CUSTOMER REQUIREMENT SPECIFICATION

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | https://lh4.googleusercontent.com/proxy/YA9Xoqs7jhpeuwrEjwhdi_EVSCDwUdpr72V-2YHZ2lz2y1FaqityK8c8RlZRTvUDEw3Y2TekyGNi07wcREil5Ez3ii80dA-DE8G6HAQjEmJVz8W32Wy2uaDAWwuZs6uPZtJp2zrUJ_Qps2T1CUmSpuPR8dk2XA=w128-h144-k-no | | | **Document Ref.:** | | |  | |
| **Version No.:** | | |  | |
| **Date :** | | |  | |
| **Project Name:** | Fruit Stalk Recognition | | | | | | |
| **Project Code:** |  | | | | | | |
| **Status:** | **Draft** / **Current** / **Superseded** | | | | | | |
| **Document Type:** | **Controlled** / **Uncontrolled** | | | | | | |
| **FRUIT STALK RECOGNITION** | | | | | | | |
| **An automated system to recognize and favor fruit picking** | | | | | | | |
| **Prepared By:** | | | | **Reviewed By:** | | | |
| **Name** | | **Date** | | **Name** | | | **Date** |
| Sandeep Pvn  01FB14ECS207 | |  | |  | | |  |
|  | | |  |
|  | |  | | **Approved By:** | | | |
| **Name** | | | **Date** |
|  | |  | |  | | |  |
|  | | |  |
| **Distribution List** | | | | | | | |
| **Project Representative(s)** | | | | | **Guide Representative(s)** | | |
|  | | | | | 1. Dr. Antony | | |

TABLE OF CONTENTS

[Definitions, Acronyms and Abbreviations 3](#_Toc43210952)

[Change History 4](#_Toc43210954)

[1.0 Introduction 5](#_Toc43210955)

[1.1 Scope 5](#_Toc43210956)

[2.0 Product Perspective 5](#_Toc43210957)

[2.1 User Characteristics 5](#_Toc43210958)

[2.2 General Constraints, Assumptions and Dependencies 5](#_Toc43210959)

[2.3 Risks 5](#_Toc43210960)

[3.0 Requirements List 6](#_Toc43210962)

[3.1 Module / Scenario 1 6](#_Toc43210963)

[3.2 Module / Scenario 2 6](#_Toc43210964)

[3.3 Module / Scenario n 6](#_Toc43210965)

[4.0 System Architecture 6](#_Toc43210961)

[5.0 User Interfaces 7](#_Toc43210970)

[6.0 External Interface Requirements 7](#_Toc43210966)

[6.1 Hardware Requirements 7](#_Toc43210967)

[6.2 Software Requirements 7](#_Toc43210968)

[6.3 Communication Interfaces 7](#_Toc43210969)

[7.0 Performance Requirements 7](#_Toc43210971)

Definitions, Acronyms and Abbreviations

1. SSH: Secure Shell
2. MAVLink: Micro Air Vehicle Link
3. FSR: Fruit Stalk Recognition
4. ESP: Electronic speed control

Change History

This section describes the details of changes that have resulted in the current CRS document.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Date** | **Document Version No.** | **Change Description** | **Reason For change** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 

# 1.0 Introduction

The purpose of this document is to give a detailed description of the requirements for the “Fruit Stalk Recognition(FSR)”. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. The main users for this project would essentially be farmers and industries pertaining to fruit farming sectors.

## Scope

The FSR is an autonomous system that aides a drone or any vision based robotic system to recognize and further navigate to a desired position/s on extractable fruits present on the tress at orchards.

There would be minimal human intervention and would fully aid the respective customer to pick fruits within their compounds. It would be a necessary step to train the closed environment for fine tuning. and probably set a few parameters during the first few trials in case of discrepancies during execution.

Furthermore, the drone needs a docking point and an inbuilt “or” cloud-based memory to store the coordinated data collected over a run. The software can later be extended to perform analytics on the data collected over time.

# Product Perspective

This product is a fully self-contained and individual product that only needs to gain exposure to the working environment.

The hardware platforms used will be a stereo camera integrator and drone kit on a stable python controllable drone.

The software platforms include python, tensor flow, opencv and a basic web portal.

The product will involve calibrating, simulating the drone and utilizing some mechanical fruit extraction method on one aspect. The other aspect would be creating a real time recognition system and also storing the virtual 3D co-ordinate data and using it for analytics using a backend database.

## User Characteristics

There are two types of users – Farmer and Administrators.

The farmers who might be having an option to set a timer using a web or phone portal or direct interface with the drone.

The administrator/Developer would calibrate the system occasionally by coding the drone kit integrated to python.

## General Constraints, Assumptions and Dependencies

**The limitations are:**

* Government privacy policy regarding flying drones.
* Battery life of the drone.
* Clarity/Resolution of camera on drone.
* Inability of the simulations to provide a perfect scenario of drone maneuvering.
* Safety and crash conditions regarding the drone and surrounding environment.

**The assumptions are:**

* The fruits have a distinct feature separating them from the rest of the foliage.
* The fruits are all accessible visually and physically to the camera and drone.
* There is power supply for drone docking and necessary network environment for communications.

# Requirements List

## Module / Scenario 1

|  |  |
| --- | --- |
| **Reqmt 1** | **Requirement** |
| Drone | A programmable drone which is compatible with drone kit library and has necessary communication models. (Along with a docking kit). Includes appropriate FCU. |

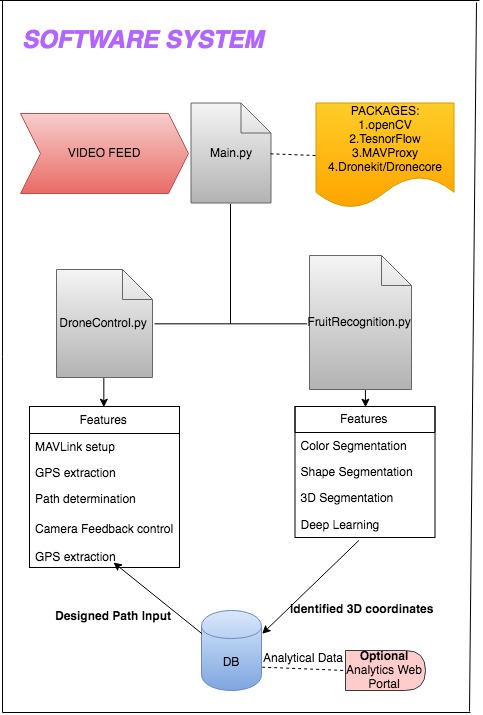
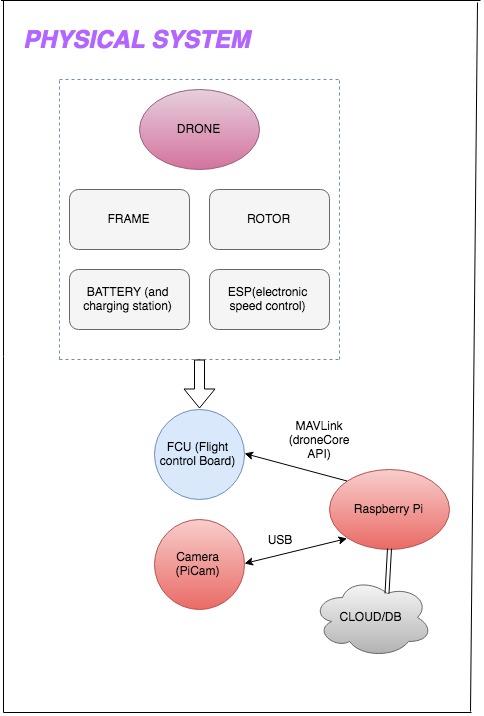
## Module / Scenario 2

|  |  |
| --- | --- |
| **Reqmt 2** | **Requirement** |
| Stereo Camera | A good resolution light weighted stereo camera with wireless interfaces. |

## Module / Scenario 3

|  |  |
| --- | --- |
| **Reqmt 3** | **Requirement** |
| Computing System | A high-power computation unit and a storage unit to perform necessary computer vision task. |

# System Architecture



# External Interface Requirements

## Hardware Requirements

The simulation can be performed using SITL and dronekit-SITL. A Drone with necessary rotors, frames, battery, landing gear, ESP (electronic speed controls) integrated to a flight control board (preferably PXFMINI/Pixhawk) and controlled using a Raspberry Pi. Stereo or Mono camera will be integrated to the raspberry Pi via USB. The drone is further communicated using ZigBee modules for manual intervention.

## Software Requirements

This is run as a OSX to raspberry PI integration.

A basic SQL database to store the 3D coordinates.

Python integrated with drone kit to control the drone.

Tensor Flow, opencv for recognition algorithms.

## Communication Requirements

The communication protocols used are MAVProxy via a MAVLink protocol and SSH channel over Wi-Fi or ZigBee.

# Performance Requirements

(Tentative to change, vision-based parameters and metrics will be implemented)

* 1 drone caters to 1 user at a time only. (1:1)
* The analytical data may be used on the cloud by multiple users. (1:N)
* The primary database will consist of configuration files and a DB of minimum records to store path and waypoints to the 3D coordinates of the fruit.
* The program can be reused on multiple drones at different locations or eventually useful to coordinate drones together.
* There is a continuous input of data to and from the drone and on-board communication system (Raspberry Pi or Pixhawk Mini) as the product is real-time specific.
* Connections might include ZigBee, Wi-Fi, SSH, telemetry.