**CHAPTER 7**

**TESTING**

Software testing is the process used to help identify the correctness, completeness, security and quality of developed computer software. This includes the process of executing the program or application with the intent of finding errors. Quality is not an absolute; it is value to some person. With that in mind testing can never completely establish the correctness of arbitrary computer software; testing furnishes a criticism or comparison that compares the state and behaviour of the product against a specification.

Testing forms the first step in determining the errors in a program. Clearly the success of testing in revealing errors in programs depends critically on the test cases. Because code is the only product that can be executed and whose actual behaviour can be observed, testing is the phase where the errors remaining from all the previous phases must be detected.

The program to be tested is executed with a set of test cases and the output of the program for the test cases are evaluated to determine if the programming is performing as expected.

**7.1 Testing Methodologies**

The following are the testing methodologies:

* **Unit Testing:** This is the first phase of testing; the different modules or components are tested individually, often performed by coder himself.
* **Integration Testing:** In this type of testing many unit tested modules are combined into subsystems, which are then tested. The goal here is to see if the modules can be integrated properly.
* **System Testing:** Here the entire software system is tested. The reference document for this process is the requirement specification and the goal is to see if the software meets the requirements. This form of testing is popularly known as black box testing.
* **Acceptance Testing:** It is performed with realistic data of the client to demonstrate that the software is working satisfactorily. It is the test conducted to determine if the requirements of a specification are met.

**7.2 Testing Criteria**

***Table 7.1:*** *Test cases for Fruit Recognition using Image Processing*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl. No** | **Test Procedure** | **Pre-Condition** | **Expected Result** | **Passed/ failed** |
| 1 | Click on feature extraction button in train phase | --- | Selected the folder of images. | Passed |
| 2 | Click on browse option under non-real time | Train the dataset | Select one image | Passed |
| 3 | Click on pre-processing button | Browse the image from the dataset | Black and white image and hsv image is displayed | Passed |
| 4 | Select segmentation button | Do the pre-processing | Three cluster images are displayed | Passed |
| 5 | Click on feature extraction button | Perform segmentation | Color histogram graph is displayed | Passed |
| 6 | Select SVM button | Features are extracted | The name of the fruit is displayed | Passed |
| 7 | Choose KNN button | Features are extracted | The name of the fruit is displayed | Passed |
| 8 | Click on realtime train option | --- | Selected the folder of images. | Passed |
| 9 | Click on camera on option | Train the real time image | The captured image is displayed | Passed |
| 10 | Select Capture button | On the camera to capture the image | Black and white image and HSV image is displayed | Passed |
| 11 | Choose SVM option | Pre-processing, Segmentation and Feature extraction is completed | The name of the fruit is displayed | Passed |