LEGOLAS Lab Notebook 2/8/24 – 5/26/24

**2/8/24 Install Legolas software on office Yoga laptop**

Install Anaconda with Python 3.11

Prompt: C:\Users\mlowe\OneDrive - Loyola University Maryland\Documents\Legolas

pip3 install rpyc

pip3 install sv\_ttk At this point, manual.py can start up.

Go to network settings on PC

Disconnect from Loyola networks

Connect to Linksys04582. press blue WPS button on Linksys.

Whirling crcle, wait. Finally see "no Internet, secured."

Network symbol looks like empty globe.

ssh pi@192.168.1.11 > raspberry Made it.

ssh pi@192.168.1.12 > raspberry

python manual.py

Reset server works

Connect via config. Had to edit config.yaml.

cell\_loc\_map: c:\Users\mlowe\OneDrive - Loyola University Maryland\Documents\Legolas\cell\_map.txt

Then Manual Mode opens.

Home works.

To set up computer:

pip install Gpy (in .ipynb file)

pip install GPyOpt

pip install paramiko

download Arduino IDE

download Logan's Github files. https://github.com/logansaar/LEGOLAS-Files/tree/main

8/8/23 Had to update numpy in order to import GPy. Use numpy-1.23.5

<https://studentsloyola-my.sharepoint.com/:w:/r/personal/arversteegen_loyola_edu/_layouts/15/Doc.aspx?sourcedoc=%7B24793729-B4A2-4409-BFB4-DF8515D8571A%7D&file=LEGOLAS%20Lab%20Notebook.docx&fromShare=true&action=default&mobileredirect=true>

2/10/24

An operating system for the independent Raspberry Pi was established. This operating system, Bullseye, allows for camera recognition and integration. Much of the documentation for camera integration with Raspberry Pi’s is more or less obsolete, as in recent years the company has switched methods and libraries.

2/20/24

Meeting between Brendan and Dr. Lowe.

2/21/24

A small cardboard tower was constructed with the purpose of being a prop to take pictures with. The tower has three levels, one level to mimic position A, one level to mimic position D, and another level at the top to provide stability and a location to house the Raspberry Pi as the pictures are taken. This was done to mimic the infrastructure of the LEGOLAS machine to assess the locations and start gathering images for the training set.

Dr. Lowe provided two longer 15-22 cables, which allow for much more freedom in positioning the camera and Pi.

3/12/24

The Arducam Camera has some warp around the edges of the pictures. This phenomenon is called radial distortion. This is to be expected of any camera, but when it comes to using these images in machine learning models, it is important for the warp to be addressed. To quantify the amount and spread of the distortion, some calibration is needed. Research was done into techniques for addressing this problem, and a method was selected. This process will include a checkerboard printed out onto a hard surface, collecting a series of images, and then finding and running a program to assess the difference between the straight lines that are expected, and the curved ones in the picture.

3/18/24

The checkerboard mentioned previously was assembled, some issue arose in the program found. This is unsurprising as it seems much of Raspberry Pi’s camera related software has depreciated. Libcamera is relatively new. Work can be done to adapt the program to current software.

3/23/24

Machine learning requires three sets of data, which can be drawn from the same database. The database will need to be split into three categories, a training set, a validation set, and a testing set at a 70/20/10 ratio. Familiarity with the camera and the commands to take pictures was reestablished. A script to take images in a series was started.

3/25/24

The previously mentioned picture taking script was finished. Pictures were taken by the Arducam camera in position D. 117 images were taken from this position with varying lighting and position of the tray well. Another script was then made to crop these pictures. Minimizing the size of an image is important for image pre-processing as it simplifies the amount of datapoints the machine learning model needs to train and learn on for an individual picture.

3/26/24

The data set of images has been established. The initial images manipulated by rotation and hue and were cropped down to only the portion necessary for our purposes. There is a total of 1,404 images that have been processed. This should allow us to properly train a model once each has been labeled. This data now needs to be labeled.

**3/27/24 (Dr. Lowe and Brendan) Recovery from incomplete installation of Debian on original Pi 7A**

Brendan said that he tried to update Pi 7A from Debian 10 to Debian 11 via a hotspot on his phone but the update was interrupted. Now 7A is missing UI elements and is not fully working.

Cannot install OpenCV on 7A, probably due to insufficient RAM.

7A and new Pi are both Raspberry Pi 4 Model B.

On 7A: buildhat 0.7.0 and rpyc 6.0.0

On 7B: buildhat 0.5.12 and rpyc 5.3.1; 803 MB RAM; python 2.7.16

Python manual.py > Connect via IP > 192.168.1.11 (7A)

* Get error “Cannot connect to pi1, try again. Expected 3, got 0.”
* Def connect\_pis in manual.py. def connect\_pi1 in core.py
* \*\*\*\*\*I normally don’t use Connect via IP\*\*\*\*\*

New Pi (Debian 11)

* client name raspberry pi 3
* ip address 192.168.1.13
* MAC D8…
* Auto\_rpyc\_server.sh works
* Configuration: SSH, VNC, serial port enabled. Same as in handout.
* Has 2 versions of Python: 2.7 and 3.9.2. On 4/3/24, I only see 3.9.2, no 2.7???

When 7A’s Buildhat hardware was inserted onto new Pi:

* To install Buildhat hardware, we removed square heat sink for Pi. Thermo-glue. Used needle-nose pliers to pull glue off.
* Python manual.py on PC
* Connect via IP
* Error: “Not enough values to unpack (expected 3 got 0)
* \*\*\*\*\*I’m not sure if Arduino was hooked up\*\*\*\*\*
* buildhat 0.7.0 and rpyc 6.0.0
* Cloned directory was done.
* Autoscript works: auto\_rpyc\_server.sh

Copy 7B SD card onto 7A SD card to try to restore 7A to Debian 10 and old software

* Insert 7A SD card into reader. Push in as far as possible.
* Reader/SD – insert into 7B USB2.0 port. Seemed to slide in OK.
* Follow instructions to copy 7B SD card onto 7A card
* After copying finished, insert 7A SD card into 7A.
* Power 7A using USB-C 3A/5V cable.
* Configure and assign a hostname to R-Pi
* System tab: hostname raspberrypi 7A
* Reboot
* Fix IP address in router
* Ifconfig
* Wlan0: 192.168.1.11 This is correct already!
* Ether: e4:5f:01:ef:9c:46 No change was done. Same as previous numbers.
* Laptop can ping 7A and ssh to it.

Put together restored 7A and Buildhat hardware. Attach Arduino to USB 2.0.

* Python manual.py
* Reset…
* Connect via Config
* Error: “There is not a Force Sensor connected to port D (Found Motor).”
* Never got “Expected 3, got 0” error.

George Hall said to use Ethernet for Linux computers. Ethernet switch was put into KH210.

**3/28 (Dr. Lowe and Brendan) Camera with new Pi, Debian 11, old Buildhat, in position of 7A.**

Cables for port D and port C are reversed on 7A.

Now 7A seems to work. Manual.py runs. Home runs.

Check if Arduino connection is causing “Expected 3, got 0 error.”

Disconnect Arduino. When power was initially turned on, got tty… error.

Insert Arduino USB into 7A USB 2.0. No more tty error.

cat /etc/os-release see OS version

getconf LONG\_BIT tells us that pi’s are 32-bit Linux

free –m indicates amount of memory in megabytes. Can also use free –h (more user-friendly).

python --version

Brendan updated packages on 7A. Takes a long time.

New Pi has 4 GB RAM. 7A and 7B seem to be 1 GB or 2GB – not enough to install OpenCV.

Meeting with Gilad, Haotong ([hliang16@umd.edu](mailto:hliang16@umd.edu)), Corey, etc (UMD and JHU). Haotong (and Sam) helped Brendan solve the errors with new Pi with Debian 11 and old BuildHat hardware.

* Nohup python /home/pi/git/rpyc/bin/rpyc\_classic.py -- host 0.0.0.0&
* There is no “3” after python. Above command is different from what is in packet on Setting Up An Automatic RPyC Server.
* Rpyc server was downgraded from 6.0.0 to 5.3.1. The lower version is present on 7A, 7B. Not sure how important this change is.
* Camera with new Pi, Debian 11, old Buildhat works. \*\*\*MAJOR DEVELOPMENT\*\*\*

Gilad will:

* email circle code for well ID. Ryan did something with color.
* Send link for wifi camera from Adafruit #4959

Part sourcing:

<https://www.amazon.com/dp/B0CLNSTV39/ref=pe_386300_440135490_TE_simp_item_image>

Arducam for Raspberry Pi Camera Module 3 Wide, 120°(D) IMX708 Autofocus Pi Camera V3 with Case, Comes with 15cm 15-22 Pin and 22-22 Pin FFC Cable

We are using 15-22 cable.

<https://www.arducam.com/product/arducam-for-raspberry-pi-zero-camera-cable-set-2-pack-11-8-30cm-ribbon-flex-extension-cables-for-pi-zerow-b0244/>

Arducam for Raspberry Pi Zero Camera Cable Set, 2 Pack 11.8″ (30cm) Ribbon Flex Extension Cables for Pi Zero&W

This is a 15-22 cable.

<https://www.amazon.com/dp/B08B6G2RFG?ref_=pe_386300_442618370_TE_sc_as_ri_0&th=1>

Raspberry Pi 4 Model B, Cana kit

Raspberry Pi 4 Extreme Kit - 128GB Edition (4GB RAM)

4/1/24 (Dr. Lowe)

Tried to ssh into Pi3 (new Pi): ssh pi@192.168.1.13

\*\*\*What is password\*\*\*Answered on 4/10/24

Also tried: ssh [greenleaf@192.168.1.13](mailto:greenleaf@192.168.1.13) and EyeOfIsengard - not successful

Also tried: monitor and keyboard plugged into new Pi. Username: greenleaf, password is set.

python manual.py

\*\*\*Cannot reset server for 192.168.1.13\*\*\*

Reset 192.168.1.12 is successful.

Connect via IP: successful for 192.168.1.13 and 192.168.1.12

Home – successful

Wooden platform, ¾" thick. Glued a small wooden block on top to stop plate.

4/10/24 (Dr. Lowe) Audrey described machine learning concept for finding circles.

New Pi (Pi3): ssh [greenleaf@192.168.1.13,](mailto:greenleaf@192.168.1.13) password: raspberry

**5/8/24 New Pi with Debian 11 in position of 7A and orig Pi with Debian 11 in position of 7B. Camera on 7B. Current configuration of PIs on Legolas after capstone project**

7A has debian 11 with 4 gb (new Pi)

7B has debian 11 with 1 gb

I installed opencv on 7A with the following command:

pip install opencv-python

I copied 7A to 7B

Installed camera onto 7B

4 toothpicks on the camera, two making a cross near the lense.

2 long toothpicks, one on top and one on the bottom.

The black axel might be hitting the white connector for the camera cable.

The 1700 pictures, they varied in orientation, lighting, background,

**5/26/24 Github repository**

Installed Git for windows 2.45.1 64-bit on Dell at home. Pick all default settings.

Copied C:\Users\mlowe\OneDrive - Loyola University Maryland\Documents\Legolas to C:\Users\mlowe\OneDrive - Loyola University Maryland\Documents\Legolas\_clone\_git

Follow instructions 2.1 Git Basics -Getting a Git Repository

git init

<https://git-scm.com/book/en/v2/Git-Basics-Getting-a-Git-Repository>

Couldn’t do:

* git add \*.c did instead git add \*.\*
* git add LICENSE
* git commit -m ‘Initial project version’

**Create Github repository**

<https://docs.github.com/en/get-started/start-your-journey/creating-an-account-on-github>

github.com

account

upper right, +, new repository…

Create a branch

Can pull and merge

Can download as zip

<https://docs.github.com/en/get-started/start-your-journey/uploading-a-project-to-github>

New repository: Legolas\_textCellLoc.git

<https://github.com/mlowe308/Legolas_textCellLoc.git>

**5/29/24 Adi started. Install Legolas software on Mac Airbook**

See User Guide for notes on installation. In CScapstone directory.

Four threads

* Understand the motors and camera of Legolas
* Gaussian processes
* Using text file of cell locations, implement GP on Legolas
* Continue CS capstone project; using camera, position the pH sensor over well.

I will work out of C:\Users\mlowe\OneDrive - Loyola University Maryland\Documents\Legolas

when using text file of cell locations