Testing Reports

Revision History:

|  |  |  |
| --- | --- | --- |
| Date | Author | Description |
| 16/10/2020 | Huining Wang | write the Testing Case |
| 20/10/2020 | Xi Chen | write the Testing Plan |
| 22/10/2020 | Guangpeng Li | update Algorithm Result |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Contents

[1. Introduction 3](#_Toc5637)

[1.1. Intended Audience and Purpose 3](#_Toc2865)

[1.2.How to use the document 3](#_Toc18849)

[2. Testing Cases 3](#_Toc3813)

[2.1. Algorithm for Target Detection and the Angle of Cobb 3](#_Toc28174)

[3. Testing Plan 3](#_Toc29099)

[3.1. The user wants to get the Cobb Angle of the spine 3](#_Toc14780)

[4. Testing Results 4](#_Toc1598)

[5. Quality Requirements 4](#_Toc6726)

[5.1. Algorithm response time 4](#_Toc31349)

## Introduction

## Intended Audience and Purpose

This document provides the testing method and results, corresponding to the requirement from the customer. It consists of 3 parts, the testing cases, the test plan, and the testing results.

## 1.2.How to use the document

You may refer to the content section for the structure of the document, in which Sec. Testing Cases collect the unit and module test information from Popcorn Algorithm team; Sec. Testing Plan shows the steps and expected results of the integration test; Sec. Results describes the real world data out of the test, and the correspondence to the requirements.

## Testing Cases

In this section, my team propose our testing cases on unit and module testing.

## Algorithm for Target Detection and the Angle of Cobb

The algorithm targets X-ray images sent from the server, marks the spine, and calculates the Cobb Angle.The algorithm is divided into two parts: the first part is to identify the location of the spine in the image based on the deep learning model; the second part is to calculate the Cobb Angle in the image.

The first table is the test case for the first part. It mainly tests the correctness and efficiency of model recognition

|  |  |  |  |
| --- | --- | --- | --- |
| Function point name | Description | Input | Expected output |
| 1.A spinal image | Test whether the model recognizes the location of the spine | A spinal image | The coordinates of the spine in this image |
| 2.A Non-spinal image | Test whether the model recognizes the location of the spine | A Non - spinal image | The coordinates of the items in the image |
| 3.large quantities of images of the spine | Test whether the model can handle large quantities of images | 100 images of the spine | The coordinates of the spine in each image |
| 4.Test model running speed | The computational model identifies the average running speed of a image | 100 images of the spine | The model identifies the average running time of an image |

The second table is the test case for the second part. It mainly tests the accuracy of Cobb Angle calculation.

|  |  |  |  |
| --- | --- | --- | --- |
| Function point name | Description | Input | Expected output |
| 5.Test the Cobb Angle calculation with a spinal image | Calculate the Cobb Angle of a spinal image | The coordinates of the spine in the test image | Cobb Angle within a reasonable range |
| 6.Test the Cobb Angle calculation with a Non-spinal image | Calculate the Cobb Angle of a Non-spinal image | The coordinates of the items in the image | An invalid number |
| 7.Test the accuracy of Cobb Angle calculation | Test with large quantities of images of the spine | large quantities of images calculated by the doctor | the accuracy of Cobb Angle calculation |

## Testing Plan

Here comes the complete testing plan for integration, referring to the workflows in the system design document.

## The user wants to get the Cobb Angle of the spine

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| WorkFlow name | Corresponding module | Input | Expected output | Result |
| Upload a image | Client |  |  |  |
| Process the image | Server |  |  |  |
| receive the image from the server | Algorithm | The image from the server | The deep learning model works |  |
| Identify the spine in the image | Algorithm | The image from the server | Coordinates of the spine | Coordinates of the spine with noise |
| Calculate the Cobb Angle | Algorithm | Coordinates of the spine | Cobb Angle of the spine | Calculate the Cobb Angle |
| return the value | Server | Cobb Angle of the spine | make a recommendation | make a recommendation based on the Cobb Angle |
| return the value | Client | Cobb Angle of the spine |  | The client receives the result |

## Testing Results

The results of the integration are listed here and you may find the correspondence to the requirements in the requirement analysist document.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case No. | Module | Result | Corresponding Requirement |
| 2.1.1 | Algorithm | 704 740 1034 1246  598 633 785 998  492 882 998 1140  562 421 821 537  562 527 821 643 | The coordinates of the spine |
| 2.1.2 | Algorithm | 562 66 927 253  598 314 714 395  30 846 147 927  172 562 466 785 | The coordinates of other items |
| 2.1.3 | Algorithm | Successfully obtain the spinal coordinates on each image. | The coordinates of the spine in each image |
| 2.1.4 | Algorithm | Return value:2.7978s | The model identifies the average running time of an image |
| 2.1.5 | Algorithm | return value:34 | Computing Angle |
| 2.1.6 | Algorithm | ValueError:  Negative Numbers or Numbers greater than 90 are invalid | Computing Angle |
| 2.1.7 | Algorithm | There is no result yet | The accuracy of Cobb Angle |

## Quality Requirements

## System response time

Testing plan: Writing a python script that generates 100 requests at the same time. And calculate the average time.

Testing results: It hasn't been tested yet.