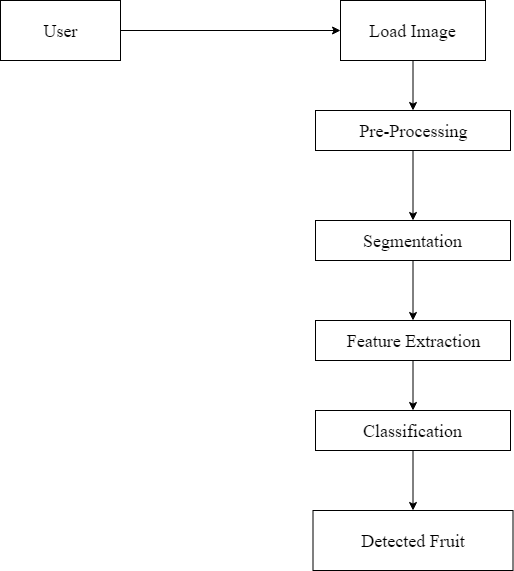
**CHAPTER 5**

**SYSTEM DESIGN**

The purpose of the design phase is to plan a solution of the problem specified by the requirement document. The design of a system is perhaps the most critical factor affecting the quality of the software, and has a major impact on the later phases, particularly testing and maintenance. The output of this phase is the design document. The design activity is often divided into two separate phases. They are system design and detailed design.

**5.1 High Level Design**

High-level design which is sometimes also called system design, aims to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results. At the end of system design all the major data structures, file formats, output formats, as well as the major modules in the system and their specifications are decided.

******

***Figure 5.1:*** *System Architecture of Fruit Recognition Using Image Processing*

Figure 5.1 shows the system architecture of Fruit Recognition using Image Processing. The user loads the Fruit image of interest. The image is pre-processed. Segmentation of pre-processed image takes place. The Feature Extraction is done using GLCM where the different features of image are extracted. The extracted image is fed as input to the SVM Classifier to classify the type of fruit present in the market.

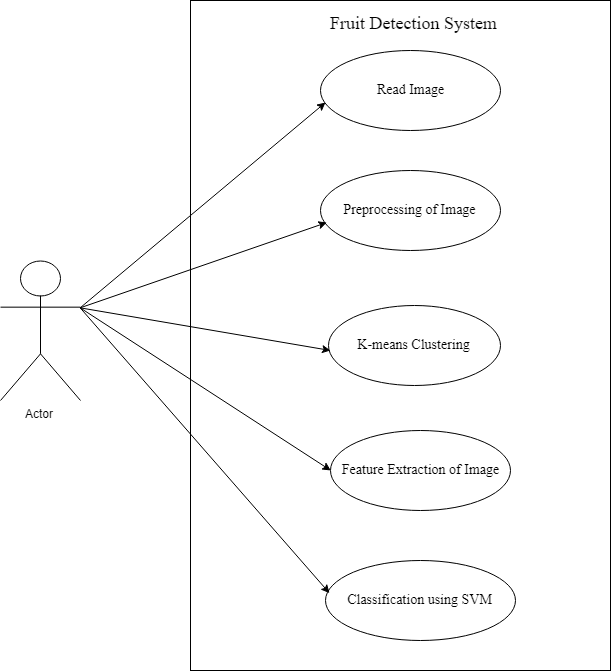
**5.2 Detailed Design**

During detailed design the internal logic of each of the modules specified in system design is decided. During this phase further details of the data structures and algorithmic design of each of the modules is specified. The logic of module is usually specified in a high-level design description language, which is independent of target language in which the software will eventually be implemented.

**5.2.1 Use Case Diagram of Fruit Recognition using Image Processing**

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. A use case diagram is a dynamic or behaviour diagram in UML. The use cases are represented by either circles or ellipses. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. Use case diagrams are valuable for visualizing the functional requirements of a system that will translate into design choices and development priorities. They also help identify any internal or external factors that may influence the system and should be taken into consideration.

There is only one actor that is user in the proposed system of Fruit Classification using SVM. The user reads the image of fruit that is to be classified. After reading the image, the system will load and pre-process the image. The image is clustered using k-means clustering algorithm by the system. The image then undergoes segmentation. Then the system will extract the features contrast, entropy, energy, homogeneity, and correlation etc of the image using GLCM. The extracted features of the image are fed to the SVM Classifier. The SVM Classifier will classify the name of fruit present in the image. The user then gets the output which specifies the name of fruit present in System.

******

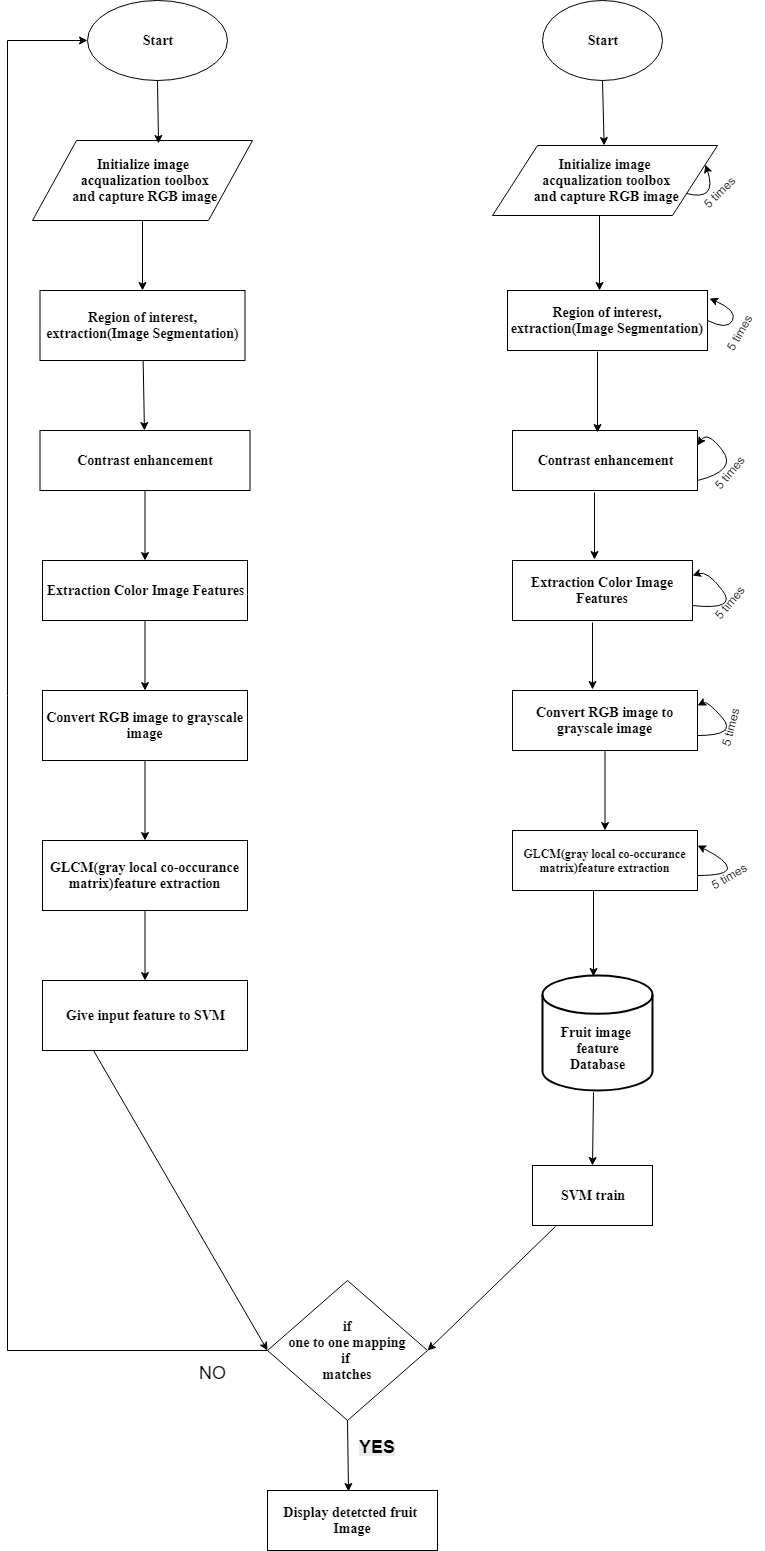
User

***Figure 5.2:*** *Use Case Diagram for Fruit Recognition using Image Processing*

**5.2.2 Data Flow Diagram of Fruit Recognition using Image Processing**

A Data Flow Diagram (DFD) is a graph showing flow of data values from their sources in objects through processes that transform them to destination in other objects .A DFD also known as “bubble chart”, has the purpose of clarifying the system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail. The bubbles represent data Transformations and the lines represent data flows in the system. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated.

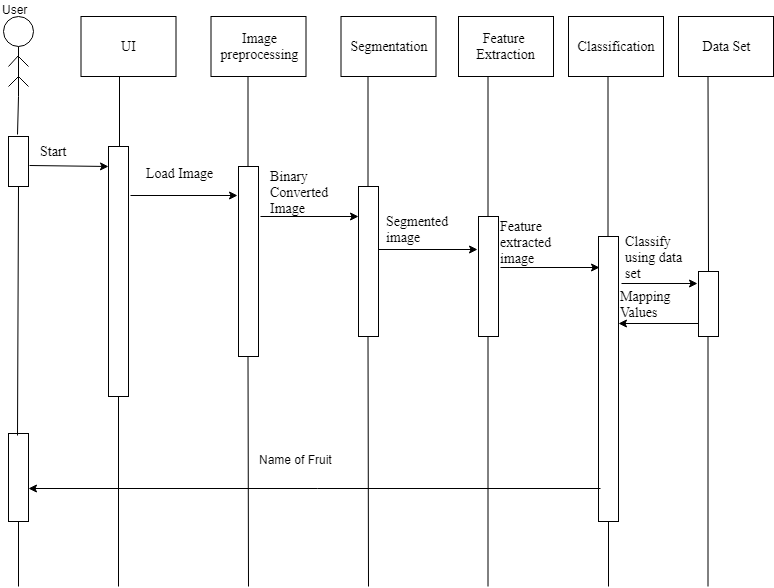
Figure 5.3 shows the data flow diagram for Fruit Recognition using Image Processing. User can give input in the form of image or captured. The image actualization capture RGB image. Next Image Segmentation is done for the particular captured image. The image obtained is extract the color features. This extracted RGB image converted into grayscale image. The Feature Extraction is done using GLCM (Gray Local Co-occurrence Matrix). The values which stored in database is taken and given input as SVM train. The input feature is also given to SVM. The condition of both test part and train part is checked, one-to-one mapping is done. The image doesn’t match the process is repeated. The input image matched then the detected fruit is displayed as output.

****

***Figure 5.3:*** *Data Flow Diagram for Fruit Recognition using Image Processing*

**5.2.3 Sequence Diagram of Fruit Recognition using Image Processing**

Figure 5.4 shows the sequence diagram of Fruit Recognition using Image Processing. The user loads the image and the application will preprocess the image. The preprocessed image is segmented in the segmentation module. The features are extracted from the segmented image in the extraction phase. The segmented image is returned to the user.



***Figure 5.4:*** *Sequence Diagram of Fruit Recognition using Image Processing*