Simulation of the Effects of Dust on Agriculture

Arianne Fels

Sierra Young

May 27th, 2024 - August 6th, 2024

Introduction

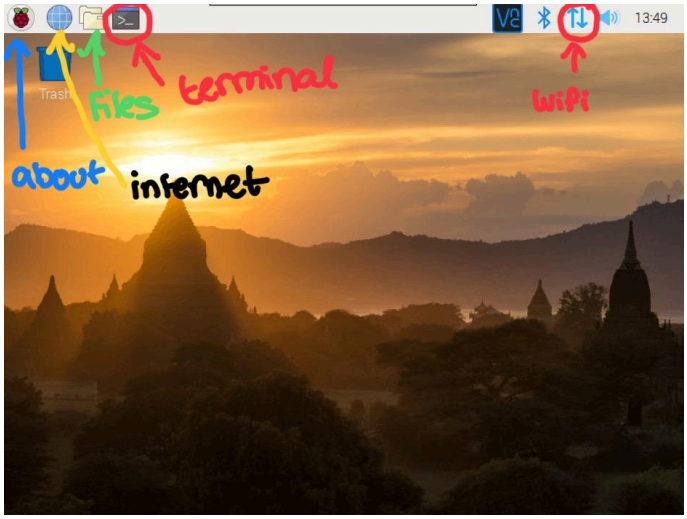
In order for Utah to continue to improve its consciousness regarding climate, droughts, and agriculture, it is critical for younger generations to be exposed to the effects of these developing problems and gain awareness.

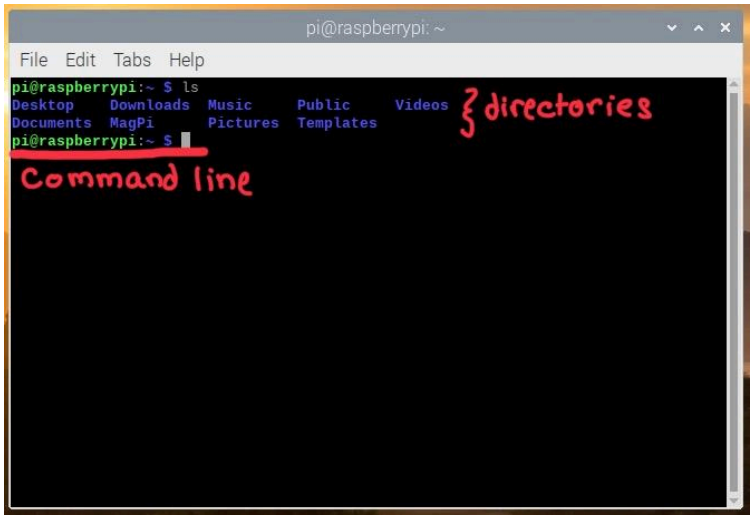
Using a RaspberryPi, along with PlantCV and other image analysis software, lettuce, chinchilla dust bath, and diatomaceous earth, the large-scale effects of dust on agriculture can be simulated for children at public libraries all over the state of Utah.

Growing lettuce in a 3 X 6 tray and using a RaspberryPi PiCamera2, we can watch the change in growth and green frequency depending on dust quantity and dust type. Although the Pi should contain all software and code to correctly complete this simulation, here are some instructions on installation, software, and Pi usage.

Before beginning to work with the Pi, it is important to connect it to the wifi once it is plugged in and turned on. Here is the RaspberryPi official documentation on how to connect to the wifi. https://www.raspberrypi.com/documentation/computers/getting-started.html#wi-fi

Upgrading the Pi





Next, we need to update the software/OS on the Pi. On the top left of the screen, there is a black square which is the icon for the terminal. Click on it. Then in what is called the "command line" (which is a green line with the name of the Pi and the directory you are currently in), you are going to type `sudo apt update` (always hit enter after typing your command) then, `sudo apt upgrade`. The terminal will load information about the upgrade, and you will type `y` to continue with the upgrade process.
Once finished, the Pi will relay a message about the upgrade being a success. You can now close this terminal window!

Installing the Virtual Environment

The rest of the installation described in this manual should already be done on the Pi. However, there is always room for error when transferring software, so if software or packages are lacking, this should help solve those errors. Also, if you would like to know more about installation processes or would like to understand the software you are using better, feel free to keep reading!

What is a Virtual Environment?

In order to run our PlantCV and OpenCV software, we will be using a virtual environment. A virtual environment is like a container with all of the Lego pieces you want for a specific project. Your computer is the big box of Legos with all the pieces for every project, but the virtual environment only holds our pieces for the project we plan on building. We can build any sort of project with an instruction manual from 1932 or 2024 without messing up our big box, since our big box only uses manuals from 2024. We can also throw away our virtual environment box with all its pieces if we wanted to without affecting the computer box. Basically, the virtual environment allows us to use software with conflicting versions, and keep our project together without causing any harm to our computer. Although it is not necessary for a project such as ours since we are not using the Pi for anything other than imaging, it is good practice.

Which Environment for the Pi?

The virtual environment we are using for our Pi's is a version of the Anaconda virtual environment. It is tailored to a 64 bit Linux system which is what the Raspberry Pi OS is. Like Windows and macOS, Linux is just another type of operating system. The macOS is surprisingly similar to Linux. To download the Miniforge environment, we need to be connected to the wifi on the Pi. Type this link exactly into any of the browsers on the Pi, and it will download the virtual environment automatically: Miniforge download

Downloading Miniforge

Now we need to check that Miniforge was downloaded correctly before we can truly install it to our system. We are going to open the terminal again and on the command line, type `cd Downloads` which means "change directory Downloads," which moves us into downloads. Nothing should happen, except "Downloads" will be added to the command line. On the next line, type `ls` which means "list files." It will now show all of the files in the Downloads directory.

If you reference the image that illustrates the command lines, the first command is “ls” and it lists all of the directories under the user. We should see `Miniforge3-Linux-aarch64.sh` in those files.

When we “cd Downloads” then “ls”, it will look similar, except with all the files within Downloads.*

Installing Miniforge

Installing software is like baking a cake from all the ingredients we downloaded, so we need to install our virtual environment. On the next line in our terminal, type `bash Miniforge3-Linux-aarch64` to install the environment. There will be a line asking you to press enter to continue the installation. Hit enter. There will be lines and lines of terms that you don’t need to read, and to jump to the bottom to agree, press “q”. There will be a prompt asking for a “yes” to accept them. Type “yes”. The installation process will then ask you if its automatic installation location is correct. Don’t worry too much about this, it is. Hit enter to confirm. Then when the installation asks for another yes, give it. We don’t say no to computers much around here!

We should get a message saying that the installation was properly completed. But it’s always important to double check before we move on.

Checking Installation

To do so, we are going to close the terminal and open a new one. In front of the command line, there should be a new word, `(base)` which means we’re in a base virtual environment and the installation worked properly. But we do not want to always open a terminal and be in a base environment, we often don’t want to be in a virtual environment at all. To undo this automatic assumption, we are going to type `conda config --set auto_activate_base false` into the command line. Conda is a command that communicates with the virtual environment. Then, close the terminal window and open a new one. We should no longer be in `(base)` anymore!

Creating the Virtual Environment

Now that our virtual environment is installed correctly, we can create the one we plan on using for our image analysis. In the terminal, type `conda create -n planty`. This command is saying create a virtual environment named (-n) whatever you would like (planty). I called my

virtual environment planty, but you may call it whatever you'd like! The creation process will be displayed, and when asked if you would like to proceed, type `y`.

Checking Proper Installation

We have our virtual environment created, however, it is not automatically activated. Before we activate it, we should check it was properly created. In the command line, type `conda info --envs`. We should see `base` and `planty` (or whatever you named yours!). If we were in planty, there would be a star by the path to planty. A path is the treasure map to our treasure chest, planty. Our computer is a house, and the directories and folders are rooms within the house. We can follow the path through the rooms like a map, and find our treasure, planty!

Activating/Deactivating the Virtual Environment

Whenever we want to enter our virtual environment, we will need to activate it. We do this by typing `conda activate planty` into a new line. We type this everytime we want to activate our environment. When planty is activated in one terminal, we can open a new terminal and it will not be in planty. If we want to deactivate planty in our terminal without simply closing out of it, we type `conda deactivate planty`. Once activated, we should see `(planty)` in front of our command lines from now on until deactivation.

Installing Imaging Libraries

Here comes the unfortunately tedious part of the installation process. Computers are very picky when it comes to installing software. If you've made it this far without your Pi blowing up angrily, congratulations! The Pi is not a great communicator, so when you do something slightly incorrectly, it often won't tell you how to fix it (at least not clearly). There are many possible errors, and hopefully none will arise, because they can be tricky to fix.

Installing pip

The first thing we need to install is pip. Pip is a package manager for Python, which allows us to install libraries that we need. Therefore, in order to install those libraries, we need pip first. Before we install, double check you are in the virtual environment! `(planty)` should be in front of the command line. Type `conda install pip` on the command line. When asked, press `y` to proceed.

Checking Python Version

Once it is installed, we want to make sure we have the correct Python version installed. Some of our libraries can only run on new versions of Python. On the command line, type `python`. We should see the version of Python installed. We want to make sure it is 3.10 or newer, which it should be automatically.

The REPL

By typing `python`, we are now in the REPL which has these three >>> arrows. The REPL stands for Read-Eval-Print Loop. It behaves like a magic board that can solve problems immediately when assigned, with no limit. When we use it, it will be used to run an import command, which is the problem it is solving. We do not need the REPL right now, so we will exit out of it with CTRL d.

Installing OpenCV

One of the most important libraries we are using is OpenCV, which has algorithms for analyzing images. We are going to use pip to install it. Type `pip3 install opencv-python` into the command line. Now comes time for the REPL to import this library. Type `python` into the command line, opening the REPL. In front of the >>>, type `import cv2`. If no error pops up, we should be in the clear! We still want to check that the version is compatible with our PlantCV library and Python version. Type `cv2.__version__` (there are two underscores on both sides of version) into the REPL. It should display a version, and we want to make sure it is at least 4.10.0.

Exit the REPL using CTRL D so we can install PlantCV and JupyterLab!!

PlantCV and JupyterLab

To install PlantCV, which is our plant analysis software library, type `pip3 install plantcv` into the command line. We will do all our imports at the end. To install JupyterLab, which is a notebook for easier manipulation of Python code, type `pip3 install jupyterlab`. Once completed, we need to install dependencies, which are packages to help our code run as smoothly as possible. Type `pip3 install ipympl` to install the ipympl dependency. Ipympl is allows us to create cool graphs! Now we can open the REPL to import all of our newly installed libraries and packages. Type `python` to open the REPL. We are going to start by importing PlantCV, so type `import plantcv` in front of the three arrows. It might take a few minutes. Once completed, which you will know by >>> appearing beneath as a new line, we can import JupyterLab by typing `import jupyterlab`. Then, on a new REPL line, type `import ipympl`.

When importing correctly, there should be no error and a new line in the REPL should appear underneath

I am using PlantCV version 4.3.1. If you install it will most likely not be this version, and there could potentially be differences

Now we can close the REPL with a CTRL d.

Now we can use JupyterLab and run our photo analysis code. After making sure we are still in our virtual environment, we type `jupyter lab` into the command line. It should then open JupyterLab as a new window in the browser! To close out of JupyterLab, CTRL c in the terminal, then type `y` to confirm the exit.

Creating VE for Running Photo Code

What is Venv?

As if one virtual environment was enough! To run the code for our picture taking, where our Pi takes a picture once per day, we need another type of virtual environment- one created by Python called a “venv”. To use the Picamera2 and to install a Python library called Schedule, we need this specific type of environment. Schedule is a library that will take our picture once per day, so it is crucial to have! When doing this, make sure you are not in your planty environment!

How to Create a Venv

To create our virtual environment, we are going to type:

`python3 -m venv --system-site-packages envi` into the terminal. “envi” can be substituted for whatever name you would like for the environment. This is just the one I chose. Next, we are going to activate our environment by typing:

`source envi/bin/activate`. Of course if you name your environment something different, “envi” will be replaced by the chosen name. Now, we should see `(envi)` in front of our command lines from now on until deactivation.

Since all of the PiCamera2 software was included in the `--system-site-packages`, we just need to install Schedule! We do this by typing `pip install schedule` on the command line. We don’t need to install pip beforehand because we are using a Python virtual environment. When completed, it should say “Successfully installed schedule” with a version.

To deactivate this environment, simply type `deactivate` on the command line.

Running our Code

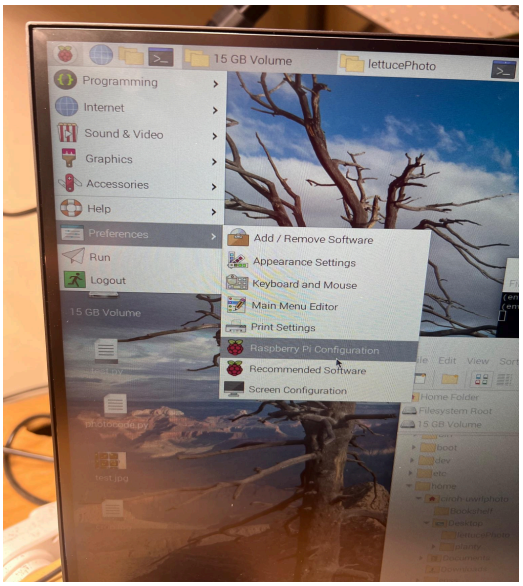
To run our photocode, we must always be in the terminal and in our “envi” environment. I will talk more about our code later, but the code used to take our once per day picture is called photocode, and it exists on the Desktop of your Pi. Before we can run it and once we are in “envi,” we need to be in the Desktop directory, so we’re gonna type `cd Desktop` on the command line. Now since we’re on the Desktop, we can run our code. We do this by typing `python photocode.py` on the command line. Nothing should appear underneath, no new command line, no message, nothing. That’s how we know it’s busy running our code!

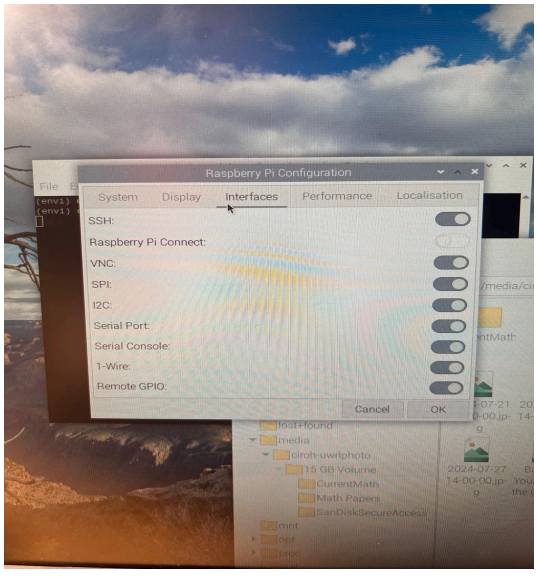
If you want to stop the code, simply CTRL D and it will cancel the operation.
The pictures are saved to a file called lettucePhotos on the Desktop

The PiCamera2 and Code

Sadly, the PiCamera2 can be just as much of a hassle as the rest of the Pi. PiCamera installation should help you with physically connecting the camera to the Pi. Sometimes (more often than not) the camera is finicky and does not like to connect easily. Don’t get discouraged!

To test that our camera is connected, we use a command called `libcamera-hello` on the command line. A box with a view from the camera should appear and disappear temporarily. If not, try turning off the Pi and plugging the camera back in. Sometimes, in cases where this does not work, we need to turn the input/output settings of the Pi off and back on. Make sure the camera is unconnected, then go to Preferences, then Raspberry Pi configuration. Then, go to Interfaces and turn off ALL the toggles. The Pi will ask to restart, say yes. Then repeat the previous steps, but toggle the Interfaces switches back on. (Don’t worry about the Raspberry Pi connect toggle). Now, connect the camera, and test again with `libcamera-hello`! If that works, we can run our code.





Pi Tips and Tricks!

- When using the terminal, if you need to copy from and paste to the command line, use CTRL SHIFT C and CTRL SHIFT V.
- ALWAYS make sure you are typing what you intend to type into the command line! We often make typos and wonder why our installations aren't working.
- If some installation isn't working, updating the Pi can often be the answer (check Updating the Pi).
- Follow instructions to a T! Computers, especially the Pi, are not lenient.
- Use the official Raspberry Pi documentation for things such as the camera attachment, turning on or off, and for any other questions you may have. ChatGPT and Stack Exchange are often great resources.
- Be patient… the Pi can be extremely slow, especially with running our analysis code.
- When running the photo code, it was attempted to use Cron instead of scheduling. However, that did not work due to the camera being "occupied." If Cron is more suitable, it can be attempted, but resources with this problem are limited.

Dusting the Lettuce

To dust the lettuce, two rows have diatomaceous earth, two have chinchilla dust bath, and two are the control rows as such:

** The lettuce was first dusted on the 28th of June, 2024, then re-dusted about once per week, as seen fit. This picture is from 17th of July, 2024.

In the following image, the red boxed plants receive 1/16 of a teaspoon of their corresponding dust, the purple boxed plants receive 1/8 of a teaspoon of their corresponding dust, and the blue boxed plants receive 1/4 of a teaspoon of their corresponding dust. That doesn't seem like a lot, but it goes a long way! Adjust as necessary, if the dust is too much or too little, but be consistent.