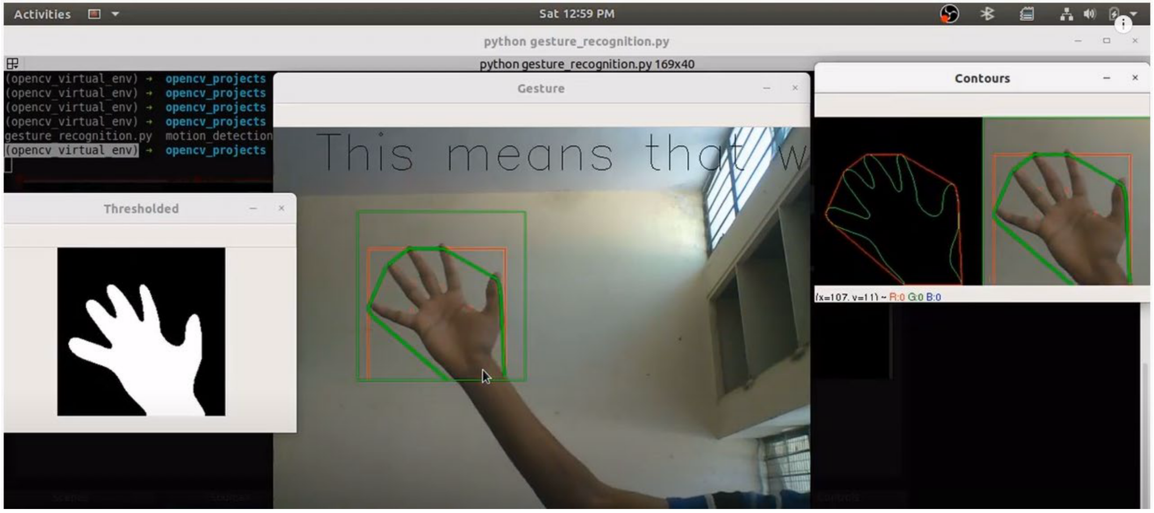
Lab 1: Computer Vision and Open CV

Learning Objectives:
This lab is intended to acquaint you using the Raspberry Pi with OpenCV, and apply that knowledge to incorporate computer vision along with more basic sensors to allow your robot to navigate more efficiently and effectively. By the end of this lab you should be able to use the RPi for basic editing of images and image tracking.

Minimum Parts Required:
Your Canakit Raspberry Pi 5 Kit, including a Pi Camera v2, various LEDs, jumper wires, and resistors.

Lab Overview

The goal of this lab is to get your RPi camera to recognize simple hand gestures and respond to changing hand signals.


Activities
Sat 12:59 PM
python gesture_recognition.py
python gesture_recognition.py 169x40
Gesture
This means that w
Thresholded
Contours
(x=107, y=11) ~ R:0 G:0 B:0

```
(opencv_virtual_env) → opencv_projects
(opencv_virtual_env) → opencv_projects
(opencv_virtual_env) → opencv_projects
(opencv_virtual_env) → opencv_projects
gesture_recognition.py  motion_detection
(opencv_virtual_env) → opencv_projects
```

You can set it up how you would like, but it should recognize at least three different hand-states and respond with a different physical output for each.

Part 1 – RPi Setup

Start off by getting your Pi setup with all the peripherals in your kit (or your own peripherals, if you prefer). After that, take half an hour at least to play around with the RPi. Make sure you can do all of the following tasks.

- Connect to the internet
- Use the command line and get a hang of the BASH commands like:
    - man, ls, pwd, cd, find, sudo, rm, mv, etc
- Connect to the Pi's GUI with your computer over wifi
    - Putty for windows for SSH. Over ethernet try raspberrypi.local instead of IP
    - TightVNC Viewer for windows Must use ::5902 after IP address
    - Tight VNC Server for RPi
    - VNC for Mac Instructions
- Transfer files between your computer and the Pi

  - Filezilla, Google Drive, or GitHub
- Write a text file or a Python file on the Pi
  - Python Example Code
- Run said files
  - python3 myCode.py

If you get the error `Is the package apt-transport-https installed?` try this link

If you have a laptop, I ***strongly encourage*** you to set it up with a remote connection (VNC, RDP, Tailscale, Raspberry Pi Connect) so that you can run your RPi without all the peripherals ("headless" mode).

Part 2 – GPIO with the RPi

If you haven't used a RPi with GPIO, use the tutorials and resources below to get your RPi reading input pins and sending output values to light LEDs, etc.

Here are some helpful links:

- RPi 5 and GPIO
- RPi.GPIO module Tutorial This is a very good general tutorial, and great for jumping the gap from Arduino to RPi.
- RPi.GPIO PWM and Outputs This is important to test out. RPi is generally not very good with PWM.

Part 3 - OpenCV

OpenCV and the Pi Camera require you to download and compile a good amount of software before you begin, but once it is all working the program is flexible and can be used to solve many problems. Before you start with OpenCV, make sure your OS is up to date. Here is a good general guide, and seems to be up-to-date. The Raspberry Pi OS (Raspbian) is continually updated, to improve it (theoretically). The latest version (v12) is Bookworm, but you may see references to version 11 (Bullseye) and older versions as well. Because we are using RPi 5's, we need to use Bookworm or Trixie. If you run into issues with OpenCV or other libraries, you can drop back to a RPi 4 if you still have one. If you don't, Prof. Reamon can probably find one for you.

Install OpenCV on Raspberry Pi 5

If you are unable to successfully compile OpenCV (sometimes it just won't work), please speak with other teams, the TAs, or Derek about cloning a functional SD card.

Installing a Raspberry Pi camera

Part 3 – Reading Hand signals

Create a program that allows your Pi Camera to recognize at least three different hand signals. You can tailor this how you would like, but an example would be for the RPi to light a red LED when you hold up your index finger (#1 signal), a green LED when you make the 'okay' sign with three fingers extended (the three-point signal in basketball), and turn them off when you make a fist. Alternatively, you could hook up a motor via a transistor and have the motor speed correspond to the number of fingers – off for a fist, slow for one finger and faster for each finger up to full speed for all five fingers up. Set up your camera in an advantageous environment to make this as simple as possible, i.e., place a black or white background behind your hand to make it as simple as possible for the camera to see your hand.

General OpenCV Tutorials: OpenCV org, PyImage, Tutorials
I recommend these for this task: Image Thresholding, Contours, Contour Detection
Here is a video tutorial on exactly what we are trying to do: Hand Gesture Recognition
And here is another using MediaPipe, which incorporates machine learning to recognize your hand with an advanced model. Hand Tracking

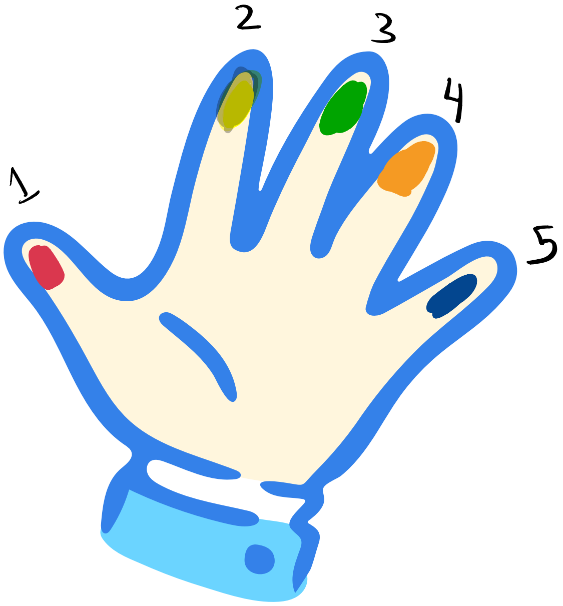
Get a signoff or submit a video of your RPi reading hand signals and responding to them.

3)

Signoff

I have witnessed Doonesbury's hand signal reader.

Witness J. M. Date 1/16/2026



GPIO			Pin #
🔴 □	Thumb	R	#13
🟡 □	Index	Y	#11
🟢 □	Middle	G	#7
🟠 □	Ring	O	#5
🔵 □	Pinky	B	#3
□			