System Verification and Validation Plan for Retinal Vessel Segmentation System (RVSS)

Xinyu Ma

February 19, 2024

1 Revision History

Date	Version	Notes
02/16/2024	1.0	Initial Release
02/19/2024	2.0	Updated based on feedback from the presentation

Contents

List of Tables

2 Symbols, Abbreviations and Acronyms

symbol	description
CNN	Convolution Neural Network
MG	Module Guide
MIS	Module Interface Specification
SRS	Software Requirement Specification
RVSS	Retinal Vessel Segmentation System
TC	Test Case
VnV	Verification and Validation

For complete symbols, abbreviations and acronyms used within the system, please refer the section 1.3 in SRS document.

This document provides an introductory blurb and roadmap of the Verification and Validation plan for the Retinal Vessel Segmentation System (RVSS). It is designed to ensure that the system meets its specified requirements and fulfills its intended goals effectively (requirements and goals can be found in the SRS document).The organization of this document starts with the General Information about the RVSS in section 3. In section 4, it describes a verification and validation plan. A system test description is provided in section 5, which contains tests for functional and nonfunctional requirements. In section 6, it provides the Unit Test Description.

3 General Information

3.1 Summary

This document introduces the validation and verification plan for the Retinal Vessel Segmentation System (RVSS). The RVSS is a medical imaging application designed to automate the process of segmenting blood vessels from fundus images. It utilizes advanced image processing and machine learning techniques, particularly Convolution Neural Network (CNN), to analyze these fundus images and accurately identify and delineate the blood vessels within the retina.

3.2 Objectives

In this section, it states what is intended to be accomplished. The objective is around the qualities that are most important for the RVSS. The following are objectives that are in scope and out of scope.

- In-Scope Objectives:
  - **Build Confidence in the Software Correctness**:To ensure that the RVSS operates according to its specifications and user requirements, accurately segmenting blood vessels from fundus images. This involves thorough testing of the segmentation algorithms for sensitivity, recall, and overall accuracy.
  - **Ensure the Software Performance**: Focus on verifying the processing speed and responsiveness to ensure that the RVSS can processes fundus images within an acceptable time without sacrificing accuracy.
- Out-of-Scope Objectives:
  - **Comprehensive Medical Usability Testing**: While comprehensive medical usability testing is valuable, it requires recruiting a large number of participants for fundus photography, which involves user privacy. Due to resource constraints, extensive medical usability testing involving diverse user groups across multiple settings would be out of scope.

– **External Libray Verification**: The RVSS relies on external libraries for some of its functionality (e.g., image processing, segmentation algorithms based on machine learning). Given the widespread use and community validation of these external libraries, we can assume that these library has already been verified by its implementation team. Thus, their in-depth verification will be out of scope.

3.3 Relevant Documentation

The relevant documentation for the RVSS includes Problem Statement which outlines the specific problem that the RVSS aims to address, System Requirements Specifications which clearly define the requirements of the system, and design documents, like MG, MIS, etc(found in Github Repository).

The problem statement document defines the scope of the problem, the needs of the stakeholders, and the goals that the RVSS seeks to achieve. It guides the development of the verification and validation plan and ensures that the verification and validation processes are aligned with the needs and requirements of the users and stakeholders. The System Requirements Specifications (SRS) document lays out the detailed requirements of the system, including both functional and non-functional requirements. The verification and validation plan directly reference the SRS document to ensure that each specified requirement is implemented correctly in the system. The verification and validation plan refers to design documents (like MG, MIS, etc) to verify that the architecture of the RVSS and each design unit have been correctly realized in the implemented system.

4 Plan

This section describes the testing plan for the Retinal Vessel Segmentation System (RVSS). It starts with the verification and validation team in section 4.1, followed by the SRS verification plan (section 4.2), design verification plan (section 4.3), implementation verification plan (section 4.5), Automated testing and verification tools (section 4.6), and Software validation plan (section 4.7).

4.1 Verification and Validation Team

The Verification and Validation Team includes 6 people who plays an important role in ensuring that the developed system meets all specified requirements and is fit for its intended use. The members of Verification and Validation plan is shown in Table 1.

Name	Role	Document	Responsibilities
Dr. Spencer Smith	Instructor/ Reviewer	SRS, VnV plan, MG + MIS	Review all documents.
Xinyu Ma	Author	SRS, VnV plan, MG + MIS	Create and manage all documents; design, develop and execute test cases.
Kim Ying Wong	Domain Expert Reviewer-Primary Reviewer	SRS, VnV plan, MG + MIS	Review all documents.
Morteza Mirzaei	Secondary Reviewer	SRS	Review the SRS document.
Nada Elmasry	Secondary Reviewer	VnV Plan	Review the VnV plan document.
Seyed Ali Mousavi	Secondary Reviewer	MG + MIS	Review the MG and MIS document.

Table 1: Verification and validation team

4.2 SRS Verification Plan

The purpose of the SRS Verification Plan is to ensure the SRS document for the RVSS is complete, accurate and meets all necessary standards and requirements. The SRS document shall be verified in the following way:

1. Conduct peer reviews by involving assigned team members (Dr. Spencer Smith, Kim Ying Wong, and Morteza Mirzaei) not directly involved in writing the SRS. This helps identify ambiguities, inconsistencies, and missing requirements. For this, the review can be performed referring the following simplified SRS checklist (The checklist refers to the SRS checklist designed by Dr. Smith).

   SRS checklist:

   - Overall qualities of documentation
     - ☐ No statement is repeated at the same level of abstraction.
     - ☐ SRS is unambiguous. At least check a representative sample.
     - ☐ SRS is consistent. At least check a representative sample.
   - Introduction
     - ☐ The purpose of the system is clearly defined.
     - ☐ Introductory blurb focuses on the problem domain.
     - ☐ The characteristics of the intended readers and reading suggestions are included and unambiguous.

- General System Description
  - ☐ System context includes a figure showing the relation between the software system and external entities.
  - ☐ System constraints are clearly outlined.
  - ☐ User characteristics are specific.
- Problem Description
  - ☐ The definition of terminology is clear.
  - ☐ The goals are written abstractly, with a minimal amount of technical language and are understandable by non-domain experts.
- Functional Requirements
  - ☐ All functionalities are described in detail.
  - ☐ Each requirement is validatable and uniquely identifiable.
- Nonfunctional Requirements
  - ☐ The nonfunctional requirements are unambiguous, abstract and verifiable.

2. Reviewers give feedback by creating issues in Github.

3. The author is responsible to response the issues created by reviewers. Also, the author need to modify the SRS document according to these suggestions.

4.3 Design Verification Plan

The Design Verification Plan aims to ensure that the design of the RVSS meets all specified requirements and is correctly implemented before proceeding to the system development phase. The Design Verification Plan shall be verified in the following way:

1. Conduct peer reviews by involving assigned team members (Dr. Spencer Smith, Kim Ying Wong, and Seyed Ali Mousavi) not directly involved in writing the MG and MIS document. For this, the review can be performed referring the following simplified MG and MIS checklist (The checklist refers to the MG Checklist and MIS Checklist designed by Dr. Smith.).

   MG and MIS checklist:

   - Module Decomposition
     - ☐ One module one secret.
     - ☐ Level 1 of the decomposition by secrets shows: Hardware-Hiding, Behaviour-Hiding and Software Decision Hiding.
     - ☐ Behaviour-Hiding modules Includes programs are related to the requirements that specified in the SRS documents.

- ☐ The Software-Decision hiding modules based on mathematical theorems, physical facts, or programming considerations and the secrets of this module are not described in the SRS.

- MG quality
  - ☐ Follow template
  - ☐ Low coupling
  - ☐ Satisfies information hiding
- MIS Quality inspection for each module
  - ☐ Consistent
  - ☐ Essential
  - ☐ General
  - ☐ Implementation independent
  - ☐ Minimal
  - ☐ High cohesion
  - ☐ Opaque (information hiding)
- MIS Completeness
  - ☐ All types introduced in the spec are defined somewhere
  - ☐ All modules in MG are in the MIS
  - ☐ All required sections of the template are present for all modules

2. Reviewers give feedback by creating issues in Github.

3. The author is responsible to response the issues created by reviewers. Also, the author need to modify the MG and MIS document according to these suggestions.

4.4 Verification and Validation Plan Verification Plan

The Verification and Validation Plan shall be verified in the following way:

1. Conduct peer reviews by involving assigned team members (Dr. Spencer Smith, Kim Ying Wong, and Nada Elmasry) not directly involved in writing the VnV plan document. For this, the review can be performed referring the following simplified MG and MIS checklist (The checklist refers to the VnV Checklist defined by Dr. Smith.).

   VnV Plan Checklist:

   - Overall qualities of documentation
     - ☐ Test cases include SPECIFIC input and EXPLICIT output.
     - ☐ Plans to quantify error for scalar values using relative error.
     - ☐ Plans to quantify error for vector and matrix values using a norm of an error vector (matrix)

- ☐ Plans are feasible (can be accomplished with resources available).
- ☐ Specific unit testing framework is given.
- ☐ Specific performance measuring tools listed (like Valgrind), if appropriate.
- ☐ Traceability between test cases and requirements is summarized (likely in a table).

2. Reviewers give feedback by creating issues in Github.

3. The author is responsible to response the issues created by reviewers. Also, the author need to modify the VnV plan document according to these suggestions.

4.5 Implementation Verification Plan

The Implementation Verification Plan outlines the approaches for ensuring that the RVSS is implemented correctly and meets all functional and nonfunctional requirements. The plan includes dynamic testing strategies and static verification of the implementation.

- Dynamic testing for the RVSS:
  - Test Cases: Test cases for all mentioned tests in section 5 will be carried out. These test cases are designed based on the functional and nonfunctional requirements listed in the SRS document.
  - Unit Testing Plan: Break down the RVSS into individual units that can be tested independently. For instance, preprocessing images, segmentation training algorithms and image loading. For each unit identified, define a set of test cases that cover all possible scenarios, including typical use cases (test the unit with valid inputs where normal operation is expected), edge cases (test the limits of the unit, such as maximum and minimum input values or stress conditions), and error conditions (test the unit with invalid inputs or conditions to ensure it handles errors gracefully).

- Static testing for the RVSS:
  - Code Walkthrough: The objective of Code Walkthrough is to manually examine the source code for logical errors, adherence to coding standards, and alignment with design specifications. This process will be performed by the author and the domain expert. Scheduled code walkthrough sessions will be conducted by the author, with participation from the development team members.
  - Code Inspection: The objective of code inspection to conduct a more formal review of the codebase to identify defects and potential improvements. The development team members will inspect the code against a predefined checklist and summarize the findings in a report.

4.6 Automated Testing and Verification Tools

Automated testing of the RVSS is conducted using the Pytest library in Python. These tests are performed by predetermining user inputs and comparing them with expected values. Pytest is used to facilitate writing and running unit tests. It is chosen for its simplicity, extensive plugin ecosystem, and strong community support.

Flake8 is used to enforces coding standards and identifies stylistic errors, making the code more consistent and easier to maintain. Continuous Integration will not be used.

4.7 Software Validation Plan

Software validation plan is beyond the scope for the RVSS system as we do not have external data for the validation of the system behavior. For external data, it studies more on the issue of model generalization, which is not the scope of our consideration.

5 System Test Description

5.1 Tests for Functional Requirements

Functional requirements for the RVSS are given the in SRS document section 5.1, which clearly defines the essential functionalities of the system and operations in processing fundus images for vessel segmentation. The tests for these functional requirements are designed to verify that each aspect of the system operates as intended, ensuring its accuracy, reliability, and usability of the system during use. There are four functional requirements for the RVSS (as shown in Figure 1), R1 and R2 are related to the image preprocessing stage, R3 is corresponding to the vessel segmentation stage, while R4 is associated with the report generation stage. Section 5.1.1 describes the image preprocessing stage related to R1 and R2, section 5.1.2 describes the vessel segmentation for R3, and section 5.1.3 describes the report generation for R4.

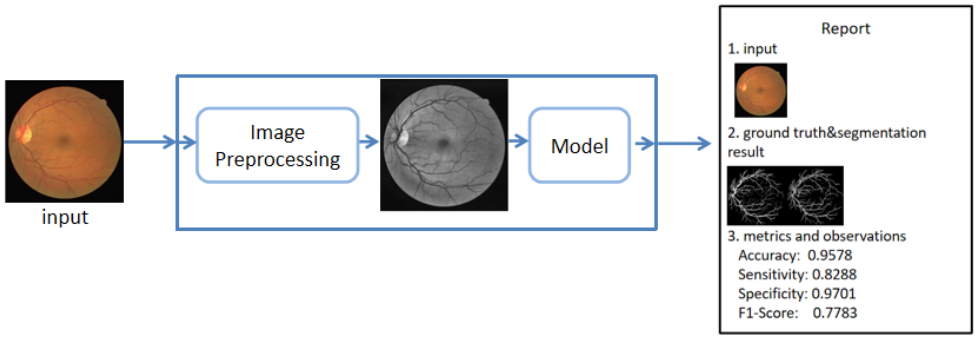

Figure 1: Overview of RVSS

5.1.1 Image Preprocessing Test

During the acquisition process of fundus images, they are often affected by external conditions, causing uneven grayscale distribution in the acquired images. In addition, the contrast between the retinal image blood vessels and the background is low, making the retinal blood vessels difficult to detect (as shown in Figure 2). The Image Preprocessing of the RVSS is important for ensuring that input fundus images are optimized for the subsequent vessel segmentation process. Effective image preprocessing improves the accuracy of the segmentation results by enhancing image quality and standardizing image features. The tests designed for this stage are tailored to verify that the preprocessing methods meet the functional requirements as outlined in the SRS document.

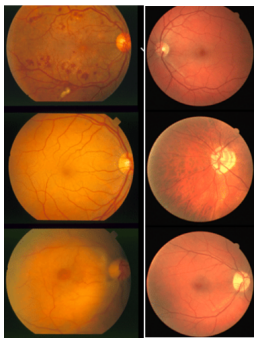

Figure 2: Fundus images under different conditions.

1. test-image-preprocessing-id1: Image Upload and Format Validation
   Control: Automatic

   Initial State: The system is ready to receive fundus images as input.

   Input: Fundus images in various formats (JPEG, PNG, TIFF and an unsupported format, e.g., GIF), as shown in Table 2

   Output: Either give an appropriate message shows that the fundus image has been uploaded successfully and is accepted by the system (the input image is in JPEG, PNG or TIFF format) or give an error message shows that the image is rejected by the system (the input image is in an unsupported format.)

   Test Case Derivation: This test case is to test the behaviour of the system when the system is supplied with inputs for area that are the original fundus images. Based on the SRS document, the RVSS should accept commonly used and medically relevant image formats (JPEG, PNG, or TIFF). The rejection of unsupported formats ensures the system's processing capabilities are focused on compatible and clinically valuable fundus image data. In test cases TC-IMAGE-1-1 to TC-IMAGE-1-3, the system produces the Uploaded successfully message, and in test case TC-IMAGE-1-4, the system produces the Unsupported format message since it is an invalid input.

ID	Input	Output
TC-IMAGE-1-1	01_test.tif	Uploaded successfully
TC-IMAGE-1-2	02_test.png	Uploaded successfully
TC-IMAGE-1-3	03_test.jpeg	Uploaded successfully
TC-IMAGE-1-4	04_test.gif	Unsupported format

Table 2: TC-IMAGE-1 - Image format input constraints tests

How test will be performed: The automation script will attempt to upload images in each specified format (JPEG, PNG, and GIF format). The system responds to each uploaded image and compares it with the expected results.

2. test-image-preprocessing-id2: Image Quality Preprocessing
   Control: Automatic

   Initial State: Fundus images in supported formats have uploaded successfully and are ready for preprocessing.

   Input: Fundus images that vary in brightness, contrast, and presence of noise.

   Output: Preprocessed images that exhibit standardized brightness and contrast levels appropriate for segmentation, with reduced noise while retaining critical details like vessel borders.

   Test Case Derivation: This test case is to test the behaviour of the system when the system is supplied with inputs for area that are the original fundus images. This test case is to test the behaviour of the system on improving the quality of the raw retinal image. The expectation that preprocessing enhances image qualities conducive to segmentation is derived from the requirement for RVSS to optimize images for analysis without losing detail. This is important for ensuring that subsequent segmentation accurately identifies blood vessels.

   How test will be performed: A set of raw fundus images will be processed through the system's preprocessing function. The output will be analyzed using image analysis software or simply observe to quantify changes in brightness, contrast, and noise levels.

5.1.2 Vessel Segmentation Test

The vessel segmentation component is the core of the RVSS, tasked with accurately identifying and delineating the blood vessels within fundus images. The tests designed for this stage are tailored to verify that the segmentation methods meet the functional requirements as outlined in the SRS document.

1. test-vessel-segmentation-id3: Segmentation Image
   Control: Automatic

   Initial State: The system has loaded the pre-trained segmentation model and is ready to receive retinal images.

   Input: Network pre-trained model parameters obtained from different rounds of training, fundus images and corresponding ground truth images with manually annotated blood vessels.

   Output: Segmented images where the retinal vessels are identified and delineated.

   Test Case Derivation: This test case is to test the behaviour of the system when the system is supplied with inputs for area that are the preprocessed fundus images. Based on the SRS document, the RVSS should output corresponding segmentation images, which are marked in binary format (e.g., vessels as 1, background as 0).

   How test will be performed: The automation script will attempt to upload different network pre-trained model parameters and images into the RVSS and collecting the output segmented images.

5.1.3 Report Generation Test

The report of the RVSS includes the segmentation outcomes and metrics (e.g., accuracy, sensitivity, specificity). The tests designed for this stage are tailored to verify that the segmentation methods meet the functional requirements as outlined in the SRS document.

1. test-report-generation-id4: Report Generation
   Control: Manual

   Initial State: Segmentation analysis of various retinal images is available for report generation.

   Input: Fundus image, output of the segmentation algorithm (i.e., segmentation images which are marked in binary format), and a range of metrics (e.g., accuracy, sensitivity, specificity).

   Output: Reports that include all relevant metrics and observations from the segmentation analysis, leaving no important information unreported.

   Test Case Derivation: The need for comprehensive reporting is justified by the system's aim to provide a full overview of the segmentation analysis.

   How test will be performed: The reports will include existing and commonly used analytical criteria to ensure that all necessary information is included and clearly presented.

5.2 Tests for Nonfunctional Requirements

Nonfunctional requirements for the RVSS are given in SRS section 5.2. There are five nonfunctional requirements the RVSS.

5.2.1 Nonfunctional: Accuracy

The accuracy in the RVSS are critical for ensuring the outputs of the system are reliable and trustworthy.

1. test-id5: Accuracy Test
   Type: Automatic

   Initial State: The system has completed the segmentation of retinal vessels from the fundus image.

   Input: The segmentation result and the images of manually annotated blood vessels by experts (ground truth).

   Output: The accuracy of the segmentation result.

   How test will be performed: The test will be performed by feeding the input images into the RVSS and collecting the output segmented images. The segmentation images will be compared against the ground truth using accuracy. Automated tools will calculate these accuracy. In addition, for unlabeled data, we can manually mark it, draw corresponding blood vessel segmentation images, and then analyze the model accuracy.

5.2.2 Nonfunctional: Robustness

Given the variability in real-world fundus images, the system must maintain high segmentation performance regardless of image quality or condition, as robustness is critical for ensuring the utility of the system in real settings.

1. test-id6: Robustness Test
   Type: Automatic

   Initial State: The system is ready to process retinal images, with the pre-trained segmentation model loaded.

   Input: Fundus images with various conditions, including low contrast, presence of noise, and varying lighting conditions.

   Output: Segmented images where the retinal vessels are identified and delineated across all tested conditions, demonstrating the robustness of the algorithm.

   How test will be performed: Each image will be processed through the pre-trained segmentation model. The segmentation images for each condition will be reviewed to assess robustness of the segmentation algorithm. Manual blood vessel annotation images (ground truth) from domain experts will be used to evaluate the segmentation results to ensure the effectiveness of the system under different image conditions.

5.2.3 Nonfunctional: Usability

Usability is a key nonfunctional requirement for the RVSS, affecting how easily healthcare professionals can interact with the system. Usability testing will involve direct user interaction with the system followed by feedback collection through surveys.

1. test-id7: Usability Test
   Type: Manual, Dynamic

   Initial State: All features of the RVSS are accessible, including image upload, segmentation processing, and report generation functionalities.

   Input/Condition:

   - Participant: volunteers recruited to test the system
   - Materials: a set of fundus images for testing, tasks or scenarios designed to guide the participants through the system's functionalities, and a usability survey for feedback collection.

   Output/Result:

   - Quantitative Data: metrics such as task completion time, error rates during task execution, and frequency of help requests.
   - Qualitative Data: Participant feedback on the system's ease of use, interface intuitiveness, satisfaction with the segmentation process time, and overall experience. This includes suggestions for improvements.
   - Survey Responses: Structured responses collected through the usability survey, addressing specific aspects of system usability and user satisfaction.

   How test will be performed:

   - Preparation: recruit volunteers, prepare a set of predefined tasks (section 7.1) that cover the key functionalities of the system, designed to simulate typical use cases, and develop a usability survey (section 7.2) that includes questions on ease of use, interface design, functionality satisfaction, and overall user experience.
   - Execution: conduct a briefing session for participants to explain the purpose of the test and provide basic instructions on using the system, then let participants complete the predefined tasks, and fill out the usability survey, providing their feedback on the system.

5.2.4 Nonfunctional: Performance

The performance test for the RVSS is designed to evaluate the efficiency of the system, ensuring that it meets the non-functional requirements for processing speed under varying loads.

1. test-id8: Performance Test
   Type: Dynamic, Automatic

   Initial State: The RVSS is running, ready to receive and process uploaded fundus images for vessel segmentation

   Input/Condition: A set of fundus images.

   Output/Result: The time and resource utilization taken to process individual images and batches of images.

   How test will be performed:

   - Preparation: select a representative set of fundus images for the test, and define performance benchmarks based on the requirements specified in the SRS document, such as acceptable processing times for single images and image batches.
   - Single Image Processing: upload an individual image to the system and measure the time and resource utilization taken from upload to the completion of the segmentation process, then record the processing time for each image and compare it against the predefined benchmarks.
   - Batch Processing: upload batches of images, and measure the total time and resource utilization taken to process each batch.
   - Draw graphs as shown in Figure 3)

Figure 3: Performance Test graphs.

5.2.5 Nonfunctional: Compatibility

The compatibility test for the RVSS is designed to ensure the system is compatible with standard operating systems used in healthcare environments.

1. test-id9: Compatibility
   Type: Manual

   Initial State: The RVSS is fully operational and configured for standard operations.

   Input: None

   Output: A report of what is running normally on the system or a log of any errors encountered.

How test will be performed: Install and run whole software in different operating systems. Also, need to ensure cases pass in all different operating system.

5.2.6 Nonfunctional: Maintainability

The maintainability test for the RVSS focuses on assessing the system's ease of modification, scalability, and adaptability to future requirements or technological advancements.

1. test-id10: Maintainability
   Type: Manual

   Initial State: The RVSS code is in a stable state, with documentation, development environment setup, and version control system access ready.

   Input: The complete source code of RVSS, code for another segmentation network architecture to be replaced.

   Output: The complete code of the system that network architecture has been replaced.

   How test will be performed: Modify the code based on predefined change requests (here replacing the segmentation network architecture is taken as an example) to simulate real maintainability scenarios.

5.3 Traceability Between Test Cases and Requirements

A traceability between test cases and requirements is shown in Table 3.

	R1	R2	R3	R4	NFR1	NFR2	NFR3	NFR4	NFR5
5.1.1	X	X							
5.1.2			X						
5.1.3				X					
5.2.1			X	X	X				
5.2.2	X				X				
5.2.3						X			
5.2.4			X				X		
5.2.5								X	
5.2.6									X

Table 3: Tracebility between test cases and requirements

6 Unit Test Description

This section will be modified after finishing the design part.

The source code for the RVSS software has following modules,

- net.py - Provides network architecture for training.
- data_load.py - Loads and shows the raw fundus images.
- data_preprocess.py - Preprocesses fundus images .
- load_params.py - Module for load the input parameters, including learning rate, training parameters, training data set, batch size, etc.
- train.py - Train the segmentation network using the training dataset.
- test.py - Verifies the performance of the trained network using the test dataset.
- report.py - Generates a report including the segmentation outcomes and metrics values.

6.1 Unit Testing Scope

Unit testing is performed for the following modules:

- Load Data
- Load Parameters
- Model Training
- Pretrained Model Testing

These modules are high priority modules that can affect the whole system if not work properly. Other modules like data preprocess and report modules are tested in the system as those have low priority static modules, so our system do not perform unit test on these module.

6.2 Tests for Functional Requirements

6.2.1 Load Data Module (M1)

A Load Data Module is assigned to loading fundus images in the folder with specified path.

1. test-data-load-id11
   Type: Automatic

   Initial State: The Load Data Module is initialized and ready to accept requests for loading data from specified sources.

Input/Condition: The path to the specified folder.

Output/Result: Show all images in the folder.

How test will be performed: Run the data_load python file to check the correctness of the file.

6.2.2 Load Parameters Module (M4)

A Load Parameters Module is used to provide all parameters required for model training.

1. test-parameter-load-id12
   Type: Automatic

   Initial State: The Load Parameters Module includes all required parameters.

   Input/Condition: N/A

   Output/Result: Pass, if parameters are loaded before model training.

   How test will be performed: Set the number of training epoch in the load_params.py file to 0, and print the values of all parameters in the train.py file. If the printed parameter values are the same as the values set in load_params.py file, the load_params.py file is correct.

6.2.3 Model Training Module (M5)

A Model Training Parameters Module is used to train the segmentation model.

1. test-model-train-id13
   Type: Automatic

   Initial State: The Model Train Module is initialized.

   Input/Condition: The training dataset.

   Output/Result: Output the value of each round of training process. For example, the output of the i-th round is "Epoch i: loss value is 0.214".

   How test will be performed: Run train.py and see the training log.

6.2.4 Pretrained Model Testing (M6)

A Pretrained Model Testing Module is used to test the performance of the pretrained segmentation network.

1. test-model-test-id14
   Type: Automatic

   Initial State: The Model Test Module is initialized.

   Input/Condition: The test dataset.

Output/Result: Output the corresponding segmentation result.

How test will be performed: Load the model weights obtained from train.py into test.py. Load the fundus images in the test dataset into the test.py file and output the segmentation results

6.3 Tests for Nonfunctional Requirements

Unit testing the non-functional requirements is beyond the scope.

6.4 Traceability Between Test Cases and Modules

A traceability between test cases and modules is shown in Table 4

	M2	M3	M4	M5	M6	M7
test-image-preprocessing-id1	X					
test-image-preprocessing-id2		X				
test-vessel-segmentation-id3					X	
test-vessel-segmentation-id4			X	X	X	
test-report-generation-id5						X
test-data-load-id11	X					
test-parameter-load-id12			X	X		
test-model-train-id13			X	X		
test-model-test-id14				X	X	

Table 4: Tracebility between test cases and module

7 Appendix

7.1 Predefined tasks in the usability test

Predefined tasks:

1. Upload a Fundus Image
   - Select the option to upload a new fundus image.
   - Choose an image from the provided set and complete the upload process.
2. Initiate the segmentation process
   - Initiate the vessel segmentation process for the uploaded image.

3. Review the segmentation image

    - Access the completed segmentation results once the process is finished.
    - Review the segmented image alongside the original, noting the delineation of retinal vessels.

4. Review the segmentation report

    - Use the option to review a report based on the segmentation results.
    - Review the report within the system, noting the presentation and clarity of information.

7.2 User survey in the usability test

Usability Survey:

- Your role:

    ☐ Medical researcher
    ☐ Segmentation algorithm researcher
    ☐ Others:

- How familiar are you with medical image segmentation?

    ☐ Very Familiar
    ☐ Somewhat Familiar
    ☐ Not Familiar

- How intuitive were the instructions for uploading and preparing images for segmentation?

    ☐ Very Intuitive
    ☐ Somewhat Intuitive
    ☐ Neutral
    ☐ Somewhat Confusing
    ☐ Very Confusing

- How would you rate the clarity of the segmentation results?

    ☐ Very Clear
    ☐ Clear
    ☐ Neutral

- ☐ Somewhat Unclear
- ☐ Unclear

- How satisfied are you with the accuracy of the segmentation results?
  - ☐ Very Satisfied
  - ☐ Satisfied
  - ☐ Neutral
  - ☐ Dissatisfied
  - ☐ Very Dissatisfied
- What is the likelihood that the RVSS will be used in medical practice?
  - ☐ Very Likely
  - ☐ Likely
  - ☐ Neutral
  - ☐ Unlikely
  - ☐ Very Unlikely
  - ☐ Comments:
- Does the system perform the segmentation tasks within an acceptable time?
  - ☐ Yes
  - ☐ No
  - ☐ Comments:
- What features would you like to see improved or added to RVSS?
  - ☐ Comments:

References