AI – Image Processing with Raspberry-Pi

Dr. Soharab Hossain Shaikh
Associate Professor, CSE, SoET, BMU

1

Artificial Intelligence

- **John McCarthy**, widely recognized as one of the godfathers of AI, defined it as **"the science and engineering of making intelligent machines."** (1956)

A branch of computer science dealing with the simulation of intelligent behavior in computers.

The capability of a machine to imitate intelligent human behavior.

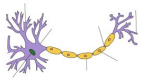
A computer system able to perform tasks that normally require human intelligence, such as visual perception, speech recognition, decision-making and translation between languages.

Source: skymind A.I. Wiki

2

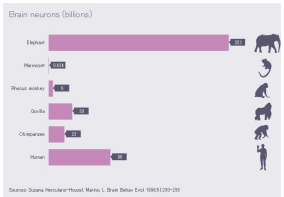
2

AI – Simulating the Power of Human Brain!!!



Connectionism: The information is in the connection – Alexander Bain, 1873.

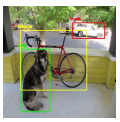
- 86 Billion neurons in human brain.
- More than 130 Trillion synaptic connections.
- The number of devices connected to the internet reached 22 billion worldwide at the end of 2018.
- Huge network of neurons interconnected with connections.
- Capable of performing massively parallel computations.



3

What computers are good/bad at and why?

- Computers are good at doing repetitive tasks for which a program can be written.
- Program < - Algorithms < - Arithmetic and/or Logical and steps
- Ex:- Playing Chess – pre-defined rules easy to code in a program.
- Ex:- Computing n-th Fibonacci number
- Ex. Computing factorial of a positive integer



- Mundane tasks like Visual Recognition – Object Detection from Images/Videos
- Navigating through a street – obstacle detection
- Machine Translation – Understanding Natural Languages - Text to Speech Conversion
- Hard to define features
- Difficult to hard-code the set of rules due to large variability in data.

4

4

Two most common tasks

- Natural Language Understanding
- Human Vision Capabilities

5

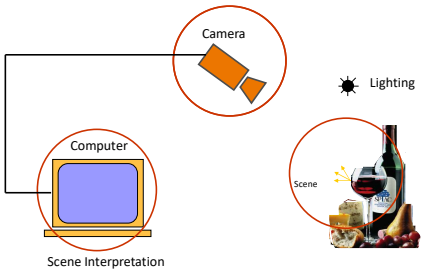
Computer Vision

- **Computer vision** is an interdisciplinary field that deals with how computers can be made to gain high-level understanding from **digital images** or **videos**.
- From the perspective of engineering, it **seeks to automate tasks that the human visual system can do.**
- Computer vision is a field that includes methods for **acquiring, processing, analyzing, and understanding images in order** to **simulate/duplicate** the abilities of human vision by electronically perceiving and understanding an image.

Image Understanding **Machine/ Robot Vision** **Image Analysis**
Video Understanding

6

Components of a Computer Vision system



Challenges

- **Change in camera view-point**
- **Illumination change in the scene**
- **Motion of objects**
- **Background clutter**
- **Occlusion**
- **Variability**

7

Image Processing (pre-processing)

Images captured by the camera need to be processed with Image Processing algorithms before being fed to sophisticated Computer Vision Systems.

8

Smart Applications

- Embedded Vision Systems
- Small/Mini (size) Devices equipped with AI capabilities
- Single board computing modules (e.g. Raspberry-Pi)

9

So…..

Today, we will see an application of image processing (video- sequence of image frames) with a credit-card-size-single-board-computer – **Raspberry-Pi**.

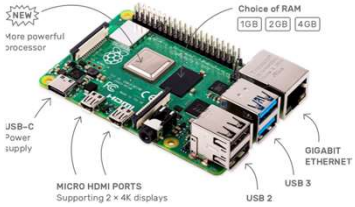
We will make it a little more interesting by making a web-app and access it with a web-browser from any machine within the same network.

10

Raspberry Pi

11

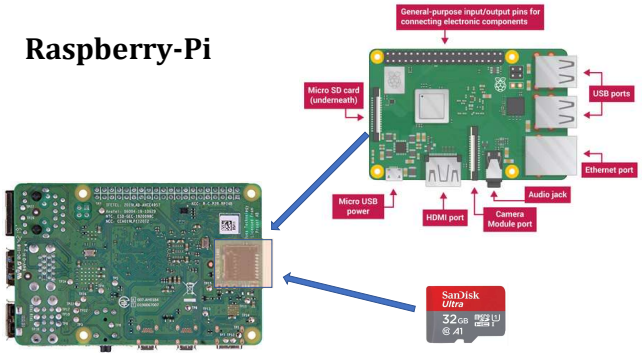
Raspberry-Pi



- Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1. 5GHz---4GB LPDDR4-2400 SDRAM
- 2. 4 GHz and 5. 0 GHz IEEE 802. 11B/g/n/ac Wireless LAN, Bluetooth 5. 0, double-true Gigabit Ethernet
- 2 × USB 3. 0 ports, 2 x USB 2. 0 Ports---2 × micro HDMI ports supporting up to 4Kp60 video resolution
- 2-lane MIPI DSI/CSI ports for camera and display---4-pole stereo audio and composite video port--Micro SD card slot for loading operating system and data storage
- Requires 5. 1V, 3a power via USB Type C or gpio-poep (power over Ethernet) enabled (requires PoE hat-not included)

12

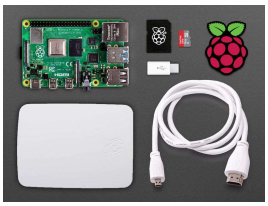
Raspberry-Pi

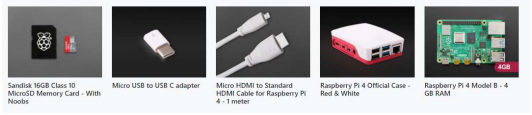


13

Accessories

Raspberry-Pi – 4 Model B - 4GB RAM Model with all the required accessories





14

Accessories

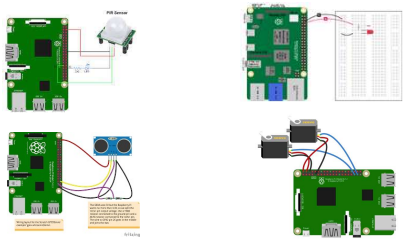
Raspberry-Pi – 3 Model B - 1GB RAM Model with all the required accessories

- Raspberry pi3 Model B
- Pi Case
- Power adapter 5.1 V with 2.5A current
- 8 GB SD card with NOOBS pre-loaded
- HDMI to HDMI adapter (To connect to HDMI monitors or HDMI TV)
- Ethernet cable of length 3ft

https://www.raspberrypi.org/documentation/installation/noobs.md#:~:text=NOOBS%20and%20NOOBS%20Lite,system%20requires%20an%20internet%20connection.

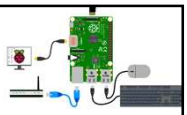
15

Connecting Sensors and Program them with GPIO



https://pypi.org/project/RPi.GPIO/
https://www.raspberrypi.org/documentation/usage/gpio/python/README.md

16



Installation and System Setup

https://github.com/soharabhossain/Raspberry-Pi/tree/main/Raspberry-Pi%20Installation%20Rasbian-OS-Imager

17

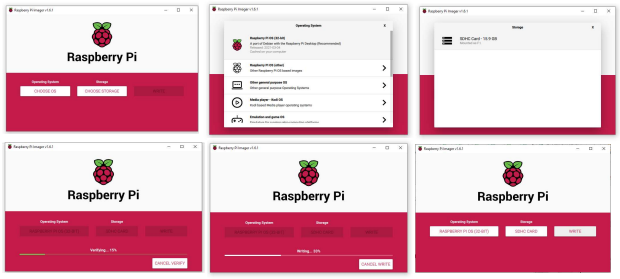
Installation & Set up

- **Download the Raspbian-OS-Imager***
- **Burn the image in the SD Card (next slide)**
- **Put the SD Card in R-Pi Board and power it up**

***Download the exe file from here:**
https://github.com/soharabhossain/Raspberry-Pi/tree/main/Raspberry-Pi%20Installation%20Rasbian-OS-Imager

18

Installation of Raspberry-Pi OS



19

How to install Raspberry-Pi-OS in the Memory Card (SD-card)

1. Install this imager in your local machine (Windows laptop).

2 Put the SD card in a suitable adapter and connect it in your local machine (laptop)

3. Run the software after installation.

4. Select an operating system (from the list) to be written to the memory card (SD card)

5. Select the memory card to write to. Let the installation proceed.

6. When done, take out the card and put it in the slot of the Raspberry-Pi module (board) and power it up.

Setting-Up:

i) You, just need to set the **time-zone/keyboard** settings.

ii) Also, configure the **wireless network** by selecting one of the available networks and providing the password for the same.

Note down the IP address of the Raspberry-Pi board.

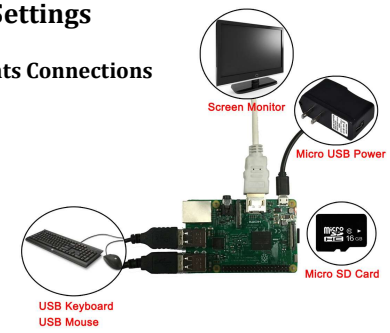
The remote desktop (GUI) login with **VNC/Remote-Desktop** is already set up.

Also, a **PuTTY**-based based login is possible (terminal/command prompt).

20

System Settings

Components Connections



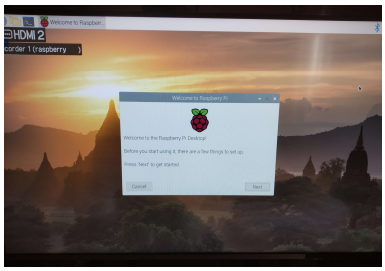
21

System Booting

- Remember to connect the R-pi board with a HDMI compatible monitor/TV
- Connect a USB Mouse and Keyboard with it
- Insert the SD Card on Raspberry-Pi board and Power it up

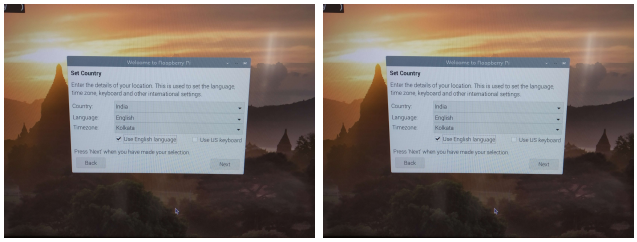
22

Booting the Raspberry-Pi for the First Time



23

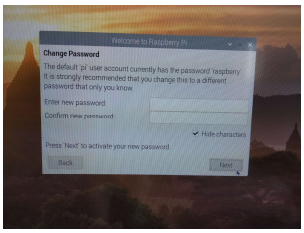
Booting the Raspberry-Pi for the First Time



Set the Country/Language/Time-Zone

24

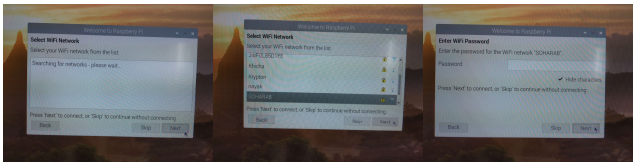
Booting the Raspberry-Pi for the First Time



Optionally you may change the password of the default user 'pi'. The default password is 'raspberry'

25

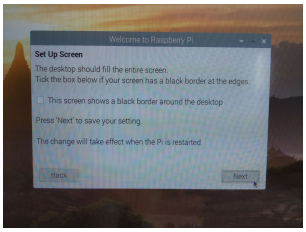
Booting the Raspberry-Pi for the First Time



Set the Wi-Fi network

26

Booting the Raspberry-Pi for the First Time



Optionally change the screen settings

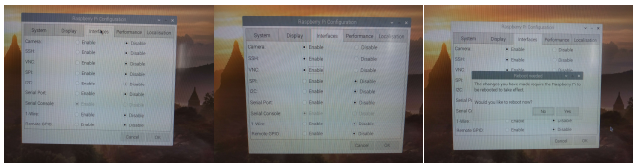
27

Booting the Raspberry-Pi for the First Time

Click here

28

Booting the Raspberry-Pi for the First Time



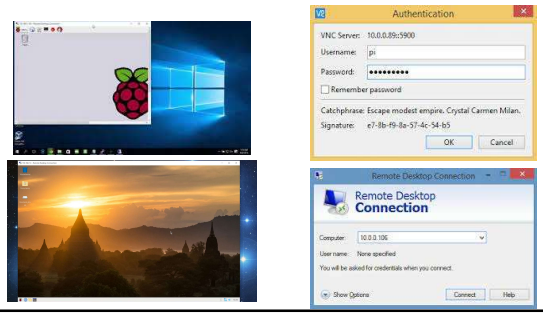
Go to the Interfaces tab. Enable the Camera, SSH and VNC.
This will enable the camera port where we want to connect the camera module.
SSH and VNC are required for remote login to the Raspbian OS.
Reboot the system.

29

Remote Desktop

30

Accessing with Remote Desktop



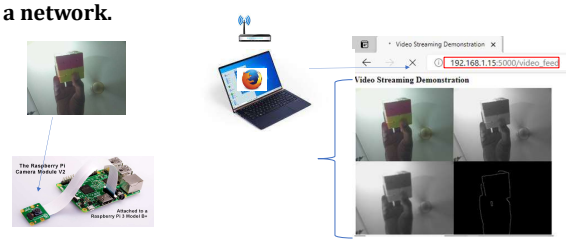
31

What are we going to do in this experiment?

32

Expected Output

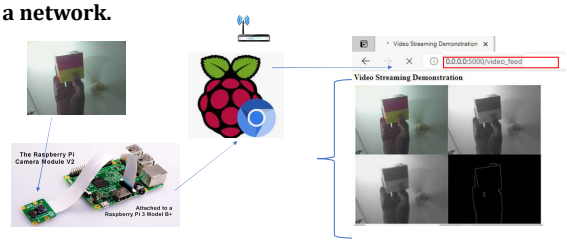
Live Stream Video along with the *processed frames* to the browser to be accessed with the IP address in a network.



33

Expected Output

Live Stream Video along with the *processed frames* to the browser to be accessed with the IP address in a network.



34

Web Application

- **Client Server Communication**
- Server stores and runs different services (The server is identified with an IP address; different services run on different ports)
- The Client sends a request to the server (located at a specific address)
- The server sends a response back to the client.

35

Web Application

- **Running a Camera Web-Application**
- The Raspberry-Pi runs a web-application (acts as a server) (***Flask*** app)
- The Client (web-browser running on a machine in the same network – laptop, smart-phone, tab etc.) sends a request to the server (IP address of the Raspberry-Pi)
- The server sends a response (processed and merged video frames) back to the client (web-browser - Firefox/Chrome/Edge etc.).

36

Image Processing

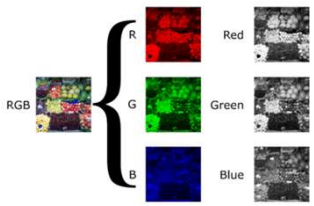
OpenCV
Open Source Computer Vision

https://docs.opencv.org/master/d9/df8/tutorial_root.html

37

RGB Image

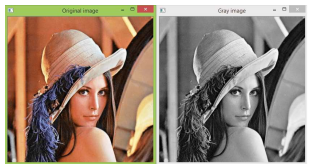
OpenCV
Open Source Computer Vision



38

RGB to Gray

OpenCV
Open Source Computer Vision



```
import cv2

image = cv2.imread('C:/Users/N/Desktop/Test.jpg')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

cv2.imshow('Original image',image)
cv2.imshow('Gray image', gray)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Grayscale = (R + G + B) / 3.

$$Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

https://docs.opencv.org/3.4/de/d25/imgproc_color_conversions.html

39

Image Blurring

OpenCV
Open Source Computer Vision


```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt

img = cv.imread('opencv-logo-white.png')

blur = cv.GaussianBlur(img,(5,5),0)

plt.subplot(121),plt.imshow(img),plt.title('Original')
plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(blur),plt.title('Blurred')
plt.xticks([]), plt.yticks([])
plt.show()
```

https://docs.opencv.org/master/d4/d13/tutorial_py_filtering.html

40

Edge Detection

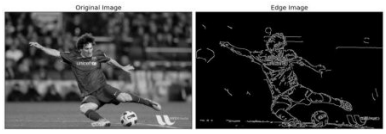
OpenCV
Open Source Computer Vision

```
import numpy as np
import cv2 as cv
from matplotlib import pyplot as plt

img = cv.imread('messi5.jpg',0)
edges = cv.Canny(img,100,200)

plt.subplot(121),plt.imshow(img,cmap = 'gray')
plt.title('Original Image'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(edges,cmap = 'gray')
plt.title('Edge Image'), plt.xticks([]), plt.yticks([])

plt.show()
```



https://docs.opencv.org/master/da/d22/tutorial_py_canny.html

41

What are the hardware components we need for this experiment?

42

All we need

Micro USB Cable for Raspberry Pi 3

Use Laptop to remotely login to the Raspberry-Pi

Raspberry Pi 3 Module

Micro SD Card

Pi-Camera Module

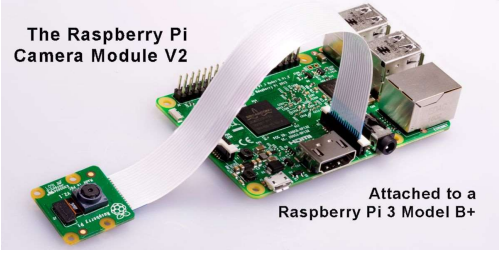
43

Camera Module

- This **5mp camera** module is capable of 1080p video and still images and connects directly to your raspberry pi. Connect the included ribbon cable to the **CSI (Camera Serial Interface)** port on your Raspberry Pi.
- The board itself is tiny, at around 25mm x 20mm x 9mm and weighing in at just over 3g, making it perfect for mobile or other applications where size and weight are important.
- The sensor has a native resolution of 5MP and has a fixed focus lens onboard. In terms of still images, the camera is capable of 2592 x 1944 pixel static images, and also supports 1080p30, 720p60 and **640x480p60/90** video.
- This module is only capable of taking pictures and video, not sound.

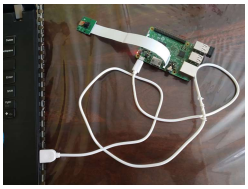
44

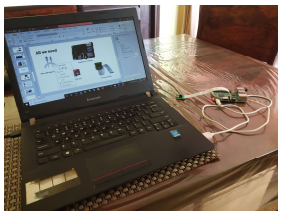
Camera connected to the R-Pi Board



45

Our Set Up – Let's make it Simple





46

Programming with Python

OpenCV
Open Source Computer Vision

Image Processing

Flask
web development, one drop at a time

Web Application

Let's check the code

Web browser (client app)

https://github.com/soharabhossain/Raspberry-Pi/tree/main/Camera%20Stream

47

```
#-------------------------------------------------------------

# Create a Flask app
app = Flask(__name__)

# What to render at the apps web address?
@app.route('/')
def index():
    """Video streaming home page."""
    return render_template('index.html')

# Video will be streamed here
@app.route('/video_feed')
def video_feed():
    """Video streaming route. Put this in the src attribute of an img tag."""
    return Response(gen(MyCamera()), mimetype='multipart/x-mixed-replace; boundary=frame')

#-----------------------------------------------
#----------------Main Program-------------------
#-----------------------------------------------

if __name__ == '__main__':

    print('\n Now running the web server.....')
    # Default port is 5000
    app.run(host='0.0.0.0', debug=True, threaded=True)
```

File to be rendered

Stream from this address will be rendered to the specified link

Port Number

IP address of the server in the local machine

48

Index.html inside templates sub-directory

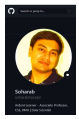
index - Notepad

File Edit Format View Help

```
<html>
 <head>
  <title>Video Streaming Demonstration</title>
 </head>
 <body>
  <h1>Video Streaming Demonstration</h1>
  <img src="{{ url_for('video_feed') }}">
 </body>
</html>
```

Stream will be taken from this source

49



Thank You

Class material will be uploaded here:
https://github.com/soharabhossain/Raspberry-Pi/tree/main/Camera%20Stream

50