

UNDERGRADUATE PROJECT REPORT

Project Title:	Automated Testing and Analysis of Artificial Intelligence Applications
Surname:	DUNZHU
First Name:	CIREN
Student Number:	201918020211
Supervisor Name:	Joojo Walker
Module Code:	CHC 6096
Module Name:	Project
Date Submitted:	May 5,2023

Chengdu University of Technology Oxford Brookes College

Chengdu University of Technology

BSc (Single Honours) Degree Project

Programme Name: Bsc Hons Software Engineer

Module No: CHC 6096

Surname: Dunzhu

First Name: CiRen

Project Title: Automated Testing and Analysis of Artificial Intelligence Applications

Student No.:201918020211

Supervisor: Joojo Walker

2ND Supervisor (if applicable) Not Applicable

Date submitted:5/5/2023

A report submitted as part of the requirements for the degree of BSc (Hons) in Software Engineering

At

Chengdu University of Technology Oxford Brookes College

Declaration

Student Conduct Regulations:

Please ensure you are familiar with the regulations in relation to Academic Integrity. The University takes this issue very seriously and students have been expelled or had their degrees withheld for cheating in assessment. It is important that students having difficulties with their work should seek help from their tutors rather than be tempted to use unfair means to gain marks. Students should not risk losing their degree and undermining all the work they have done towards it. You are expected to have familiarised yourself with these regulations.

<https://www.brookes.ac.uk/regulations/current/appeals-complaints-and-conduct/c1-1/>

Guidance on the correct use of references can be found on www.brookes.ac.uk/services/library, and also in a handout in the Library.

The full regulations may be accessed online at <https://www.brookes.ac.uk/students/sirt/student-conduct/>

If you do not understand what any of these terms mean, you should ask your Project Supervisor to clarify them for you.

I declare that I have read and understood Regulations C1.1.4 of the Regulations governing Academic Misconduct, and that the work I submit is fully in accordance with them.



Signature

Date :5/5/2023

REGULATIONS GOVERNING THE DEPOSIT AND USE OF OXFORD BROOKES UNIVERSITY MODULAR PROGRAMME PROJECTS AND DISSERTATIONS

Copies of projects/dissertations, submitted in fulfilment of Modular Programme requirements and achieving marks of 60% or above, shall normally be kept by the Oxford Brookes University Library.

I agree that this dissertation may be available for reading and photocopying in accordance with the Regulations governing the use of the Oxford Brookes University Library.



Signature

Date :5/5/2023

Acknowledgment

First of all, I would like to thank my parents from the bottom of my heart for the endless love and care they have shown me. No matter what I needed or where I went, they were always there to support me without any demands in return. I am grateful to my lovely family, and family is a place I can always turn to.

I would also like to thank my supervisor, DR JOOJO Walker, for his guidance and assistance in completing my thesis. He is a very responsible teacher, and whenever I encountered professional academic problems, I was able to complete my thesis successfully through his guidance and feedback. He devoted himself with great care and energy to the writing of my dissertation, including his advice on the use of words and guidance on the structure of my dissertation. I would also like to thank my teacher Javier JIA, for always encouraging me when I was down and motivating me to finish my studies at university.

I would also like to thank my friends I met at university, Cole, Robin, Terry and Tony, who always kept me up to date with academic discussions, which encouraged the completion of my undergraduate studies.

I would also like to thank all those who helped me, cared for me and wished me well. The achievement of the thesis belongs to us and is a testament to our collaboration, our hard work, perseverance and eternal friendship.

Table of content

Declaration	3
Acknowledgment.....	4
Table of content.....	5
Abstract.....	7
Abbreviations	8
Glossary	9
Chapter 1 Introduction.....	10
1.1. Background	11
1.2. Aim.....	11
1.3. Objective:.....	11
1.4. Project Overview	12
1.4.1 Scope.....	12
1.4.2 Audience.....	12
Chapter 3 Methodology	18
3.1 Approach for Automated Testing.....	18
3.1.2 Approach for OpenCV.....	22
3.2 Technology	30
3.3 Project Version Management	30
Chapter 4 Result	31
Chapter 5 Professional Issues	44
5.1 Project management.....	44
5.1.1 Activities	44
5.1.2 Schedule	45
5.1.3 Project Data Management	46
5.1.4 Project Deliverable	46
5.2 Risk analysis	47
5.3 Professional Issues.....	50
5.3.1 Legal issues.....	50
5.3.2 Social issues	51
5.3.3 Ethical issues	51
5.3.4 Moral issues	52
5.3.5 Environmental issues	52

Chapter 6 Conclusion	54
6. Reference	56
6.3 Formatting Requirements	59
6.4 Written Presentation	59
Appendices	59

Abstract

This paper is an automated test-based artificial intelligence program, on the basis of which its advantages and disadvantages are summary. An OpenCV-based face-mask detection method is proposed. The AI program is first tested using Selenium IDE, converting the script into Junit for testing. In terms of image processing, image enhancement is performed using thresholding, grey scale histogram equalization, which improves the problem of poor face recognition due to the effect of exposure variation. Using median filtering to reduce the effect of noise, the results show that histogram equalization, with median filtering, improves the recognition of faces.

Keywords: Automated Testing, Face Mask Detection, OpenCV, Selenium IDE, Histogram Equalization, Median Filtering

Abbreviations

Abbreviation	Definition
AI	Artificial Intelligence
TDD	Test Driven Development
PE	Processing Element
DFT	Design For Testability
EML	Empirical Model Learning
MMU	Matrix Multiply Unit
HADA	Hard Ware Dimensioning of Algorithms,
IDE	Integrated Development Environment
CB	Color BG2Gray
GPU	Graphics processing unit
CPU	Central Processing Unit
HAAR	Har cascade detector
Junit	Java Unit
APS	Artificial Intelligence Planning to software testing
AJAX	Asynchronous javascript and xml
XPATH	XML Path Language
CSS	Cascading Style Sheets
IMG	Image
API	Application Programming Interface
KNN	K-Nearest Neighbor

Glossary

OpenCV - OpenCV is a computer database developed under the BSD license which can be freely used in various operating systems such as Windows, Linux and Mac. it is a highly scalable and lightweight library. OpenCV provides rich interfaces such as MATLAB and Python and has had a significant impact on image processing, facial recognition, tracking and measurement. It is widely used in real-time applications.

Selenium IDE -Selenium IDE is a tool for automated testing that allows you to input test data into the system under test, compare the expected results with the actual results and generate some detailed test reports. selenium is used to perform automated tests on web browsers.

Grayscale Histogram -Grayscale Histogram is a simple and effective image enhancement technique that alters the histogram of an image to change the greyscale of individual pixels in the image, with obvious benefits for blurred or overexposed images.

Image segmentation –Image segmentation is the process of segmenting an image to predict which category or object each pixel in the image belongs to, and deep learning and machine learning have used this technique to solve image recognition problems.

Cascade Classifier -Cascade classifier is divided into multiple weak and strong classifiers, and in a cascade classifier, for each input image, the cascade seeks through each classifier to filter the images

K-Nearest Neighbor- K nearest neighbor algorithm is Given a training dataset, for a new input instance, find the K nearest instances to the instance in the training dataset and classify the input instance into this class if the majority of these K instances belong to a certain class.

Chapter 1 Introduction

This report mainly summarizes the progress of the project. Chapter 1 briefly describes the background to the project and the overall objectives of the development project. Stating the mission of the project, an overview of the project and the target population of the development project. Chapter 3 describes all the progress that has been designed and achieved thus far, as well as the tools and resources for the development of the project. Chapter 4 is a summary of project management, describing the user's unfinished or completed goals, the use of a schedule to show completed and unfinished displays, and what tools to use to manage the user's code. Chapter 5 assesses professional issues and risks, analyzes risks to current progress, anticipates possible risks in the future, and discusses legal and ethical issues related to the project.

1.1. Background

Artificial intelligence is a method of researching and developing human intelligence. At present, this technology has achieved success in many fields such as computer vision, robot and automatic driving. Traditional testing, that is, manual testing, cannot meet the current market demand [6], and it usually leads to poor product quality, delayed release, unsatisfactory user experience, and even loss of revenue [10]. Turning to automation is an effective way to execute tests. It helps testers in their daily work, speeds up the test execution cycle, greatly reduces labor, and provides error free quick feedback. With the rapid development of complex modern software systems, there is an urgent need for some automated test systems [1].

Most AI applications run in the cloud or data center. As consumers generate large amounts of data, some large enterprises have to develop automatic test systems to improve data response to events [2]. The method of AI technicians is isolated [4]: models and algorithms are developed on a test bed isolated from the real world environment, and there is no larger system and background to jointly develop them. This is currently a challenge for AI automated testing.

The key stage of software development is system testing. Using this method in manual technology is expected to become a dynamic research field to solve or solve the challenges facing mankind today.

1.2. Aim

Carry out automated analysis and testing of state-of the artificial intelligent applications to ascertain its effectiveness.

1.3. Objective:

The objectives are as follows:

1. Test and analysis already existing Artificial intelligence applications.
2. Generate conclusions on effective methods for building Artificial intelligence applications.
3. Employ the comprehensive functions learnt from the already existing system to

build the Facemask detection system.

4. Carry out comprehensive testing and evaluation of the facemask detection system.

1.4. Project Overview

The main objective is to use selenium to test different AI applications, analysis and evaluate application capabilities.

1.4.1 Scope

This paper demonstrates the practicality of combining a novel approach with a test-driven approach in the context of testing artificial intelligence programs using the Selenium IDE. The novel approach is mainly applied to the integration testing of Junit. Based on the above approach, the advantages of the AI program are summarized and the above advantages are then summarized to develop a system for face mask recognition using OpenCV. OpenCV is widely recognized and used by the research community as it has many packages for processing images and has remarkable results in image processing. The scope of this paper is limited to the testing of the Selenium IDE and the development of a face mask recognition system using the OpenCV library.

1.4.2 Audience

The target audience for this paper is test engineers/for individuals who are also researching the same subject area. The implications of this research may be broader, for example in the context of web testing, and in particular the application of image recognition. The testing methods described in this paper may be directly applicable to those involved in software testing and network testing, and engineers who are able to apply this range may find value in being cost effective and efficient. With regard to thresholding and the processing of grey-scale histogram equalization in image processing. The aim of this is to better process image information and make it a more valuable processing method. When completed, the code will be

published for further development by others.

Chapter 2 Background Review

At present, there are various ways to test different AI programs. For instance, building a model or choose an existing architecture to test the application of AI [6]

Method 1: Use a new architecture of selective partial scanning to test AI accelerator applications [2]. In this architecture, the structural test mode is applied in a functional way, which reduces the testability problem of the array to the testability problem of a single processing element (PE). MMU (256 * 256 matrix cell) is used for each PE set, and the sum of each PE in the array is added at the end.

Finally, a new architecture model is synthesized (Figure1). Comparing the new architecture test with the traditional full scan architecture test (DFT), it is found that the test time and test power are more effective.

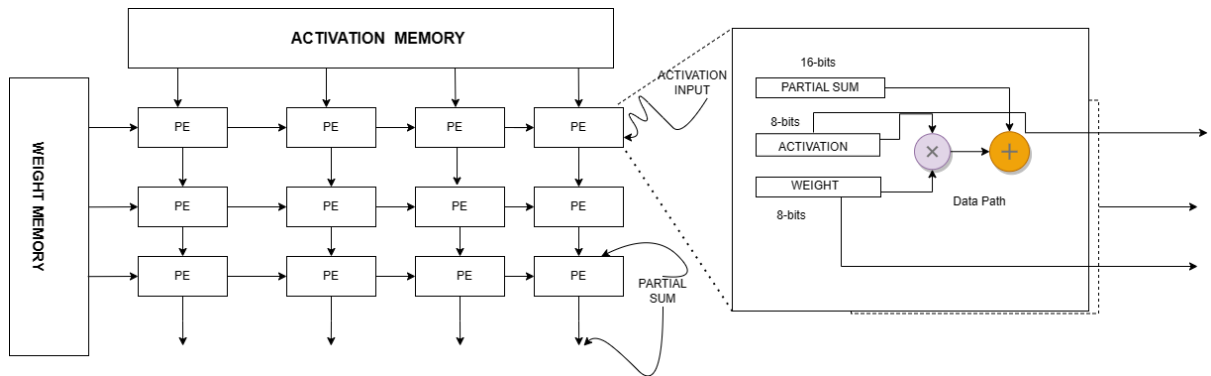


Figure1: New Architecture model of PE

Method 2: proposing a HADA which is an automatic tool for the size of the hardware algorithm [2], and using this tool to test external input.

The test tool is labeled as a black box (Figure2), accepting a set of AI algorithm features or some user-defined constraints (budget time and cost) as input, and the output result is the optimal hardware resource dimension required to run the algorithm.

Method 2 (2)

An additional model is proposed to construct the overall architecture of HADA in the EML framework [2] (Figure3), which is composed of data-driven components, domain knowledge components, and user specified objective functions and constraints. Data driven components and domain knowledge components form the

core.

Based on such a model, different ML regression models in the ensemble are used for encoding, and specific algorithms are used as input to the test set. The output result is the time when the algorithm finds the solution.

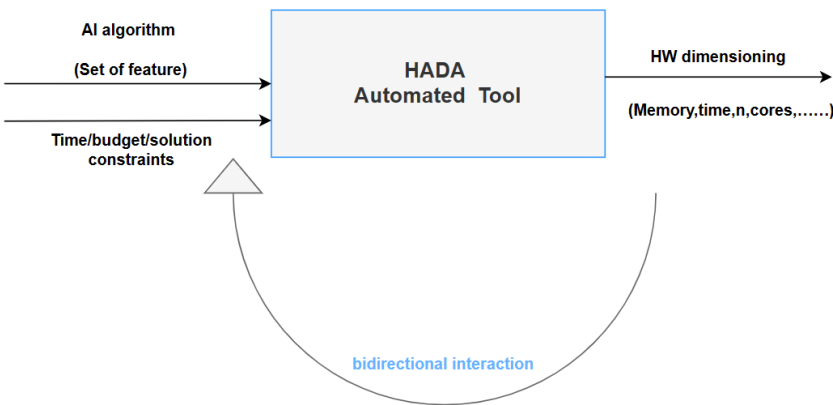


Figure 2: HADA Model

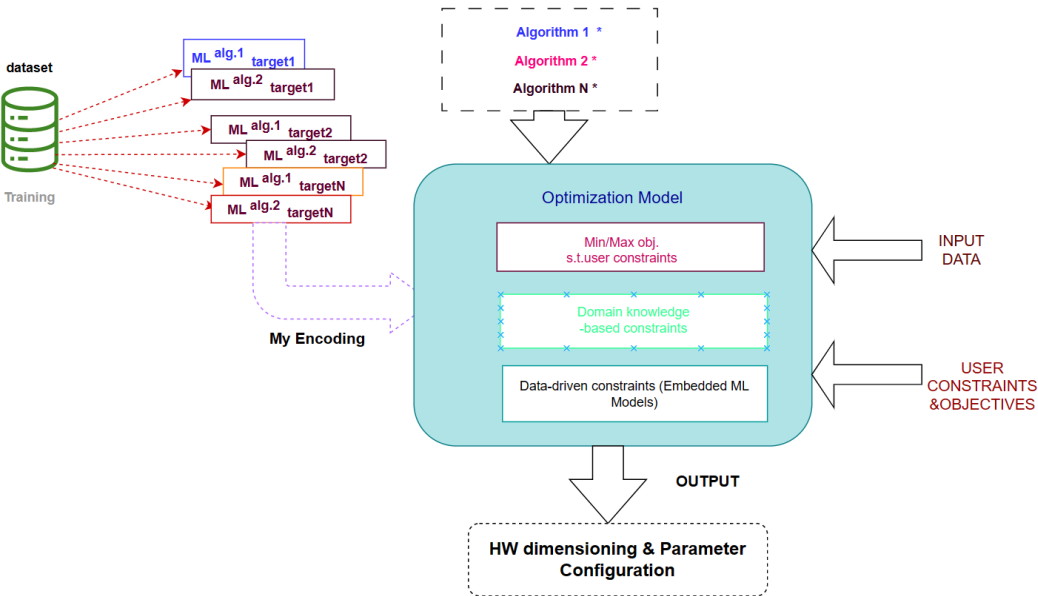


Figure3: Optimization Model

Method 3: A test automation framework that applies AI planning technology to

software system testing, namely APS framework is proposed [1]. It consists of a planning generator and an AI planner (Figure 4), the software parameters are defined as $\{S, A, T, C, f\}$, where O is the collection of operators, s_0 is the initial state, and g is the desired state for a particular programming problem.

The planning parameters $\{O, s_0, g\}$ are used as standard inputs to get a new sum through AI planner.

The test method is shown in Figure5, S represents the state set of the function, A = the operation using the function, such as create and append T represents the function of the test object, C represents that no new object is created or destroyed.

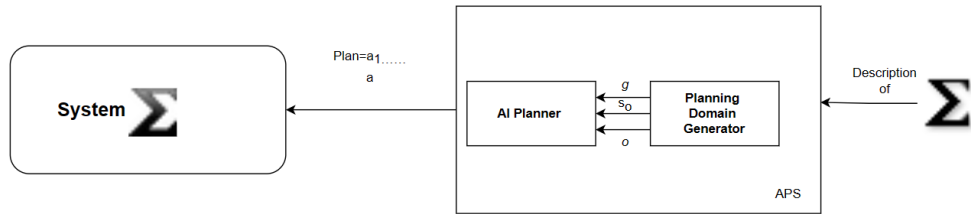


Figure 4: APS Framework

Finally, map the result to the output of Figure 4 through the test method in Figure 5.

```

V-function
  Length::List → natural
  ith::List x i: positive → element
  ! i [1<=i<=Length] → raise Bad Index
o-function
  create :: → List
  create().Length()=0
  append :: L::List x e::element x i:natural → List
  ! [0<=i<=L.length()] → raise Bad index
  L.append(i,e.Length()) = L.length()+1
  L.append(i,e).ith(j) = if j<=i → L.ith(j)
  j=i+1 → L.ith(j-1)
  end if
  remove :: L::List x i:natural → List
  ! [1<=i<=L.length()] → raise Bad index
  L.remove(i).length()=L.length()-1
  L.remove(i).ith(j) = if j<i → L.ith(j)
  j>=i → L.ith(j+1)
  end if
  delete ::List

```


Figure 5: APS Test Method

According to Figure 6, according to the comparison of the above methods, it is found that the above three methods need to be tested by establishing a model or selecting a pattern of part of the architecture. In summary, the method of selecting selenium is better.

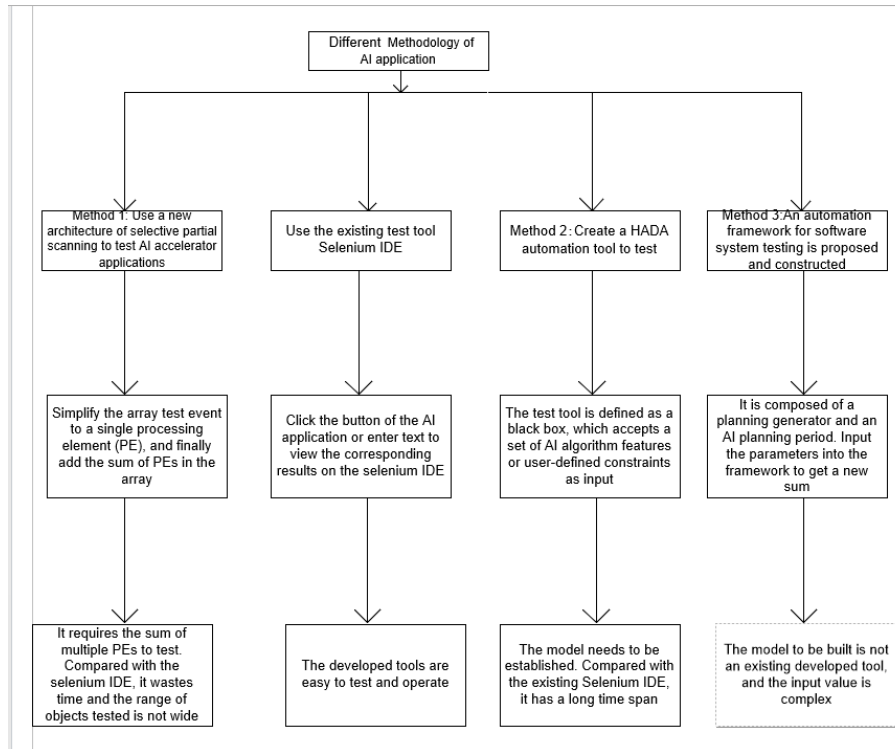


Figure 6: compare with different method

Chapter 3 Methodology

3.1 Approach for Automated Testing

The Selenium IDE is an open source online automated testing tool included in the Selenium Suite [12] [15] [21]. It is a fundamental tool for script creation. It has a very intuitive user interface and is a Fire Fox extension. It includes a recording function that enables users' activities to be captured and exported as scripts that may be reused. Invest in test automation training for beginners with stringent requirements, script authoring, tool acquisition, and software feature analysis [11]. It only needs to record the interactions with the browser to generate test cases [17] [19] [21]. No programming logic is required to build the test foundation.

At present, test cases are executed utilizing web browsers. Selenium manages and logs these tests generally. Selenium can record sensitive clusters in order to identify and standardize comparable actions [5]. [16]. Selenium IDE, for instance, executes and logs tests for three unique login and registration script pages (Figure 7). There will be a record of the values and accounts registered. The Selenium IDE is able to complete each testcase for a variety of AI application testing. [8] It is acknowledged that web testing is a subfield of programming and that the nodes of testing websites are quite complex. [14] Specify that this data processing procedure requires a robust and secure device. Selenium IDE can be assembled and tested in tandem with many IDE versions and can eliminate node redundancy. According to experts [9], Selenium IDE will likely be employed in the future to overcome concerns with online testing. It expedites the testing and optimization of sensitive website regions.

TABLE I. Selenium Test ST1

	Command	Target	Value
e1	Open	/magento/index.php/	
e2	clickAndWait	link=Log In	
e3	Type	Email	abc@xyz.com
e4	Type	Pass	Mypswd1
e5	clickAndWait	send2	
e6	clickAndWait	//img[@alt='Magento Commerce']	
e7	clickAndWait	link=Acer Ferrari 3200 Notebook Computer PC	
...			
e11	clickAndWait	link=Log Out	

Figure 7: Selenium Test

The Selenium IDE mostly searches or crawls the homepage using CSS and Link Text [13][18], However there is a disadvantage: the Selenium IDE cannot detect the located components, and the response time in the real test script is delayed or the system crashes and cannot identify the elements. In the actual test script, the author discovered that the running speed of the artificial intelligence program on the test extranet would become slower than the running speed of the artificial intelligence program in China, and the selenium IDE could not find the elements of CSS, LINKTEXT or id, so the XPath method was used to quickly locate the elements, and the XPath was required to enter text on some programs listed in the web element interaction page, because the page has been changing all the time. XPath has a high level of expressiveness and universality, and each new version of the program requires the locator to be reviewed and maybe altered. [7] In some circumstances, XPath is argued to be the sole viable solution. Only the built-in XPath of the individual Selenite IDE may be used to locate the element when the identification is absent or when numerous test suites are executed in succession. Despite the fact that classical XPath can rapidly identify the position of items, it will result in a loss for some businesses when testing software using the web test suite. [7] When it comes to the fact that XPath localization is particularly fragile, it is often considered that the vulnerability of XPath locators hinges mostly on the fact that

they are created by tools. In contrast to CSS- and id-localization, some development tools are unable to localize web components due to XPath's high error rate (Figure 9).

Name:	John	
Surname:	Doe	
Gender:	Male	
Phone:	123456789	Target Element


```

<html>
<body>
  <table id="userInfo">
    <tr><td>Name: </td><td title="name"> John</td></tr>
    <tr><td>Surname: </td><td title="surname"> Doe</td></tr>
    <tr><td>Gender: </td><td title="gender"> Male</td></tr>
    <tr><td>Phone: </td><td title="mobile"> 123456789</td></tr>
  </table>
</body>
</html>

```


Tool	XPath Locators	Robustness	✓ robust	✗ broken
FirePath	✗	/html/body/table/tr[3→4]/td[2]		
FirePath	✗	//*[@id="userInfo"]/tr[3→4]/td[2]		
Chrome	✗	//*[@id="userInfo"]/tr[3→4]/td[2]		
XPath Helper	✗	/html/body/table[@id="userInfo"]/tr[3→4]/td[@title="mobile"]		
XPath Checker	✗	id("userInfo")/tr[3→4]/td[2]		
ROBULA+	✓	//*[contains(text(),'123456789')]		

Figure 9 XPath error– Ver. 2 – Page, Source, Locators

It is evident that XPath and XPath checker are not identified as such website elements. An algorithm, the resilient locator algorithm, is developed to tackle this issue. It is the advancement of ROBULA with the addition that a single node employs several XPaths for localization and prioritized measurements to improve the code. By constructing a current algorithm (Figure 10) it is possible to detect the distinct items using the variable, the list, and the Boolean function values.

```

1. XPath ROBULA+(XPath abs, Document d)
2. {
3.   Element e = eval(abs, d);
4.   List<XPath> xpList = ["//*"];
5.   while (true)
6.   {
7.     XPath xp = xpList.removeFirst();
8.     List<XPath> temp = [];
9.     temp.append(transfConvertStar(xp));
10.    temp.append(transfAddID(xp));
11.    temp.append(transfAddText(xp));
12.    temp.appendAll(transfAddAttribute(xp));
13.    temp.appendAll(transfAddAttributeSet(xp));
14.    temp.append(transfAddPosition(xp));
15.    temp.append(transfAddLevel(xp));
16.    for (XPath x : temp)
17.    {
18.      if (uniquelyLocate(x, e, d)) return x;
19.      else xpList.append(x);
20.    }
21.  }
22. }

List<XPath> eval(XPath abs, Document d):
  returns the element in d selected by the XPath locator abs

Boolean uniquelyLocate(XPath x, Element e, Document d):
  TRUE iff eval(x, d) contains only e

```

Figure 10 New Algorithm about XPath Pseudocode of ROBULA

The results of this method can be seen (Figure 11) which elements can be located on the web using this algorithm, and XPath can locate elements and get successful results on the selenium IDE.

Name:

Surname:

Gender:

Phone:

John

Doe

Male

123456789

← Target Element

```
<html>
<body>
  <table id="userInfo">
    <tr><td>Name: </td><td title ="name">      John</td></tr>
    <tr><td>Surname:</td><td title ="surname">    Doe</td></tr>
    <tr><td>Gender: </td><td title ="gender">      Male</td></tr>
    <tr><td>Phone:  </td><td title ="mobile"> 123456789</td></tr>
  </table>
</body>
</html>
```

Tool		XPath Locators	Robustness	✓ robust	✗ broken
FirePath Abs	✗	/html/body/table/tr[3]/td[2]			
FirePath Rel	✗	//*[@id="userInfo"]/tr[3]/td[2]			
Selenium IDE	✗	//table[@id="userInfo"]/tr[3]/td[2]			
Montoto	✓	//td[text()='123456789']			
ROBULA	✓	//td[@title="mobile"]			
ROBULA+	✓	//*[contains(text(),'123456789']			

Figure 11 New Algorithm result Pseudocode of ROBULA

3.1.2 Approach for OpenCV

OpenCV is an open-source computer vision library that contains hundreds of computer vision algorithms and has a modular structure, which means the program has a range of shared or static libraries. OpenCV supports a number of assembly languages, such as C++, Python, and Java, and may be used on many systems.

OpenCV face detection includes image reading, image pre-processing, Cascade Classifiers, and obtaining the final result. The flow chart is shown in Figure12

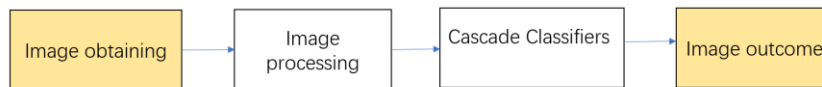


Figure 12 Face Detection Flowchart

The process of image processing consists of three parts: grayscale processing, thresholding and histogram visualization, all of which are beneficial in the application of face recognition development (Figure 13).

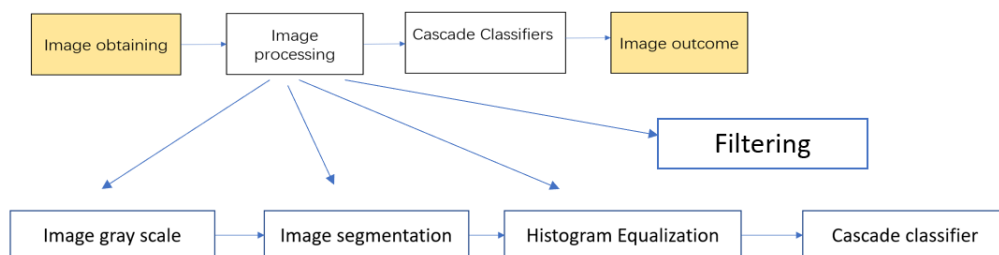


Figure 13 Image processing

A. Image Grayscale

Pixel colors are determined by R, G, and B components. Each component's values range from 0 to 255, and the range of pixel changes in a color image is $256 \times 256 \times 256 = 16777216$ colors. interference data. The CB (COLOR_BGR2GRAY) function included in the OpenCV package is used to convert color images to

grayscale. With the `CVCOLOR ()` function, the image is converted from one color space to another while maintaining the same bit depth as the source image.

B. Image segmentation

Image segmentation processing is to segment the information of an image, so that the segmented area becomes a meaningful region of information, this report then needs to divide the processed image, the image is generally processed using image binarization, the processed image has two colors values i.e. black and white, thresholding eliminates noise according to the actual situation, it is a technique to eliminate pixels from the image when the pixels exceed a fixed value, in developing face mask recognition The function `cv2.threshold(IMG,THRESH,MAXVAL,TYPE)` is used to set all pixels in the image that are less than thresh to black and vice versa.

C. Histogram Equalization

This report adds histogram uniformity to cope with the effect of poor image detection caused in different environments, the use of histogram uniformity improves image quality as well as accuracy, the image histogram is in fact the frequency of pixels in the relative grey scale range (0-255) and the statistical plot given based on the detection results is called a histogram. The histogram offers clear feedback on the intensity distribution of the picture pixels in the image, the histogram is used to increase the contrast of the image by expanding the range of the pixel intensity distribution, hence the sharpness of the image.

The histogram equalization formula is as follows:

M is the number of gray scale, N represents the number of pixels of the picture, histogram for a gray scale image where van is $[0, m-1]$. M is a digital image with 8bit display bit depth representing a maximum range of 256 and a gray scale of 0-255, and a gray scale $r(k)=255$ there are $n(k)$ pixels. Using $n(k)/n$ it is possible to derive the grey level $r(k)$ in the digital image to obtain the grey density function H . By counting the density of all the grey levels. it is possible to draw a histogram, and in the case of a histoscore plot it is desired to find a change $s=T(r)$ ($0<r<1$) The probability density

function of s is homogeneous and $T(r)$ satisfies two conditions. $t(r)$ is positive and monotonically increasing in the interval when $0 < r < 1$, $0 < T(r) < 1$. Condition 1 Denote $M = T(r)$. R is the grey scale value of the input image and M is the grey scale value after the transformation. the transformed image grey scale histogram needs to satisfy two conditions.

$$H(x) = \frac{m-1}{s} \sum_{i=0}^k n_i \quad (1)$$

Figure 14(a) Histogram equalization formula

Authors suggest that in extreme environments such as haze, rain, darkness or bright light, the application of the histogram equalization algorithm can optimize the processing of face images in harsh environments, with relative robustness for edge detection.

D. Filtering

Noise information inevitably exists in face images. In order to improve the effectiveness of facial recognition, face images must be pre-processed in order to facilitate the elimination of noise information in the image, and the next image enhancement can make the image clearer and be used for different image enhancement algorithms for the characteristics and effects of face images. This report uses median filtering to enhance face images, eliminate noise from images and improve image quality, thus improving the recognition rate and discrimination accuracy of images.

D(a) Median filtering

Median filtering is commonly used in image pattern recognition to effectively eliminate pushed noise and preserve the sharpness of image edges and features of image detail and it is a very non-linear smoothing technique that sets the grey value of each pixel point to the median of the grey values of all pixels within a certain neighborhood window of that shop. It does this by using a two-bit sliding template that sorts the pixels within the slab according to the size of the pixel value, generating monotonically rising or falling two-bit data according to the arrangement.

Algorithm 1: Median filtering algorithm

- (1) Select a sliding window, i.e. $(2n+1) \times (2n+1)$ (Ordinary is 3×3 or 5×5).

(2) Slide the window along each pixel in the row or column of the face image (usually from left to right, top to bottom), and make the center of the window coincide with a pixel in the face image.

(3) Read the grayscale values of each corresponding pixel in the window.

(4) Sort pixels by grayscale.

(5) Select the median of this group as the output pixel value.

(6) Use this median instead of the grayscale value of the center pixel of the window.

D(b)mathematical formula

$$g(x, y) = \text{median}\{f(x - i, y - j)\}, (x, y) \in S \quad (2)$$

Figure 14(b): Median filtering mathematical formula

In this mathematical formula, $g(x, y)$ is the pixel grayscale value (x, y) preprocessed through median filtering; $F(x, y)$ is the pixel grayscale value (x, y) before preprocessing through median filtering; S - Template window; I - horizontal size of the window; J - The vertical size of the window.

E. Cascade Classifier

A cascade classifier based on HAAR features is used for object detection, which has a good detection rate, accuracy and fast processing speed for hairy inputs. The face detection method is very sensitive to face recognition and the presence of small cells. In OpenCV, based on fast target detection and machine learning methods, the cascade function is trained from a large number of positive and negative images and then used to detect objects in the images. The classifier is essentially trained on a large number of positive images (face images) and negative images (images without faces). How it actually works is that if a face is identified, a process of image subtraction morphology is used in the classification detection of faces. Such subtraction occurs on each acquisition image in the classifier, but this subtraction occurs on each acquisition image that is classified and also the images with less error are collapsed and or summed, which is ultimately called a weak classifier. All the weak classifiers are wrapped and merged together to form the

strong classifier, in which the desired features are then extracted, such as the features of the face, i.e. the nose or mouth, HAAR functions like a convolution kernel, each feature value is obtained by subtracting the pixel synthesis of the white rectangle from the pixel synthesis of the black rectangle, in cv. Cascade classifiers have xml files for detecting the nose, but they are mainly for Open CV own face detection model.

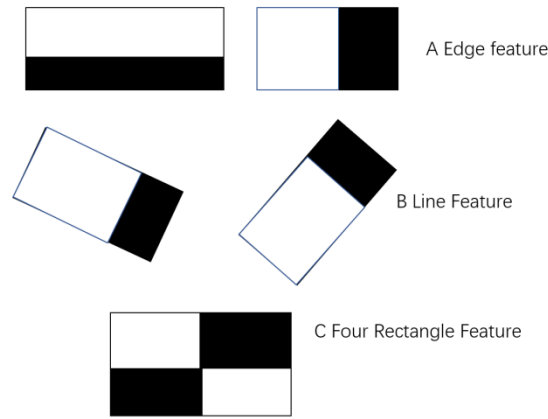


Figure 14 Eigenvalue processing

E(a)Integral graph construction algorithm:

$$V_i = \sum_{j=1}^n u_{ij} W_j \quad i = 1, 2, \dots; m \quad (3)$$

Figure 15 Integral graph Algorithm

The integral image algorithm defines a rectangle with (i,j) representing the cumulative sum in the row direction and initialized as Figure15, and $V_i=(i,j)$ representing an integral image, scans the image line by line, calculates the cumulative sum in the row direction for each pixel (i,j) at the beginning by denigrating it, and when the scan reaches the bottom right pixel of the image, the integral image is well constructed. The image is read by integrating the entire image. Figure 16 shows the classifier's face recognition process, in which most of the image is a non-face region, the window is checked by the classifier cascade method

to see if it contains a face region, if not, it is discarded at once, and if there is a possible face region these features are grouped into different stages of the classifier and used one by one.

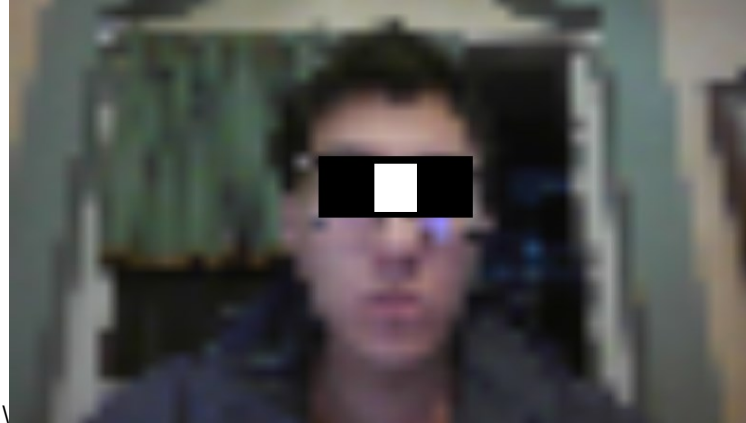


Figure 16 Classifier face recognition process

As shown in Figure 17 Traversing from left to right and finally summing to obtain the image as Eigenvalue Identification, it can be clearly seen that HAAR first scans the background of the image and stops when the scanner sweeps in the lower right corner of the green box to determine the position of the eyes and the mouth in the case of the green box.

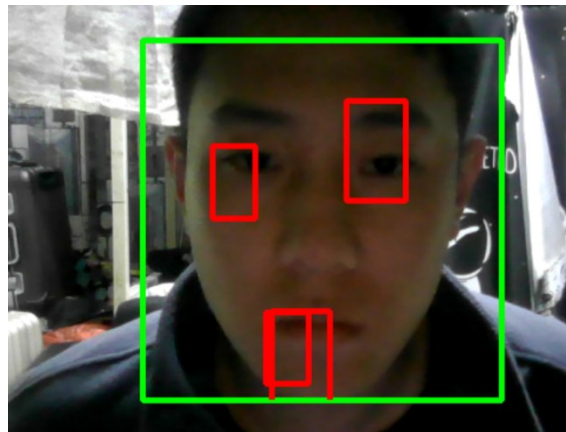


Figure 17 Eigenvalue Identification

D(b) Detect Multiscale

The multiscale detection program identifies every face and stores the face data as a vector while processing the grayscale picture. The goal of processing is a vector of rectangles, each holding the detected item; numerous faces can be detected in a single picture; hence, multiple rectangles must be applied to the image for recognition. how

can faces be reliably identified? This is a really important query. While the size of the picture is lowered column by column, the scale of face detection does not grow as vast as in Figure 20, this approach is analogous to constructing a pyramid by sampling nearby pixels from the image. Just identifying faces in a picture is the objective. Using the least neighbor approach, the detector first detects body areas and then picks body regions by applying a filtering operation to these candidate regions, filtering each candidate region by the number of least neighbors. If none of the object's neighbors are present, no object parameters will be collected; nonetheless, having too many or too few parameters will impair the detector's capacity to identify entities. Too many parameters, for instance, Similar to the KNN technique based on deep learning, too many parameters will yield fewer entities with high precision, whereas too few parameters would yield more entities.

$$T = (x_1, y_1), (x_1, y_1), \dots, (x_N, y_N) \quad (4)$$

Figure 18 sets

The procedure begins by establishing several sets of sets as figure 18, with x representing the vector of data and y representing the feature values of the data. In face recognition, x indicates the length or breadth of the face, whereas y represents the organ that the user wishes to identify. For instance, in the Iris dataset, x represents the length or width of the leaves and y represents the kind of Iris. How a computer may obtain the feature values of an item using a particular algorithm, the picture input is in pixels, and the computer uses a special method to extract the feature values of the object. In essence, an assortment OpenCV retrieves the feature values through a manner similar to KNN, in which the picture is traversed, and if the feature value requested by the user is a nose, the computer does the full image of a nose. The third parameter of the method it provides is user input, the purpose of which is to indicate that each feature value must be detected at least n times before it can be considered a real target, its mathematical formula is depicted in Figure 18.

$$\begin{aligned} d(A, B) &= \sqrt{(x_{1a} - x_{1b})^2 + (x_{2a} - x_{2b})^2 + \dots + (x_{3a} - x_{3b})^2} \\ &= \sqrt{\sum_{i=1}^n (x_{ia} - x_{ib})^2} \end{aligned} \quad (5)$$

Figure 19 Distance Algorithms

The Euclidean formula is used to measure spatially focused distances, which is

calculated simply by squaring the x and y values from the origin to the target point

Figure 19. Figure 20 details how this can be used to determine the category of eigenvalues, with small circles and triangles as different eigenvalues, circles as nose eigenvalues and triangles as mouth eigenvalues, with an elected value of 3. If the closest one is a triangle, then the eigenvalue belongs to the nose category and the selected value is 5. If the number of small circles is higher than the number of triangles then the circle is selected as the eigenvalue.

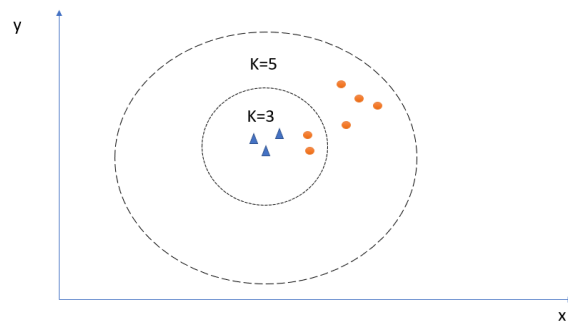


Figure 20 Core Approach

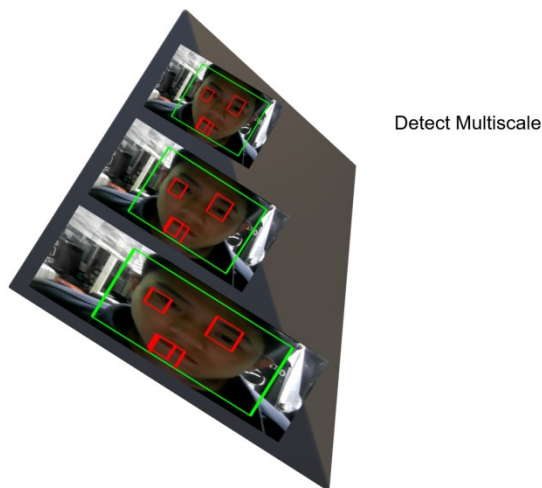


Figure 21 Scale diagram

3.2 Technology

Software

1. Development platform Selenium IDE to test. For instance, click some button as the objective thing and record the result.
2. Use PyCharm to build the facemask model.
3. Using selenium IDE to convert scripts to Java language
4. Visualize data using PyCharm (code to image)
5. Using the Mat Plot lib to plot the diagram.

Hardware:

1. Computer (GPU, CPU, Camera)
2. Shared GitHub repository to manage the several versions of the test files script and project code.
3. Upload weekly report in the Fei Shu repository.
4. GitHub link text: <https://github.com/charliejacklove/project>

3.3 Project Version Management

Use GitHub Desktop to create a branch and name it Project. In this branch, create a private folder named code because all codes are uploaded on GitHub (note: users have been created for GitHub)

Chapter 4 Result

This report found several problems, problem 1: selenium IDE is not sensitive to the target named id and Link Text crawl, how to solve the above problem, the following is the solution to the missing locator method. Solution1: The authors found that the id and Link Text were replaced with a different form, for example, the id or Link Text was replaced with an XPath or CSS locator target that was less likely to be lost and the script eventually ran successfully. Problem 2: When the locator element is XPath and CSS, the selenium IDE still cannot find the location of the locator element. Solution 2: Use the verify command to check if the element exists on the page, if it is found, selenium will automatically perform the next operation. Solution 3: Check if the element is visible by manually i-checking the HTML or the command FINDELEMENT in the selenium IDE console or use the WAITFORELEMENT command to wait for the element to appear before testing. Problem 4: Run time exceeds 30,000ms to terminate the test, Solution 4: Debug the speed of the test, because when testing the site, the page is not even updated with the positioning of the element, selenium can't find the positioning of the element to report an error. The user needs to manually click on the button called test speed in the automation tool to set the speed to the lowest possible level to find the positioned element and complete the test. If the above problem is still solved, it is recommended to change to a new browser, such as Google Chrome or Firefox, as they run a little faster than other browsers. Or use a testing framework: use a testing framework such as JUnit to manage tests and make them run faster.

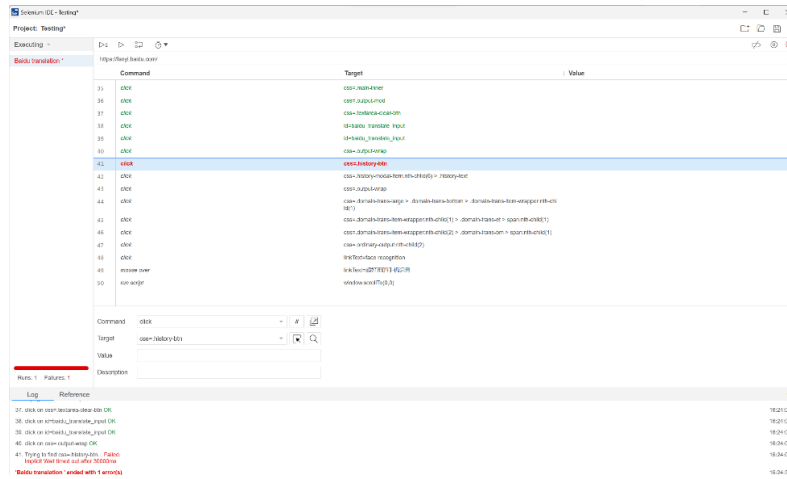


Figure 22 Selenium IDE Source code (2022) Over Time Testing

Use Solution 1 change the ID locator element to XPath element

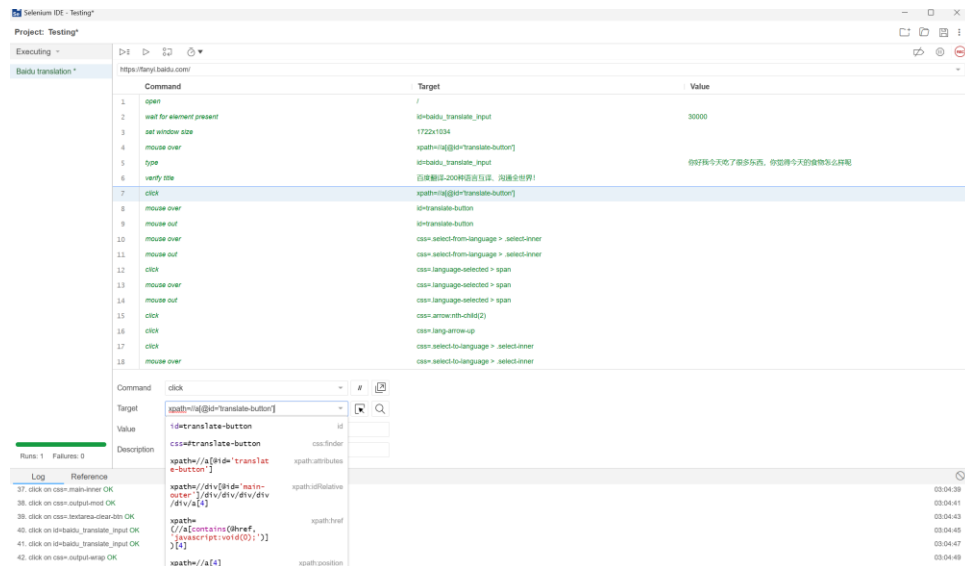


Figure 23 Selenium IDE Source code(2022) id change to XPath

Use solution 2 add the verify to ensure element whether is existence

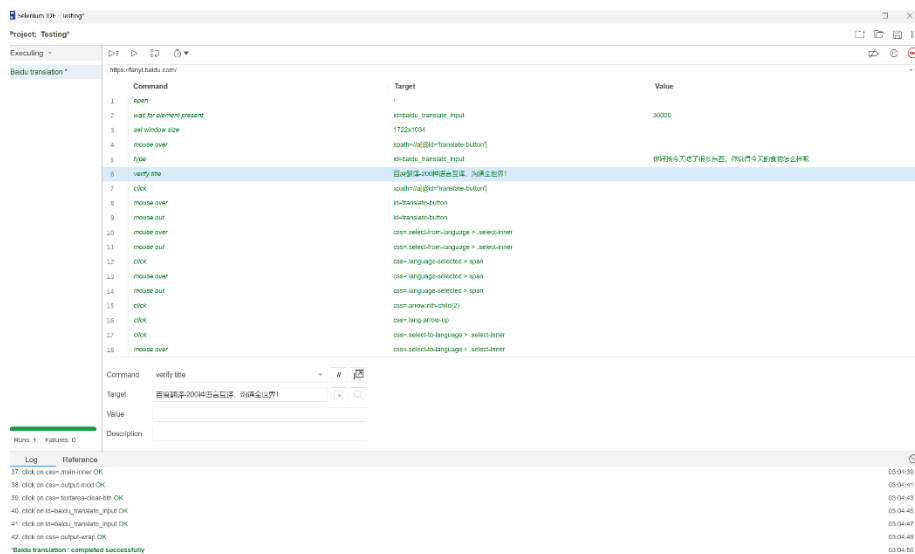


Figure 24 Selenium IDE Source code(2022) add verify method

Use solution 3 wait for element to solve the issue

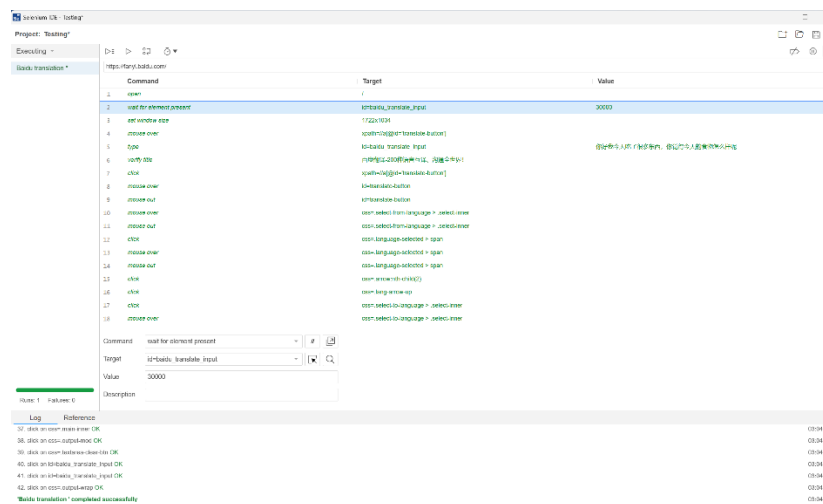


Figure 25 Selenium IDE Source code (2022) wait for element method

Use solution 4 Adjusting the speed of running tests

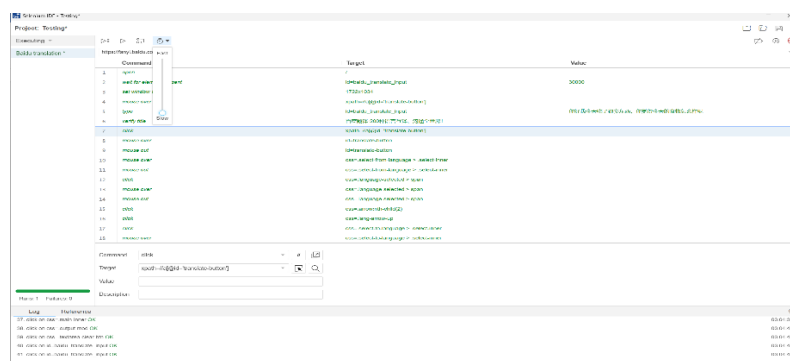


Figure 26 Selenium IDE Source code(2022) Adjusting speed

Use solution 5 Using the testing framework: script change to Junit

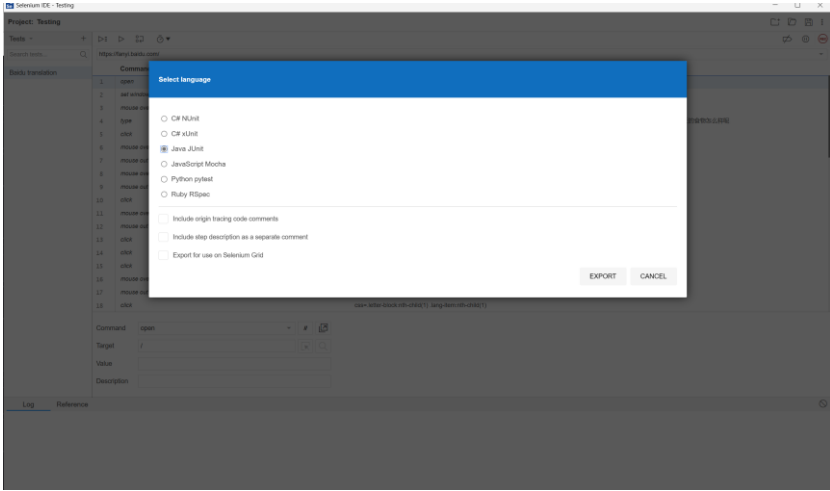


Figure 27 Selenium IDE Source code (2022) script change to Junit

Finally, through the above methods. It designs and tests five different AI programs, uses the selenium IDE to test the functions on the website, and finally converts the code on the selenium IDE into the Java Unit language [15]. Figure 28 is about the selenium test. which can see that the script runs successfully. Solve the above problems through the above solutions.

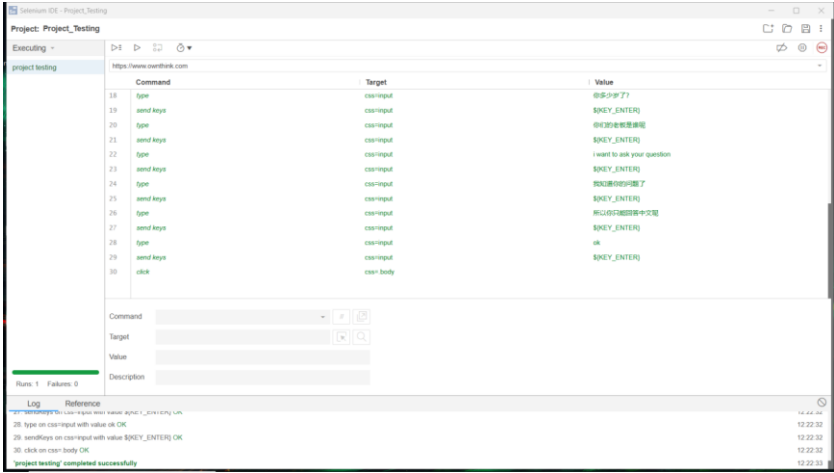


Figure 28 Selenium IDE Source code (2022) Testing AI application

After summarizing the above-mentioned testing methods of selenium IDE, I found that this testing tool still has some limitations. When users test some special websites, such as uploading images for text recognition or translation, running the script will directly skip this part of the content, which will make the observer think that this part is incomplete and the receiver cannot judge whether the product is

qualified by this test. In addition, the selenium IDE is not sensitive to positioning elements such as ids and link text. Although there may be some changes on the page that cause the text of the element to change, it is not possible to get the correct result for some positioning elements that have not changed. In terms of debugging: Debugging the selenium IDE's test scripts is very difficult. Debugging requires running the entire script from scratch, which is fatal for developing tests for large projects as it takes a long time and many software companies do not use this tool for testing websites. From limited testing scope: The Selenium IDE is not suitable for testing complex web applications. It has limited support for dynamic content, AJAX and JavaScript-intensive web pages. This limitation makes it difficult to test complex scenarios that require advanced scripting capabilities. From limited reporting capabilities: The Selenium IDE has limited reporting capabilities. It provides basic reporting features such as pass/fail status, execution time and error messages. However, it does not provide detailed reports with charts, graphs and other advanced features. For locating elements, the frequent use of XPath locality has an impact on automated testing and can lead to test script time overflow, as the testing tool needs to locate all XPath elements of a page element when testing a website.

For testing frameworks, converting scripts to Junit will result in errors when running the test program (Figure 29).

Test case	
Test case id	1
Test case Purpose	Test data of user method is wrong
Input action	Wrong result
Expected output/action	Testcase can't find element
Actual output/Action	Wrong

Table 1

```

十二月 29, 2022 7:31:24 下午 org.openqa.selenium.devtools.CdpVersionFinder find
警告: Unable to find an exact match for CDP version 108, so returning the clos
十二月 29, 2022 7:31:24 下午 org.openqa.selenium.devtools.CdpVersionFinder find
信息: Unable to find CDP implementation matching 108.
十二月 29, 2022 7:31:24 下午 org.openqa.selenium.chromium.ChromiumDriver lambda
警告: Unable to find version of CDP to use for . You may need to include a dep

```

Figure 29 Selenium IDE Source code (2022) Testing wrong

Limitations: It uses the Selenium WebDriver API to communicate with the web browser and interact with the elements on the page. In addition, Junit's test code needs to be modified when elements of the page change to use an element locator that is compatible with the current page structure, which requires the user to update and test the code in a timely manner. How do you solve the problem of Junit not finding the location element? Author proposed two methods.

Method 1 In order to solve the problem that Java unit cannot find elements. the author edited a new algorithm (Figure 30). In this process, the author added a for loop and a list to access and iterate all the elements of the page.

Test case	
Test case id	1
Test case Purpose	Test data of user method 1 successful to run it
Input action	Change the code and use for loop
Expected output/action	Testcase can run and successful to get result
Actual output/Action	Success

Table 2

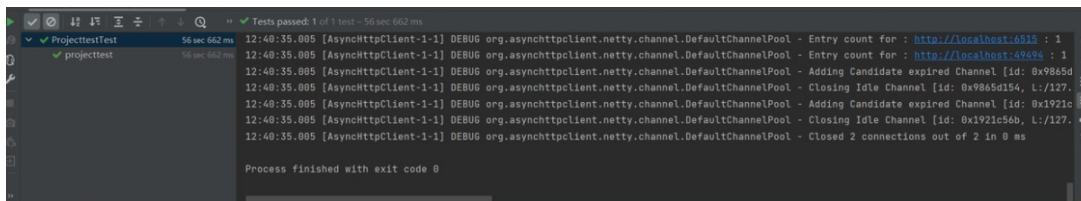


Figure 30 IntelliJ IDEA source code (Version 2022.1) successful result.

Outcome: successful to get result

Method 2: The second method is to construct a function to make the server "sleep". The compiler will look for this element during this period of time (Figure 32). However, this method also has some disadvantages. When the input value is too

large, the compiler will automatically report an error and display the stagnation time is too long. When the input value is too small, the compiler does not have enough time to find the location element. Method 1 and Method 2 are compared, Method 1 can quickly locate the elements and takes less time and memory.

Test case	
Test case id	2
Test case Purpose	Test data of user method 2 successful to run it
Input action	Change the code and using wait for element and waiting 30000ms
Expected output/action	Testcase can run and successful to get result
Actual output/Action	Success

Table 3

Outcomes: successful to get result

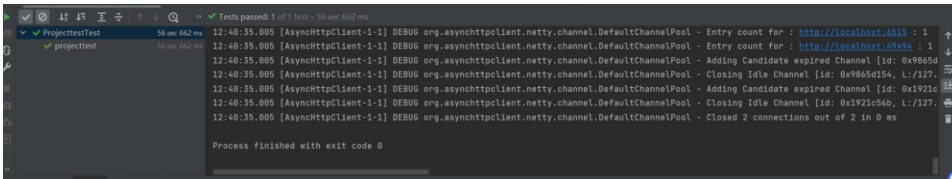


Figure 33) IntelliJ IDEA (Version 2022.1) successful result

The authors have found several advantages in using Junit testing. Integration: Selenium IDE scripts can be easily integrated with the Junit testing framework to enable testing and analysis of results. Reliability: Junit testing ensures consistent execution of Selenium IDE scripts, which reduces the likelihood of runtime errors and improves overall software quality, and JUnit testing provides quick feedback on the status of tests, enabling users or developers to identify and resolve issues. However, Junit has certain drawbacks and its functionality is limited: Junit is mainly focused on unit testing and it does not provide comprehensive testing functionality. Complexity: Junit testing requires additional code and can be time and resource intensive which is not practical for smaller projects.

This report uses TDD (Test-driven development is the core practice and technology in agile development, and it is also a design method. The principle of TDD is to write unit test case code before developing functional code. The test code determines what product code needs to be written. The basic idea of TDD is to promote the whole development through testing. However, test-driven development is not only a simple test work, but also a requirement analysis and design, The process of quality control quantification.) test to divide the test into failed test, successful test and refactoring. Use selenium IDE to test different AI websites. The following are different tests and evaluation (Form1). Unit test is a unit test of a script to verify whether each part can reach the next step. When testing a website, the selenium IDE is used for unit test to make the program run one by one. For example, what happens when you click the next button or hyperlink when testing a website, and what happens when you enter text, Integration testing tests multiple scripts to verify whether each function tested can reach the expected results. Use selenium IDE to test the website, click on the page or enter text to observe whether each function can run normally and record.

Objects	TDD	Test Method	Evaluation Method
Each function	Successful test	Unit Testing	Functionality availability
Each function	Successful test	Integration Testing	Functionality availability
One function	Failed test	Usability Testing	Functionality failure
Each function	Failed test	Stress testing	Functionality failure

Table 4

For the processing of face mask recognition images, grey-scale processing, threshold histograms, and grey-scale histograms are all effectively verified with the aim of better discriminate whether a face is wearing a mask.

Why use greyscale processing, because greyscale images contain only luminance information, each pixel is represented by only one value, therefore the size of a

greyscale image takes up less memory than a color image, this makes the processing of greyscale images faster, secondly greyscale images have better contrast, the value of each pixel in a greyscale image represents the luminance of the pixel, therefore the contrast of a greyscale image is more pronounced, in addition to that Grayscale images reduce the dimensionality of the data, as they have only one channel and can therefore be reduced to two dimensions, reducing the complexity of computation and storage. Grayscale processing improves robustness, as grayscale images are more robust to changes in illumination and citation, therefore improving the stability of the face recognition mask algorithm and reducing false positives and missed detections due to changes in illumination. The following are images of the greyscale processing (Figure 34).



Figure 34 compare with grayscale image(left) & source image(right)

The goal of image segmentation is to extract meaningful information from an image by grouping similar pixels together and separating different regions. There are various methods of image segmentation, including thresholding. Thresholding entails setting a threshold and classifying each pixel as foreground or background based on whether it is above or below the first threshold when it comes to intensity. In face mask recognition, thresholding is a good processing method that divides the image split into two parts: foreground and background. Thresholding compares the grey scale values of all pixel points of the image with a set threshold. The pixels below the threshold are divided into two categories: those below the threshold are used as the background and those above the threshold are used as the foreground. The specific steps are as follows: 1. Determine the threshold value:

set it according to the characteristics of the image and determine it by manual adjustment. 2. The image is binarization, dividing the pixel points in the image into two categories based on the pixel points Pixel points below the threshold are set to 0 Pixel points above the threshold are set to 255.



Figure 35 Thresholding processing

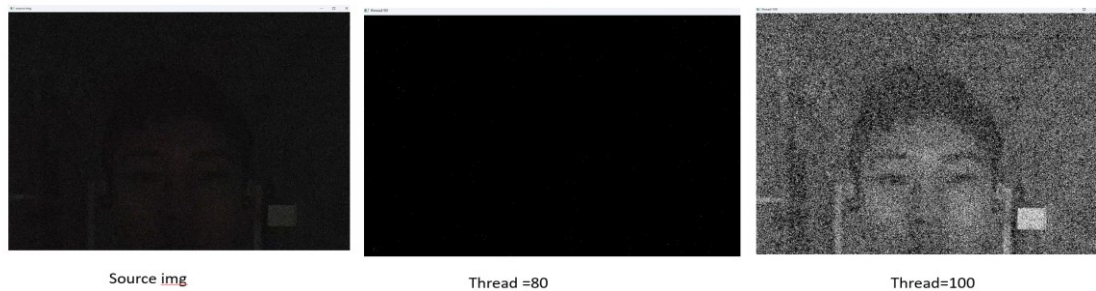


Figure 36 black background face image

In different environments, the threshold processing needs to be debugged, the above picture was taken by the author in a dark situation, after the threshold processing, we found that the threshold value at 90, the processed image will have a lot of noise, in the dark background, the noise of the image may be recognized as brighter elements than the surrounding, when the threshold value is 100 the effect of the processed picture is even clearer than the original picture, but the author However, the authors do not recommend the use of images with a threshold of 100 as a parameter for face recognition, because such images are usually not referred to, firstly the face is not correctly recognition and secondly too much noise will spoil the subsequent steps of processing the image, based on the above conclusions the authors draw the following conclusions. 1. Adjust the background image of the camera reasonably well. Avoid

taking pictures in too dark or bright environments. In the case of an appropriate background image, adjust the size of the threshold appropriately, too much threshold leads to unclear faces and too little threshold makes it difficult to see the facial features. This is a conclusion based on the thread.

Thresholding still has some limitations, especially when the background of the image is a highly variable environment such as lighting and noise, and the authors use histogram equalization to solve the problem of unprocessed thresholds. Histogram equalization is a method of enhancing the contrast of an image by redistributing the distribution of luminance values in the image to give a more even distribution. Figure 37 is a visualization of the image using mat lab plot, with the image background in Figure 36. In the straight cube diagram, the horizontal coordinate represents the interval of the numerical processing range, the vertical coordinate represents the number of data in the interval, Figure 37 shows that the maximum value of the vertical coordinate is 125000, the maximum value of the horizontal coordinate is 200, the picture on the left is the picture of grayscale processing averaging, the picture on the right is the picture of grayscale processing, the author found that in the picture on the right, the data direction is more uniform but the number of data is less than the picture on the left, the ordinary grayscale after the photo processing is not good, firstly, the number of data is less, secondly, the data is too concentrated. Then this picture can't be used as a picture to identify the face.

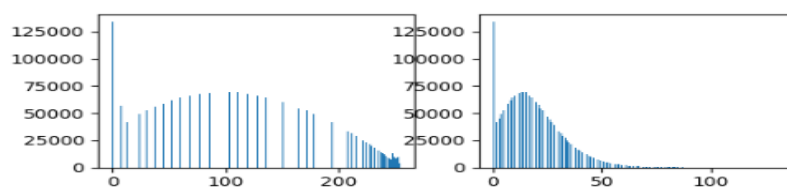


Figure 37 Comparison of grey-scale equalization histograms and grey-scale histograms

Figure 38 then shows the grey-scale histogram in a normal environment. A comparison with Figure 37 shows that the number of pixels in the horizontal coordinate is higher in the normal environment, which means that the quality of the soil is good. This can be used as a reference image for the project and the histogram shows a bimodal distribution as well as peaking. In addition, the peak of the histogram reflects the grey scale value with the largest number and percentage of images, and the larger peak represents the image that has the highest number and percentage of pixels in the histogram. This means that the histogram can be used as an important reference for face mask recognition.

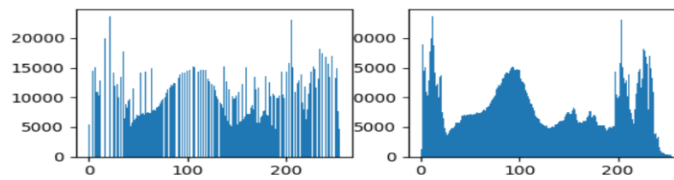


Figure 38 grey-scale histogram in normal diagram

However, the data analysis alone is not sufficient to demonstrate the effect of the straight cube plot equalization, the visualization of the straight cube plot only shows that the graph can take the results of the face mask recognition, but for a more intuitive representation the authors show the results of the run directly in this report.

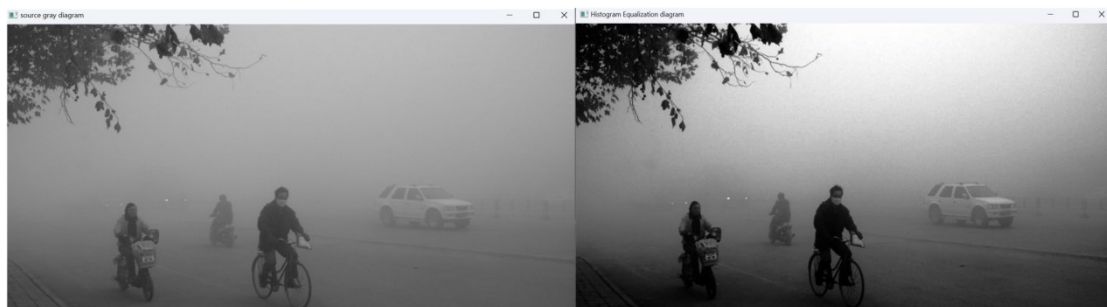


Figure 39 Compare with Histogram diagram (Right) & Gray image (Left)

The demonstration in Figure 39 shows that the greyscale histogram has a significant

effect, firstly, the figure shows the flow of people as well as vehicles in a hazy background, the greyscale processed image cannot be clearly seen as a picture of a human face, the right image is a processed image. both the human image and the vehicles are effectively identified, so the authors suggest that in extreme environments such as haze, rain, darkness or bright light, the application of the histogram equalization algorithm can optimize the processing of face images in harsh environments, with relative robustness for edge detection.

Median filtering is commonly used in applications such as image denoising, edge detection, and feature extraction. It is a simple and effective way to remove noise from an image while preserving edges and other important features.

The following is the result of the median filtering process. You can clearly see that the left plot is sharper than the right plot, and that the noise in the left plot has been perfectly processed. The noise is much reduced and you can see the faces in the image.

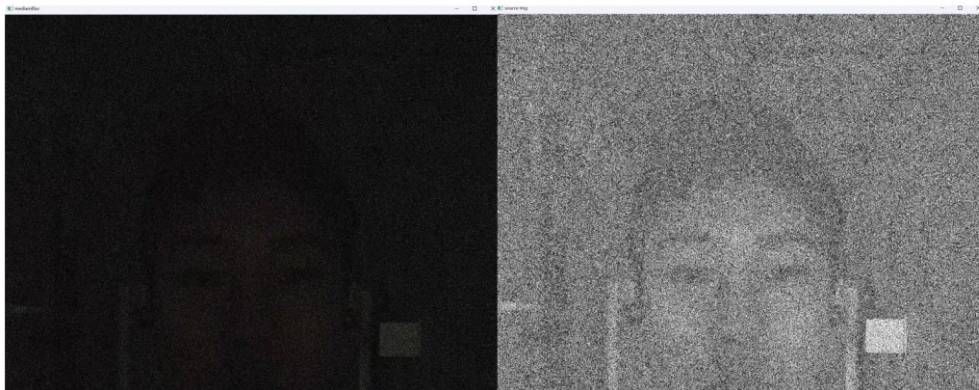


Figure 40 Median filtering process(left) & gray scale image(right)

Figure 40 shows the results of the median filtering process. You can clearly see that the left image is much sharper than the right image and that the noise in the left image has been perfectly processed. The noise is much reduced and the face is visible in the image but the median filter has some drawbacks, the planting filter works by replacing the value of a pixel with the median of the neighbourhood pixels. This can also lead to a loss of detail in the image, especially in areas with sharp edges or transitions. In terms of computational complexity, median filtering requires sorting and collating adjacent pixels for each pixel in the image, which is

computationally expensive for large images or for real-time applications, and cannot handle non-uniform noise, which cannot always occur in practice, so other noise processing techniques may be more effective in dealing with non-uniform noise.

Chapter 5 Professional Issues

5.1 Project management

Project management can be a challenge because there is a lot of time spent on certain tasks, which can result in too little time left to complete subsequent tasks. With the right tools and time certain management techniques can be utilized to alleviate these challenges. TDD is a development model that has at its core the idea of designing test code before developers implement features and writing product features based on the listed test code, which provides a clear structure to the project and allows for testing before the code is developed.

5.1.1 Activities

Objective	Activities
Test and analysis already exiting AI applications	<ul style="list-style-type: none">i. Investigate existing AI applications in different areaii. Analyze and compare the functions of different AI applicationiii. Make a table and record the different functions of the application
Generate conclusions on effective methods for building AI applications	<ul style="list-style-type: none">i. Click the special button on AI applicationii. Enter text in the AI application text box and view it in seleniumiii. the response time of this function in seleniumiv. the value of the function in selenium and record it.v. Make a table and record the conclusion

Employ the comprehensive functions learnt from the already existing system to test the Facemask detection system.	<ul style="list-style-type: none"> i. Divide and summarize the functions of existing tables ii. Building the model iii. Apply the recorded method to established facemask application iv. Analyze and record some information such as corresponding events and accuracy
Carry out comprehensive evaluation of the facemask detection system.	<ul style="list-style-type: none"> i. The test results are displayed graphically ii. The corresponding results of different AI applications are analyzed and evaluated, iii. Discuss the future trend of this achievement

The Gantt chart below reflects the completion of all tasks, start and end times and durations. There are some discrepancies for all objectives but they do not make much difference as the tasks can be completed in parallel

5.1.2 Schedule

Task name	Assigned	Start date	End date	Creation date	Duration	Status
		2022/10...	2023/04...	2023/03/03	1184h	
1 Selection of Essay The...	CD CIRE...	2022/10...	2022/10...	2023/03/03	112h	● Done
2 Establishment of tables ...	CD CIRE...	2022/10...	2022/10...	2023/03/03	80h	● Done
3 Prepare Project Propo...	CD CIRE...	2022/10...	2022/11...	2023/03/03	104h	● Done
4 Survey existing opera...	CD CIRE...	2022/10...	2022/10...	2023/03/03	80h	● Done
5 Generate conclusion a...	CD CIRE...	2022/11...	2022/11...	2023/03/03	120h	● Done
6 Testing & analysis result	CD CIRE...	2022/11...	2023/03...	2023/03/03	560h	● Done
7 Evaluation Testing result	CD CIRE...	2022/11...	2022/12...	2023/03/03	224h	● Done
8 Prepare Interim Report	CD CIRE...	2022/11...	2022/12...	2023/03/03	232h	● Done
9 Write Final report	CD CIRE...	2023/01...	2023/04...	2023/03/03	648h	● In pr...
10 Create Poster	CD CIRE...	2023/01...	2023/02...	2023/03/03	320h	● Done

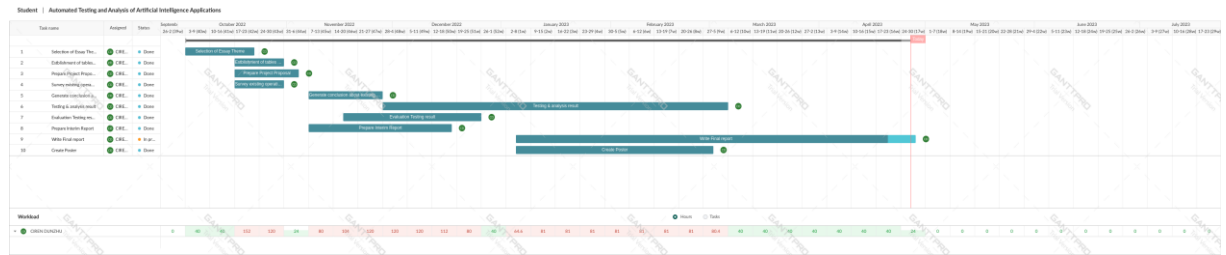


Figure 41 Gantt chart to show tasks and start time and duration time, status

- a) Selection of essay theme - 2 week (Finished)
- b) Establishment of tables and investigation of relevant background - 1 week (Finished)
- c) Prepare Project Proposal - 2 week (Finished)
- d) Survey existing operation system about AI - 2 week (Finished)
- e) Generate conclusion about existing AI system - 4 week (Finished)
- f) Testing & analysis result -14 week (Finished)
- g) Evaluation Testing result - 8 week (Finished)
- h) Prepare Interim Report - 3 week (Finished)
- i) Write Final Report - 11 week (Unfinished)
- j) Create Poster - 8 week (Finished)

Total effort – 26 weeks

5.1.3 Project Data Management

planning documents: sites of FEISHU service folders/other files:

1. Weekly meeting log: progress, follow-up process, supervisor's comments
2. User Story: Planning and Review.
3. Literature: useful links or find new chapters and put them into folders.
4. Report: create the single document to store progress report and proposal.

5.1.4 Project Deliverable

Submitted files: The project proposal, progress report and final report have all been

Submitted to a shared document on FEISHU.

Submitted code: folder created on GitHub.

Unsubmitted documents: Poster presentation file and academic poster

5.2 Risk analysis

Risk analysis based on the current level of project progress: success of resolved risks and response strategy, and changes to the project plan due to risks

This table is arranged as below

Risk type - Types of risks that can occur in a project

Risk - Possible risks in regards to the risk type

Likelihood - The likelihood of the risk occurring as a value between 1 and 5. 1 indicates the lowest likelihood, 5 indicates a high likelihood of occurrence.

Impact - The impact of risk on the project. 1 represents no/very small risk, 5 indicates significant impact on the project.

Response strategy - This shows a plan to reduce the chance of a risk occurring or to reduce the impact of the risk.

to mitigate the impact of the risk.

Risks Type	Risks	Likelihood	Impact	Response Strategy
Loss of data	Loss of software (1)	1	5	Create version control system and upload code to allow for recent versions of code to be recovered.
	Loss of documentation (2)	1	5	Create FEISHU Folder to store all documentation relating to project.
	Loss of code (3)	1	5	The project code is backed up on GitHub, which also ensures that no project code is lost.
Deadline Missed	Poor time management (4)	1	2	List all the tasks to be done and use the Gantt chart which has allowed extra time
	Sprint plan is not well defined (5)	3	4	The sprint plan defines the actions to be taken throughout the process. Throughout the project, the sprint plan may not follow the pre-defined plan of the project

				and to mitigate this, the Americas plan is refactored to allow for enhanced and clear path to completion.
	Test time takes longer than expected (6)	3	4	Complete testing as early as possible and it is recommended that you repeat testing of different applications on a weekly basis.
Technical	Selenium IDE cannot find web element (7)	2	5	Study the different positioning elements, change the id and LINKTEXT to XPATH or CSS positioning elements to ensure that the Selenium IDE finds the elements
	Junit could not find the location element (8)	2	4	Traversal of positioned elements using the new algorithm.
	Face mask recognition does not clear recognition of face (9)	2	5	Use the OPENCV-Python extra library, which has functions for accurate face recognition of faces
	Unable to visualize data for face mask (10)	2	3	Visualize data using corresponding functions.
	Project to complex (11)	1	5	Research project extensively and discuss with the supervisor to ensure the project is achievable.

These risks are categories by severity according to the likelihood and impact of each risk on the project, as shown in the table below.

Key	Meaning	Description
Green	Low/Little Risk	These risks will have little impact on project and recovery from risk will be simple.
Yellow	Moderate Risk	These risks will have a noticeable impact on the project development and will take time to recover from.

Red	High/Great Risk	These risks have a significant impact on the project completion schedule and can lead to missed deadlines or project development failures. These risks persist throughout the development process.
-----	-----------------	--

This is a diagram of the task corresponding to the impact and likelihood.

Impact ->	1	2	3	4	5
Likelihood					
1		4			1,2,3,11
2			10	8	7,9
3				5,6	
4					
5					

5.3 Professional Issues

5.3.1 Legal issues

law is the sum of various codes of conduct formulated or recognized by the national legislature and guaranteed by the national force. In legal terms, face mask recognition technology has privacy and data protection implications. The use of such technology to collect and process facial images and other biometric data of individuals may infringe on the privacy of individuals. It is therefore important to ensure that relevant privacy legislation is complied with when using this technology.

Privacy: Face mask recognition technology requires the collection and processing of personal facial feature data, which may infringe on the privacy and personal information rights of individuals. Therefore, the use of this technology requires compliance with relevant privacy laws and regulations, such as the General Data Protection Regulation (GDPR) of the European Union and the Privacy Act of the United States. In addition, measures need to be taken to ensure the security and confidentiality of personal data, such as encryption, de-identification.

Discrimination issues: Face mask recognition technology may have a discriminatory effect on certain groups of people, such as African-Americans and women, who are relatively less discriminated against. Therefore, when using this technology, there is a need to ensure that the technology is fair and equal to avoid any discriminatory results.

Legal responsibility: When using face-mask recognition technology, there is a need to ensure compliance with relevant laws, regulations and standards. There may be financial and legal liabilities for individuals and organizations in the event of technical failures or errors. Therefore, when using this technology, relevant legal liability mechanisms need to be put in place, with appropriate risk management and contingency plans in place.

Data security: Face mask recognition technology requires the collection and processing of large amounts of personal data, and therefore measures need to be put in place to ensure the security and protection of the data. This may involve aspects such as data encryption, data backup, data security audits and data access controls. When using this technology, data security compliance needs to be ensured and relevant data protection laws and regulations, such as the EU's General Data Protection Regulation (GDPR) and the US Privacy Act, need to be adhered to.

5.3.2 Social issues

Making Contributions to the welfare of society and other categories, and recognize that all people are stakeholders of computers

This principle involves the quality of life of all people. It confirms that the members of the computing profession (individual and collective) have the obligation to use their skills to benefit the society, its members and the surrounding environment. This obligation includes the promotion of basic human rights and the protection of the sovereignty of each human being. The three basic goals of computing professionals are to minimize the negative consequences of computing, including threats to health, safety, personal security and privacy. When the interests of multiple groups conflict, more attention should be paid to the needs and priorities of those less favorable groups.

5.3.3 Ethical issues

Ethics refers to the principles, norms or requirements that people should follow when dealing with various social relations. Face mask recognition technology can lead to discrimination and unfair treatment. For example, because the technology may not accurately identify the facial features of certain people, it may discriminate against certain groups, such as ethnic minorities or people with disabilities. The use of such technology may exacerbate inequalities that already exist.

5.3.4 Moral issues

The responsibility to respect privacy applies to computing professionals in a particularly profound way. The technology can collect, monitor and exchange individual information quickly and cheaply, and is often not affected by the understanding of affected groups. Therefore, computing professionals should be familiar with various privacy definitions and forms, and should understand the rights and responsibilities related to the collection and use of personal information. The use of facial mask recognition technology also involves issues of personal freedom and social responsibility. Some people may believe that this technology violates their freedom, while governments or other institutions may believe that using this technology is for the overall safety and welfare of society. Therefore, it is necessary to balance individual rights and social responsibility when using this technology.

5.3.5 Environmental issues

The use of face mask recognition technology may increase the consumption of energy and resources. This technology requires significant computing and storage resources, which may result in an environmental burden. It is therefore important to consider how this technology can be used to reduce its environmental impact.

Energy consumption: Face mask recognition technology requires significant computing resources, such as high-performance servers. These devices consume large amounts of energy, resulting in the emission of carbon dioxide and other greenhouse gases. Therefore, measures should be taken to reduce energy consumption and carbon emissions in the development and use of the technology.

E-waste: The use of face-mask recognition technology may lead to an increase in e-waste. For example, a large number of electronic devices and components may be required in the development and use of the technology, which may be obsolete or become obsolete within a short period of time, resulting in a large amount of e-waste. To reduce this impact, measures should be taken to promote sustainable

electronic product design and recycling systems.

Environmental pollution: The development and use of face-mask recognition technology may generate a number of harmful chemicals and pollutants. For example, during the production of electronic equipment and components, harmful gases and waste water may be released, thus polluting the environment. Therefore, measures should be taken to reduce the emission and disposal of pollutants in the development and use of the technology.

Chapter 6 Conclusion

6.1 Reflection and conclusion

In this project, the authors look at the future of artificial intelligence programs in automated testing. With this project, the authors summarize the advantages of automated testing and develop a face recognition mask system that not only recognizes faces accurately but also improves the efficiency of detection. The authors used fly-by-wire to track the different sprint phases, which correspond to each other and help a lot in development. The low cost of this project shows that it can be extended to other areas such as deep learning or machine learning. Currently, Apple has a faster processor and better access to resources than the OpenCV library, but they also have to deal with more data. The concept of this project is also shown by the authors in the article.

Today, more and more attention is being paid to the development of artificial intelligence technology. In practical applications, automated testing and analysis techniques are being widely applied to AI programs as a way of ensuring their quality and reliability. This report concludes that, firstly, automated testing improves the efficiency of AI applications, and that automated testing circumvents the time and labor costs associated with manual testing. Automated testing and analysis can improve maintainability. When the code of an AI application changes, testing and analysis need to change accordingly. Manual testing and analysis can lead to missed test cases and errors, which can affect maintainability. Automated testing and analysis reduce these problems and makes testing and analysis easier when coding changes. After testing current AI programs through automation, the benefits were summarized and eventually applied to face mask recognition. After continuous testing and optimization, the mask recognition application was perfectly developed and applied. Overall, automated testing offers many benefits for AI, including increased accuracy and efficiency, reduced costs, and improved user experience. In

the future, look forward to the development of more sophisticated testing and analysis methods that will lead to more reliable and efficient AI programs.

6.2 Future Work

Future work in face mask recognition by more advanced Raspberry Pi or using deep learning knowledge to re-update my project by more complex demonstrations to explore the recognition of multiple faces, data processing of multiple photos and spending time developing and researching some more applicable algorithms, exploring and researching different model architectures to improve the quality of face detection images. If the project is longer, then the next step of the project is to use TF(TensorFlow)or Yolov5 to process the image dataset, train the dataset, and predict the value of the loss. In terms of automated testing, add multiple testing methods and update new algorithms in order to reduce the time and space complexity of the code.

6. Reference

- [1] Gupta, M., Fu, J., Bastani, F. B., Khan, L. R., & Yen, I. L. (2007). Rapid goal-oriented automated software testing using MEA-graph planning. *Software Quality Journal*, 15(3), 241–263. <https://doi.org/10.1007/s11219-007-9018-3>
- [2] de Filippo, A., Borghesi, A., Boscarino, A., & Milano, M. (2022). HADA: An automated tool for hardware dimensioning of AI applications. *Knowledge-Based Systems*, 251. <https://doi.org/10.1016/j.knosys.2022.109199>
- [3] Solangi, U. S., Ibtesam, M., Ansari, M. A., Kim, J., & Park, S. (2021). Test Architecture for Systolic Array of Edge-Based AI Accelerator. *IEEE Access*, 9, 96700–96710. <https://doi.org/10.1109/ACCESS.2021.3094741>
- [4] Lavin, A., Gilligan-Lee, C. M., Visnjic, A., Ganju, S., Newman, D., Ganguly, S., Lange, D., Baydin, A. G., Sharma, A., Gibson, A., Zheng, S., Xing, E. P., Mattmann, C., Parr, J., & Gal, Y. (2022). Technology readiness levels for machine learning systems. *Nature Communications*, 13(1). <https://doi.org/10.1038/s41467-022-33128-9>.
- [5] Xu, D., Xu, W., Bavikati, B. K., & Wong, W. E. (2012). Mining executable specifications of web applications from Selenium IDE tests. *Proceedings of the 2012 IEEE 6th International Conference on Software Security and Reliability, SERE 2012*, 263–272. <https://doi.org/10.1109/SERE.2012.39>
- [6] Haldar, S., Vijaykeerthy, D., & Saha, D. (2021). *Automated Testing of AI Models*. <http://arxiv.org/abs/2110.03320>.
- [7] M. Leotta, A. Stocco, F. Ricca and P. Tonella, "Reducing Web Test Cases Aging by Means of Robust XPath Locators," *2014 IEEE International Symposium on Software Reliability Engineering Workshops*, 2014, pp. 449-454, doi: 10.1109/ISSREW.2014.17.

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6983884&isnumber=6983760>

[8] I. Altaf, J. A. Dar, F. u. Rashid and M. Rafiq, "Survey on selenium tool in software testing," 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), 2015, pp. 1378-1383, doi: 10.1109/ICGCIoT.2015.7380682.

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6915770&isnumber=7051309>

9 Lal, A., & Kumar, G. (2021). Intelligent Testing in Software Industry. 2021 12th International Conference on Computing Communication and Networking Technologies, ICCCNT 2021. <https://doi.org/10.1109/ICCCNT51525.2021.9580012>

10. M. A. Umar and C. Zhanfang, "A Study of Automated Software Testing: Automation Tools and Frameworks", *Int. J. Comput. Sci. Eng.*, vol. 8, no. 06, pp. 217-225, 2019.

11. Kumar, D., & Mishra, K. K. (2016). The Impacts of Test Automation on Software's Cost, Quality and Time to Market. *Procedia Computer Science*, 79, 8–15. <https://doi.org/10.1016/j.procs.2016.03.003>.

12. *Selenium IDE. Open source record and playback test automation for the web*, May 2021, [online] Available: <https://www.selenium.dev/selenium-ide/>.

13 Nguyen, V., To, T., & Diep, G. H. (2021). Generating and selecting resilient and maintainable locators for Web automated testing. *Software Testing Verification and Reliability*, 31(3). <https://doi.org/10.1002/stvr.1760>.

14. A. Holmes and M. Kellogg, "Automating functional tests using Selenium," AGILE 2006 (AGILE'06), 2006, pp. 6 pp.-275, doi: 10.1109/AGILE.2006.19.

15. Leotta, M., Clerissi, D., Ricca, F., & Spadaro, C. (2013). Repairing selenium test cases: An industrial case study about web page element localization. *Proceedings - IEEE 6th International Conference on Software Testing, Verification and Validation*,

ICST 2013, 487–488. <https://doi.org/10.1109/ICST.2013.73>

16. Gojare, S., Joshi, R., & Gaigaware, D. (2015). Analysis and design of selenium webdriver automation testing framework. *Procedia Computer Science*, 50, 341–346.

<https://doi.org/10.1016/j.procs.2015.04.038>

17. Sunman, N., Soydan, Y., & Sözer, H. (2022). Automated Web application testing driven by pre-recorded test cases. *Journal of Systems and Software*, 193.

<https://doi.org/10.1016/j.jss.2022.111441>

18. Hammoudi, M., Rothermel, G., & Tonella, P. (2016). Why do Record/Replay Tests of Web Applications Break? *Proceedings - 2016 IEEE International Conference on Software Testing, Verification and Validation, ICST 2016*, 180–190.

<https://doi.org/10.1109/ICST.2016.16>

19. Leotta, M., Clerissi, D., Ricca, F., & Tonella, P. (2013). Capture-replay vs. programmable web testing: An empirical assessment during test case evolution. *Proceedings - Working Conference on Reverse Engineering, WCRE*, 272–281.

<https://doi.org/10.1109/WCRE.2013.6671302>

20. S. R. Choudhary, D. Zhao, H. Versee, and A. Orso. Water: Web application test repair. In *Proceedings of the 1st International Workshop on End-To-End Test Script Engineering, ETSE 2011*, pages 24-29. ACM, 2011.

21. E. Collins and V. de Lucena. Software test automation practices in agile development environment: An industry experience report. In *Proceedings of the 7th International Workshop on Automation of Software Test, AST 2012*, pages 57-63. IEEE, 2012.

6.3 Formatting Requirements

6.4 Written Presentation

Appendices