UROP Code Summary

Written up explanations of codes in this folder, including instructions on how to implement them and bugs/issues with some of them

Live Streaming Codes

1. Webstreaming_Remote_Final.py
    a. Store it on the Rasperry Pi
2. WebStreaming_iFrames_Final.html
    a. Contains iFrames code with each of the raspberry pi's IP address
    b. There are 16 iFrames, each one can contain a live stream of one of the pis
    c. Put the Rasperry Pi IP addresses in the src variable

```
<iframe src="http://192.168.0.217:8000/index.html"

</iframe>

<iframe src="http://192.168.0.218:8000/index.html"

</iframe>

<iframe src="http://192.168.0.219:8000/index.html"
```

3. WebStreaming_Local_Final.py
    a. Stored on local machine that's SSHed to Pi's
    b. Running it will automatically run the WebStreaming_Remote_Final.py as well WebStreaming_iFrames_Final.html and will open up the live streaming pi's on your default browser
    c. Change Address variable to smallest last three-digit Rasperry IP address
    d. Change value in the while loop argument to +1 of the largest last three-digit Rasperry Pi IP address
    e. Change the circled part of the command variable to what the name of the Webstreaming_Remote_Final.py file is titled on the raspberry Pi's, and make sure the final name is the same on all the pi's
    f. Put your source file location of your iFrames document inside of the webbrowser.open command

```
def main():
    hosts = []
    address = 217
    while address < 219:
        hosts.append('192.168.0.' + str(address))
        address += 1

    #hosts = ['192.168.0.218', '192.168.0.219', '192.168.0.217']

    threads = []
    command = 'python3 WebStreaming'
    #command = 'sudo restart now'
    #command = 'sudo shutdown now'
    for host in hosts:
        thread = threading.Thread(target=workon, args=(host,command))
        thread.start()
        threads.append(thread)
    for t in threads:
        t.join
    time.sleep(5)
    webbrowser.open('file:///C:/Users/jamis/OneDrive/Desktop/Live%20Streaming%
```

4. **BUGS/THINGS TO FIX**
    a. When running, the system works; however, it is impossible to end the live stream without restarting the pi's, which takes about 10 seconds. Finding a way to cut the livestream would save a lot of time.
    b. When running many streams on the iFrames can cause lag. When running the iFrames on Microsoft Edge there was never any lag, but Google Chrome and Mozzilla Firefox both had problems running many streams at a time.

Raw Image Converters

1. Method 1
    a. Yet to be implemented
    b. Source: https://github.com/6by9/dcraw
2. Method 2
    a. Implemented
    b. Source: https://github.com/6by9/dcraw
    c. Instructions to Implement
        i. In Raspberry Pi Terminal run *raspistill --raw -o image_name.jpg* to make a jpg image with the raw data attached to it
        ii. Store get_raw_images.c on Raspberry Pi
        iii. To build in Raspberry Pi Terminal run *gcc -o get_raw_images -O4 get_raw_images.c -lm -ljasper -ljpeg -llcms2*
        iv. To install necessary libraries run *sudo apt-get install libjasper-dev libjpeg8-dev gettext liblcms2-dev*
        v. Run *./dcraw image_name.jpg* to get raw image. You will end up with image_name.ppm. Store that image on local computer
    d. BUGS/ISSUES
        i. Raw Images turn out with a greenish tint. When the .ppm image is compared with .jpg in MatLab the raw image appears less sharp than the jpg despite having raw data and being a larger file. Must search if there is a flaw in the raw image converter or if the image needs to be demosaiced or if some other flaw. I believe
3. Method 3
    a. Implemented
    b. Source: https://www.raspberrypi.org/forums/viewtopic.php?t=146310&start=25
    c. Insructions to Implement
        i. In Raspberry Pi Terminal run *raspistill --raw -o image_name.jpg* to make a jpg image with the raw data attached to it
        ii. Store raw_image.c on Raspberry Pi
        iii. To build in Raspberry Pi Terminal run *gcc -o raw_image -O4 raw_image.c -lm -ljasper -ljpeg -llcms2*

iv. Run *./dcraw image_name.jpg* to get raw image. You will end up with image_name.dng. Store that image on local computer

d. Bugs/Issues

i. Same issue as with Method 2

MatLab Raw Image Tester

1. Compare_Raw_JPG_Images.m
   a. Insert name of image, both the raw and jpg formats in the circled sections below
   b. Run and compare the two images
   c. Using a speckled image like either of the ones shown below is a good way to test the sharpness of the raw image format

```
1 -    I = imread('insert image name.raw');
2 -    J = rgb2gray(I);
3 -    figure, imshow(J)
4 -    A = imread('insert image name.jpg');
5 -    B = rgb2gray(A);
6 -    figure, imshow(B)
```