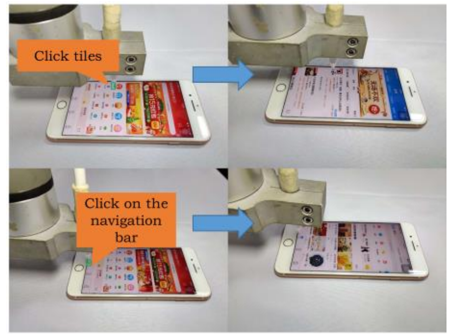
Figure 6. OCR recognition results:(a)Eleme;(b)China Daily.

## B. *Eleme Test Results*

Eleme is a Chinese food ordering application. The system tested the homepage of Eleme on Android mobile devices and iOS mobile devices.When the homepage function is tested, the system can successfully complete the test process. The test result is shown in Fig. 7.




(a)

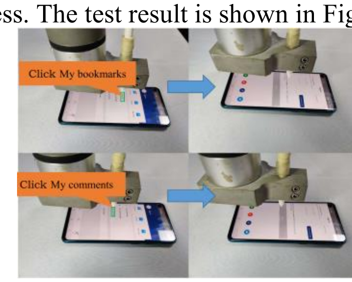



(b)

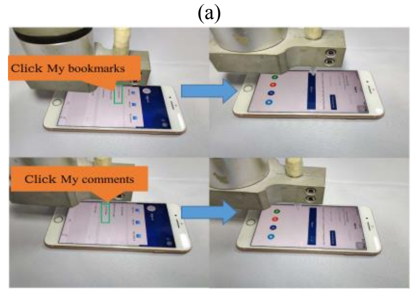
Figure 7. Eleme homepage test results: (a) Android device; (b) iOS device.

## C. *China Daily Test Results*

China Daily is an English application. This system also carried out automated tests on China Daily on Android mobile devices and iOS mobile devices. When testing China Daily's user center, the system can successfully complete the testing process. The test result is shown in Figure 8.




(a)




(b)

Figure 8. China Daily User Center Test Results: (a) Android device; (b) iOS device.

# IV. CONCLUSION

This paper proposes a new automated testing method for mobile applications, which is based on the technique of robot control and imaging processing. In monitoring mobile devices, wireless projection technology is used to replace traditional cameras, which not only improves the recognition rate, but also saves costs. When locating elements, this article uses OCR algorithm and normalized correlation matching algorithm. In terms of performing operations, a high-precision UR3 robotic arm equipped with a capacitive pen is used for click operations. In order to verify the system performance, this article uses Eleme and China Daily as the test software, and the related test scripts is written. Experimental results show that the proposed system can test both Chinese application software and English application software. During the testing process, when the software does not show the expected interface, the system will output an error log and save a screenshot of the error interface, which is convenient for developers to maintain the software. However, the test script needs to be maintained when the APP is updated, which increases the test cost. Therefore, how to reduce maintenance costs is the next research direction of this article.

## ACKNOWLEDGMENT

This work was supported by the funding of National Natural Science Foundation of China (Grant No. 61901227), Nanjing University of Posts and Telecommunications (NY217021, NY218014).

## REFERENCES

[1] Arnatovich Y L, Wang L, Ngo N M, et al. " An automated approach to exercising mobile application GUIs using symbiosis of online testing technique and customated input generation," Practice and Experience, 2018, pp.1107-1142.

[2] Alotaibi A A, Qureshi R J. "Novel framework for automation testing of mobile applications using Appium," International Journal of Modern Education and Computer Science, 2017, pp.34.

[3] Fu H D, Wang Z Z. "Research on Automatic Testing System of Vehicles Steering Angle," Advanced Materials Research, 2013,pp.1827-1830.

[4] Si-Jia L I, Xi W, Xiao-Zhu G. "UI automation testing framework based on Appium," Science & Technology Vision, 2018(6) , pp.141-142.

[5] Hao S, Liu B, Nath S, et al. " programmable UI-automation for large-scale dynamic analysis of mobile apps," Proceedings of the 12th annual international conference on Mobile systems, applications, and services. ACM, 2014,pp.204-217.

[6] Ki T, Simeonov A, Park C M, et al, " Fully automated ui testing system for large-scale android apps using multiple devices," Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services. ACM, 2017, pp. 185-185.

[7] Coppola R, Raffero E, Torchiano M. "Automated mobile UI test fragility: an exploratory assessment study on Android," Proceedings of the 2nd International Workshop on User Interface Test Automation. ACM, 2016, pp.11-20.

[8] Zadgaonkar. "Robotium automated testing for Android," Packt Publishing, 2013.

[9] Sinaga A M, Wibowo A, Silalahi A, et al. "Performance of Automation Testing Tools for Android Applications," 2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE). IEEE, 2018, pp. 534-539.

[10] Yang-Yang Z, Yong-Hong H, Bao-Liang W. "Application of automatic test tool Robotium for Android," Information Technology, 2015, pp.198-200.

[11] Shah G, Shah P, Muchhala R. "Software testing automation using appium," International Journal of Current Engineering and Technology, 2014, pp. 3528-3531.

[12] Mao K, Harman M, Jia Y. "Robotic testing of mobile apps for truly black-box automation," IEEE Software, 2017, pp. 11-16.

[13] Lyu P, Liao M, Yao C, et al. "An end-to-end trainable neural network for spotting text with arbitrary shapes," Proceedings of the European Conference on Computer Vision (ECCV). 2018, pp. 67-83.

[14] Hamad K, Kaya M. A "Detailed Analysis of Optical Character Recognition Technology," 2016.

[15] James Conder. "Gaussian smoothing filter," Gaussian Filter.

[16] Wei-Bo W. "Study on Edge Detection Method," Computer Engineering and Applications, 2006.

[17] Arbelaez P, Maire M, Fowlkes C, et al. "Contour detection and hierarchical image segmentation," IEEE transactions on pattern analysis and machine intelligence, 2010, pp.898-916.

[18] Zhen-kun W, Wei-Zong Z, Peng-fei L, et al. "A robust and discriminative image perceptual hash algorithm," 2010 Fourth International Conference on Genetic and Evolutionary Computing. IEEE, 2010, pp. 709-712.