**CHAPTER 6**

**SYSTEM IMPLEMENTATION**

System Implementation is the stage where the theoretical design is converted into a working system, the new system may be totally new, replacing an existing manual, or automated system or it may be a major modification to an existing system. The system is implemented using Python and rainfall data set.

**6.1 Programming Languages and Libraries Used**

**Python**

Python is an [interpreted](https://en.wikipedia.org/wiki/Interpreted_language) [high-level](https://en.wikipedia.org/wiki/High-level_programming_language) [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language). Python's design philosophy emphasizes [code readability](https://en.wikipedia.org/wiki/Code_readability) with its notable use of [significant indentation](https://en.wikipedia.org/wiki/Off-side_rule). Its [language constructs](https://en.wikipedia.org/wiki/Language_construct) as well as its [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) approach aim to help [programmers](https://en.wikipedia.org/wiki/Programmers) write clear, logical code for small and large-scale projects.

Python is [dynamically-typed](https://en.wikipedia.org/wiki/Dynamic_programming_language) and [garbage-collected](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigms), including [structured](https://en.wikipedia.org/wiki/Structured_programming) (particularly, [procedural](https://en.wikipedia.org/wiki/Procedural_programming)), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) and [functional programming](https://en.wikipedia.org/wiki/Functional_programming). Python is often described as a "batteries included" language due to its comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library).

Python is a [multi-paradigm programming language](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language). [Object-oriented programming](https://en.wikipedia.org/wiki/Object-oriented_programming) and [structured programming](https://en.wikipedia.org/wiki/Structured_programming) are fully supported, and many of its features support [functional programming](https://en.wikipedia.org/wiki/Functional_programming) and [aspect-oriented programming](https://en.wikipedia.org/wiki/Aspect-oriented_programming) (including by [meta-programming](https://en.wikipedia.org/wiki/Metaprogramming) and [meta-objects](https://en.wikipedia.org/wiki/Metaobject) (magic methods)). Many other paradigms are supported via extensions, including [design by contract](https://en.wikipedia.org/wiki/Design_by_contract) and [logic programming](https://en.wikipedia.org/wiki/Logic_programming).

Python uses [dynamic typing](https://en.wikipedia.org/wiki/Dynamic_typing) and a combination of [reference counting](https://en.wikipedia.org/wiki/Reference_counting) and a cycle-detecting garbage collector for [memory management](https://en.wikipedia.org/wiki/Memory_management). It also features dynamic [name resolution](https://en.wikipedia.org/wiki/Name_resolution_(programming_languages)) ([late binding](https://en.wikipedia.org/wiki/Late_binding)), which binds method and variable names during program execution.

**Requests**

Requests is a python module that allows us to send HTTP/1.1 requests extremely easily. There’s no need to manually add query strings to your URLs, or to form-encode your PUT & POST data.

We use this module to make a API request to the data set provider and get the dataset as a response.

**Scikit-learn**

Scikit-learn is a [free software](https://en.wikipedia.org/wiki/Free_software) [machine learning](https://en.wikipedia.org/wiki/Machine_learning) [library](https://en.wikipedia.org/wiki/Library_(computing)) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) [programming language](https://en.wikipedia.org/wiki/Programming_language). It features various [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis) and [clustering](https://en.wikipedia.org/wiki/Cluster_analysis) algorithms including [support vector machines](https://en.wikipedia.org/wiki/Support_vector_machine), [random forests](https://en.wikipedia.org/wiki/Random_forests), [gradient boosting](https://en.wikipedia.org/wiki/Gradient_boosting), [k-means](https://en.wikipedia.org/wiki/K-means_clustering) and [DBSCAN](https://en.wikipedia.org/wiki/DBSCAN), and is designed to interoperate with the Python numerical and scientific libraries [NumPy](https://en.wikipedia.org/wiki/NumPy) and [SciPy](https://en.wikipedia.org/wiki/SciPy).

Scikit-learn uses [NumPy](https://en.wikipedia.org/wiki/NumPy) extensively for high-performance linear algebra and array operations. Furthermore, some core algorithms are written in [Cython](https://en.wikipedia.org/wiki/Cython) to improve performance. Support vector machines are implemented by a Cython wrapper; logistic regression and linear support vector machines by a similar wrapper. Scikit-learn integrates well with many other Python libraries, such as [Matplotlib](https://en.wikipedia.org/wiki/Matplotlib) and [plotly](https://en.wikipedia.org/wiki/Plotly) for plotting, [NumPy](https://en.wikipedia.org/wiki/NumPy) for array vectorization, [Pandas](https://en.wikipedia.org/wiki/Pandas_(software)) dataframes, [SciPy](https://en.wikipedia.org/wiki/SciPy), and many more.

**Numpy**

NumPy is a library for the [Python programming language](https://en.wikipedia.org/wiki/Python_(programming_language)), adding support for large, multi-dimensional [arrays](https://en.wikipedia.org/wiki/Array_data_structure) and [matrices](https://en.wikipedia.org/wiki/Matrix_(math)), along with a large collection of [high-level](https://en.wikipedia.org/wiki/High-level_programming_language) [mathematical](https://en.wikipedia.org/wiki/Mathematics) [functions](https://en.wikipedia.org/wiki/Function_(mathematics)) to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by [Jim Hugunin](https://en.wikipedia.org/wiki/Jim_Hugunin) with contributions from several other developers. In 2005, [Travis Oliphant](https://en.wikipedia.org/wiki/Travis_Oliphant) created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is [open-source software](https://en.wikipedia.org/wiki/Open-source_software) and has many contributors.

**Pandas**

Pandas is a [software library](https://en.wikipedia.org/wiki/Software_library) written for the [Python programming language](https://en.wikipedia.org/wiki/Python_(programming_language)) for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and [time series](https://en.wikipedia.org/wiki/Time_series). It is [free software](https://en.wikipedia.org/wiki/Free_software) released under the [three-clause BSD license](https://en.wikipedia.org/wiki/3-clause_BSD_license). The name is derived from the term "[panel data](https://en.wikipedia.org/wiki/Panel_data)", an [econometrics](https://en.wikipedia.org/wiki/Econometrics) term for [data sets](https://en.wikipedia.org/wiki/Data_set) that include observations over multiple time periods for the same individuals. Its name is a play on the phrase "Python data analysis" itself.

**Matplotlib**

Matplotlib is a [plotting](https://en.wikipedia.org/wiki/Plotter) [library](https://en.wikipedia.org/wiki/Library_(computer_science)) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) programming language and its numerical mathematics extension [NumPy](https://en.wikipedia.org/wiki/NumPy). It provides an [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) [API](https://en.wikipedia.org/wiki/API) for embedding plots into applications using general-purpose [GUI toolkits](https://en.wikipedia.org/wiki/GUI_toolkit) like [Tkinter](https://en.wikipedia.org/wiki/Tkinter), [wxPython](https://en.wikipedia.org/wiki/WxPython), [Qt](https://en.wikipedia.org/wiki/Qt_(software)), or [GTK](https://en.wikipedia.org/wiki/GTK).

**Pyqt5**

PyQt5 is the latest version of a GUI widgets toolkit developed by Riverbank Computing. It is a Python interface for Qt, one of the most powerful, and popular cross-platform GUI library. PyQt5 is a blend of Python programming language and the Qt library. It is a popular C++ framework for writing GUI applications for all major desktop, mobile, and embedded platforms (supports Linux, Windows, MacOS, Android, iOS, Raspberry Pi, and more).

**6.2 Methods for Flood Prediction**

**KNN (K-Nearest Neighbor)**

Classification KNN is a nearest neighbor classification model where you can change both the distance matrix and the number of nearest neighbour. It stores training data, can use the model to compute the resubstitution prediction. This model can be convenient because training a classifier occurs in one step and classification in other steps.

**Logistic Regression**

Logistic Regression is a machine learning algorithm that predicts the probability of a categorical dependent variable. It is a statistical way of analyzing a set of data that comprises more than one independent variable that determines the outcome. The outcome is then measured with a dichotomous variable. The goal of this algorithm is to find the best model to describe the relationship between a dichotomous characteristic of interest and a set of independent variables.

**SVM (Support Vector Machine)**

It is a supervised learning algorithm which can used for binary classification or regression. It is a coordinate of individual observations. It is based on decision planes which defines decision boundaries. It also separated the set of objects having different class. a. This classifier is chosen as it is incredibly versatile in the number of different kernel functions that can be applied, and this model can yield a high predictability rate. SVM is one of the most popular and widely used clustering algorithms. It belongs to a group of generalized linear classifiers and is considered as an extension of the perceptron.

**Decision Tree**

Decision Trees (DTs) are a non-parametric supervised learning method used for [classification](https://scikit-learn.org/stable/modules/tree.html#tree-classification) and [regression](https://scikit-learn.org/stable/modules/tree.html#tree-regression). The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation. Decision trees are commonly used in [operations research](https://en.wikipedia.org/wiki/Operations_research), specifically in [decision analysis](https://en.wikipedia.org/wiki/Decision_analysis), to help identify a strategy most likely to reach a [goal](https://en.wikipedia.org/wiki/Goal), but are also a popular tool in [machine learning](https://en.wikipedia.org/wiki/Decision_tree_learning).

**6.2 Procedure for Fruit Recognition**

Step 1: Start

Step 2: Input image

Step 3: Image pre-processing

Step 4: Segmentation

Step 5: Feature Extraction

Step 6: Training the SVM and KNN

Step 7: Submit the new Fruit images to the trained SVM, and predict the output

Step 8: Stop

**6.2.1 Procedure for Image Pre-Processing**

Step 1: Start

Step 2: Convert original image to grey scale

Step 3: Convert grey scale to binary image

Step 4: Stop

**6.2.2 Procedure for SVM**

Step 1: Set up training data.

Step 2: Set up SVM’s parameter such as the type of SVM

Step 3: Train the SVM.

Step 4: Regions classified by the SVM.

Step 5: Support vector

**6.2.3 Procedure for KNN**

Step 1: Set up training data.

Step 2: Set up KNN’s termination criteria of the algorithm etc.

Step 3: Load the data

Step 4: Initialize the value of k

Step 5: For getting the predicted class, iterate from 1 to total number of training data points

1. Calculate the distance between test data and each row of training data. Here we will use Euclidean distance as our distance metric since it’s the most popular method. The other metrics that can be used are Chebyshev, cosine, etc.

2. Sort the calculated distance in ascending order based on distance values

3. Get top k rows from the sorted array

4. Get the most frequent class of these rows

5. Return the predicted class

Step 6: Train the KNN.

Step 7: Regions classified by the KNN.

Step 8: Support vector

**6.2.4 Procedure for Fruit Recognition in Non-Realtime**

Step 1: Start

Step 2: Train the dataset by selecting feature extraction option.

Step 3: Browse the image from dataset for testing.

Step 4: Then

Select Pre-processing button to perform preprocessing of the selected fruit image.

Step 5: Then

Select Segmentation button to perform Segmentation of the selected fruit image.

Step 6: Then

Select Feature Extraction button to perform Extraction of features of the selected fruit image.

Step 7: if button is equal to SVM then

SVM algorithm is used for recognition.

Step 8: if button is equal to KNN then

KNN algorithm is used for recognition.

Step 9: End.

**6.2.5 Procedure for Fruit Recognition in Realtime**

Step 1: Start

Step 2: If Camera is ON Then

Capture the fruit image then

Pre-processing of captured fruit image, then

Segmentation of captured fruit image, then

Feature Extraction of captured fruit image

Step 3: if option is equal to SVM then

SVM algorithm is used for recognition.

Step 4: if option is equal to ROC then

Graph with precision-recall is displayed

Step 5: End

**6.2.6 Procedure for Multi-Fruit Recognition in Realtime**

Step 1: Start

Step 2: If Camera is ON Then

Capture the fruit image then

Crop the single fruit image, then

Pre-processing of captured fruit image, then

Segmentation of captured fruit image, then

Feature Extraction of captured fruit image

Step 3: if option is equal to SVM then

SVM algorithm is used for recognition.

Step 4: End

**6.3 Flowchart for Fruit Recognition using Image Processing**

User Interface

Options

Preprocessing

Feature Extraction

Segmentation

Capture fruit image

Select image

Train captured dataset

Train Dataset

Real Time

Non-Real Time

Display Fruit Name

KNN

SVM

Recognition

***Figure 6.3:*** *Flowchart for Fruit Recognition using Image processing*

A system flowchart symbolically shows how data flows throughout a system and how event controlling decisions are made. Initially the image is input, and the pre-processing of the input image takes place followed by segmentation. Later the images are classified according to their extracted feature. Finally, the name of the fruit is detected. The steps involved are shown in figure 6.3.