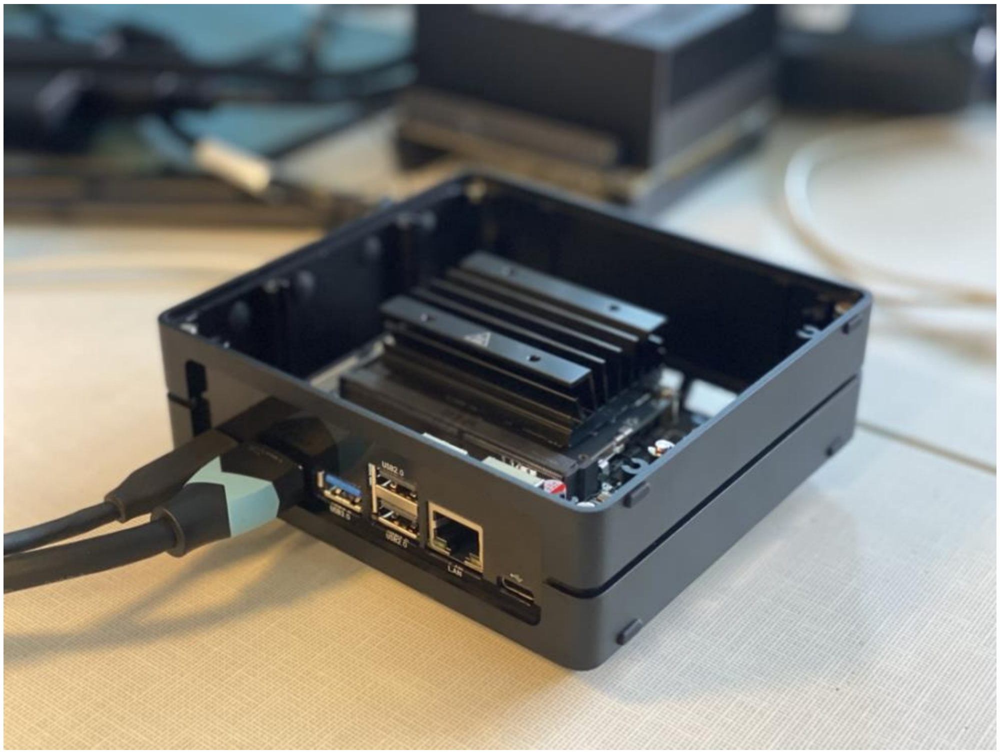
Week 5
CSI Camera Setup and Data Collection
Tuesday, February 7th 5:30 PM – 6:45 PM

DUE: Tuesday, February 14th



DESCRIPTION

Red, Green, Blue (RGB) camera sensors are popular for acquiring imagery data for various applications across disciplines. Imagery data are important for developing image processing, machine learning, and deep learning models for computer vision applications. Furthermore, as computational resources are becoming readily available via edge devices for deploying solutions, it is important to interface camera sensors on the edge for acquiring and analyzing imagery data. The purpose of this assignment / lab is for students to:

1. Use the CSI camera sensors
2. Connect the CSI camera
3. Interface with the edge device
4. Obtain live feed
5. Acquire images
6. Acquire videos
7. Obtain a small dataset

Instructors:
Dr. Vetria Byrd (vlbyrd@purdue.edu)
Dr. Dharmendra Saraswat (saraswat@purdue.edu)
Aanis Ahmad (ahmad31@purdue.edu)

PREREQUISITES

1. Complete Lab 1
2. Install the SSD expansion
3. Obtain the CSI camera sensors.

LEARNING OBJECTIVES

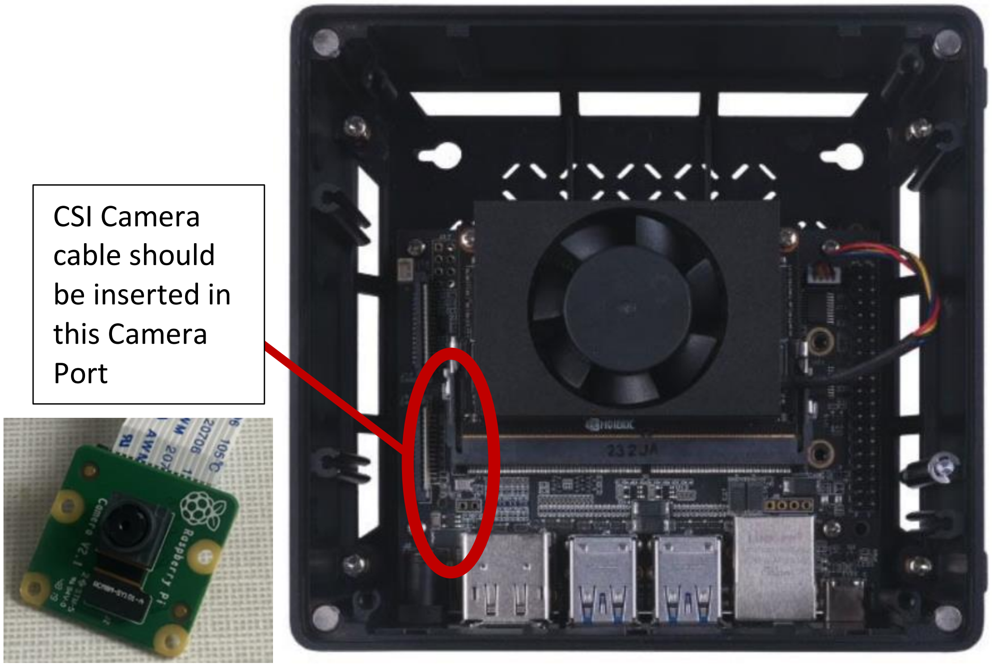
By the end of this course, students will:

1. Connect the CSI Camera correctly
2. Install the relevant libraries required to interface with the camera
3. Obtain live feed from the CSI Camera
4. Acquire images using the camera sensor
5. Acquire a short video using the camera sensor

ASSIGMENT

Part 1: Setup

1. Connect the CSI camera to the jetson module
    a. This is an ArduCam v2 CSI camera which the edge device is capable of interfacing with out of the box
2. Each module consists of two camera ports
3. Put the camera into the port closer to the power outlet (as shown below)



CSI Camera cable should be inserted in this Camera Port

4. Lift the tab carefully (these can break easily)
5. Insert the camera cable with the **pins facing** the Jetson module
6. Lock the port into place after inserting the camera
7. **Restart** the edge device
8. Open the terminal by right clicking anywhere on the Desktop
9. To check that the camera is working, enter the following commands in the terminal:
    a. Type nvgstcapture-1.0 and then press "enter"
        i. You may type j and press "enter" to capture one image
        ii. For capturing images using a timer, type jx5000 and press "enter" for taking images every 5 seconds
        iii. For capturing a specified number of images, type jx5 and press "enter" for taking 5 images
        iv. Type q and press "enter" to exit
    b. The images will be saved to the "Home" directory
        i. Reference: https://developer.nvidia.com/embedded/learn/tutorials/first-picture-csi-usb-camera
10. Use Python for interfacing with the camera:
    a. Open the terminal and enter the following commands:
        i. Type cd and then press "enter" to go back to the main directory
        ii. Type mkdir camera and then press "enter" to create a new directory called "camera"
        iii. Type cd camera and then press "enter" to enter the camera directory
        iv. Type git clone https://github.com/JetsonHacksNano/CSI-Camera and then press "enter"
        v. Type cd CSI-Camera and then press "enter"
        vi. Type python simple_camera.py and then press "enter" to run the sample Python code for obtaining live camera feed
    b. Reference: https://jetsonhacks.com/2019/04/02/jetson-nano-raspberry-pi-camera/
11. Use following command for installing a library for examining camera capabilities:
    a. Type sudo apt-get install v4l-utils and then press "enter"
    b. Type cd and then press "enter"

c. Type v4l2-ctl --list-formats-ext -d /dev/video0 and then press “enter” to few camera pixel formats, image sizes, and frame rates

d. Reference: https://jetsonhacks.com/2022/02/02/in-practice-usb-cameras-on-jetson/

Part 2: Homework Tasks

1. Obtain 10 images and save them to your folder using the approach from Step 9:

    a. Type nvgstcapture-1.0 and then press “enter”

    b. For capturing a single image, press j followed by “enter on the keyboard

    c. For exiting the camera, press q followed by “enter

2. Modify a Python code to acquire 10 images from the camera with a 5 second interval:

    a. Navigate to the folder that was created in step 10 (b):

        i. Type cd and then press “enter” to go back to the main directory

        ii. Type cd camera and then press “enter” to enter the camera directory

        iii. Type cd CSI-Camera and then press “enter”

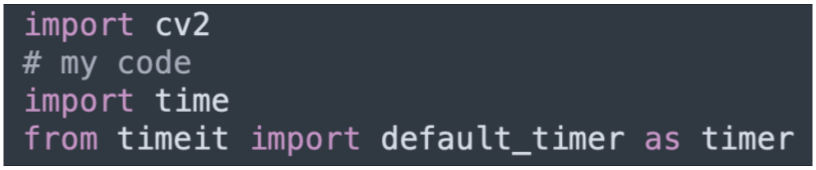
        iv. Within this directory, you are required to modify the “simple_camera.py” Python file.

        v. Using the terminal, type vi simple_camera.py and press “enter”

        vi. To modify the code, press i on your keyboard

        vii. Using the arrow keys, you may navigate through the code

        viii. Below import cv2, add the following two lines to import the appropriate libraries



```
import cv2
# my code
import time
from timeit import default_timer as timer
```

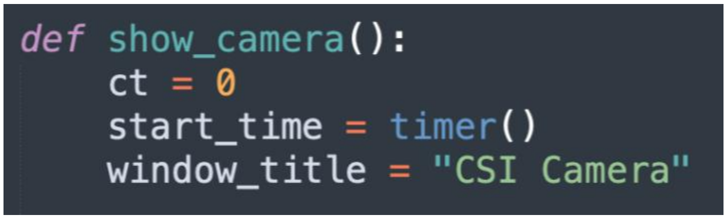
        ix. Now scroll down to the function definition for “show_camera” and add two lines as shown in the image below:

            1. First we create a counter variable and set it as zero by typing:

                a. ct = 0

            2. Then, you need to record the starting time by typing:

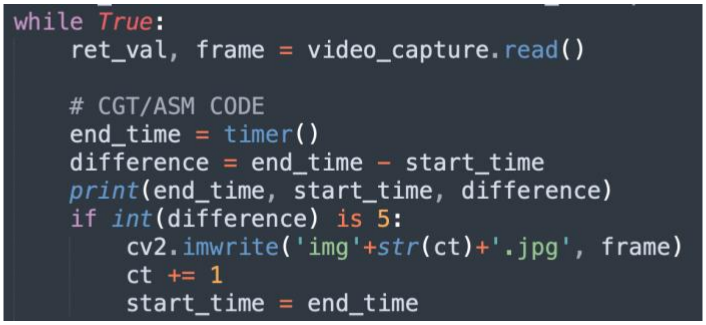
                a. start_time = timer()



```
def show_camera():
    ct = 0
    start_time = timer()
    window_title = "CSI Camera"
```

x. Within this “show_camera” function, scroll down to the “While loop” (While loop will run as long as you don’t close the program) and add the code under the comment “CGT/ASM Code” as shown below:

1. First the end timer is defined by typing:
    a. end_time = timer()
2. The difference between the times is calculated and stored in a new variable called “difference” by typing:
    a. difference = end_time - start_time
3. Now an “if” conditions is created to check if whether or not 5 seconds have passed by typing:
    a. if int(difference) is 5:
4. If 5 seconds have passed, the code will them same a frame as an image while you are viewing the live feed by typing:
    a. cv2.imwrite('img'+str(ct)+'.jpg', frame)
5. The counter variable “ct” must be updated to change the image name. If “ct” is not updated, the new images will override existing images. Update the counter variable by typing:
    a. ct += 1
6. Now, the “start_time” will be set to the “end_time” so that we calculate the 5 seconds for acquiring another image. Redefine the “start_time” variable by typing:
    a. start_time = end_time



```
while True:
    ret_val, frame = video_capture.read()

    # CGT/ASM CODE
    end_time = timer()
    difference = end_time - start_time
    print(end_time, start_time, difference)
    if int(difference) is 5:
        cv2.imwrite('img'+str(ct)+'.jpg', frame)
        ct += 1
        start_time = end_time
```

xi. Your code should look like the image above

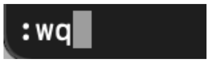
xii. Don't modify any more code

xiii. On the keyboard, first press "escape" as this will take you out of writing / editing mode.

xiv. Now enter the "colon" button which looks like ":"

xv. Then type wq and press "enter" (wq means save and quit)

xvi. It should look like the image below



```
:wq
```

3. Copy and paste the 10 images acquired from Homework step 1 and 10 images acquired from Homework step 2 in two separate word documents. Then submit each document with the images and the Code file that you have modified on Brightspace.

Note: There are many examples available for Raspberry Pi and Jetson Nano CSI camera examples that will work with the current module.

REFERENCES / ADDITIONAL RESOURCES

1. Very useful GitHub repository with multiple examples: https://github.com/JetsonHacksNano/CSI-Camera
2. Easy image capture: https://developer.nvidia.com/embedded/learn/tutorials/first-picture-csi-usb-camera
3. Tutorials and explanations:
    a. https://www.youtube.com/watch?v=dHvb225Pw1s
    b. https://www.youtube.com/watch?v=1lJUb1vmnx8
4. The following approach can also help acquire images easily:
    a. https://github.com/thehapyone/NanoCamera
5. Python to obtain images:
    a. https://elbruno.com/2020/12/11/opencv-open-a-video-file-🎥-and-save-each-frame-as-a-png-🖼-file-to-a-folder-📂python/
    b. https://theailearner.com/2018/10/15/extracting-and-saving-video-frames-using-opencv-python/
6. How to use the timer function:
    a. https://stackoverflow.com/questions/7370801/how-do-i-measure-elapsed-time-in-python