Software Test Design (STD)

Introduction

This Software Test Design (STD) document provides detailed test cases for verifying the functionality of the ThermoML system - a machine learning-based tool for detecting inflammation in hands through thermal imaging. The test cases are derived from the system's Software Test Plan (STP), and cover all key features including image preprocessing, landmark detection, filtering, registration, classification, and web interface validation.

Test Cases

The following table provides the detailed test cases derived from the STP. The structure is organized by functional areas and includes both preprocessing and web validation components.

Test Case ID	Description	Preconditions	Test Steps	Expected Result	Actual Result
TC-1.1	Load optical and thermal images from dataset	Dataset is accessible	1. Launch preprocessing pipeline 2. Select image pair 3. Run loading function	Both optical and thermal images are successfully loaded	Passed successfully, images loaded correctly
TC-1.2	Apply contrast enhancement (CLAHE) to optical image	Optical image is loaded	1. Load optical image 2. Apply CLAHE function	Contrast-enhanced image is returned	Passed successfully, contrast was enhanced

TC-1.3	Segment hand region using SAM model	Thermal/optical images loaded	1. Run SAM model on optical image 2. Verify segmented mask	Segmented mask covers the hand area	Succeeded in approximately 93% of cases
TC-2.1	Detect landmarks on optical hand image	Segmented image of hand is available	1. Run MediaPipe on optical image	21 landmarks are detected and plotted	Passed successfully, all landmarks detected correctly on optical images
TC-2.2	Detect landmarks on thermal hand image	Hand in thermal image is visible	1. Run MediaPipe on thermal image	21 landmarks are detected (partial detection acceptable)	Partially failed - some thermal hands had fewer than 21 landmarks
TC-2.3	Visual rendering of detected landmarks	Landmarks detected	1. Display all landmarks on respective images	Landmarks clearly visualized	Passed successfully
TC-2.4	Calculate palm center on palmar hand	Hand is palmar-facing and landmarks detected	1. Run palm center function 2. Visualize intersection	Palm center is marked correctly	Passed successfully, palm center was correctly identified on palmar-facing hands only
TC-3.1	Scale and pad thermal image	Thermal image loaded	1. Resize and pad thermal image to match optical size	Scaled thermal image matches physical dimensions	Passed successfully

TC-3.2	Reject hands with <21 landmarks	MediaPipe returns fewer than 21 points	1. Run filtering function 2. Check result	Hand is discarded from processing	Passed successfully, hands with fewer than 21 landmarks were filtered out
TC-3.3	Match thermal and optical images by filename	Image pairs named consistently	1. Scan directory 2. Match filenames	Thermal-optical pairs matched correctly	Passed successfully
TC-4.1	Visual validation of registration alignment	Optical & thermal images + SAM mask available	1. Apply registration 2. Overlay mask or landmarks 3. Display result	Overlays align hand contours correctly between thermal and optical views	Registration overlays aligned correctly in all tested images.
TC-4.2	Numerical validation of registration alignment	Landmarks detected on both image types	1. Apply registration 2. Compare landmark positions 3. Compute distances	Average distance between corresponding landmarks is under 10 pixels	Average distance was below 10 pixels in most cases.
TC-4.3	Registration failure handling	Poor alignment conditions present	1. Apply registration 2. Detect alignment error 3. Skip image	Image is excluded from pipeline if alignment fails	Passed successfully
TC-5.1	Extract 4-channel input patch for joint	Registered thermal + hand center + mask available	1. Generate patch per joint including thermal,	Valid 4-channel patch extracted	Joint classification models achieved an average

					mask, masked thermal, distance map		accuracy of approximately 95.66% on the test datasets.

TC-5.2	Train 32 classification models	Patches available for each joint	1. Train 16 models per hand (excluding fingertips)	All 32 models trained	Passed successfully
TC-5.3	Validate file structures and formats	Patches and folders structured	1. Scan file hierarchy 2. Verify naming convention and image format	All files meet format expectations	Passed successfully
TC-6.1	Run inference for one joint using 4-channel patch	Model and patch available	1. Feed patch into corresponding model 2. Record output	Inflammation status predicted	Feature not yet implemented at the time of testing
TC-6.2	Check consistency across identical inputs	Same patch used multiple times	1. Feed patch multiple times 2. Compare outputs	Same output returned each time	Passed successfully
TC-6.3	Check latency of joint prediction	Model and patch ready	1. Measure time to predict output for one joint	Latency is within acceptable range	Passed successfully
TC-6.4	Visualize model outputs on hand image	Prediction results available	1. Map predictions to joint locations 2. Render	Each joint shows correct status (color)	Feature not yet implemented

			output overlay		at the time of testing
TC-7.1	View previously uploaded images	User is logged in	1. Go to 'history' 2. Select session	Images from session are shown	Feature not yet implemented at the time of testing
TC-7.2	Trigger image processing from frontend	User uploads image via web app	1. Upload image 2. Trigger preprocessing	Processing pipeline starts automatically	Passed successfully
TC-7.3	Full pipeline triggered via UI	Image uploaded and visible	1. Click process button 2. Run pipeline end-to-end	Image processed and output shown	Passed successfully
TC-7.4	Display per-joint results in frontend	Classification finished	1. Open results screen 2. Verify each joint display	Each joint status displayed clearly	Feature not yet implemented at the time of testing
TC-7.5	Verify auth and navigation functionality	User logged in	1. Navigate via top menu 2. Switch tabs 3. Log out	All UI flows function as expected	Passed successfully