

Raspberry Pi Tutorial

Phumiphat Charoentananuwat 6372093721

Nisawan Ngambenjavichaikul 6372066821

Software and Hardware Requirement

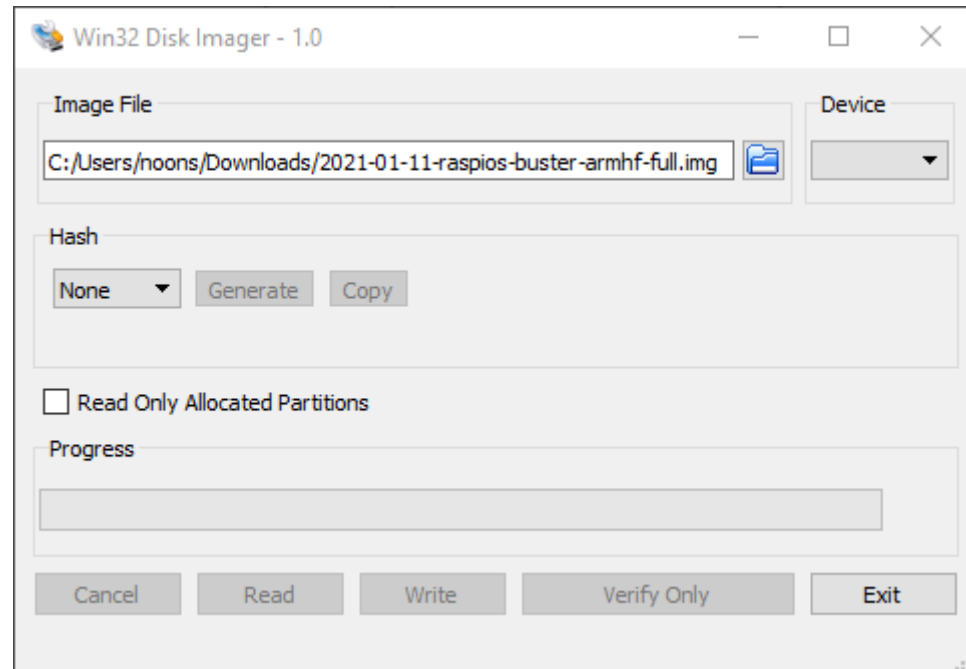
- Win32DiskImager
- Visual Studio Code with Remote-SSH Extension (Optional)
- VNC Viewer (Optional)
- SD card
- Keyboard
- Mouse
- HDMI monitor

Workflow

- Install Raspbian OS into SD card
- Setup system
 - Enable camera
 - Enable ssh (Optional)
 - Enable VNC (Optional)
- Install OpenCV
- Test camera and OpenCV
- Implement laplacian and canny edge detection

Install Raspbian OS

- Download Raspbian image from <https://www.raspberrypi.org/software/operating-systems/#raspberry-pi-os-32-bit>
- Install by Win32DiskImager



Hardware setup



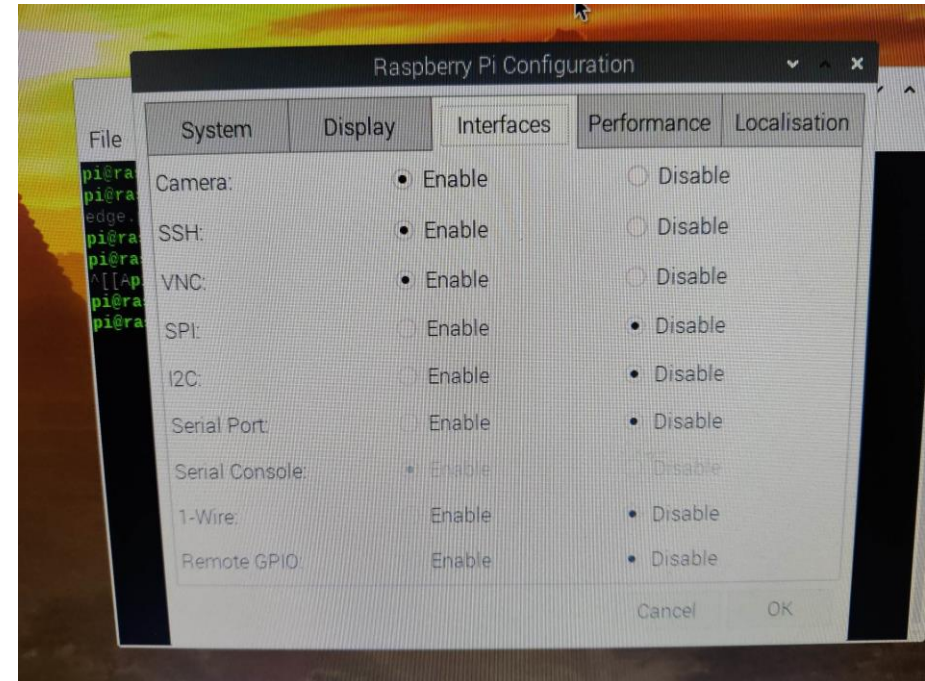
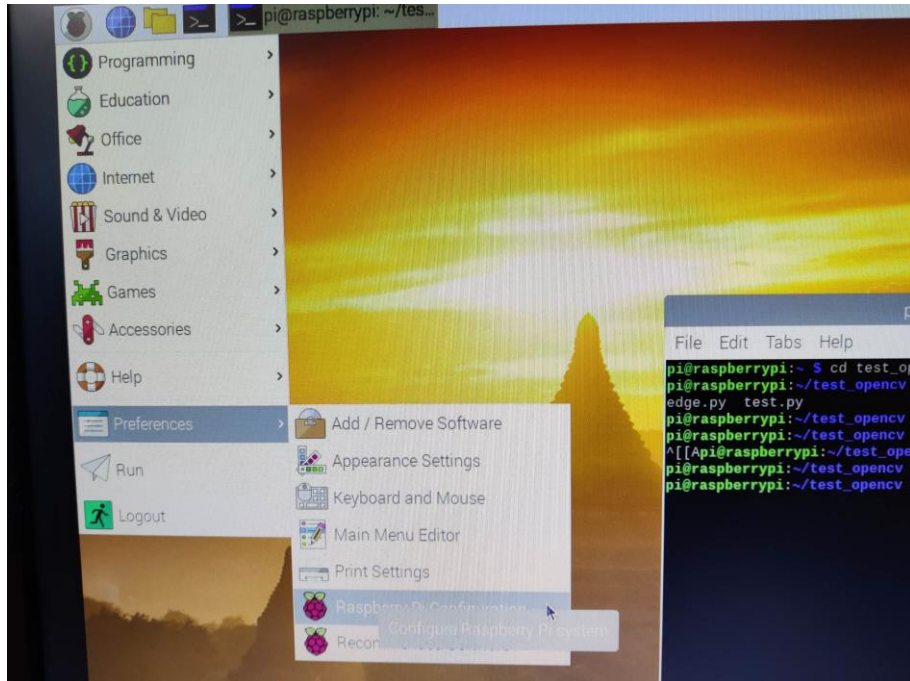
Power supply

HDMI monitor

usb keyboard

usb mouse

System setup

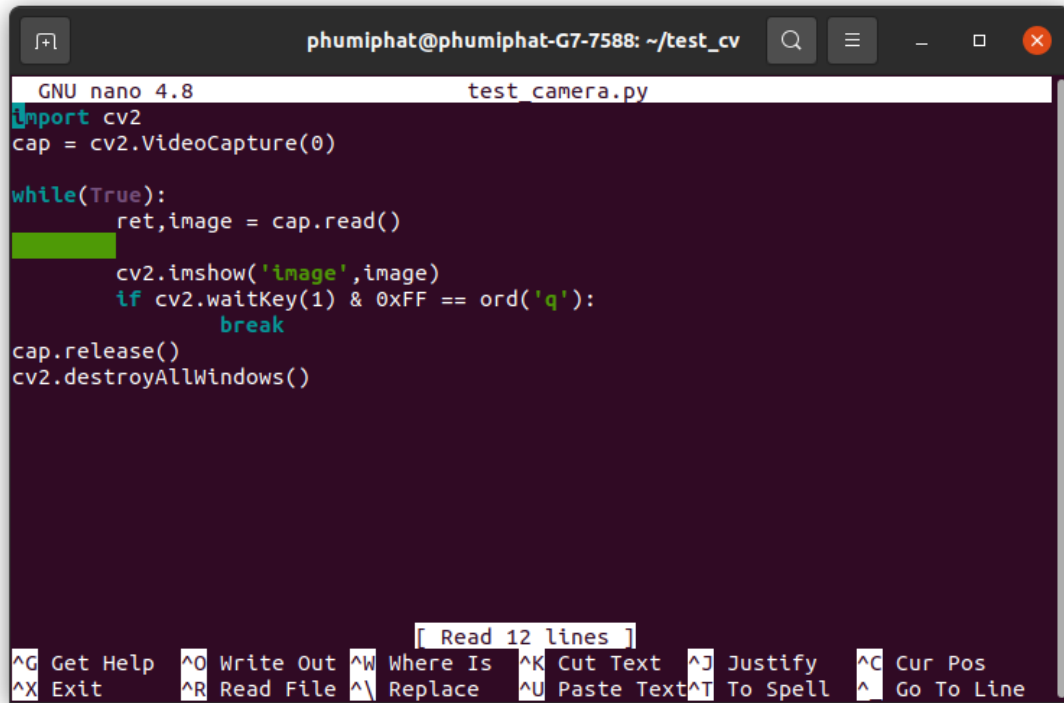


Install OpenCV

- System update
 - `sudo apt-get update`
 - `Sudo apt-get upgrade`
 - `Sudo rpi-update`
- Install dependencies image and video io
 - `sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev -y`
 - `sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev -y`
 - `sudo apt-get install libxvidcore-dev libx264-dev`
- Install HiGUI
 - `Sudo apt-get install libgtk2.0-dev libgtk-3-dev -y`
 - `Sudo apt-get install libatlas-base-dev gfortran -y`
- Install OpenCV
 - `Sudo apt-get install python3-opencv`

Experiment (1)

- Test picamera and OpenCV

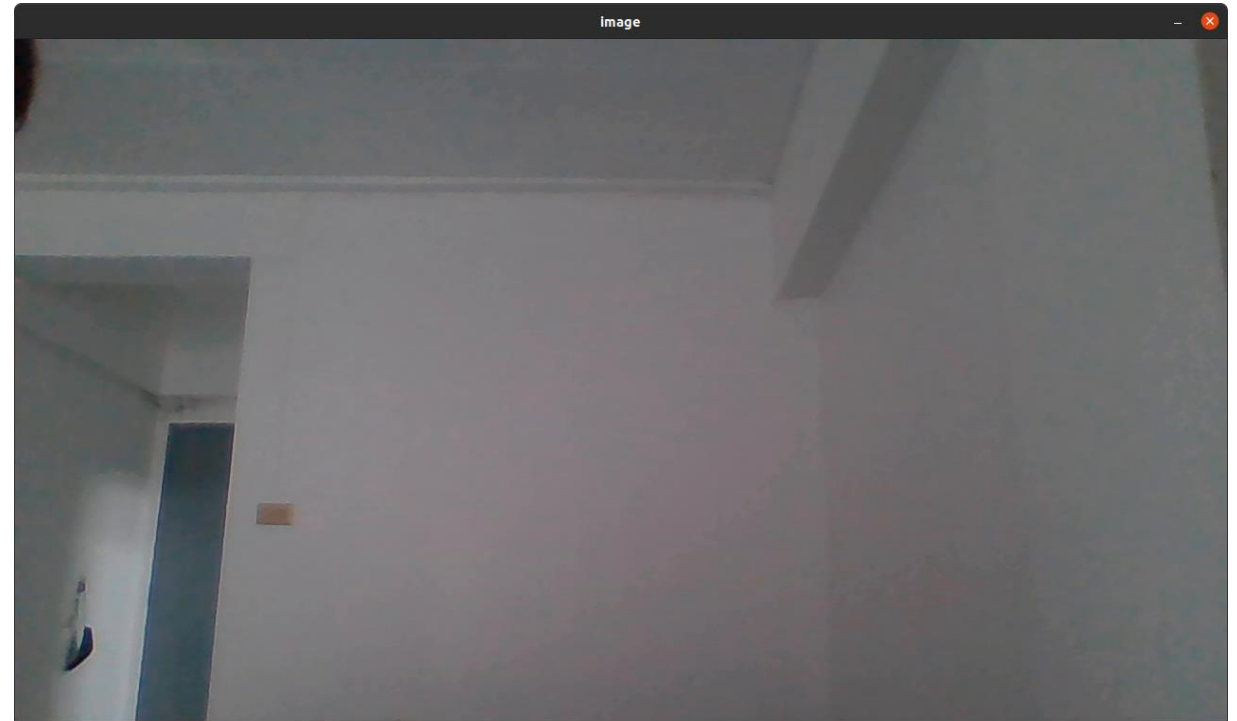


The screenshot shows a terminal window with the title bar "phumiphat@phumiphat-G7-7588: ~/test_cv". Inside, the GNU nano 4.8 editor is open, editing the file "test_camera.py". The code in the file is as follows:

```
import cv2
cap = cv2.VideoCapture(0)

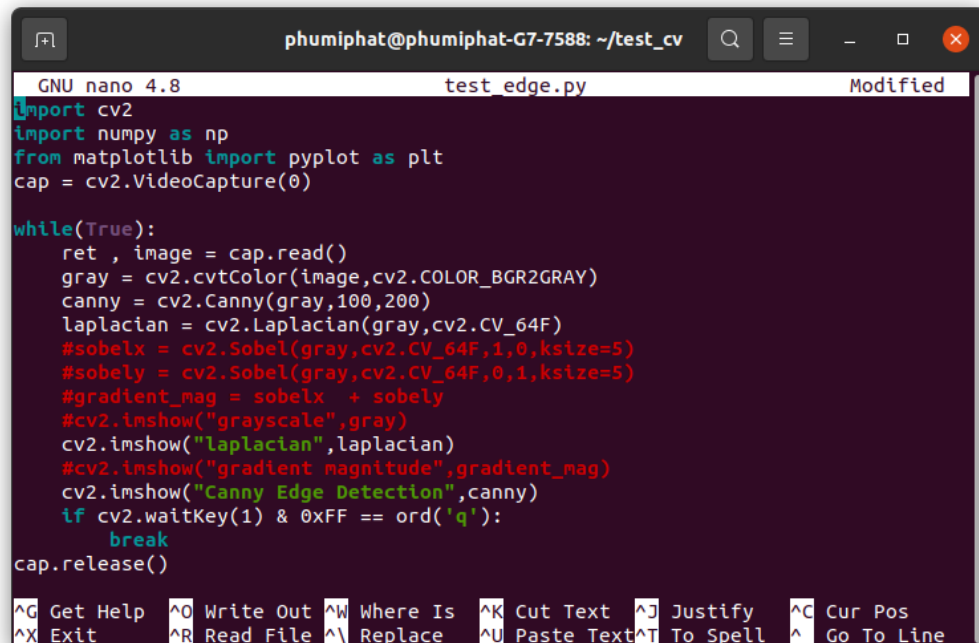
while(True):
    ret,image = cap.read()
    cv2.imshow('image',image)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

At the bottom of the terminal, there is a status bar with various keyboard shortcuts: ^G Get Help, ^O Write Out, ^W Where Is, ^K Cut Text, ^J Justify, ^C Cur Pos, ^X Exit, ^R Read File, ^_ Replace, ^U Paste Text, ^T To Spell, and ^_ Go To Line. A small box above the status bar indicates "Read 12 lines".



Experiment (2)

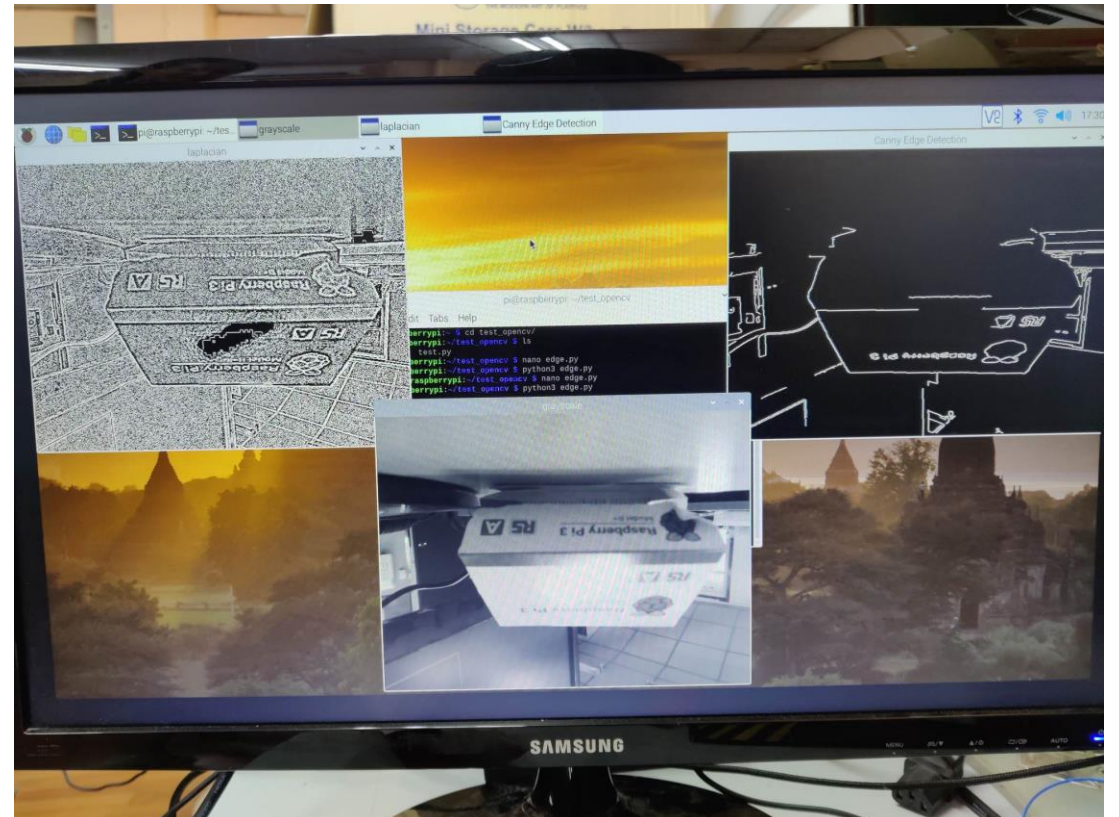
- Implement laplacian and canny edge on raspberry pi



```
phumiphat@phumiphat-G7-7588: ~/test_cv
GNU nano 4.8      test_edge.py      Modified
import cv2
import numpy as np
from matplotlib import pyplot as plt
cap = cv2.VideoCapture(0)

while(True):
    ret, image = cap.read()
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    canny = cv2.Canny(gray, 100, 200)
    laplacian = cv2.Laplacian(gray, cv2.CV_64F)
    #sobelx = cv2.Sobel(gray, cv2.CV_64F, 1, 0, ksize=5)
    #sobely = cv2.Sobel(gray, cv2.CV_64F, 0, 1, ksize=5)
    #gradient_mag = sobelx + sobely
    #cv2.imshow("grayscale", gray)
    cv2.imshow("laplacian", laplacian)
    #cv2.imshow("gradient magnitude", gradient_mag)
    cv2.imshow("Canny Edge Detection", canny)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File ^_ Replace  ^U Paste Text ^T To Spell ^_ Go To Line
```



Demo video : <https://www.youtube.com/watch?v=3j-d8IbfJ8I>