

ENME 489Y – Remote Sensing: Spring 2018

Department of Mechanical Engineering

Due Date Thursday, March 1st, 2018**Submission Information**

- Submit Question #2 selfie.jpg via email to mitchels@umd.edu by 9:30 am
- Submit .pdf response to Question #3 via Gradescope by 9:30 am

*Introduction to OpenCV and object tracking with Raspberry Pi*Question #1 (nothing to submit)

Here we install VNC on the Raspberry Pi, allowing you to see your Raspberry Pi's desktop remotely, using your laptop's keyboard and touchpad (or mouse) as if you were connected directly to your RPi. This VCN connection is not required to proceed with the course; i.e. feel free to continue using a monitor and keyboard/mouse to connect directly to your RPi. However, using the VNC connection will reduce the materials you bring along when deploying your sensor on campus.

Navigate over to:

<https://learn.adafruit.com/adafruit-raspberry-pi-lesson-7-remote-control-with-vnc/installing-vnc>

and work through the Adafruit tutorial to install VNC on your RPi. You'll need to install a VNC client on your laptop, such as:

<https://www.realvnc.com/download/viewer/>

Thinking ahead: given the targets of interest you've identified on campus, what materials will you need to bring with you to map the target? Start thinking now about power, cabling, etc.

Additional resource: checkout <https://www.youtube.com/watch?v=S6tnNBlqtNI>

Question #2 (20 points)

Navigate over to Adrian Rosebrock's PyImageSearch.com and follow the step-by-step instructions to install OpenCV on your Raspberry Pi:

<http://www.pyimagesearch.com/2015/07/27/installing-opencv-3-0-for-both-python-2-7-and-python-3-on-your-raspberry-pi-2/>

To begin, verify which version of Raspbian your Raspberry Pi is running by typing

```
cat /etc/os-release
```

into a terminal window. Then follow the appropriate installation guide on PyImageSearch. For example, if your Raspberry Pi has Raspbian Jessie, you'll want to follow:

<https://www.pyimagesearch.com/2016/04/18/install-guide-raspberry-pi-3-raspbian-jessie-opencv-3/>

A few comments as you install OpenCV on your RPi:

Historically, this has taken students a significant amount of time to complete (usually an entire evening). Please plan accordingly.

Your RPi must be connected to the Internet to download the required software packages

Respond by typing *Y* (for yes) when prompted throughout the installation process

After `sudo rpi-update` reboot your RPi to ensure all firmware updates have taken place

When asked to update `~/.profile`, open `.profile` in the main RPi directory by typing `nano .profile`

Once OpenCV is installed, take a new selfie with your RPi...but now take the selfie using Python! To take your Python selfie, navigate to the following site and follow the step-by-step instructions:

<http://www.pyimagesearch.com/2015/03/30/accessing-the-raspberry-pi-camera-with-opencv-and-python/>

Specifically, complete the following sections:

Step 4: Installing picamera (*remember to work on either **cv3** or **cv**, as listed on the site)

Step 5: Accessing a single image of your Raspberry Pi using Python and OpenCV

To create a new Python file, type `nano test_image.py` into the command line, enter the code displayed on the site, then hit `ctrl + X` to exit and type *Y* (for yes) to save changes.

In order to save your selfie to your RPi, add the following line of code towards the end of `test_image.py`, after `cv2.imshow()`:

```
cv2.imwrite("selfie.jpg", image)
```

Run your code by typing `python test_video.py` in the terminal window (double-check you are working on `cv` or `cv3`). Once complete, email your RPi Python selfie to Dr. Mitchell at mitchels@umd.edu, naming your image file as ***lastname_firstname.jpg***.

“Where is my selfie located?” Click Menu > File Manager and the *selfie.jpg* should be located in /home/pi.

Congratulations! You’re now **programming in Python on your Raspberry Pi!**

Finally, complete Step 6: Accessing the video stream of your Raspberry Pi using Python and OpenCV, which will give you a live feed of the RPi camera video stream. This will be useful experience for Question #3 of this assignment.

Note 1: ensure your keyboard is in US format by clicking on Menu > Preferences > Keyboard and Mouse

Under the Keyboard tab, click on Keyboard Layout... > Country: United States > Variant: English

Note 2: every time you restart the RPi or open a new command window, be sure to reload your virtual environment by typing:

```
source ~/.profile
```

```
workon cv3
```

Note 3: Have fun with this exercise! **Creative selfies always encouraged.**

Question #3 (20 points)

Modern Advanced Driver Assistance Systems (ADAS) typically consist of camera, radar, and lidar subsystems. A primary advantage of the camera subsystem is the ability to **sense colors** as the automobile drives down the road, while the radar and lidar subsystems provide a 3-dimensional representation of the automobile's environment. This enables the ADAS to detect features such as stop lights, road signs, and lane markings. For example, checkout Civil Maps' "Cognition for Cars" demo:

<https://www.civilmaps.com/>

<https://civilmaps.com/cognition-for-autonomous-cars-using-6d-localization/>

Here we perform object tracking with OpenCV to simulate an ADAS camera. Head to <https://github.com/oneshell/enme489y> and download the Python scripts *stoplighttracking.py* and *video_play_file.py*. These scripts have been developed to (1) detect the presence of a colored circle, the color of which can be specified by the user, (2) monitor the position of the circle in real-time as it translates across the RPi camera field of view, and (3) play a video.mp4 file in Python.

We will use a similar Python script for the semester project (provided to you in a few weeks), in which OpenCV tracks the (x,y) coordinates of the laser imaged onto the Raspberry Pi camera frame. Based on the geometry of the lidar sensor, this (x,y) coordinate is translated into a real-time range measurement from the sensor to the target of interest.

Using the stop light schematic provided at right (on your laptop, tablet, printed paper, etc), take a minimum 30 second video clip of yourself tracking the green light using the provided scripts and your Raspberry Pi. Upload the video to your YouTube account (which can be accomplished directly from your RPi) then upload a single page .pdf to Gradescope providing the link to the video. An example video is available here:

<https://youtu.be/HLXRjq7DXw>

Note 1: Have fun with the recording! And consider using the clip for your semester project video!

Just for fun: if you have a tennis ball, move/toss the ball across the camera field of view to watch it track automatically. To get an idea of the processing power (or lack thereof) of the RPi as compared to your laptop, try running the following code in PyCharm and track the tennis ball with your laptop webcam:

<https://github.com/oneshell/enme489y/blob/master/objecttrackingwebcam.py>

