Planta Python Programming Assignment

Table of Contents

Introduction

Planta makes robotic devices that operate in greenhouses, taking pictures of the greenhouse crop while moving over them, also known as scanning. These pictures are then uploaded to our cloud storage for further analysis.

Some of our robotic ADI devices have multiple cameras, which capture images simultaneously whilst scanning the crops. These simultaneously taken pictures are kept together in a folder. This collection of images in a folder is what we call a **snapshot**.

Once taken, snapshots are often processed before we upload them to the Planta ADI cloud storage.

Below, a simplifying diagram can be found to visualize this process:

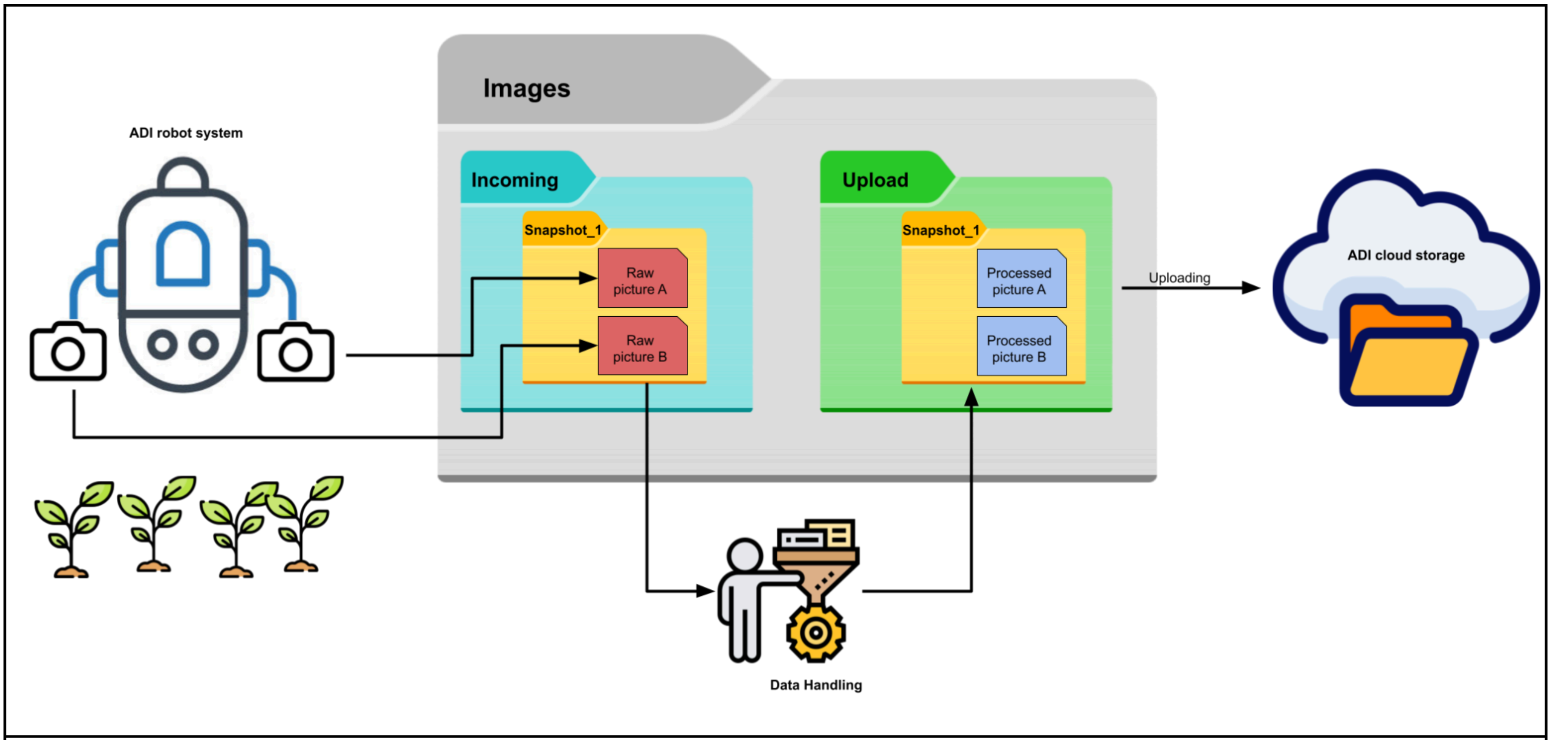

Figure 1: picture taking to upload process

This assignment is about the **Data Handling** subprocess of the above defined process.

The goal of this assignment is to test your python programming skills within a certain time frame, around 3 – 4 hours. We will pay attention to the following:

- How easy is it to read and understand your code?
- How easy is it for a colleague to contribute to this code base?
- What good practices are being applied?
- Does the code include bugs or bad practices?

We have created a very minimal Python scaffold which you can build upon in order to complete the assignment. The scaffold can be used as a starting point but feel free to rewrite existing code. The provided code is not necessarily of high quality or easily extended.

You are also allowed to rewrite the whole code, if you think that you can do it better!

Within the **Data Handling** subprocess, we want you to implement a couple of features and fixes. If you have time left, you can start on the bonuses.

Data Handling Process

The following section outlines the current process implemented within the scaffold.

Waiting for Raw Snapshots

As also outlined in the above diagram (Figure 1), the provided scaffold waits for snapshots to arrive in the **incoming** folder. We have provided four example snapshots in this folder, named: **snapshot_1, snapshot_2, snapshot_3** and **snapshot_4**. Each snapshot will contain images taken by the ADI robotic system.

Note, that not all snapshots created by our system will be complete. This is the case when:

- Not all snapshots will have exactly 2 images. For example, **snapshot_3** only has 1 image.
- Images are corrupt.

In these cases, the snapshot should not be processed.

Processing Snapshots

After the images arrive in the **incoming** folder, processing steps are applied. In the provided scaffold, there is only 1 processing step. This step watermarks the images with a small logo, also available in the repository, in the top left corner. Once successfully processed, the original incoming snapshot gets deleted.

Saving Processed Snapshots for Upload

After completion of the processing steps it saves the modified images in the **upload** folder, in a folder with the same name as the original snapshot.

Features, Fixes and Bonuses

This section lists what we would like you to implement within the data handling process.

Features

1. Add the following image processing steps:
    a. Compress all images with a configurable quality.
    b. Downscale all images to 500x500 pixels.
    c. Rotate image_b of each snapshot 180 degrees.
2. The ability to handle different file names.

Fixes

1. Snapshot **snapshot_4** has a corrupted image. This snapshot including the images are copied to the **upload** folder. Change the code so that it doesn't move snapshots with corrupted images to the **upload** folder.

Bonuses

1. Add the ability to get a summary of the state of all snapshots (both raw and processed) .
2. Additional unit test to improve code coverage.
3. If you can think of a cool feature, don't hesitate to add it and tell us about it!

Deliverables

Once either the allocated time is up, or you think that you have proved yourself to the best of your ability, we expect to find the following within a .zip file that you send back to us:

Working Code

Within the allocated time, attempt to complete as many of the features, fixes and bonus as possible, keeping in mind what we will be paying attention to.

Documentation

While working on the assignment you will be making design decisions. Do not hesitate to use the readme to enlighten us on the choices that you have made and how to use the code that you have written. Do this in the `explanation_and_reasoning.md` file that is sent with the assignment code.