

INFORMATICS INSTITUTE OF TECHNOLOGY

In Collaboration with

UNIVERSITY OF WESTMINSTER

**LifeVeda: Recognizing Ayurvedic Medicinal plants under natural background conditions.**

A Product Specification, Design and Prototype by

Ms. Minoli Jayasiri

w1867669/20210558

Supervised by

Ms. Malsha Fernando

Submitted in partial fulfilment of the requirements for the BSc (Hons) Computer Science degree at the University of Westminster.

**February 2024**

**Abstract**

Ayurveda is a traditional healthcare system in Sri Lanka that dates to more than 3000 years. Ayurveda uses raw materials obtained from the nature to produce medicine that can be used to treat various health conditions and diseases. Plant leaves are one of the main ingredients used to prepare Ayurvedic medicine. Plant leaves used in the process of preparing Ayurvedic medicine is normally collected from gardens and forests. Sometimes the individuals who collect hem may not be able to recognize the Ayurvedic plants leaves correctly which could result in the use of incorrect leaf types in the preparation of the medicine. This could cause the medicine to be ineffective or even toxic in some conditions. Furthermore, Ayurvedic plants get destroyed unintentionally when gardens and forests get cleared as many individuals are not able to recognize Ayurvedic medicinal plants on their own this could lead to the endangerment and extinction of these valuable plants.

To solve this problem the author has implemented a system that is able to recognize Ayurvedic medicinal plants using their leaves when in their natural environment. The author took two approaches in implementing this system to find the best solution. The first method involved the building of a Convolutional Neural Network architecture (CNN) from scratch the other method involved the training of multiple pre-trained models using transfer learning and combining them by using the averaging technique to form an Ensemble model to recognize the leaf of Ayurvedic plants in their natural backgrounds.

During the initial testing, accuracy was used as the evaluation metric to determine the performance of the two models. The CNN model gave an accuracy of 50% and the Ensemble model showed higher performance with an accuracy of 96.88%.

**Keywords:** Convolutional Neural Network (CNN), Transfer learning, Ensemble Learning, Ayurveda

**Subject Descriptors:**

* Computing methodologies -> Artificial Intelligence -> Computer vision -> Computer vision problems -> Object Recognition
* Computing methodologies -> Machine learning -> Machine learning approaches -> Neural networks

TABLE OF CONTENTS

[CHAPTER 01: PROBLEM 3](#_Toc158505468)

[1.1Chapter Overview 3](#_Toc158505469)

[1.2 Introduction 3](#_Toc158505471)

[1.3 Problem Domain 3](#_Toc158505473)

[1.3.1 Ayurvedic Medicine in Sri Lanka 3](#_Toc158505474)

[1.3.2 Image Recognition 9](#_Toc158505475)

[1.3.3 Plant recognition 9](#_Toc158505476)

[1.4 Problem Definition 10](#_Toc158505477)

[1.4.1 Problem Statement 10](#_Toc158505478)

[1.5 Research Aim 11](#_Toc158505479)

[1.6 Research Objectives 11](#_Toc158505480)

[1.7 Novelty of the Research 13](#_Toc158505481)

[1.7.1 Problem Novelty 13](#_Toc158505482)

[1.7.2 Solution Novelty 13](#_Toc158505483)

[1.8 Research Gap 13](#_Toc158505484)

[1.9 Contribution to the Body of Knowledge 14](#_Toc158505485)

[1.9.1 Contribution to the problem domain 14](#_Toc158505486)

[1.9.2 Contribution to the research domain 14](#_Toc158505487)

[1.10 Research Challenge 15](#_Toc158505488)

[1.11 Chapter Summary 15](#_Toc158505489)

[CHAPTER 02: SOFTWARE REQUIREMENT SPECIFICATION 16](#_Toc158505490)

[4.1 Chapter Overview 16](#_Toc158505491)

[4.4 Rich Picture Diagram 16](#_Toc158505492)

[2.3 Stake Holder Analysis 17](#_Toc158505493)

[2.3.1 Stake Holder Onion Model 17](#_Toc158505494)

[2.3.2 Stake Holder Viewpoints 17](#_Toc158505495)

[2.4 Selection of Requirement Elicitation Methodologies 19](#_Toc158505496)

[2.5 Discussion of Findings 20](#_Toc158505497)

[2.5.1 Literature Review 20](#_Toc158505498)

[2.5.2 Survey 20](#_Toc158505499)

[2.5.3 Prototyping 27](#_Toc158505500)

[2.5.4 Summary of Findings 28](#_Toc158505501)

[2.6 Context Diagram 28](#_Toc158505528)

[2.7 Use Case Diagram 29](#_Toc158505529)

[2.8 Use Case Descriptions 30](#_Toc158505530)

[2.9 Requirements 31](#_Toc158505531)

[2.9.1 Functional Requirements 32](#_Toc158505532)

[2.9.2 Non-Functional Requirements 32](#_Toc158505533)

[2.10 Chapter Summary 33](#_Toc158505534)

[CHAPTER 03: DESIGN 34](#_Toc158505535)

[3.1. Chapter Overview 34](#_Toc158505536)

[3.2 Design Goals 34](#_Toc158505538)

[3.3 System Architecture Design 35](#_Toc158505539)

[3.3.1 Tiered Architecture Diagram 35](#_Toc158505540)

[3.3.2 Discussion of tiers 35](#_Toc158505541)

[3.4 System Design 36](#_Toc158505542)

[3.4.1 Choice of Design Paradigm 36](#_Toc158505543)

[3.5 Design Diagrams 37](#_Toc158505544)

[3.5.1 Sequence Diagram 37](#_Toc158505545)

[3.5.2 Data Flow Diagram – Level 1 38](#_Toc158505546)

[3.5.3Data Flow Diagram – Level 2 38](#_Toc158505547)

[3.5.4 System Process Flow Chart 39](#_Toc158505548)

[3.5.5 User Interface Design 39](#_Toc158505549)

[3.6 Chapter Summary 39](#_Toc158505550)

[CHAPTER 04: INITIAL IMPLEMENTATION 40](#_Toc158505552)

[4.1 Chapter Overview 40](#_Toc158505553)

[4.2 Technology Selection 40](#_Toc158505555)

[4.2.1 Technology Stack 40](#_Toc158505556)

[4.2.2 Dataset Selection 41](#_Toc158505557)

[4.2.3 Development Frameworks 41](#_Toc158505559)

[4.2.4 Programming Languages 42](#_Toc158505560)

[4.2.5 Libraries 42](#_Toc158505561)

[4.2.6 Integrated Development Environments(IDE) 42](#_Toc158505562)

[4.2.7 Summary of Technology Selection 42](#_Toc158505563)

[4.3 Implementation of the Core Functionality 43](#_Toc158505564)

[4.3.1 Data Augmentation 43](#_Toc158505566)

[4.3.2 Implementation of Model 45](#_Toc158505567)

[4.5 Chapter Summary 48](#_Toc158505568)

[CHAPTER 05: CONCLUSION 49](#_Toc158505570)

[5.1 Chapter Overview 49](#_Toc158505571)

[5.2 Deviations 49](#_Toc158505573)

[5.2.1 Scope related deviations 49](#_Toc158505574)

[5.3 Initial Test Results 50](#_Toc158505575)

[5.3.1 Initial Test Results for the CNN model 50](#_Toc158505577)

[5.3.2 Initial Test Results for the Ensemble model 50](#_Toc158505578)

[5.4 Required Improvements 50](#_Toc158505579)

[5.4.1 Required Improvements for the CNN model 50](#_Toc158505580)

[5.4.2 Required Improvements for the Ensemble model 51](#_Toc158505581)

[5.5 Demo of the Prototype 51](#_Toc158505582)

[5.6 Chapter Summary 51](#_Toc158505583)

[Appendix 54](#_Toc158505584)

# LIST OF TABLES

[Table 1: Research objectives 10](#_Toc158582365)

[Table 2: Stakeholder Viewpoints (Self-Composed) 16](#_Toc158582366)

[Table 3: Requirement Elicitation Methods (Self-Composed) 17](#_Toc158582367)

[Table 4: LR Findings (Self-Composed) 17](#_Toc158582368)

[Table 5: Survey (Self-Composed) 24](#_Toc158582369)

[Table 6: Summary of Findings (Self-Composed) 25](#_Toc158582370)

[Table 7: Use Case Descriptions of Upload Image (Self-Composed) 27](#_Toc158582371)

[Table 8: Use Case Descriptions of View Ayurvedic Plants (Self-Composed) 28](#_Toc158582372)

[Table 9: Requirements (Self-Composed) 28](#_Toc158582373)

[Table 10: Table 3: Functional Requirements (Self-Composed) 29](#_Toc158582374)

[Table 11: Non-Functional Requirements (Self-Composed) 30](#_Toc158582375)

[Table 12: Design Goals (Self-Composed) 31](#_Toc158582376)

[Table 13: Libraries Utilized (Self-Composed) 40](#_Toc158582377)

[Table 14: Summary of Technologies (Self-Composed) 40](#_Toc158582378)

[Table 15: Schedule Deviations (Self-Composed) 47](#_Toc158582379)

# LIST OF FIGURES

[Table 1: Research objectives 10](#_Toc158582411)

[Table 2: Stakeholder Viewpoints (Self-Composed) 16](#_Toc158582412)

[Table 3: Requirement Elicitation Methods (Self-Composed) 17](#_Toc158582413)

[Table 4: LR Findings (Self-Composed) 17](#_Toc158582414)

[Table 5: Survey (Self-Composed) 24](#_Toc158582415)

[Table 6: Summary of Findings (Self-Composed) 25](#_Toc158582416)

[Table 7: Use Case Descriptions of Upload Image (Self-Composed) 27](#_Toc158582417)

[Table 8: Use Case Descriptions of View Ayurvedic Plants (Self-Composed) 28](#_Toc158582418)

[Table 9: Requirements (Self-Composed) 28](#_Toc158582419)

[Table 10: Table 3: Functional Requirements (Self-Composed) 29](#_Toc158582420)

[Table 11: Non-Functional Requirements (Self-Composed) 30](#_Toc158582421)

[Table 12: Design Goals (Self-Composed) 31](#_Toc158582422)

[Table 13: Libraries Utilized (Self-Composed) 40](#_Toc158582423)

[Table 14: Summary of Technologies (Self-Composed) 40](#_Toc158582424)

[Table 15: Schedule Deviations (Self-Composed) 47](#_Toc158582425)

# LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| Acronym | Description |
| CNNs | Convolutional Neural Networks |
| SVMs | Support Vector Machines |
| SSAD | Object Oriented Analysis and Design |
| OS | Operating System |

# **CHAPTER 01: PROBLEM**

## **1.1Chapter Overview**

## This chapter aims to give the reader an insight into the problem that the author is going to solve through this research and the gaps in the existing systems that have been proposed to solve this problem. The objectives and the scope of the research and the anticipated challenges in implementing the solution will also be outlined in this chapter.

## **1.2 Introduction**

## Ayurveda holds a significant place as a traditional medical system in Sri Lanka. Ayurvedic medicine is used by many citizens of Sri Lankan. Ayurveda makes use of raw materials obtained from the nature to prepare medicine that is used for the treatment of many health conditions and diseases. Natural ingredients like plants, plant leaves, flowers, roots, minerals etc. are used in the preparation of Ayurvedic medicine. It is important that the raw materials used in the preparation of Ayurvedic medicine are of good quality for the medicine to be effective. One of the main ingredients used for the preparation of Ayurvedic medicine are leaves of Ayurvedic plants. Plant leaves are collected from gardens and forests.

## As these raw materials are collected by humans, they may collect incorrect plant types as well which can cause the medicine to not work as desired. Also, Ayurvedic plants get destroyed during clearance of gardens and forests a most of the individuals are not aware of the identity and value of Ayurvedic plants. This can lead to the extinction of Ayurvedic plants which could reduce the supply of the enough medicine for the individuals who use Ayurveda for the treatment of health conditions.

## As a solution to this the author has identified an application to recognize ayurvedic medicinal plants in their natural environment would be beneficial in aiding the Ayurvedic medical system in Sri Lanka.

## **1.3 Problem Domain**

### **1.3.1 Ayurvedic Medicine in Sri Lanka**

Ayurveda, originating in India, is a traditional system of medicine in Sri Lanka. Ayurveda has been practiced in Sri Lanka for more than 3000 years. In Ayurveda various parts of plants such as leaves, roots, fruits, flowers, and barks are used for the treatment of diseases. Among these, leaves play a significant role in the formulation of ayurvedic medicines. Plants used in ayurvedic medicine are mostly collected from forests and gardens.

To collect the correct plants needed for the preparation of the medicines which are used to cure diseases, one should be able to accurately identify them but those who collect these plants are not properly trained for it. There is a risk of human error in this situation. Incorrect plants used in the preparation of the medicine used for treatment of diseases can cause the medicine to be ineffective. This might also result in a decline of trust among individuals towards the principles of Ayurvedic healthcare.

Certain plants and herbs are at risk of disappearing because the forests they grow in are being destroyed (Weragoda, 1980). Since individuals aren't familiar with these plants, they might unintentionally remove them. If there was a method to easily identify these plants, it could contribute to their preservation.

### **1.3.2 Image Recognition**

Image recognition is a subset of computer vision, computer vision involves utilizing computers to interpret digital images. A fundamental aspect of computer vision is image recognition, which aids in identifying and classifying components within images. Image recognition enables machines to recognize and identify objects, individuals, elements, and other factors within images. In image recognition labelled images can be trained using a machine learning model, once model is trained it can be used to predict the class of previously unseen images. Image recognition technology automatically extracts features of the image and performs image classification ([Yinglong](https://ieeexplore.ieee.org/author/37089736255), 2022).

In recent years rapid developments have been made in image recognition technology. Among these the most popular traditional machine learning models are Support Vector Machines (SVMs), a type of supervised machine learning algorithm that is used for classification and regression tasks and Bag of Features Models, a classification method that uses unordered collection of image features.

However, the latest advancement in the field of image recognition is deep learning. It has introduced more accurate and efficient models for image recognition. In simple words, deep learning is a subset of machine learning that mimics our brain’s functioning that enables systems to learn without human intervention. Convolutional Neural Networks and transformer-based models are two significant deep learning algorithms that are used in recognizing images. The key feature of CNNs is that they do the feature extraction themselves using convolutional layers unlike traditional machine learning models where features must be extracted manually. The Transformer architecture was first introduced in 2017 in the paper “Attention Is All You Need” (Vaswani, 2017).

### **1.3.3 Plant recognition**

Plant recognition involves training computer systems to automatically recognize and classify plants mostly using their leaves, flowers, fruits etc. Identifying plants through image recognition has been an area of research for many years now. Often leaves have been used for the classification of plants. A leaf can be characterized by its color, its texture, and its shape (Sharma and Gupta, 2015).

The process of detection consists of various stages, including image pre-processing, image segmentation, image enhancement, and localization. The effective implementation of pre-processing is important as it significantly reduces computation time and noise, resulting in higher accuracy (Pushpanathan et al., 2020).

## **1.4 Problem Definition**

Medicinal plants are gaining more attention in the pharmaceutical industry as they tend to have less harmful effects and are cost effective compared to modern medicine (Pushpanathan et al., 2020). Ayurveda is a traditional system of healing in Sri Lanka where plants and various plant components including leaves, roots, bark, flowers, and fruits, are widely employed as fundamental ingredients in the preparation of ayurvedic medicine. As a result of post-pandemic inflation, the prices of Western medicines have significantly risen. Additionally, with the occurrence of recent cases of western medicine toxicity in Sri Lanka, an increasing number of individuals have turned to Ayurvedic medicine as an alternative remedy for various ailments. Currently the plants used in the preparation of ayurvedic medicine are collected manually, and individuals involved in the collection process often lack professional training in accurately identifying medicinal plants (Jayalath et al., 2019). In the scenario where one needs to collect these plants, seeking guidance from someone with expertise in their identification becomes essential. Improper collection of plants can lead to the ineffectiveness of Ayurvedic medicine. Certain Ayurvedic plants also thrive within home gardens. Facilitating easy recognition of these plants is important, as otherwise, they might be removed unintentionally due to limited knowledge regarding them. Therefore, an application that could recognize ayurvedic medicinal plants under natural environmental conditions could contribute significantly to both the collection and preservation of these plants, thereby supporting the Ayurvedic medical industry.

### **1.4.1 Problem Statement**

Ayurvedic medicinal plants are currently collected manually which can lead to the collection of incorrect plants due to human error, furthermore, due to limited knowledge about these plants, they often face inadvertent destruction.

## **1.5 Research Aim**

*The aim of this research is to design, develop and evaluate a system that can detect and identify Ayurvedic Medicinal plants in their natural environment using image processing.*

Further elaborating on the aim, the proposed system would help classify ayurvedic plants in their natural habitat with the use of image processing technology. A suitable deep learning model will be developed utilizing available deep learning architectures that would be able to accurately perform image classification. This system mainly targets the Ayurvedic Medicine industry in Sri Lanka. This could be used to assist in the process of collecting Ayurvedic medicinal plants. This application could also be used by patients undergoing ayurvedic medical treatments as some medicines are required to be prepared by patients on their own according to the advice of the doctor.

## **1.6 Research Objectives**

|  |  |  |  |
| --- | --- | --- | --- |
| **Research Objectives** | **Description** | **Learning Outcomes** | **Research Questions** |
| Literature Review | R01: Identifying gaps and limitations in existing research on the chosen domain.  R02: Research on techniques that are used for plant recognition using image processing.  R03: Analyze the algorithms, machine learning models and technologies that have been used in previous research to solve this problem.  R04: Research on methods that can be used to recognize objects with natural backgrounds. | L02, L04, L06 | RQ2, RQ3 |
| Requirement Elicitation | R05: Get feedback and opinions on proposed system through surveys and interviews.  R06: Get insights from domain and industry experts and analyze them to build a suitable application.  R07: Gather data on Ayurvedic medicinal plants that can be used for the research. | L02, L03, L04, L05 | RQ1 |
| Design | R08: To design the machine learning classification model required to build the proposed system.  R9: To design the frontend for the proposed system.  R11: To design the backend for the proposed system. | L01, L03, L06 | RQ2, RQ3 |
| Implementation | R12: To develop an appropriate machine learning model to classify the ayurvedic plants accurately.  R13: To develop the core functionalities of the proposed system.  R14: To develop an application with all proposed functionalities. | L01, L03, L05, L07 | RQ2, RQ3 |
| Evaluation | R15: Test the created plant recognition model using evaluation metrics, calculate accuracy, precision, recall and f1 score to evaluate the performance of the model.  R16: Check if the training dataset has a class imbalance.  R17: Compare created plant recognition model with existing systems.  R18: Ensure each software component operates as expected individually and the system works as intended as a whole.  R19: Carry out usability testing to determine how well a user can use the created web application in a real-world scenario.  R20: To get feedback from industry and domain experts on the final prototype. | L04 | RQ2, RQ3 |

Table 1: Research objectives

## **1.7 Novelty of the Research**

### **1.7.1 Problem Novelty**

The problem identified by the author is that there are no existing systems developed that are able to perform image classification based of leaf images of Ayurvedic medicinal plants in Sri Lanka that are captured in their natural background. All existing systems requires users to capture the image of the leaf by placing it in a white background.

### **1.7.2 Solution Novelty**

As the solution to the identified problem the author has implemented two different techniques. One technique involves the development of a CNN model from scratch to perform classification of the Ayurvedic medicinal plants. The other method involves the building of an Ensemble model developed using the averaging technique which involves the combination of outputs of several base models to give the final classification output.

## **1.8 Research Gap**

Research has been carried out for the recognition of ayurvedic plants in Sri Lanka. An approach using Transfer learning techniques has achieved an accuracy of 95.5% for a dataset with 5 classes and 369 images in total (Azeez and Rajapakse, 2019). One attempt using CNNs achieved an accuracy of 90% ( Jayalath et al., 2019). Another attempt was carried out using scanned images of leaves, this was trained using a convolutional neural network based on AlexNet and the model achieved a validation accuracy of 97.71% (Jayanka and Fernando, 2020).

Although these approaches perform well and have good test accuracies, it’s important to note that all leaves were captured against a white background when creating the dataset used to train the models. So, these models may not work well under natural environment conditions, varying light conditions and noisy backgrounds could lead to poor performance and misclassification. These limitations would be addressed in this research project.

## **1.9 Contribution to the Body of Knowledge**

The proposed research project contributes to the problem domain and research domain as follows.

### **1.9.1 Contribution to the problem domain**

This project aims to create an application that would recognize ayurvedic medicinal plants under natural environment conditions. This application holds significant value in the preparation of ayurvedic medicine as it enables the accurate recognition of medicinal plants used in the production process. Given the limited awareness regarding ayurvedic medicinal plants, the preservation of these valuable plants is often compromised due to human activities, therefore this application holds the potential to mitigate this issue by aiding the recognition of medicinal plants contributing to their conservation.

### **1.9.2 Contribution to the research domain**

The proposed research project will utilize image processing technology to recognize ayurvedic medicinal plants, therefore several concepts in machine learning will be used in the development process. The findings discovered during this research can be used by other researchers for evaluation and future research. Recognizing plants in their natural setting remains a challenge in the machine learning domain till date. Limited research has been done regarding the recognition of plants in their natural environments. This study aims to introduce an architecture that leverages existing deep learning models to precisely classify plants in their natural habitats. This research could demonstrate that deep learning models can be used in more complex real-world scenarios as well rather than under controlled conditions. The outcome of this research could also contribute to the development of more robust architectures in this domain.

## **1.10 Research Challenge**

During the implementation of this research project the author will encounter several challenges. Listed below are some of the challenges identified so far.

* As there is no publicly available dataset the author would have to create a dataset by capturing images of ayurvedic medicinal plants.
* Research would have to be done to select the most efficient machine learning algorithm for accurate image classification.
* Since the application would be developed to recognize ayurvedic medicinal plants in natural environment conditions the author would have to develop a technique to effectively distinguish the plants from their backgrounds.
* Machine learning methods like edge detection and image segmentation would have to be used to separate the foreground and the background, another challenge would be to select the best algorithm for this task.

## **1.11 Chapter Summary**

The chapter presents the user with an overview of the research, including the identified problem in the Ayurvedic Medicinal system of Sri Lanka and the proposed solution. It justifies the need to provide a solution to this problem by using the latest technologies in machine learning and how this would be a contribution to both the Ayurvedic Medicinal industry and the field of plant recognition using deep learning and the novelty of the research proposed by the author.

# 

# **CHAPTER 02: SOFTWARE REQUIREMENT SPECIFICATION**

## **4.1 Chapter Overview**

This chapter gives an insight to the requirements the author has identified for this research project, the methods the requirements were gathered, and the conclusions drawn from the gathered requirements. The potential stakeholders of the system and the way they interact with the system have also been identified. Additionally, the functional and non-functional requirements and the use case diagram and its descriptions have also been outlined in this chapter.

## **4.4 Rich Picture Diagram**

A diagram of a life cycle

Description automatically generated

Figure 1: Rich Picture Diagram (Self-Composed)

## **2.3 Stake Holder Analysis**

### **2.3.1 Stake Holder Onion Model**

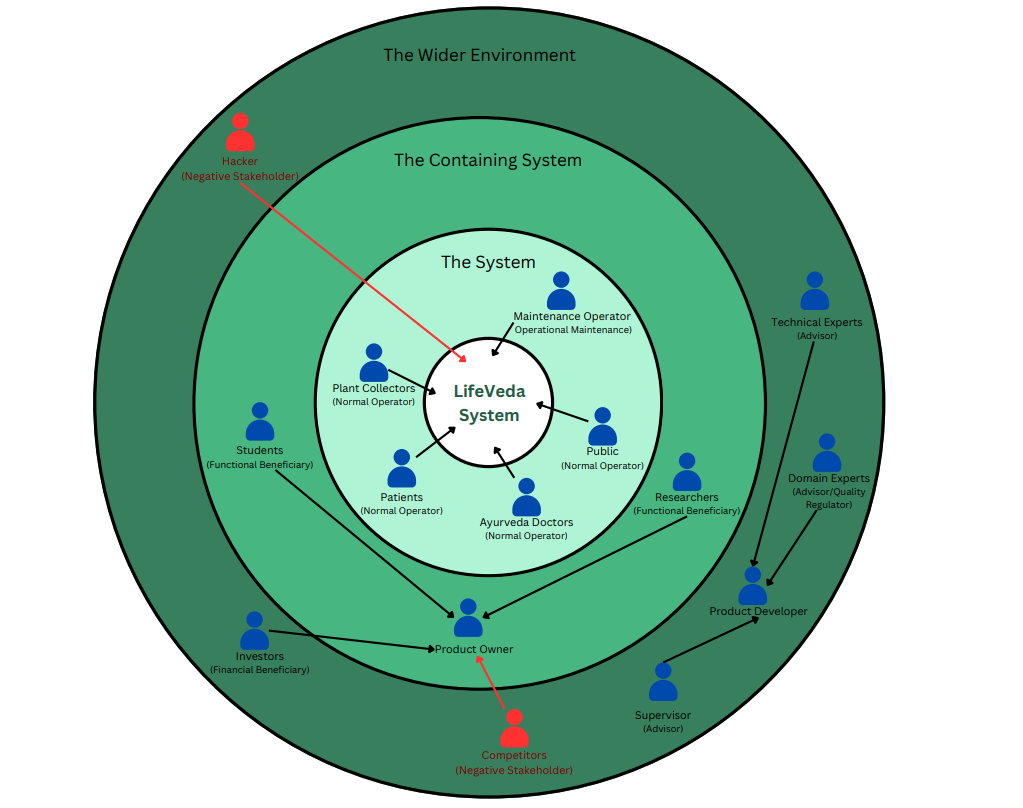


Figure 2: Onion Model (Self-Composed)

### **2.3.2 Stakeholder Viewpoints**

|  |  |  |
| --- | --- | --- |
| **Stakeholder** | **Role** | **Description** |
| **The System Stakeholders** | | |
| Plant collectors | Normal operator | Uses the LifeVeda application to identify plants with an Ayurvedic value. |
| Ayurveda doctors | Normal operator | Uses the LifeVeda application to validate the identity of collected plant leaves. |
| Public | Normal operator | Uses the LifeVeda application to identify Ayurvedic plants to use for home remedies and to identify plants to avoid unintentional destruction. |
| Patients | Normal operator | Uses the LifeVeda application to identify Ayurvedic plants to prepare medicine advised by Ayurvedic doctors. |
| Maintenance operator | Operational Maintenance | Maintains the LifeVeda system and investigates user inquiries. |
| **The Containing System Stakeholders** | | |
| Students | Functional beneficiary | Explore the techniques used in the research to use it in other problem domains. |
| Researchers | Functional Beneficiary | Investigates the techniques used in the research to further improve it. |
| Product owner | Functional beneficiary | Owns the product, communicates with other stakeholders, and takes decisions regarding the system. |
| **The Wider Environment Stakeholders** | | |
| Hackers | Negative stakeholders | May steal data and disrupt the system. |
| Competitors | Negative stakeholders | Build better systems that outperform the LifeVeda application and decreases its market value. |
| Investors | Financial beneficiary | Provide funding to the system and gains profit from investing. |
| Product Developer | Developer | Develops and improves the system. |
| Supervisor | Advisor | Guides the developer and gives feedback to make improvements. |
| Technical experts | Advisor | Gives expert opinions and feedback helpful in developing the system. |
| Domain Experts | Advisor/Quality regulator | Provides information on the chosen domain helpful for the system development and evaluates the system. |

Table 2: Stakeholder Viewpoints (Self-Composed)

## **2.4 Selection of Requirement Elicitation Methodologies**

Requirement Elicitation is used to gather requirements needed for the development of this research project. The table below shows the requirement elicitation methods that were used and discusses the rationale behind the choices.

|  |
| --- |
| **Method 01: Literature Review** |
| A literature review helps identify research gaps and problems in existing systems implemented to identify Ayurvedic medicinal plants in Sri Lanka. Identifying a good research gap to carry out the research is a vital component of a research project. The author reviewed research papers of existing systems that have been implemented in the chosen problem domain to identify their limitations. Possible techniques available for the implementation of the system were also identified through this. |
| **Method 02: Survey** |
| There is no specific target audience for this application . The purpose of the system is to help recognize Ayurvedic medicinal plants and their properties. This system can be used by the individuals who collect Ayurvedic plants, doctors who prepare Ayurvedic medicine, patients who were advised doctors to use Ayurvedic medicine prepared at home and the public to identify these plants to preserve them if available in their home gardens. Since this system if useful in many ways it is important to gather insights and requirements from a large group of people to get their ideas. The author can determine which age group to target from the results of the survey. |
| **Method 03: Prototyping** |
| It is important to implement an effective novel technique in order to address the identified research gap. ML models used for image recognition tasks tend to underperform when images are noisy. To build a robust system which is able to identify edges and distinguish between the foreground and background to identify Ayurvedic medicinal plants in their natural environments, continuous developments, testing and fine-tuning is required therefore prototyping was chosen as a requirement elicitation method. |

Table 3: Requirement Elicitation Methods (Self-Composed)

## **2.5 Discussion of Findings**

### **2.5.1 Literature Review**

|  |  |
| --- | --- |
| **Findings** | **Citation** |
| The leaves of plants contain lot of features, and these can be automatically learned by DL algorithms. | Kanda, Xia and Sanusi, 2021 |
| Data augmentation technique can be used to increase the size of datasets and the use of augmented images increases model performance. | Kanda, Xia and Sanusi, 2021 |
| Small datasets can cause CNNs to overfit therefore CNNs require large datasets for training. | Kanda, Xia and Sanusi, 2021 |
| The use of DL models for feature extraction gives better performance than using hand-crafted features. | Kanda, Xia and Sanusi, 2021 |
| The use of Deep Ensemble Learning can give better performance than using individual CNN models. | Chompookham and Surinta, 2021 |
| Model did not perform well when tested on images of leaves captured in their natural environment. | Jayanka and Fernando, 2020 |
| Reasons for poor quality of Ayurvedic medicine and absence of enough supply is related to the lack of knowledge in recognizing these plants and their extinction. | Kankanamalage et al., 2014 |

Table 4: LR Findings (Self-Composed)

### **2.5.2 Survey**

|  |  |
| --- | --- |
| **Question** | Have you used Ayurvedic medicine before? |
| **Aim of the Question** | To identify the percentage of individuals who have use Ayurvedic medicine. |
| **Observation**  Forms response chart. Question title: Have you used Ayurvedic medicine before?. Number of responses: 114 responses. | |
| **Conclusion**  It can be observed that 89.5% of the participants have used Ayurvedic medicine before. It can be concluded that Ayurveda is an important medicinal system in Sri Lanka that most people trust and use therefore addressing a problem in this domain is beneficial to the society. | |
| **Question** | Do you believe that the Ayurvedic medicinal system in Sri Lanka is important? |
| **Aim of the Question** | To prove the importance of Ayurveda. |
| **Observation**  Forms response chart. Question title: Do you believe that the Ayurvedic medicinal system in Sri Lanka is important?. Number of responses: 114 responses. | |
| **Conclusion**  All participants agreed that the Ayurvedic medicinal system in Sri Lanka is important. The high level of agreement among participants suggests that Ayurveda plays a crucial role in the health and well-being practices of individuals and it is essential to address the problem identified in this domain. | |
| **Question** | Are you aware that plant leaves are used as an ingredient in the preparation of Ayurvedic medicine? |
| **Aim of the Question** | To identify whether the target audience knows that leaves are used for the preparation of Ayurvedic medicine. |
| **Observation**  Forms response chart. Question title: Are you aware that plant leaves are used as an ingredient in the preparation of Ayurvedic medicine?. Number of responses: 114 responses. | |
| **Conclusion**  It is visible that 97.4% know that leaves are used for the preparation of Ayurvedic medicine. It is crucial that the target audience know the significance of medicinal leaves in Ayurveda because for users to engage with the proposed system they must know that leaves are used for the preparation of Ayurvedic medicine. | |
| **Question** | Are you able to identify Ayurvedic plants on your own? |
| **Aim of the Question** | To prove that there is a need for a system to help identify Ayurvedic plants. |
| **Observation**  Forms response chart. Question title: Are you able to identify Ayurvedic plants on your own?. Number of responses: 114 responses. | |
| **Conclusion**  The proposed system would not be of much use if all participants could identify Ayurvedic plants on their own. The system aims to assist the target audience in recognizing Ayurvedic plants by allowing them to upload images of leaves, as they are not able to recognize them on their own. It is evident from the responses to this survey question that 78.1% of the participants are not able to identify Ayurvedic plants on their own and 20.2% participants are only able to recognize some Ayurvedic plants. | |
| **Question** | To which plant do you think this leaf belongs to? |
| **Aim of the Question** | To find out if the given Ayurvedic leaf can be identified by the target audience. |
| **Observation**  Forms response chart. Question title: To which plant do you think this leaf belongs to? . Number of responses: 114 responses. | |
| **Conclusion**  It is evident that only 8.8% of participants could identify the correct identity of the images. This concludes that the majority of the target audience is not able to recognize Ayurvedic medicinal plants on their own. | |
| **Question** | Could you identify the plant to which the above shown leaf image belongs to? |
| **Aim of the Question** | To find out whether the audience identified the leaf or whether they guessed it. |
| **Observation**  Forms response chart. Question title: Could you identify the plant to which the above shown leaf image belongs to?. Number of responses: 114 responses. | |
| **Conclusion**  The responses to this question show that 93.9% of the target audience was not able to correctly recognize to which the plant the shown leaf image belonged to. This validates the need for a system to help recognize Ayurvedic medicinal plants in Sri Lanka. | |
| **Question** | If there were Ayurvedic medicinal plants in your home gardens, would you preserve them? |
| **Aim of the Question** | To identify if the target audience is willing to preserve Ayurvedic plants. |
| **Observation**  Forms response chart. Question title: If there were Ayurvedic medicinal plants in your home gardens, would you preserve them?. Number of responses: 114 responses. | |
| **Conclusion**  The responses to this survey question show that 92.1% are willing to preserve Ayurvedic medicinal plants and there is a chance that the other 7.9% of the participants are willing to preserve them. This shows that if the participants could recognize Ayurvedic medicinal plants available in their home gardens through this application they are willing to preserve them which would help prevent the extinction of these valuable Ayurvedic plants. | |
| **Question** | Have you ever heard of a system that could recognize Ayurvedic plant leaves while in their natural environment?  If yes, what is that system or application called? |
| **Aim of the Question** | To identify if there are similar systems and what they are called. |
| **Observation**  Forms response chart. Question title: Have you ever heard of a system that could recognize Ayurvedic plant leaves while in their natural environment? . Number of responses: 114 responses.  Plantum app | |
| **Conclusion**  It is evident that 97.4% of the participants have not heard of a system that could recognize Ayurvedic medicinal plants in their natural environment. This validates the research gap of this research and out of the other 2.6% only one participant had mentioned the name of such a system. The name of the application the participant has mentioned is ‘Plantum app’. However, the mentioned application is not designed to recognize Ayurvedic medicinal plants of Sri Lanka. | |
| **Question** | Do you believe that a system that could recognize Ayurvedic medicinal plants and display their properties while in their natural background would be useful? |
| **Aim of the Question** | To validate the usefulness of the proposed system. |
| **Observation**  Forms response chart. Question title: Do you believe that a system that could recognize Ayurvedic medicinal plants and display their properties while in their natural background would be useful?. Number of responses: 114 responses. | |
| **Conclusion**  All participants of this survey have agreed to this survey question. This concludes that a system that could recognize Ayurvedic medicinal plants in their natural environment is highly beneficial to the society so it is important to build a system that could accurately classify Ayurvedic medicinal plant of Sri Lanka in their natural background. | |

Table 5: Survey (Self-Composed)

### **2.5.3 Prototyping**

The author implemented CNNs from scratch and fine-tuned the hyperparameters to obtain the best accuracy and to reduce overfitting. It was found that CNNs can extract features and distinguish between the foreground and background well on their own. Multiple pre-trained models were trained using the transfer learning approach to find the models which are most appropriate for this research. The convolutional layers in pre-trained models were used as feature extractors and fully connected layers were added by the author for classification. It was found that using transfer learning was a better option than implementing CNNs from scratch because the dataset was not large enough for CNNs to perform well. The 3 best performing models were further fine-tuned to increase their performance and they were used to build an ensemble model using the averaging technique. Combining outputs from individual models and forming an ensemble model showed higher performance rather than using individual models.

### **2.5.4 Summary of Findings**

|  |  |  |  |
| --- | --- | --- | --- |
| **Finding** | **Literature Review** | **Survey** | **Prototyping** |
| Validate problem domain and research gap. | **Checkmark with solid fill** | **Checkmark with solid fill** |  |
| Using DL is better in image recognitions tasks than using traditional ML methods. | **Checkmark with solid fill** |  | **Checkmark with solid fill** |
| Convolutional Layers of pre-trained models act good feature extractors. | **Checkmark with solid fill** |  | **Checkmark with solid fill** |
| CNNs underperform when larger datasets are not used for training. | **Checkmark with solid fill** |  | **Checkmark with solid fill** |
| Ensemble models show greater performance than individual models. | **Checkmark with solid fill** |  | **Checkmark with solid fill** |
| The proposed system is useful. |  | **Checkmark with solid fill** |  |
| Ayurveda is a significant medicinal system in Sri Lanka. | **Checkmark with solid fill** | **Checkmark with solid fill** |  |
| Displaying the uses of the leaf of the plants that were recognized is important. |  | **Checkmark with solid fill** |  |

Table 6: Summary of Findings (Self-Composed)

## **2.6 Context Diagram**

The figure below is a context diagram that shows the interactions of the system with its external entities.

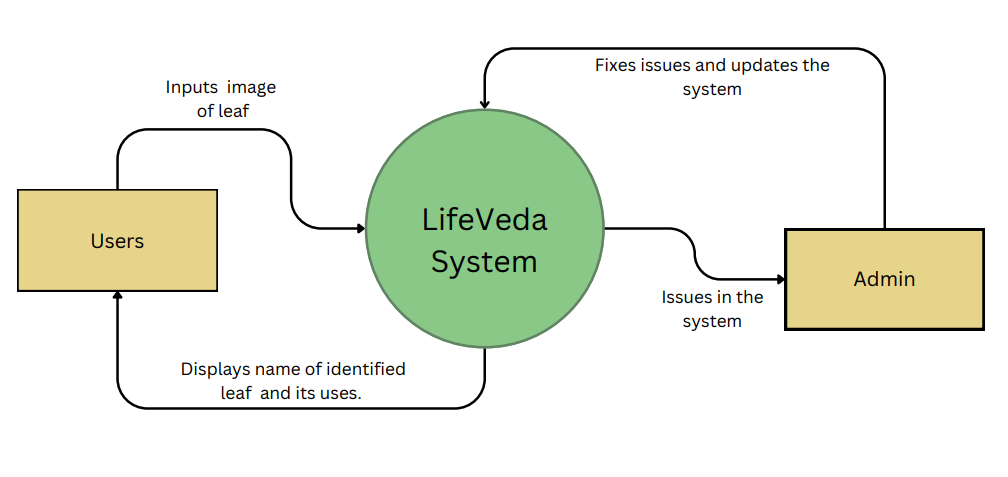


Figure 3: Context Diagram (Self-Composed)

## **2.7 Use Case Diagram**

A screenshot of a diagram

Description automatically generated

Figure 4: Use Case Diagram (Self-Composed)

## **2.8 Use Case Descriptions**

The descriptions of two main use cases are available in the tables below, the rest of the descriptions can be viewed in **Appendix A.**

|  |  |  |
| --- | --- | --- |
| **Use case name** | Upload image | |
| **ID** | UC 01 | |
| **Description** | The users can upload an image of a leaf to the system in order to get the result. | |
| **Participating actors** | User | |
| **Pre-conditions** | Image should be of a valid format (jpg,jpeg,png) and an internet connection should be available. | |
| **Extended use cases** | Display image format error | |
| **Included use cases** | Display plant name and uses | |
| **Main flow** | **Actor** | **System** |
| 1. Uploads an image of leaf. | 2. Checks if the image is of a valid format else displays and error.  3. If the image format is valid it is input into the ensemble model. |
| **Alternative flows** | None | |
| **Exceptional flows** | If a technical issue occurs during the use case it will result in a failure. | |
| **Post-conditions** | The uploaded image is input into the ensemble model to get the result. | |

Table 7: Use Case Descriptions of Upload Image (Self-Composed)

|  |  |  |
| --- | --- | --- |
| **Use case name** | View Ayurvedic plan list | |
| **ID** | UC 02 | |
| **Description** | The users is able to view a list of Ayurvedic plants in Sri Lanka and their properties. | |
| **Participating actors** | User | |
| **Pre-conditions** | None | |
| **Extended use cases** | None | |
| **Included use cases** | None | |
| **Main flow** | **Actor** | **System** |
| 1. Selects the option to view the Ayurvedic plant list.  4. Views the list of Ayurvedic plants. | 2. Fetches the list of Ayurvedic plants and their properties from the database.  3. Displays the list of Ayurvedic plants. |
| **Alternative flows** | None | |
| **Exceptional flows** | If the list could not be retrieved from the database an error message would be displayed. | |
| **Post-conditions** | View each Ayurvedic plant in the list. | |

Table 8: Use Case Descriptions of View Ayurvedic Plants (Self-Composed)

## **2.9 Requirements**

|  |  |
| --- | --- |
| Priority Level | Description |
| Must Have (M) | These features are mandatory to be developed in the system. |
| Should Have (S) | These features are important but is not mandatory to be developed. |
| Could Have (C) | These features are not important, and it is optional to develop these features. |
| Will Not Have (W) | These features will not be a part of the system. |

Table 9: Requirements (Self-Composed)

## **2.9.1 Functional Requirements**

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Priority Level** | **Use Case** |
| User should be able to upload an image into the system. | M | UC 01 |
| The format of the image uploaded by the user must be validated and an error message must be displayed if image with an incorrect format is uploaded. | M | UC 04 |
| System should display the plant of the leaf image which was uploaded by the user. | M | UC 05 |
| System should display the properties and uses of the image which was uploaded by the user. | M | UC 05 |
| User should be able to view a list of Ayurvedic plants in Sri Lanka. | S | UC 02 |
| User should be able to view details of any selected plant. | S | UC 03 |
| Admin should be able to update the model used for image classification. | C | UC 06 |
| Real time recognition of leaf image. | W | UC 01 |
| User should be able to capture image and upload it. | C | UC 01 |

Table 10: Table 3: Functional Requirements (Self-Composed)

## **2.9.2 Non-Functional Requirements**

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Category** | **Priority Level** |
| The image classification model should give the correct outputs therefore the accuracy of the model should be high. | Performance | M |
| The system should not take a long time to display the results of the classification model. | Performance | S |
| The application must be easy to use so that any technical or non-technical user can use it without difficulty. | Usability | M |
| A good security system to avoid the corruption of the system by hackers or viruses. | Security | M |
| The application must have a user-friendly UI. | Usability | S |

Table 11: Non-Functional Requirements (Self-Composed)

## **2.10 Chapter Summary**

This chapter focused on giving the reader an idea of the requirements of this research project and the possible stakeholders and their interactions through the rich picture diagram and stakeholder analysis. Furthermore, it focused on defining the requirement elicitation methodologies used by the user and the findings from those methods. The context diagram gives the reader an overview of how the system interacts with its users.

# **CHAPTER 03: DESIGN**

## **3.1. Chapter Overview**

## This chapter seeks to provide the reader with an understanding of the design objectives in our research project and an outline of the architecture of the system. Depending on the chosen design paradigm, the data flow diagrams have also been shown in a detailed manner. The system flow chart and the initial design of the UI was also depicted in this chapter.

## **3.2 Design Goals**

|  |  |
| --- | --- |
| **Design Goal** | **Description** |
| Accuracy | The main purpose of this research project is to build a system that can recognize Ayurvedic plants in order to prepare Ayurvedic medicine, learn about various Ayurvedic plants or identify Ayurvedic plants to preserve them and prevent their extinction. Therefore, it is crucial that this system gives out an accurate output. Achieving a higher classification accuracy is one of the main goals of this research project. |
| Usability | There is no specific target audience for this application, it can be used by any individual therefore the system should be designed in such a way that it is easy to be used by individuals under any age category and profession. |
| Quality | Since this application is designed to recognize Ayurvedic plants in their natural background an individual with knowledge on this research domain may doubt if the classification was made based on the actual leaf or the background used when training. Therefore, it is important to implement a feature to give the user and idea of based on which features of the leaf the classification was made to increase the trust of users in the application. |
| Scalability | There are large number of Ayurvedic plants in Sri Lanka therefore the application must be implemented in a way that it is possible to update the model by increases the number of classes the model can identify. |

Table 12: Design Goals (Self-Composed)

## **3.3 System Architecture Design**

### **3.3.1 Tiered Architecture Diagram**

A diagram of a data processing process

Description automatically generated

Figure 5: Tiered Architecture Diagram (Self-Composed)

### **3.3.2 Discussion of tiers**

#### **3.3.2.1 Presentation Tier**

The presentation tier represents what is visible to the user and where the user interacts with the system.

* Upload image – user can upload the image of the leaf using this option to be input into the classification model to get the output.
* Display results – the user can select this option in order to see the results of the classification and the properties and uses of the identified plant.
* View list of Ayurvedic plants – this option can be used by the user to get an output of the list of Ayurvedic plants in Sri Lanka and their properties.

#### **3.3.2.2 Logic Tier**

The logic tier is where the core functionalities of the system take place.

* Backend API – This is the backend of the application which will be implemented. Once the user selects and option the functionality related to that option will be executed here. When user selects the upload image option, the format of the image will be validated and if correct will be passed onto the image classifier. Data corresponding to the output of the classification will be fetched from the database and passed on to the presentation layer.
* Image classifier – the validated image will be preprocessed and input into the image classification model. The model will predict the class of the image.

#### **3.3.2.3 Data Tier**

The data required for the system is represented using the data tier.

* Ayurvedic plant details dataset – this dataset contains the information about the list of Ayurvedic plants in Sri Lanka. When the user selects the option to view this from the backend API this data is fetched and displayed to the user in the presentation layer.
* Ayurvedic plant image dataset – this dataset contains the images of the classes that were used to train the Ayurvedic leaf classification model.

## **3.4 System Design**

### **3.4.1 Choice of Design Paradigm**

As the design paradigm for this research project the Structured Systems Analysis and Design ( SSAD) will be followed.

## **3.5 Design Diagrams**

### **3.5.1 Sequence Diagram**

A diagram of a computer program

Description automatically generated

Figure 6: Sequence Diagram (Self-Composed)

### **3.5.2 Data Flow Diagram – Level 1**

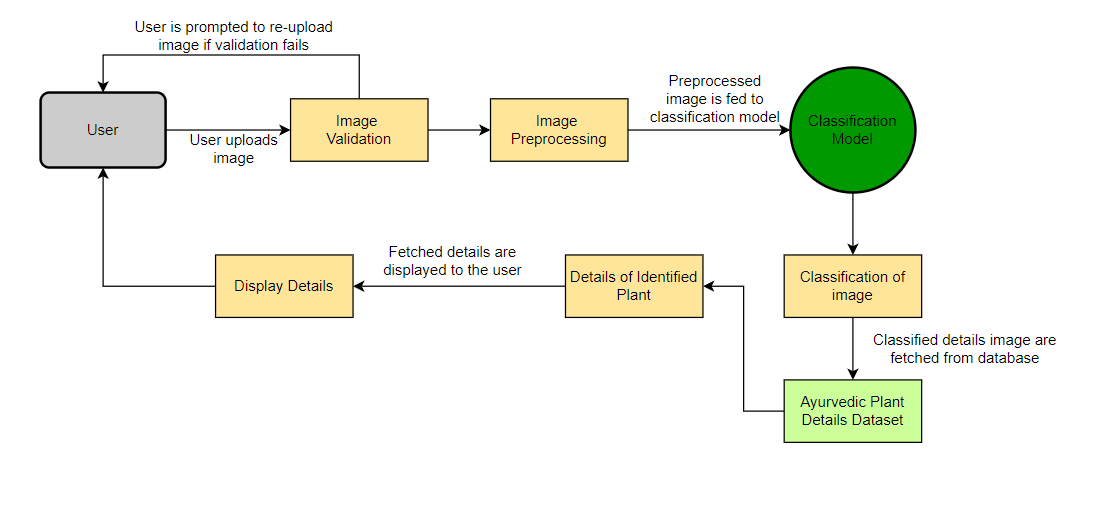


Figure 7: Data Flow Diagram Level 1 (Self-Composed)

### **3.5.3Data Flow Diagram – Level 2**

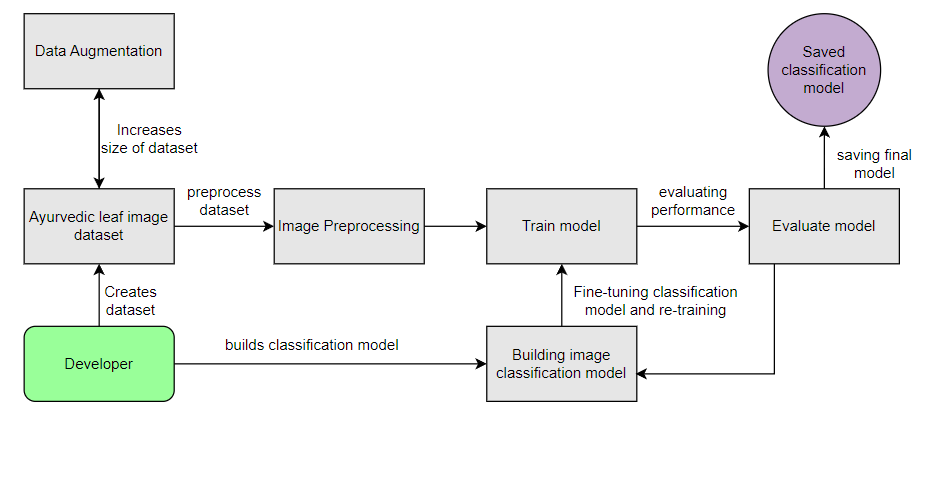


Figure 8: Data Flow Diagram Level 2 (Self-Composed)

### **3.5.4 System Process Flow Chart**

A diagram of a program

Description automatically generated

Figure 9: System Flow Chart (Self -Composed)

### **3.5.5 User Interface Design**

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

## A screenshot of a computer Description automatically generated A screenshot of a computer Description automatically generated

Figure 10: UI design of web application (Self-Composed)

## **3.6 Chapter Summary**

## This chapter outlines the design goals, system architecture, and data flow of the system using data flow diagrams. A system flow chart is employed to illustrate the functionality of the system. Furthermore, the design of the user interface for the proposed web application is presented within this section.

# **CHAPTER 04: INITIAL IMPLEMENTATION**

## **4.1 Chapter Overview**

## This chapter provides a summary of the technology stack utilized for the implementation of the system and the technologies that are expected to be used for the development of the user interface of the system. The selection of the dataset used for training the model has also been discussed. Additionally, the chapter outlines the implementation of the system's core functionality.

## **4.2 Technology Selection**

### **4.2.1 Technology Stack**

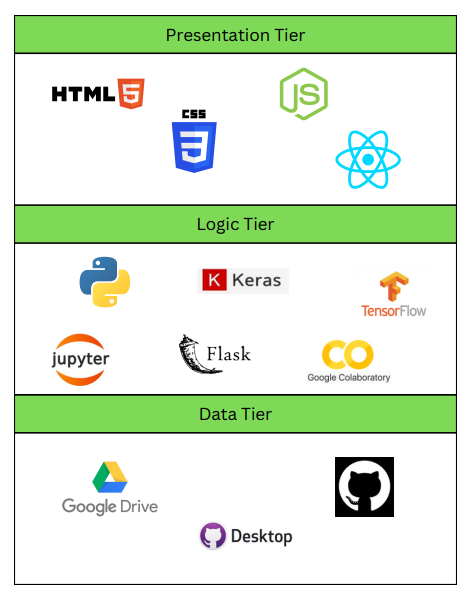


Figure 11: Technology Stack (Self-Composed)

### **4.2.2 Dataset Selection**

### According to the aim of this research project in order to train the ML model, a dataset containing leaf images of Ayurvedic medicinal plants captured in their natural background is essential. However as there were no such datasets publicly available the author had to create a dataset. The dataset consists of 8 different Ayurvedic medicinal plants, available in home gardens, that were captured while in their natural background. The images were captured from multiple angles under various light conditions. About 100 images from each class, 800 in total, were captured and to further increase the size of the dataset data augmentation was used. The final dataset contains 2080 images in total, 260 images per class. To prove that the model is generalized a separate dataset was created for testing. The test dataset contained 20 images per class, 160 in total. The test dataset was created by capturing images from different plants to the ones that were used for training.

### **4.2.3 Development Frameworks**

#### **4.2.3.1 Web Development Framework**

Flask and Django were the two web development frameworks the author had to choose from as Python was used for the implementation of the core functionality of the research project. Out of the two Flask is simpler and easier to learn compared to Django (Ghimire, 2020). Django is more suitable for larger projects with complex features and functionalities and Flask is preferred for smaller and simpler applications (Kinsta, 2023). As this research project serves the purpose to recognize Ayurvedic medicinal plants and does not contain very complex functionalities and features Flask appeared as the more suitable web development framework. Therefore, it was decided that Flask would be used as the web development framework.

#### **4.2.3.2 Model Development Framework**

The widely used framework for the development of DL models are PyTorch and TensorFlow. TensorFlow has Keras integrated in it which is not available in PyTorch. This makes it easier to build and train DL models (Boesch, 2023). Therefore, TensorFlow was chosen as the framework for the development of the model.

### **4.2.4 Programming Languages**

To develop the core functionality, which is the DL model, python was chosen as the main programming language. Python is the most widely used programming language for ML developments. It has many inbuilt libraries which makes the development process easier. JavaScript was chosen for the development of the user interface of the system.

### **4.2.5 Libraries**

|  |  |
| --- | --- |
| **Library** | **Justification** |
| numpy | It was used to convert image data into arrays to be input into the matplotlib function. |
| matplotlib | It was used to plot images from the dataset and the graphs for accuracy and loss of training and validation. |
| keras | It was used to import the necessary layers to build the DL models. |

Table 13: Libraries Utilized (Self-Composed)

### **4.2.6 Integrated Development Environments(IDE)**

The IDE used for the developments was Google Colab. This research involves the development of an image classification model. Image classification models take a very long time when trained on a CPU. Google Colab provides the access to GPUs freely, which makes the training process faster. This was the main reason for the choice of Google Colab as the IDE.

### **4.2.7 Summary of Technology Selection**

|  |  |
| --- | --- |
| **Component** | **Tools** |
| Development Frameworks | Flask, TensorFlow |
| Programming Languages | Python, JavaScript |
| Libraries | NumPy, |
| IDE | Google Colab, Jupyter Notebook |
| Version Control | GitHub |

Table 14: Summary of Technologies (Self-Composed)

## **4.3 Implementation of the Core Functionality**

## As a solution to the problem identified the author implemented two different techniques to find the technique that works best. One method includes the building of a CNN architecture from scratch to classify the Ayurvedic medicinal plants. The other method involves the training of multiple pre-trained models using transfer learning and combining the best performing models to form an ensemble model which takes the average of the predictions made from the individual models to give the final classification output. Both models underwent the same steps for data preprocessing. All images used for training were RGB images.

### **4.3.1 Data Augmentation**

The dataset used for training and testing of both models was created by the author. The train dataset contains 8 classes. The initial dataset with captured images contained about 800 images, the size of the dataset was further increased by using data augmentation. The final dataset used for the implementation of the models contain 2080 images of Ayurvedic leaves in total, out of these 480 images were used for validation and the rest of the 1600 images were used for training. The split was made by taking 80% of the images for training and 20% for validation. A separate dataset was created for testing which contains 160 images in total.

A computer screen shot of a program code

Description automatically generated

Figure 12: Data augmentation code

The above figure shows the python function which was used to produce the augmented images. Various parameters were passed like rotation, adjusting of brightness and horizontal and vertical flips in order to create augmented images and these parameters were constantly changed to ensure that images from various angles and lighting conditions were generated. The code was written in such a way that the images chosen randomly would only be used once to produce images using augmentation.

Figure 13: Sample augmented images from each class

As part of data preprocessing, all images were resized into a size of 224x224.



A screenshot of a computer program

Description automatically generated

Figure 14: Data Preprocessing

### **4.3.2 Implementation of Model**

#### **4.3.2.1 CNN Model**

The CNN model was built with a total of 11 layers. The first layer in the model is a rescaling layer, it scales all the pixel values in each image in the training dataset into a value between 0 and 1. The next 6 layers are convolutional and max pooling layers. Convolution layers are used for the extraction of features, convolutional layers with filters of size 64, 128 and 256 respectively have been utilized. Max pooling layers have been used to reduce the size of the output image from each convolutional layer.

A computer screen shot of a program code

Description automatically generated

Figure 15: CNN model code

#### **4.3.2.2 Ensemble Model**

The base models used for the implementation of the ensemble model were developed using transfer learning. Pre-trained models were fine-tuned to meet the requirements of this research project. The feature extraction layers in the pre-trained model were loaded and were set to be untrainable. This is because these layers have already been trained and there was no need to re-train the layers. The loaded layers were then flattened, and classification layers were added to perform classification of the classes available in the training dataset.

A screenshot of a computer program

Description automatically generated

Figure 16: ResNet50 trained using transfer learning

A screenshot of a computer program

Description automatically generated

Figure 17: VGG16 trained using transfer learning

A screenshot of a computer program

Description automatically generated

Figure 18: EfficientNetB5 trained using transfer learning

The ensemble model was built using the averaging technique. Three individual models were trained using transfer learning and then was combined by taking the average of the individual models to give the final classification output. All the best performing pre-trained models were found through literature review and all of them were trained to find out which models give the best accuracy and then the three models which gave the highest accuracy were chosen to build the ensemble model. The code for this approach is available in **Appendix C.**

A screen shot of a computer

Description automatically generated

Figure 19: Ensemble model code

## **4.5 Chapter Summary**

## This chapter demonstrates the implementation of the core functionality of the system and the technologies and programming language utilized in its development. Code snippets from the development process are also included for illustration.

# **CHAPTER 05: CONCLUSION**

## **5.1 Chapter Overview**

## This chapter explores the features integrated into the prototype and discuss how they differ from the initially outlined scope in the project proposal. The discussion extends to deviations in the schedule of deliveries and initial test results of the models. Finally, the improvements that will be made by the completion of the minimum viable product are stated.

## **5.2 Deviations**

### **5.2.1 Scope related deviations**

In the project proposal which was submitted by the author it was mentioned that the system would be designed to identify five types of Ayurvedic medicinal plants however the system was implemented to classify eight types of Ayurvedic medicinal plants using their leaves. The validation of the format of the images could not be implemented due to time constraints. This will be implemented as part of the MVP. The option to allow users to view information about other Ayurvedic plants will also be implemented for the MVP.

**5.2.2 Schedule related deviations**

The Gantt chart during the initial proposal of the project is available in **Appendix D.** Up to the completion of the project proposal there were no deviations from the schedule, the deviation from the schedule after that is listed in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Expected Delivery Dates** | **Actual Delivery Date** | **Status** |
| Literature Review | Start – 24/07/2023  End – 31/10/2023 | Start – 05/11/2023  End – 25/11/2023 | Completed |
| Requirement Gathering | Start – 01/11/2023  End – 27/11/2023 | Start – 22/01/2024  End – 05/02/2024 | Completed |
| Designing of the System | Start – 27/11/2023  End – 24/01/2024 | Start – 20/01/2024  End – 04/02/2024 | Completed |
| Selection of Tools & Technology | Start – 25/11/2023  End – 01/01/2024 | Start – 10/11/2023  End – 01/01/2024 | Completed |
| Testing & Evaluation | Start – 01/02/2024  End - 01/03/2024 | Start – 01/01/2024  End - Ongoing | In progress |
| Dissertation & Documentation | Start – 01/03/2024  End – 13/05/2024 | Ongoing | In progress |

Table 15: Schedule Deviations (Self-Composed)

## **5.3 Initial Test Results**

## Two models were implemented as a solution to the problem. The dataset used for testing was different to the dataset used for training. The initial test results for the two models are as follows.

The accuracy was taken as the evaluation metric for both models.

### **5.3.1 Initial Test Results for the CNN model**

A screen shot of a computer code

Description automatically generated

Figure 20: Initial results (CNN model)

In the initial testing of the CNN model, it gave out an accuracy of 50% for the testing dataset used.

### **5.3.2 Initial Test Results for the Ensemble model**

A screenshot of a computer program

Description automatically generated

Figure 21: : Initial results (Ensemble model)

In the initial testing of the Ensemble model, it gave out an accuracy of 96.88% for the testing dataset used.

## **5.4 Required Improvements**

### **5.4.1 Required Improvements for the CNN model**

The initial test accuracy for the CNN model is 50% which is a very low value for a good classification model. The CNN model requires fine-tuning to increase the overall test accuracy. Furthermore, more preprocessing would be done for the dataset by adding edge detection techniques to distinguish between the leaf and its background.

### **5.4.2 Required Improvements for the Ensemble model**

* All base models used for the ensemble model were fine-tuned in the same manner, as an improvement each individual model would be repeatedly fine-tuned and tested to increase their performance.
* In transfer learning it is recommended to preprocess the dataset in the same manner the initial dataset used for training of the models were preprocessed therefore the dataset would be preprocessed in the required manner for each model.
* There are many techniques that can be used to form deep ensemble models. Currently the ensemble model in the prototype was created using the averaging techniques. Other ensemble techniques will also be implemented to find the technique which gives the best performing model.

## **5.5 Demo of the Prototype**

**Upload an unlisted video to YouTube of a demonstration of the prototype. (must be a YouTube link).**

Duration: 10 minutes

Problem: 3 minutes (max)[like a pitch]

Solution: 5 minutes (explain your solution with a demonstration)

Improvements you can do: 2 minutes (a reflection of your solution)

Also provide a link to the code.

(You must run your software at this point of time. If you can’t run, people will conclude you don’t have a prototype)

## **5.6 Chapter Summary**

In this chapter, the author outlines the deviations in functionality between the prototype and the features proposed in the initial project proposal. The enhancements that will be made to the system are discussed and the links to the demonstration of the prototype and code are also listed.

**REFERENCES**

Weragoda, P.B. (1980). The traditional system of medicine in Sri Lanka. Journal of Ethnopharmacology, 2(1), pp.71–73. doi:https://doi.org/10.1016/0378-8741(80)90033-1.

‌Li, Y. (2022). Research and Application of Deep Learning in Image Recognition. [online] IEEE Xplore. doi:https://doi.org/10.1109/ICPECA53709.2022.9718847.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I. (2017). Attention Is All You Need. [online] arXiv.org. Available at: https://arxiv.org/abs/1706.03762.

Sharma, S & Gupta, C 2015, 'A review of plant recognition methods and algorithms', International Journal of Innovative Research in Advanced Engineering (IJIRAE), vol. 2, no. 6, pp. 2349-2163.

Pushpanathan, K., Hanafi, M., Mashohor, S., & Fazlil Ilahi, W. F. 2020, 'Machine learning in medicinal plants recognition: A review', Artificial Intelligence Review, doi: 10.1007/s10462-020-09847-0.

Jayalath, D. 2019, 'Identification of Medicinal Plants by Visual Characteristics of Leaves and Flowers', doi:https://www.researchgate.net/profile/Dasuni-Nawinna/publication/337534280

Azeez, Y and Rajapakse, C 2019, 'An Application of Transfer Learning Techniques in Identifying Herbal Plants in Sri Lanka.' [online] Available at: https://ieeexplore.ieee.org/document/8842681 [Accessed 4 Oct. 2023].

Senanayake, U. and De Silva, D., 2022, Identifying Medicinal Plants and Their Fungal Diseases, Available at: https://ieeexplore.ieee.org/document/10002624 [Accessed 4 Oct. 2023].

Jayanka, M. and Fernando, T.G.I. (2020). Recognising Ayurvedic Herbal Plants in Sri Lanka using Convolutional Neural Networks. Vidyodaya Journal of Science, [online] 23(01). doi:<https://doi.org/10.31357/vjs.v23i01.4680>.

Kanda, P.S., Xia, K. and Sanusi, O.H. (2021) ‘A Deep Learning-Based Recognition Technique for Plant Leaf Classification’, *IEEE Access*, 9, pp. 162590–162613. Available at: https://doi.org/10.1109/ACCESS.2021.3131726.

Chompookham, T., Surinta, O., 2021. Ensemble Methods with Deep Convolutional Neural Networks for Plant Leaf Recognition. https://doi.org/10.24507/icicel.15.06.553

Kankanamalage, T.N.M., Dharmadasa, R.M., Abeysinghe, D.C., Wijesekara, R.G.S., 2014. A survey on medicinal materials used in traditional systems of medicine in Sri Lanka. Journal of Ethnopharmacology 155, 679–691. https://doi.org/10.1016/j.jep.2014.06.016

Ghimire, D. (2020). *Comparative study on Python web frameworks: Flask and Django*. Www.theseus.fi. <https://www.theseus.fi/handle/10024/339796>

Kinsta®. (2023). *Flask vs Django: Let’s Choose Your Next Python Framework*. [online] Available at: <https://kinsta.com/blog/flask-vs-django/>.

‌ Boesch, G. (2023). *Pytorch vs Tensorflow: A Head-to-Head Comparison*. [online] viso.ai. Available at: https://viso.ai/deep-learning/pytorch-vs-tensorflow/#:~:text=In%20general%2C%20TensorFlow%20and%20PyTorch.

‌

# **Appendix**

**Appendix A**

**USE CASE – View properties of selected plant.**

|  |  |  |
| --- | --- | --- |
| **Use case name** | View properties of selected plant. | |
| **ID** | UC 03 | |
| **Description** | The user can view the properties of individual plants that are selected. | |
| **Participating actors** | User | |
| **Pre-conditions** | None | |
| **Extended use cases** | None | |
| **Included use cases** | Display plant name and uses. | |
| **Main flow** | **Actor** | **System** |
| 1. Selects the option to view the properties of a selected plant. | 4. View the displayed details of the selected plant. |
| **Alternative flows** | None | |
| **Exceptional flows** | If the details of the selected plant could not be retrieved from the database an error message would be displayed. | |
| **Post-conditions** | None | |

**USE CASE – Display image format error**

|  |  |  |
| --- | --- | --- |
| **Use case name** | Display image format error | |
| **ID** | UC 04 | |
| **Description** | If the uploaded image is not of an appropriate format the system will display an error message. | |
| **Participating actors** | None | |
| **Pre-conditions** | Image must be uploaded by user. | |
| **Extended use cases** | None | |
| **Included use cases** | None | |
| **Main flow** | **Actor** | **System** |
| 2. Views error message.  . | 1. Displays error message. |
| **Alternative flows** | None | |
| **Exceptional flows** | None | |
| **Post-conditions** | User uploads an image with the correct format. | |

**USE CASE – Display plant name and uses**

|  |  |  |
| --- | --- | --- |
| **Use case name** | Display plant name and uses | |
| **ID** | UC 05 | |
| **Description** | The system displays the name of identified plant and its uses. | |
| **Participating actors** | None | |
| **Pre-conditions** | Image must be uploaded by user. | |
| **Extended use cases** | None | |
| **Included use cases** | None | |
| **Main flow** | **Actor** | **System** |
| 1. View the displayed properties of plant and its uses. | 1. Displays the properties and uses of the image uploaded by the user. |
| **Alternative flows** | None | |
| **Exceptional flows** | If plant name is not found in the database and error message will be displayed to the user. | |
| **Post-conditions** | None | |

**USE CASE – Update model**

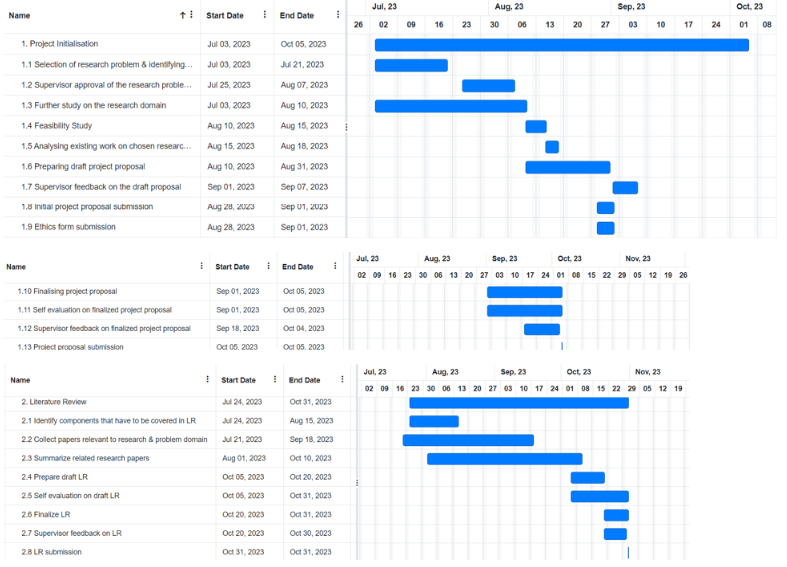
|  |  |  |
| --- | --- | --- |
| **Use case name** | Update model | |
| **ID** | UC 06 | |
| **Description** | The admin updates the deep ensemble model. | |
| **Participating actors** | Admin | |
| **Pre-conditions** | Dataset used to train model must be preprocessed. | |
| **Extended use cases** | None | |
| **Included use cases** | None | |
| **Main flow** | **Actor** | **System** |
| 1. Changes hyperparameters of the model.  4. Re-trains the model. | 2. Gets updates ensemble model. |
| **Alternative flows** | None | |
| **Exceptional flows** | None | |
| **Post-conditions** | Updated model is used for image classification. | |

**APPENDIX B**

**A screenshot of a graph

Description automatically generated**

**APPENDIX C**



A screenshot of a calendar

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated