High speed script execution for GUI Automation using Computer Vision

M Narayana[1], Raghu Ram Reddy N[2], Hyndavi Reddy N[3]
[1,3]Department of Electronics and Communication Engineering, Vardhaman College of Engineering, India
[2]Department of Electronics and Communication, Jayaprakash Narayan College of Engineering, India

Article Info

Article history:

Received Apr 11, 2018
Revised Aug 17, 2018
Accepted Sep 2, 2018

Keywords:

Automation
Computer vision
GUI

ABSTRACT

Software testing by using open source tool like Selenium windows applications cannot be automated, citrix based applications, flash websites and games. Computer vision based automation tools can be used to automate these kinds of applications. These automation tools works based on screenshots of GUI objects like button, radio button, text box, images, dropdowns etc. In this paper a prototype of automation tool has been developed which can execute the automation scripts much faster than existing tools like Sikuli, which takes much time to run the Automation scripts. The execution time can be reduced by using this proposed tool.

Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Hyndavi Reddy N,
Department of Computer Science Engineering,
Jayaprakesh Narayan College of Engineering, India.
Email: raghutripler@gmail.com

1. INTRODUCTION

Software is everyware in this business world. Software testing is the process of validating and verifying that a software program or product meets the business, functional, technical and user requirement that guides the design and development. It verifies whether an application or program works as expected. It is a process that takes place throughout the software development life cycle. It is very important to ensure the quality of software.

Automated software testing [1] is essential because manual testing of all the test cases, all the UI elements, all the positive and negative scenarios is time and cost consuming. It is difficult to test a scenario for a number of users and with huge test data. It does not requires human intervention and can be executed unattended, which increases speed of execution and helps to increase the test coverage. Automated software testing can be used to do various kinds of testing like Smoke testing, Regression testing which reduces lot of manual efforts.

The popular tools in GUI automation [2] testing are Selenium [3], QTP [4], Test complete, CodedUI, Ranorex, Telerik Test studio etc. Table 1 shows that these tools can be used to automate both windows and Web applications [5]. Selenium is the most popular open source automation tool used to automate web application. This tool cannot be used to automate flash websites where the object Id's not exposed for identifying the objects. Websites with video players, Music players, Games are some examples of flash websites. To overcome this limitation, Sikuli [6] can be used to automate the flash websites and Games.

Sikuli works based on Template matching [7] using screen shots [8] of GUI objects [9]. But the limitation of Sikuli is it takes much time for object recognition and fails to identify the object like button, labels, dropdown etc. Because of this issue there are more chances of Test failure. As a result of this test execution takes more time than expected. This also increases the time for Test Automation engineers to develop automated test cases [10].

Table 1. Comparison of Various Automation Tools and Usage

Automation Tool	Windows Application	Web Application	Flash objects Automation
Selenium	No	Yes	No
QTP	Yes	Yes	No
Test Complete	Yes	Yes	No
CodedUI	Yes	Yes	No
Ranorex	Yes	Yes	No
Sikuli	Yes	Yes	Yes

2. RELATED WORKS

For software testing there are many types of tools like Selenium, QTP, Test Complete etc. These kinds of tools can be used for Automating Web and Windows Applications. These tools are based on the object properties like Id, class name, Tag name, text etc. Computer vision based automation tools like Sikuli uses images of various GUI objects. Sikuli can be used to perform various actions like click, double click, entering the text in text box, drag and drop etc.

Sikuli is used to automate the test cases using the screen shots of GUI objects such as button, links, radio buttons, dropdowns etc. by using image recognition technique. It interacts with the GUI by finding the position of object in GUI using template. Sikuli automates the GUI interactions [11-12] of keyboard and mouse events using Visual patterns. The input to the Sikuli is the screenshot of various web element or windows element. Sikuli will do various actions such as click, double click, drag and drop etc. by finding the location of the elements in the user interface. The screenshots of the various elements are stored in the project folder and passed as parameter for doing various GUI interactions.

Even Window based applications can also be automated using Sikuli. It provides very friendly Sikuli-script.jar, which can be easily used together with Selenium Web Driver. The screenshots of the various elements are stored in the project folder and passed as parameter for doing various GUI interactions as shown in Figure 1. We can even automate Adobe Video/Audio player, Flash Games on the websites using Sikuli [1].

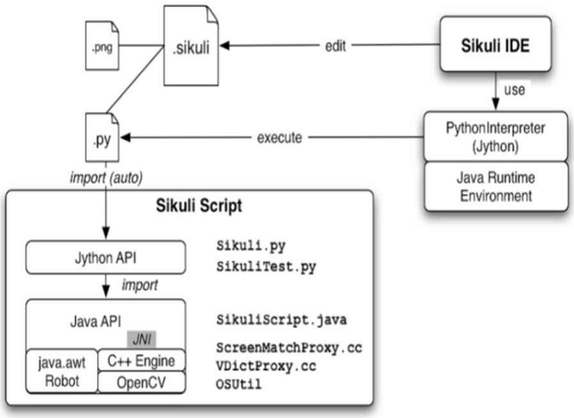


Figure 1. Sikuli block diagram

Selenium is the most popular open source automation tool which is widely used in Automation testing [13]. Selenium Web driver identifies the objects using xpath [14], css selectors, using id, name, class name etc. When an object property is changed then the test fails as Selenium failed to identify the object. In this case even the object exist the test failed. To avoid this kind of situations computer vision-based tools can be very helpful to overcome this kind of problems. We can automate what we are seeing on the screen. It provides, simple API i.e. all the test cases are automated using the screenshots of the objects. Sikuli automates the GUI interactions of keyboard and mouse events using Visual patterns. The input to the Sikuli is the screenshot of various web element or windows element.

3. PROPOSED SOLUTION

In this paper a prototype of computer vision based automation tool has been developed using C++ and OpenCV. In this there are two modes if you want to execute the script faster screenshot is converted to gray scale which reduces the calculations, Template matching algorithm of computer vision is used to detect the objects like button drop down radio button checkboxes and images etc. The proposed framework can successfully handle various kinds of GUI objects based on screenshots as shown in Figure 2. The proposed prototype can be used for automating windows applications, WPF Applications, Games and Citrix based Applications. By executing several tests, 25 test cases are executed using both Sikuli and the proposed tool in the similar environment the proposed tool has taken only 29 minutes whereas Sikuli has taken around 35 minutes.

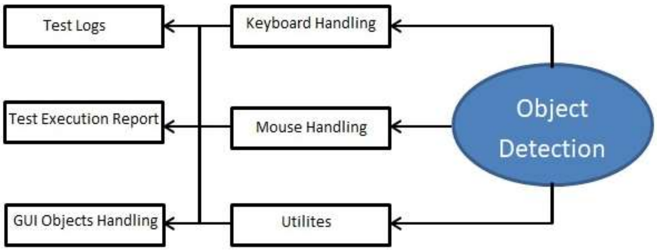


Figure 2. Proposed block diagram

In computer vision Template matching is a technique for finding the area in the image which matches with the template image. Here we need two images. First image is the Source image (I) and second image is the Template Image (T). Source image is used to find the matching area with the template image [15]. Template image is the input image for comparing with the Source image. The main objective is to find the highest matching area of template image (T) in the Source image. To identify the matching area the template image is compared with source image by sliding the image patch. Here patch is moved one pixel at a time from left to right and up to down. At each location a metric is calculated for T over I and the metric is stored in the result matrix R [16]. Normalized cross correlation [17] is used for finding the Matrix R. The exact match of the template is identified using Matrix R. Chrome icon image is the input image. When the template image is compared with source image using R Matrix, the object is identified and highlighted with green color. The highest matched area is identified, and the midpoint of the patch is calculated in the source image. By using this point various actions are performed used keyboard and mouse for various GUI Interactions [18].

$$R(x,y) = \frac{\sum_{x',y'}(T'(x',y').I'(x+x',y+y'))}{\sqrt{\sum_{x',y'}T'(x',y')^2.\sum_{x',y'}I'(x+x',y+y')^2}}$$

$$T'(x',y') = T(x',y') - 1/(w.h).\sum_{x'',y''}T(x'',y'')$$

$$I'(x+x',y+y') = I(x+x',y+y') - 1/(w.h).\sum_{x'',y''}I(x+x'',y+y'')$$

3.1. Handling keyboard

Keyboard is handled by simulating keystrokes using virtual key codes of various buttons in the keyboard like ctrl(VK_CONTROL), Enter(VK_RETURN), SHIFT key(VK_SHIFT), SPACEBAR(VK_SPACE), Numeric keypad 0 key(VK_NUMPAD), ALT key(VK_MENU), etc. By using this various actions are performed like entering the text in the text box, clearing the values in the Text box, for generating Keystroke of Enter button etc. Various Utilities has been created to perform various actions such as copy paste, cut, Navigating forward and backward in the browser refreshing the current page etc.

3.2. Handling mouse

Basic function of mouse such as left click and right click are generated once after locating the object using template matching algorithm [8] based on the inputs from the user. Midpoint of the template is identified. Various actions such as drag and drop, Left click, Right click, double click etc. can be performed for automating an application. User can also click on the desired location of the template as shown in Figure 3. Template is divided into various quadrants Q1, Q2, Q3, and Q4 as shown in Figure 4. For example, if user wants to click on the first quadrant Q1 midpoint, Q1 has to be passed as a parameter in Click function.

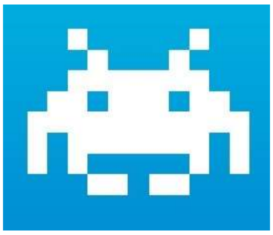


Figure 3. Template image

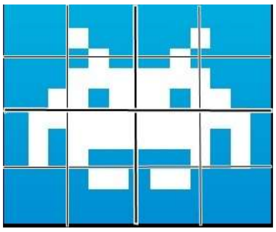
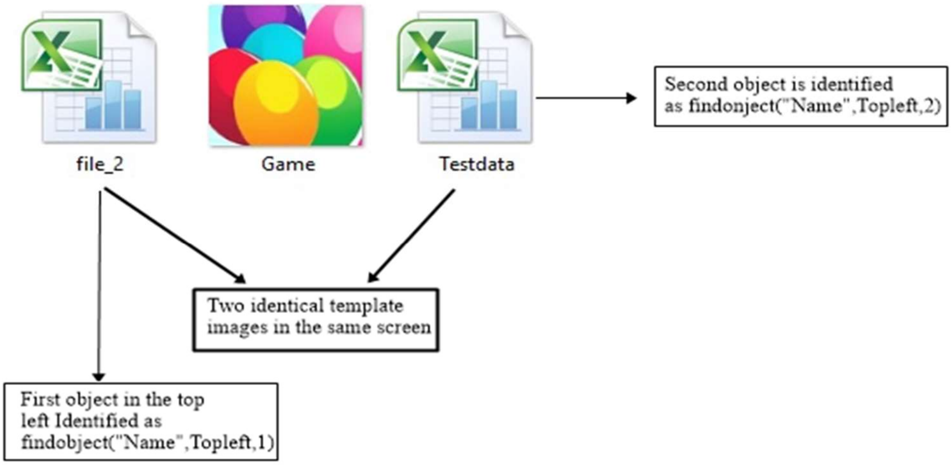


Figure 4. Template quadrants

3.3. Handling multiple objects

For few test scenarios there is a chance where there are multiple identical template objects in the same screen as shown in Figure 5. For this case the desired object can be identified based on the position of the image. Top, bottom, left, right parameters can be used to identify the objects based on the position in the source image. If the above conditions does not satisfy, object can be identified based on the position of the images from left to right. If there are two objects side by side in the top left of the screen then the object is identified based on the count of the objects. The two images are identified as top left first object, top left second object respectively.



Second object is identified as findonject("Name",Topleft,2)

Two identical template images in the same screen

First object in the top left Identified as findobject("Name",Topleft,1)

Figure 5. An example of handling identical multiple objects in the same screen

4. RESULTS AND DISCUSSIONS

For execution of automated test cases, the proposed model is compared with popular open source tool Sikuli. For Execution of each test case time has been calculated. The test has been conducted on both windows and web application. In Table 2 test cases 1-4 are related to Windows application and Test case 5-7 are related to web application. For Sikuli the total time of execution of automated scripts is 425.09 seconds and for proposed model it has taken 406.04 seconds. Based on the below results the proposed model is much faster than the existing open source Sikuli. By using the proposed model test script development time also can be minimized.

Table 2. Sikuli and Proposed Model Test Execution Speed Comparison

Test Case number	Sikuli	Proposed model
Test case 1	4.54 sec	4.31 sec
Test case 2	50.33 sec	47.89 sec
Test case 3	22.58 sec	21.23 sec
Test case 4	90.73 sec	86.26 sec
Test case 5	73.13 sec	70.46 sec
Test case 6	96.26 sec	92.77 sec
Test case 7	87.52 sec	83.12 sec
Total time in sec	425.09 sec	406.04 sec

5. CONCLUSION AND FUTURE SCOPE

In this paper a novel Automation framework has been developed using computer vision, which can automate windows applications, web applications, flash websites and Citrix based applications at high speed compared to the other open source tools like Sikuli. The object detection speed on GUI can be increased by using the proposed model. The Test execution time for running the automated test cases can be reduced which increases accuracy and reduces Time out issue.

There is a need in test automation for the development of a framework which is independent of object properties like Id's, xpath and screenshot of objects. Objects should be identified based on the text of the objects and its visual features. Future work focuses on identifying the GUI objects using text based on various Machine learning and deep learning algorithms which helps to reduce the time to automate the application. By using machine learning concepts objects can be identified by training the screenshots with the classifiers.

REFERENCES

[1] D. Rajya Lakshmi, S. Suguna Mallika, "A Review on Web Application Testing and its Current Research Directions, " *International Journal of Electrical and Computer Engineering*, vol. 7, pp. 2132-2141, August 2017

[2] Leshed, G., E. M. Haber, T. Matthews, and T. Lau, "CoScripter: Automating & Sharing How-to Knowledge in the Enterprise, " *Conference on Human Factors in Computing Systems*, p. 1719-1728, 2008.

[3] S. Inderjeet, and T. Bindia, "Comparative Analysis of Open Source Automated Software Testing Tools: Selenium, Sikuli and Watir," *International Journal of Information & Computation Technology*, vol. 4, pp. 1507-1518, 2014.

[4] Monika Sharma, Rigzin Angmo, Web based Automation Testing and Tools, *International Journal of Computer Science and Information Technologies*, vol. 5, pp. 908-912, 2014.

[5] Bolin, M., M. Webber, P. Rha, T. Wilson, and R. C. Miller, "Automation and Customization of Rendered Web Pages," *Symposium on User Interface Software Technology*, pp. 163-172, 2014.

[6] C. Tsung-Hsiang, Y. Tom, C. M. Robert, "GUI Testing Using Computer Vision," *CHI 10th Conference on Human Factors in Computing Systems*, pp. 1535-1544, 2010.

[7] J. Hyunjun, L. Sukhoon, B. Doo-Kwon, "An Image Comparing-based GUI Software Testing Automation System," *World Congress in Computer Science, Computer Engineering, and Applied Computing*, 2012.

[8] Y. Tom, C. Tsung-Hsiang, C. M. Robert, "Sikuli: Using GUI Screenshots for Search and Automation," *The 22nd annual ACM symposium on User interface software and technology*, pp. 183-192, 2009.

[9] S. L. M. Jeffrey, "User Interface Computation as a Contextualized Approach for Introductory Computing Instruction," *The 9th Annual International ACM Conference on International Computing Education Research*, pp. 179-180, 2013.

[10] K. Dea-Kwang, and L. Lee-Sub, "Reverse Engineering from Exploratory Testing to Specification-based Testing," *International Journal of Software Engineering and Its Applications*, vol. 8, pp. 197-208, 2014.

[11] M. Gharavi-Alkhansari, "A Fast Globally Optimal Algorithm for Template Matching Using Low-resolution Pruning," *IEEE Trans. Image Process*., vol. 10, pp. 526-533, Apr. 2001.

[12] L. Maurizio, S. Andrea, R. Filippo, and T. Paolo, "Automated Generation of Visual Web Tests from DOM-based Web Tests," *ACM/SIGAPP Symposium on Applied Computing*, April, 2015.

[13] V. Andriychenko, L. Ying-dar, C. T. National, "Automatic Functionality and Stability Testing Through GUI of Handheld Devices," *CiteSeerx*, 2011.

[14] Nisha Gogna, "Study of Browser Based Automated Test Tools WATIR and Selenium," *International Journal of Information and Education Technology*, Vol. 4, pp. 336-339, August 2014.

[15] Rajithkumar B K, H.S. Mohana, "Template Matching Method for Recognition of Stone Inscripted Kannada Characters of Different Time Frames Based on Correlation Analysis," *International Journal of Electrical and Computer Engineering* ,Vol. 4, pp. 719-729, October 2014.

[16] Qiao Zhang, Huijie Gao, Zhen Kang, "Fast Template Matching with Partial Skipping Using Sub-template," *Audio Language and Image Processing (ICALIP) 2012 International Conference*, 2012, pp. 889-892.

[17] Wanli Ouyang, Federico Tombari, Stefano Mattoccia, Luigi Di Stefano, W.K. Cham, "Performance Evaluation of Full Search Equivalent Pattern Matching Algorithms," *Pattern Analysis and Machine Intelligence IEEE Transactions*, vol. 34, pp. 127-143, 2012.

[18] C. Tsung-Hsiang, "Using Graphical Representation of User Interfaces as Visual References," *The 24th annual ACM symposium adjunct on User interface software and technology*, pp. 27-30, 2011.

BIOGRAPHIES OF AUTHORS

Dr. M. Narayana is a professor in Vardhaman college of Engineering, Hyderabad, Telangana. He received B.Tech. from G. Pullareddy College of Engineering, Kurnool, SKU, Anantapur and M.Tech. from JNTUH, Hyderabad, AP, India and received Ph.D from JNTUA, Anantapur, AP. He has seventeen years of experience in teaching undergraduate and post graduate students, and guided more than 43 undergraduate and more than 34 postgraduate thesis. He has published 32 papers in International Journals, 6 papers in International Conferences and 1 paper in National Conferences. He has presented research papers in international and national conferences. His research interests are in the areas of signal and image processing, segmentation, pattern recognition, content-based image retrieval, Biometrics and Biomedical Engineering.

Raghu Ram Reddy N is currently working as a Software Developer in a reputed product based company in Bangalore. He received B. Tech and M. Tech from Jaya Prakash Narayan college of Engineering, Mahabubnagar, Telangana. He has published 2 papers in International Journals. His research interests are in the areas of Computer Vision, Segmentation, Machine Learning, Pattern recognition and Neural Networks.

Hyndavi Reddy N is currently working as a Software Developer in a reputed service based company in Bangalore. She received B.Tech from Jaya Prakash Narayan college of Engineering, Mahabubnagar, Telangana. Her research interests are in the areas of Artificial Intelligence, Computer Vision, Segmentation, Machine Learning, Pattern recognition and Neural Networks.