# Introduction

At present, people's demand for sports and fitness is increasing, and the application trend of health sports combined with high technology, online and big data is obvious. According to the Research report on China's Sports and Fitness Market in the third quarter of 2020, under the influence of the epidemic, residents' fitness and health issues have become a national social topic.So we decided to build a motion counting system to monitor our exercise.

# Test summary

## System Introduction

Our system is a real-time fitness counting system, which can automatically tally user data or verify the quality of exercise performed and make real-time counts. Our system can count some exercises that do not require equipment, such as Push-up. The count can be done in real time by holding a phone in front of a portrait, making sure the whole body is visible and unshaded.

## Test environment

|  |  |
| --- | --- |
| content | Specific instructions |
| cpu | 1.6 GHz 双核Intel Core i5 |
| os | macOS |
| Test software | Jupyter Notebook |
| Test environment | python3.9 |
| test dependencies | mediapipe/opencv-python/numpy/unittest |
| client | Safari browser |

## Testing process

|  |
| --- |
| Test case 1: |
| Image key point acquisition test |
| Testing purposes: |
| Whether the test system can normally obtain the joint point information of the input image |
| Test Conditions: |
| The environment required for installation and testing: opencv-python, mediapipe, tqdm, time, matplotlib, unittest |
| Testing process:  Import the toolkit, import the model, read in the image, input the model, obtain the prediction result, and obtain the pixel coordinates of key points |
| Desired result:  Can get 33 key points of the input image and output the coordinates of the key points |
| Test Results:  fail |
| Reason for failure:  The image format read by opencv is BGR, and the visualization format of matplotlib is RGB, so you need to convert BGR to RGB |
| Test date: May 23, 2023 Testers：summer |

Test script:：



test\_negative\_numbers (\_\_main\_\_.Test\_Pose) ... Fail

test\_positive\_numbers (\_\_main\_\_.Test\_Pose) ... OK

=================================================================

ERROR: test\_negative\_numbers (\_\_main\_\_.Test\_LookImg)

----------------------------------------------------------------------

Traceback (most recent call last):

File"/var/folders/rg/y2cj7c0d381gv6x04106261r0000gn/T/ipykernel\_30525/3102635351.py", line 5, in test\_negative\_numbers

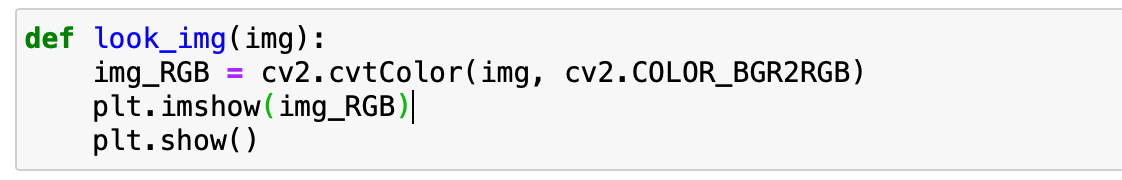
self.assertEqual(look\_img(cv2.imread('person1.jfif')))

TypeError: assertEqual() missing 1 required positional argument: 'second'

----------------------------------------------------------------------

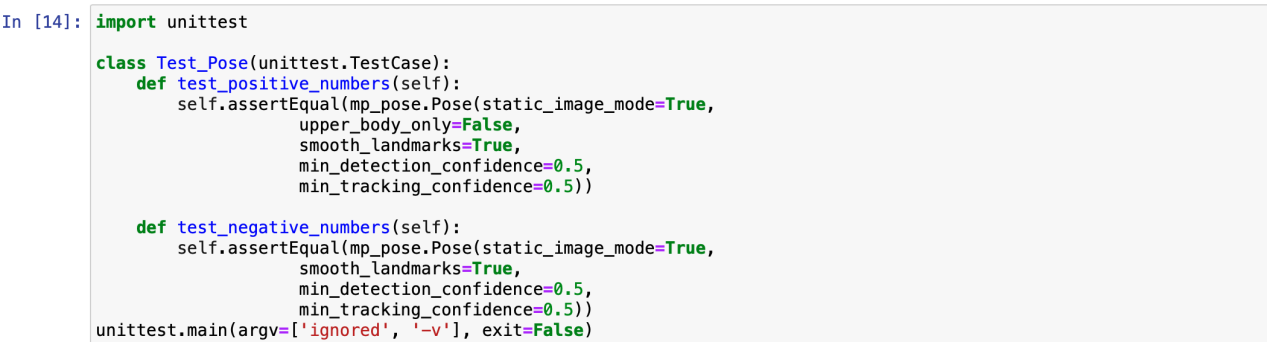
Ran 4 tests in 1.697s

modified script：



|  |
| --- |
| Test case 2:  Image key point acquisition test |
| Testing purposes:  Whether the test system can normally obtain the joint point information of the input image |
| Test Conditions:  The environment required for installation and testing: opencv-python, mediapipe, tqdm, time, matplotlib, unittest |
| Testing process:  Import the toolkit, import the model, read in the image, input the model, obtain the prediction result, and obtain the pixel coordinates of key points |
| Desired result:  Can get 33 key points of the input image and output the coordinates of the key points |
| Test result: failed |
| Reason for failure:  Different versions of mediapipe require different parameters for importing solution |
| Test date: May 23, 2023  Testers：summer |

test script：



test\_negative\_numbers (\_\_main\_\_.Test\_Pose) ... ERROR

test\_positive\_numbers (\_\_main\_\_.Test\_Pose) ... OK

=================================================================

ERROR: test\_negative\_numbers (\_\_main\_\_.Test\_LookImg)

----------------------------------------------------------------------

Traceback (most recent call last):

File"/var/folders/rg/y2cj7c0d381gv6x04106261r0000gn/T/ipykernel\_30525/3102635351.py", line 5, in test\_negative\_numbers

self.assertEqual(mp\_pose.Pose(static\_image\_mode=True,

upper\_body\_only=False,

smooth\_landmarks=True,

min\_detection\_confidence=0.5,

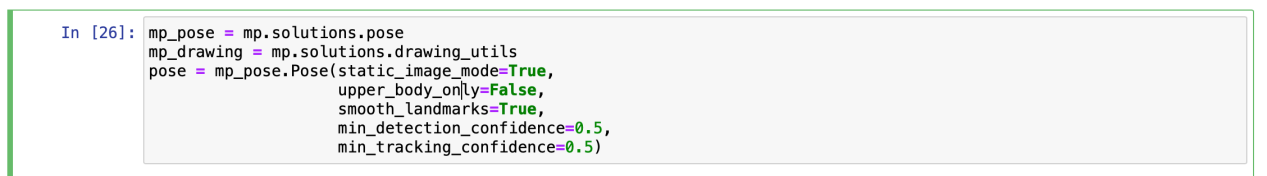
min\_tracking\_confidence=0.5))

TypeError:Pose.\_\_init\_\_() got an unexpected keyword argument 'upper\_body\_only’

----------------------------------------------------------------------

Ran 1 tests in 2.527s

modified script：



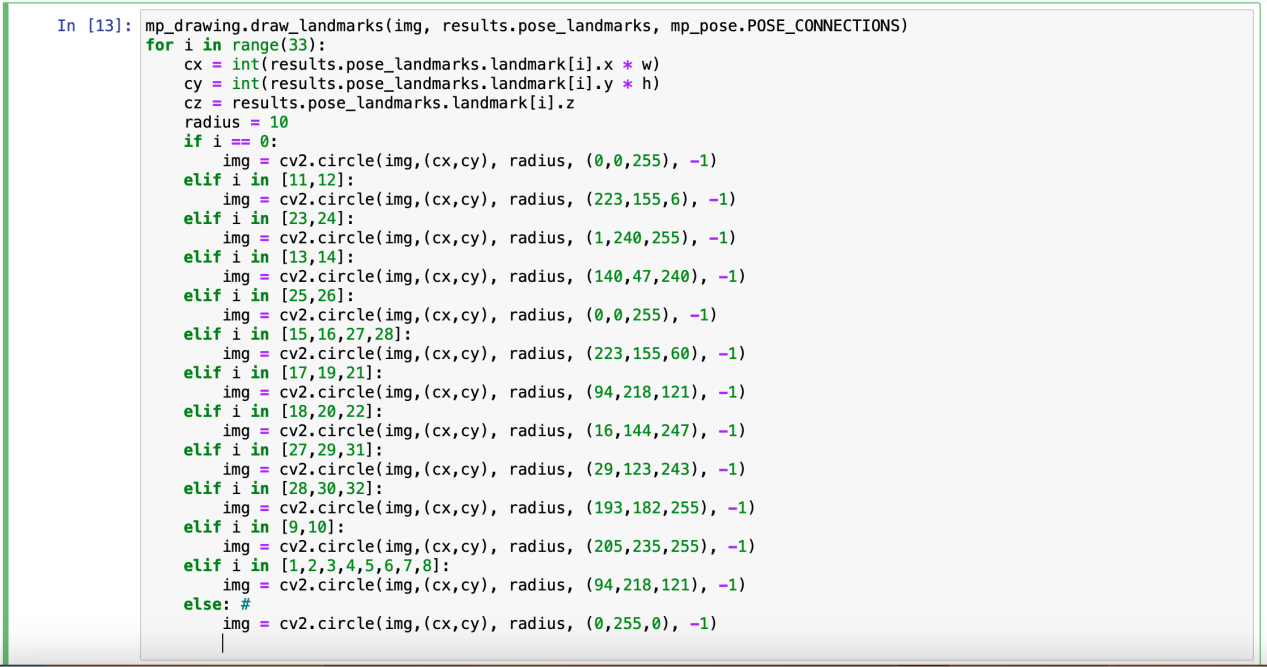
|  |
| --- |
| Test case 3:  Image key point acquisition test |
| Testing purposes:  Whether the test system can normally obtain the joint point information of the input image |
| Test Conditions:  The environment required for installation and testing: opencv-python, mediapipe, tqdm, time, matplotlib, unittest |
| Testing process:  Import the toolkit, import the model, read in the image, input the model, obtain the prediction result, and obtain the pixel coordinates of key points |
| Desired result:  Can get 33 key points of the input image and output the coordinates of the key points |
| Test result: passed, as shown in the figure |
| Test date: May 23, 2023 Tester: summer |



The key point coordinate information is shown in the figure：



Test script：



.

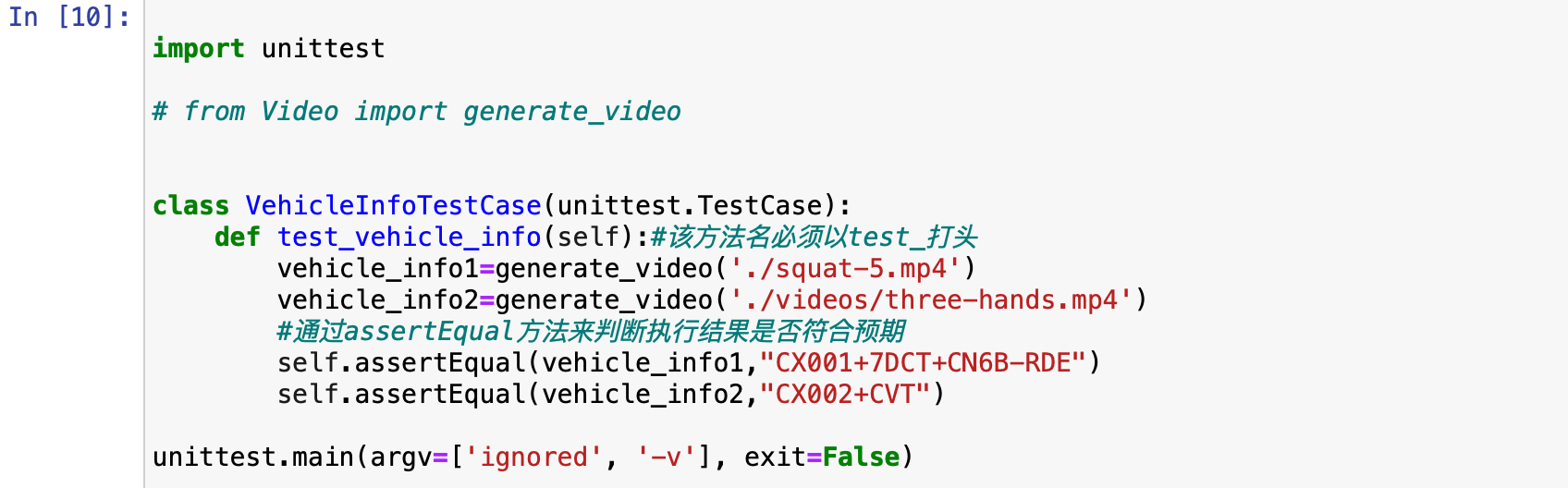
----------------------------------------------------------------------

Ran 1 test in 0.000s

OK

|  |
| --- |
| Test case 4:  Video Processing Test |
| Testing purposes:  Test whether the system can normally obtain the joint point information of the input video |
| Test Conditions:  The environment required for installation and testing: opencv-python, mediapipe, tqdm, time, unittest |
| Testing process:  Import toolkit, import model, input model, process single frame data, input video, process video frame by frame |
| Desired result:  Ability to output video with keypoint annotations |
| Test result: failed |
| Reason for failure:  There is a problem with the parameters passed in, and the video file cannot be obtained |
| Test date: May 23, 2023 Tester: summer |

Test script:





FAIL

=================================================================

FAIL: test\_vehicle\_info (\_\_main\_\_.VehicleInfoTestCase)

----------------------------------------------------------------------

Traceback (most recent call last):

File "/var/folders/rg/y2cj7c0d381gv6x04106261r0000gn/T/ipykernel\_30830/2967315508.py", line 11, in test\_vehicle\_info

self.assertEqual(vehicle\_info1,"CX001+7DCT+CN6B-RDE")

AssertionError: None != 'CX001+7DCT+CN6B-RDE'

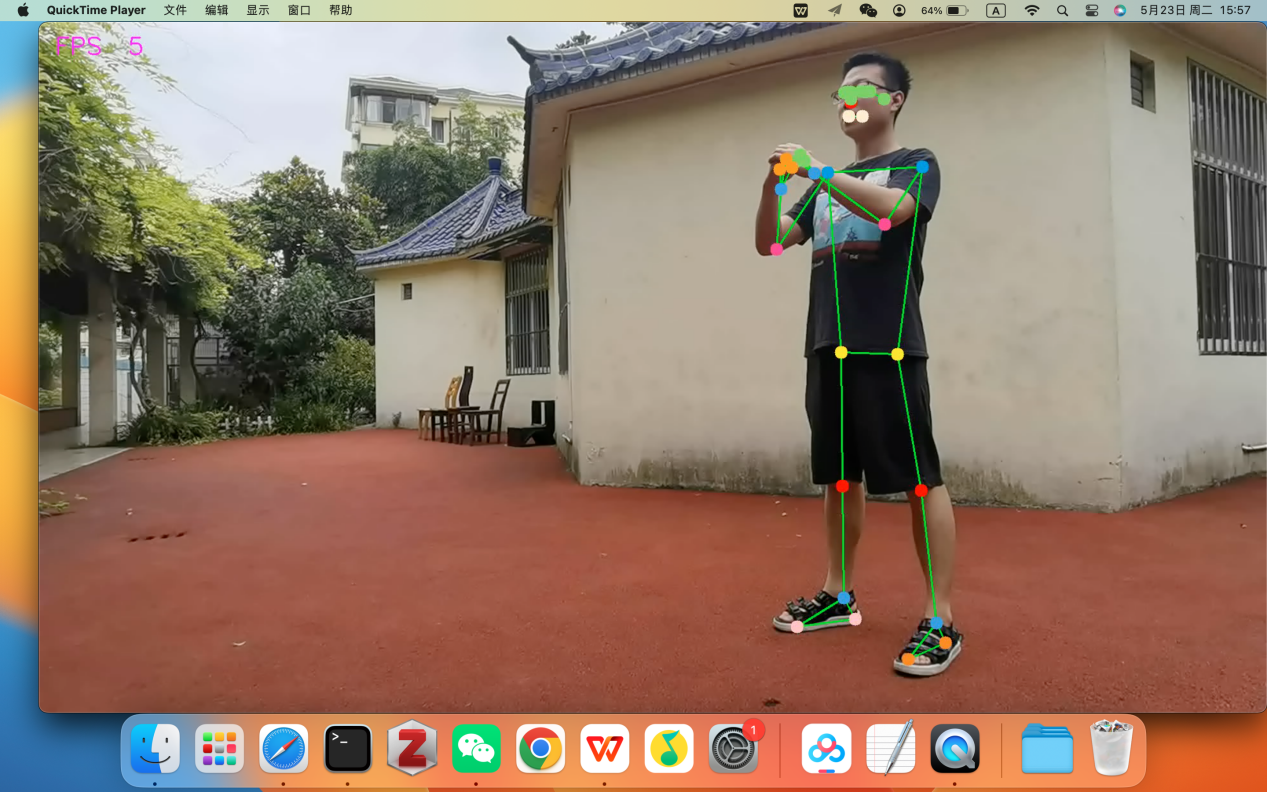
----------------------------------------------------------------------

Ran 1 test in 23.110s

FAILED (failures=1)

|  |
| --- |
| Test case 5:  Video Processing Test |
| Testing purposes:  Test whether the system can normally obtain the joint point information of the input video |
| Test Conditions:  The environment required for installation and testing: opencv-python, mediapipe, tqdm, time, unittest |
| Testing process:  Import toolkit, import model, input model, process single frame data, input video, process video frame by frame |
| Desired result:  Ability to output video with keypoint annotations |
| Test result:  passed, as shown in the figure |





Test script：



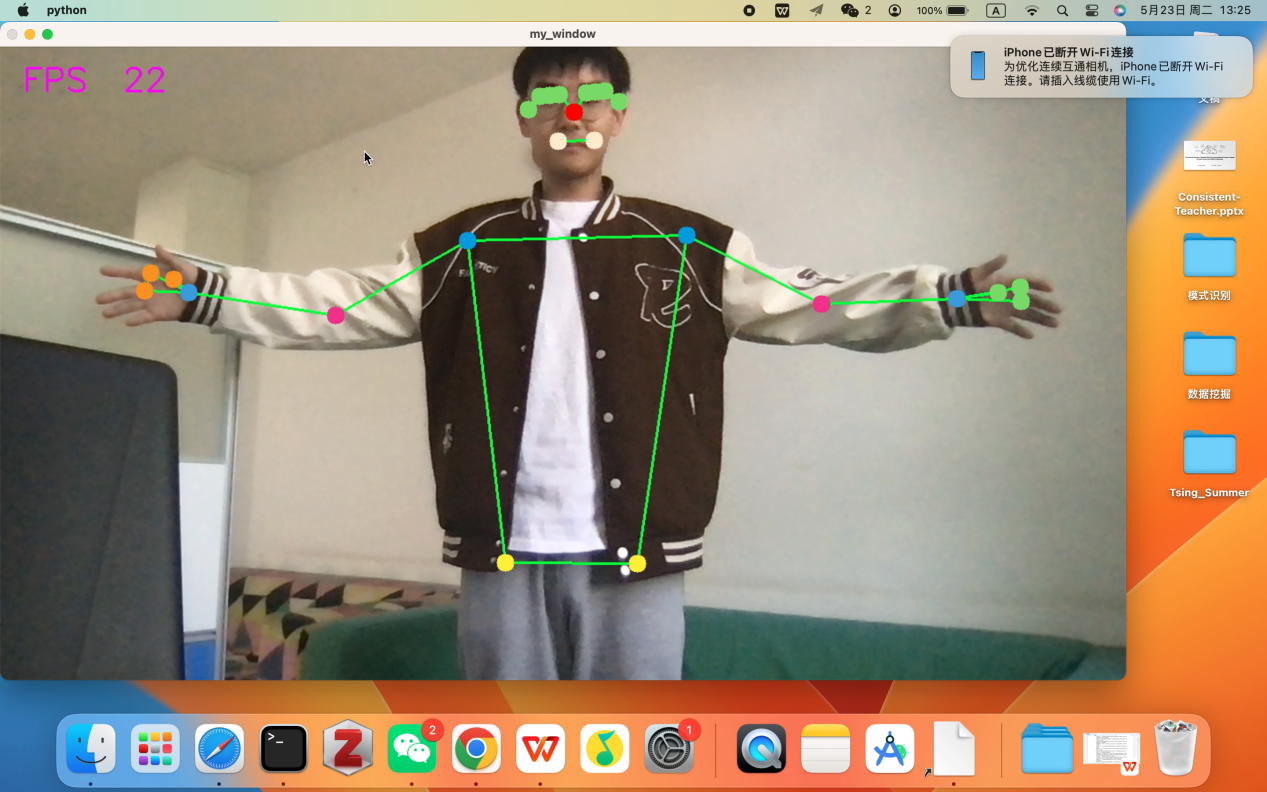
.

----------------------------------------------------------------------

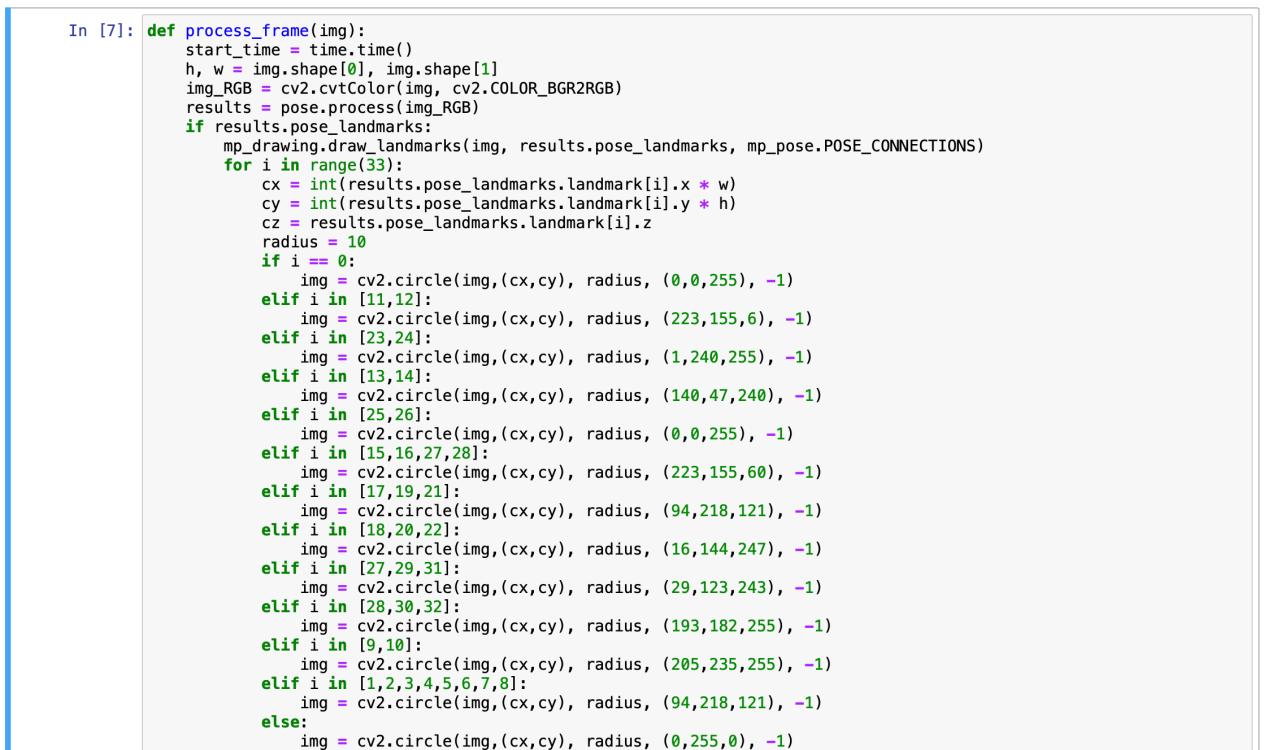
Ran 1 test in 24.540s

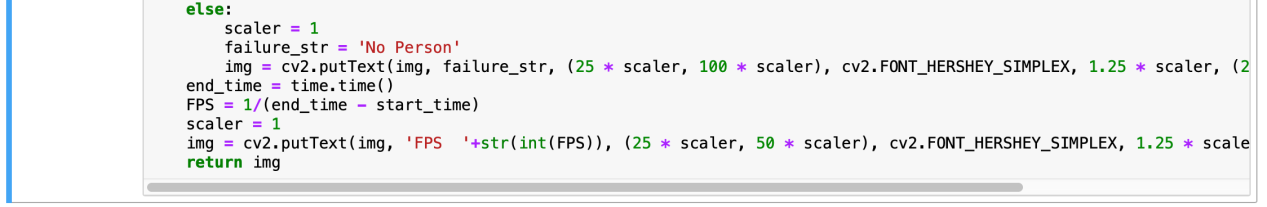
OK

|  |
| --- |
| Test case 6:  Camera real-time test |
| Testing purposes:  Test whether the system can normally obtain device camera input and detect key points |
| Test Conditions:  The environment required for installation and testing: opencv-python, mediapipe, tqdm, time |
| Testing process:  Import toolkit, import model, input model, process single frame data, call camera to get each frame data |
| Desired result:  Successfully invoked the device camera, which can output images in real time and display key point information |
| Test result: passed, as shown in the figure |



Test script：







.

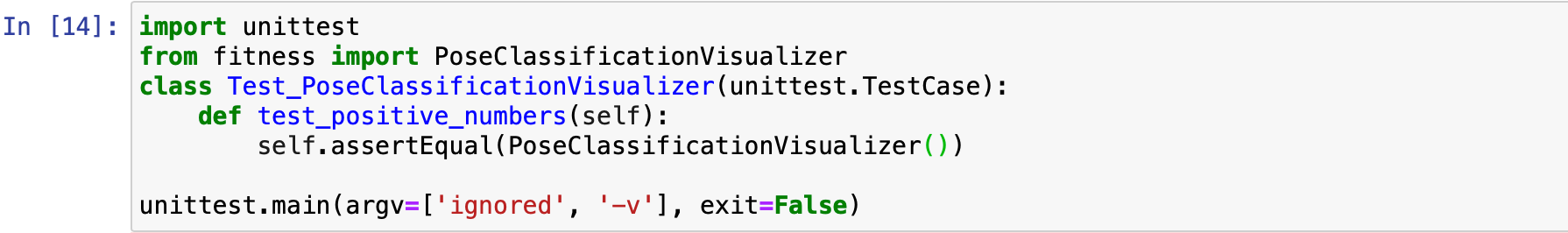
----------------------------------------------------------------------

Ran 1 test in 20.106s

OK

|  |
| --- |
| Test case 7:  Pose Classification Test |
| Testing purposes:  Whether the test system can normally classify and visualize human body poses |
| Test Conditions:  The environment required for installation and testing: opencv-python, mediapipe, tqdm, time, unittest |
| Testing process:  Build the test category and write the test code for the target function |
| Desired result:  Output OK |
| Test result: failed |

Test script:



E

=================================================================

ERROR: /Users/spinulose/Library/Jupyter/runtime/kernel-e0eb3d93-32d1-46db-b345-4ca6bbb5a455 (unittest.loader.\_FailedTest)

----------------------------------------------------------------------

MessageError: Error: credential propagation was unsuccessful

----------------------------------------------------------------------

Ran 1 test in 6.901s

FAILED (errors=1)

1. **Overall Evaluation Conclusion**

From May 18 to May 25, 2023, it lasted for 7 days in total. 3 functions were tested, 7 test cases were executed, 3 bugs were found in the test, and they were corrected. This test is tested from multiple dimensions such as requirement compliance, functional correctness, performance indicators, operational stability, interconnection, documentation, usability, maintainability, compatibility, and risk assessment, all of which meet the requirements.

This version meets the online conditions and can be used online;

This version has a quality risk and can only be used with restrictions on the Android side.