

CS 1010



Getting started with Raspberry Pi and PiCamera

Prof. Kartik Bulusu, CS Dept.

```
import cv2
```

```
Import picamera
```

Teaching Assistants:

Marshall Thompson, CS Dept.

Jonathan Garcia, MAE Dept.

Matthew Dionne, CS and EMSE Dept.

Learning Assistants:

Josie Libbon, CS Dept.

Josh Rizika, CS Dept.

Miles Grant, CS Dept.

Addy Irankunda, Physics Dept.

Talia Novack, CS Dept.

Fred Kamgang, CS Dept.



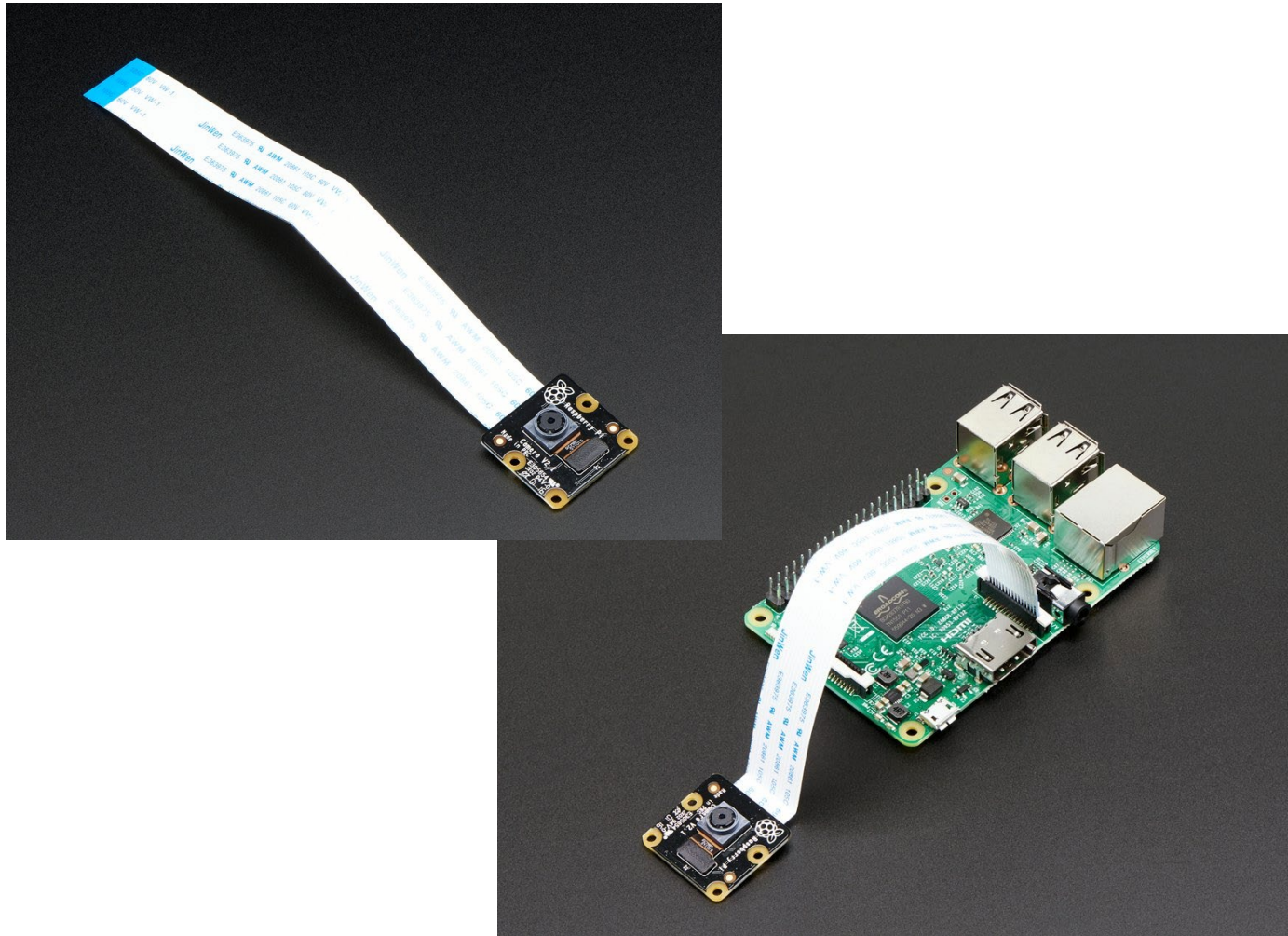
School of Engineering
& Applied Science

THE GEORGE WASHINGTON UNIVERSITY

Fall 2022

Photo: Kartik Bulusu

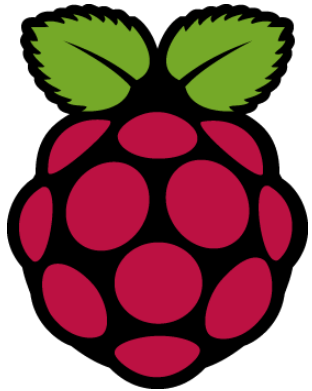
Introducing the Pi NoIR Camera



- 8 megapixel native resolution high quality Sony IMX219 image sensor
- 3280 x 2464 pixel static images
- Capture video at
 - 1920 x 1080 p30
 - 1280 x 720 p60
 - 640 x 480 p90 resolutions
- No Infrared (NoIR) filter
 - Infrared photographs or photographing objects in low light (twilight) conditions

Source:

<https://www.adafruit.com/product/3100#description>



picamera 1.13



Raspberry Pi hardware and Camera hardware interactions
Low-level image acquisition and processing

Computer vision on the Raspberry Pi



picamera 1.13



```
# import the necessary packages
```

```
from picamera.array import PiRGBArray
from picamera import PiCamera
import time
import cv2
```

picamera 1.13

```
# initialize the camera and grab a reference
# to the raw camera capture
```

```
camera = PiCamera()
camera.resolution = (320, 240)
rawCapture = PiRGBArray(camera)
```

```
# allow the camera to warmup
time.sleep(0.1)
```

picamera 1.13

```
# grab an image from the camera
camera.capture(rawCapture, format="bgr")
image = rawCapture.array
```



```
# display the image on screen and wait for a keypress
cv2.imshow("Image", image)
cv2.imwrite("savedImage.png", image)
cv2.waitKey(0)
```


Goal of the lab segment

Co-work

- Observe, ask and try in groups

Make

- Build-a-hack
- Use Pi NoIR Camera to acquire an images
- import OpenCV library

Perform basic image processing functions using OpenCV



```
3 import cv2
4 import numpy as np
5
6 #IMAGE COLOR FILTERING
7
8 img = cv2.imread('/home/pi/Desktop/Fall2022/Week4-OpenCV_Camera/savedI
9
10 retval, threshold = cv2.threshold(img, 12, 255, cv2.THRESH_BINARY)
11
12 grayscale = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
13
14 retval2, threshold2 = cv2.threshold(grayscale, 12, 255, cv2.THRESH_BINARY)
15
16 gaus = cv2.adaptiveThreshold(grayscale, 255, cv2.THRESH_BINARY, 3, 1, 0, 0, 0, 0, 0, 0)
17
18 retval2, otsu = cv2.threshold(grayscale, 125, 255, cv2.THRESH_BINARY, cv2.THRESH_OTSU)
19
20 cv2.imshow('original', img)
21 cv2.imshow('threshold', threshold)
22 cv2.imshow('threshold2', threshold2)
23 cv2.imshow('gaus', gaus)
24 cv2.imshow('otsu', otsu)
25 cv2.waitKey(0)
26 cv2.destroyAllWindows()
```

```
Shell x
===== RESTART =====
>>> %Run picamera_opencv_test1.py
===== RESTART =====
>>> %Run picamera_opencv_test7.py
```

