B EE 525 Embedded System Design

Lab2: Interfacing Camera with Rpi & ML inference

Madhava Vemuri, Ph.D.
Assistant Professor, EE
E&M, School of STEM

Winter, Quarter'25

Lab and Project Deadline

Week	Mon	Description	Wed	Description
8	Feb 24	Linux Programming	Feb26	Lab1:GPIO Pins
9	Mar3	Final Project Discussion **Lab1 Due by Mar 5 Midnight**	Mar6	No Class Only Office Hours (Virtual and in person)
10	Mar10	No Class Only Office Hours (Virtual and in person)	Mar 13	No Class Only Office Hours
11	Mar17	Project Report + Video Submission **Due on Mar 16th Midnight (Sunday)** **No Extensions on Final Submission after**		

Lab Objectives

> Connect the **camera module** to the RPi and take pictures

> Use the system for **machine learning (ML) inference**

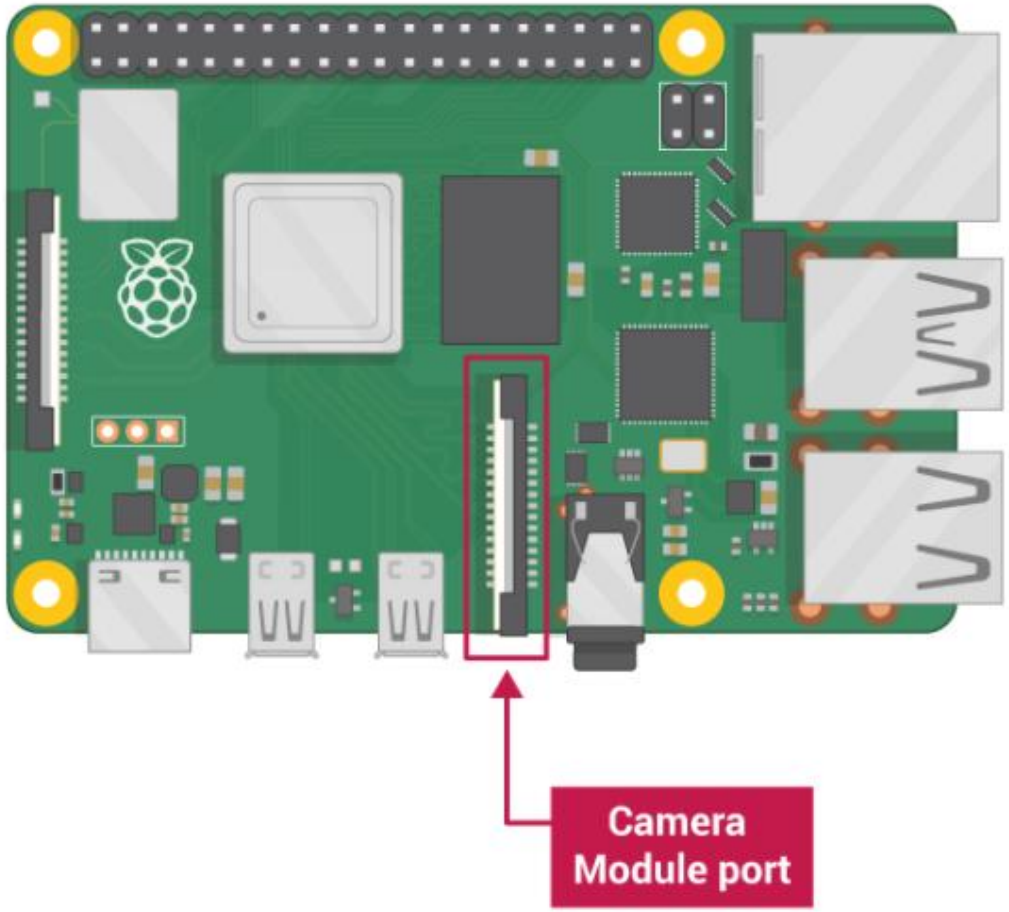
Textbook: Molly, “Exploring Raspberry Pi ”

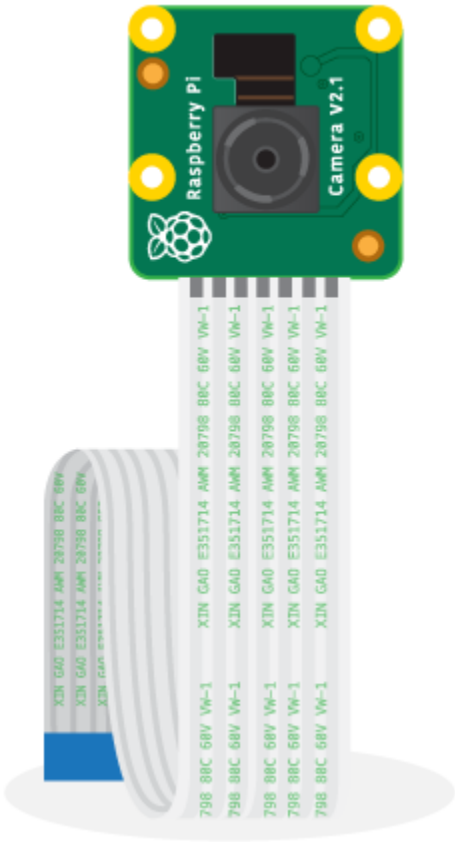
Materials Needed

- RPi
- Camera module
- Software code

Connect the Camera Module

> Prepare your **RPi** and **camera module**

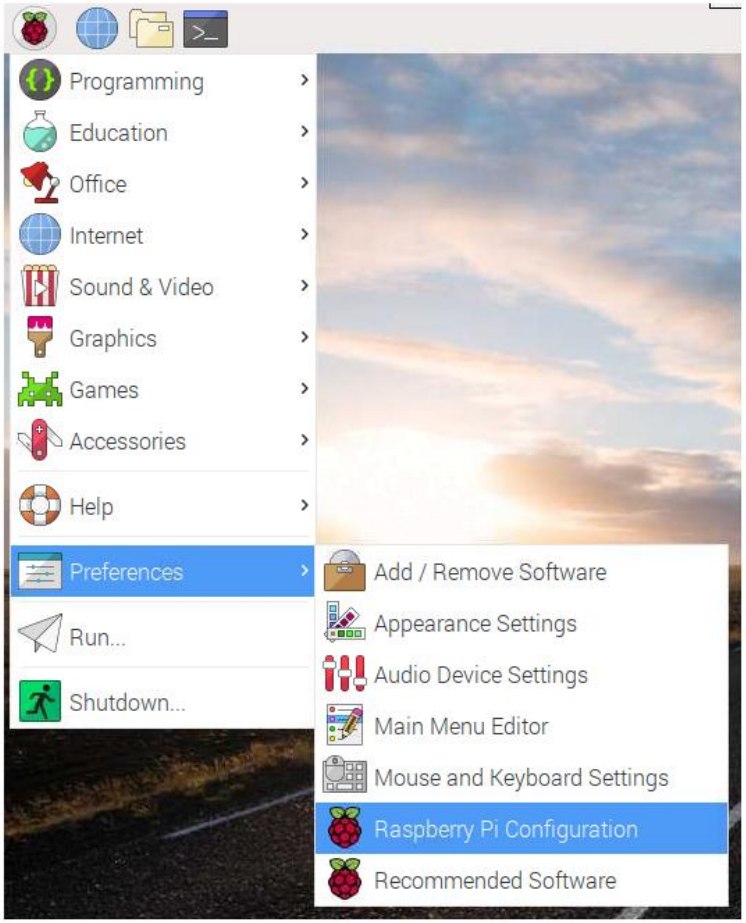


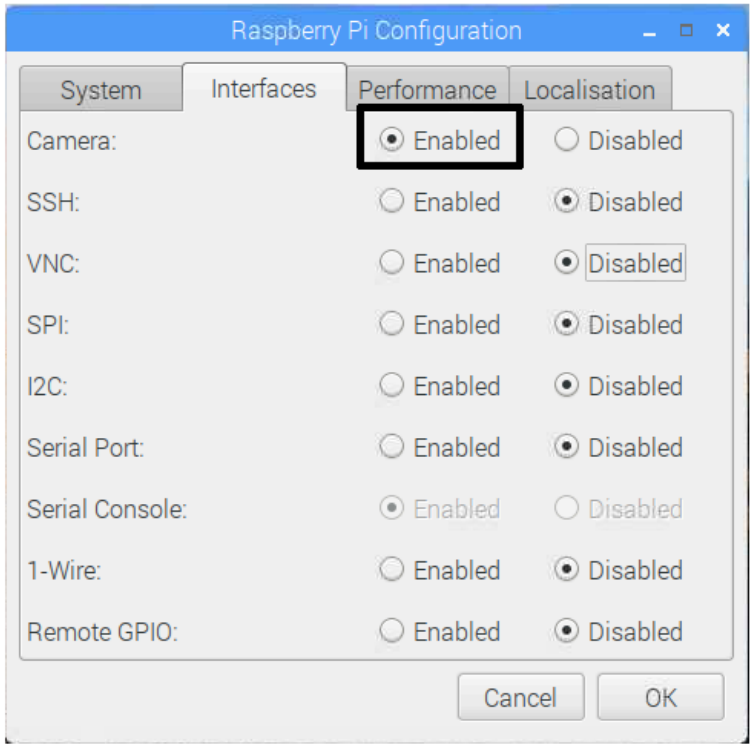


[Source] https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/

Connect the Camera Module (cont.)

> Configuration: method 1 (using GUI)



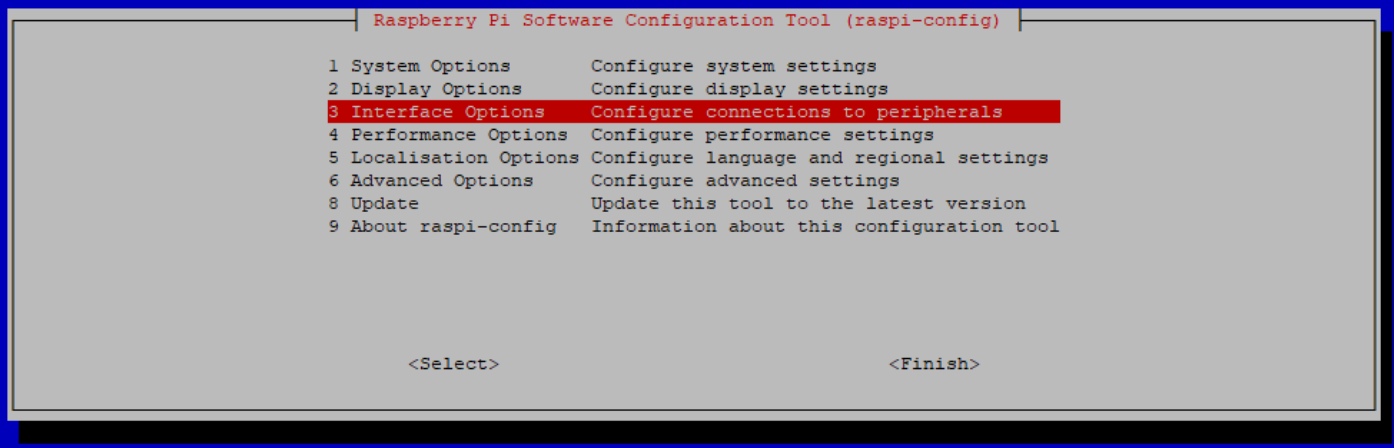


You may need to reboot your RPi

[Source] https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/

Connect the Camera Module (cont.)

- Configuration: method 2 (using a command)
  - pi@raspberrypi:~ $ **sudo raspi-config**
  - Select **Interfacing Options** > **P1 Camera** and enable your camera



You may need to reboot your RPi

Taking a picture using Python Code

- Install the library before running the code

```
sudo apt-get install python3-picamzero
```

- Import the **picamzero** library in your python script
  - `from picamzero import Camera`
- Create a camera object using Camera class
  - `cam = Camera()`
- Start a preview on the camera
  - `cam.start_preview()`
- Wait for 4 seconds before taking the picture for
  - `sleep(4)`
- Use take_photo() function to capture and save a image in a specific location
  - `cam.take_photo(f"{path_to_dir}/{file_name}")`
- End the camera preview
  - `cam.stop_preview()`

Taking a picture using Python Code

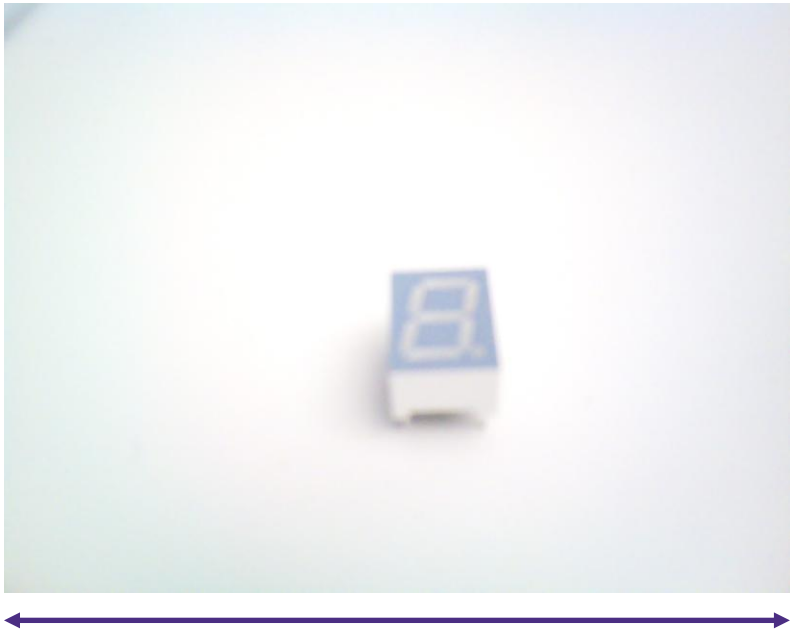
- Install the library before running the code

```
sudo apt-get install python3-picamzero
```

- Import the **picamzero** library in your python script
  - `from picamzero import Camera`
- Create a camera object using Camera class
  - `cam = Camera()`
- Start a preview on the camera
  - `cam.start_preview()`
- Wait for 4 seconds before taking the picture for
  - `sleep(4)`
- Use take_photo() function to capture and save a image in a specific location
  - `cam.take_photo(f"{path_to_dir}/{file_name}")`
- End the camera preview
  - `cam.stop_preview()`

Taking a picture using Python Code

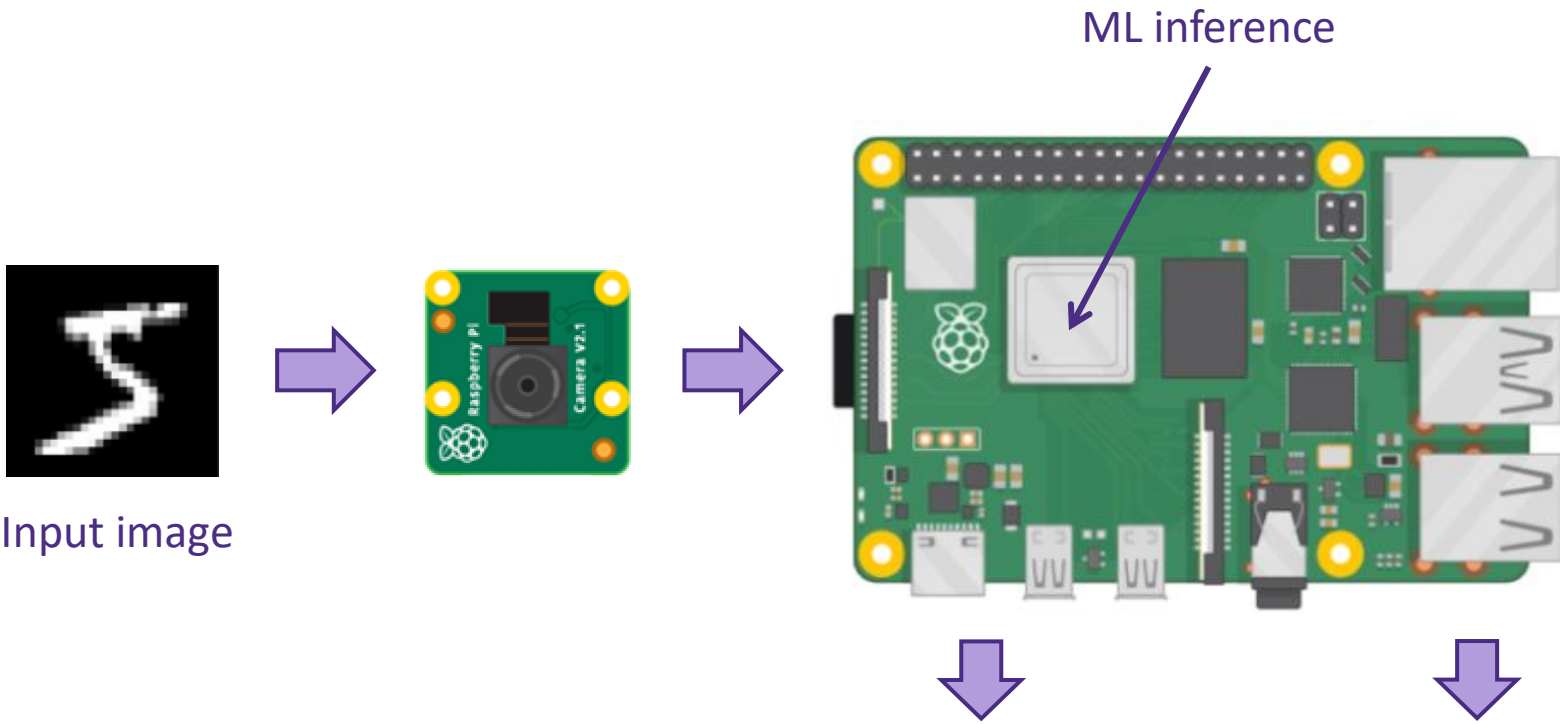
```
> python capture_image.py
```



2592 X 1944 pixel

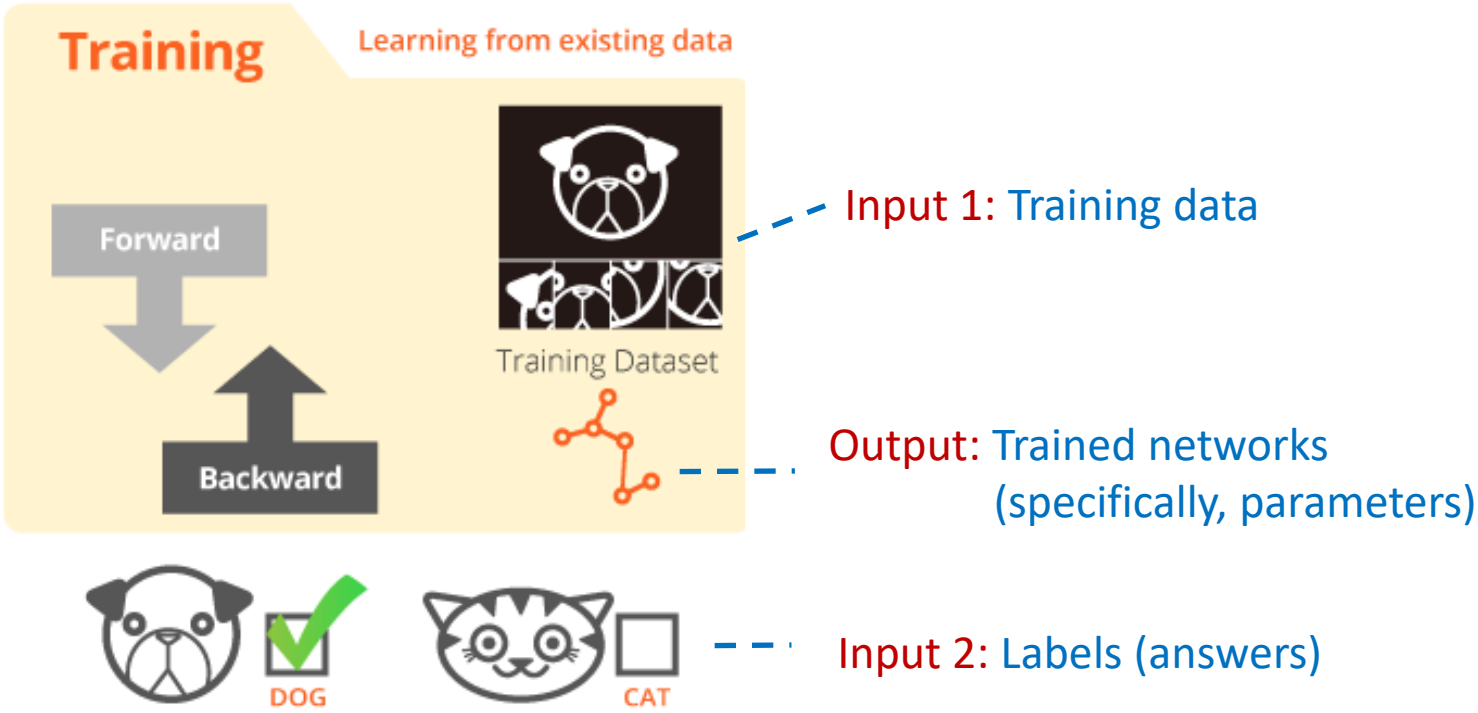
Our Specific Goal Today

> **Capture** an input image using the **camera module** and **classify** it by using **ML** software program running on the RPi



Background: Supervised ML

> A basic ML process is divided into two phases:
  - **Training** and **Inference**

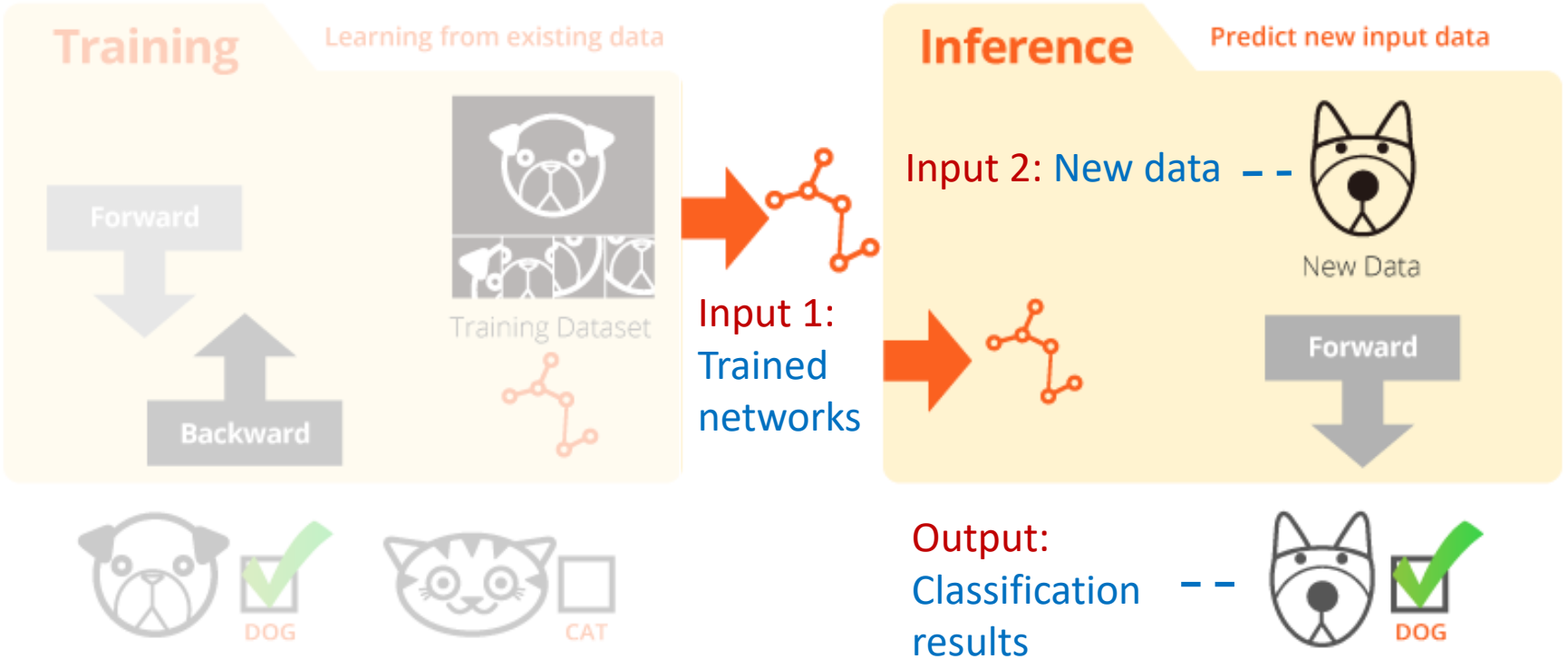


[Source] https://www.qnap.com/solution/openvino/en/

Background: Supervised ML (cont.)

> A basic ML process is divided into two phases:

– **Training** and **Inference**

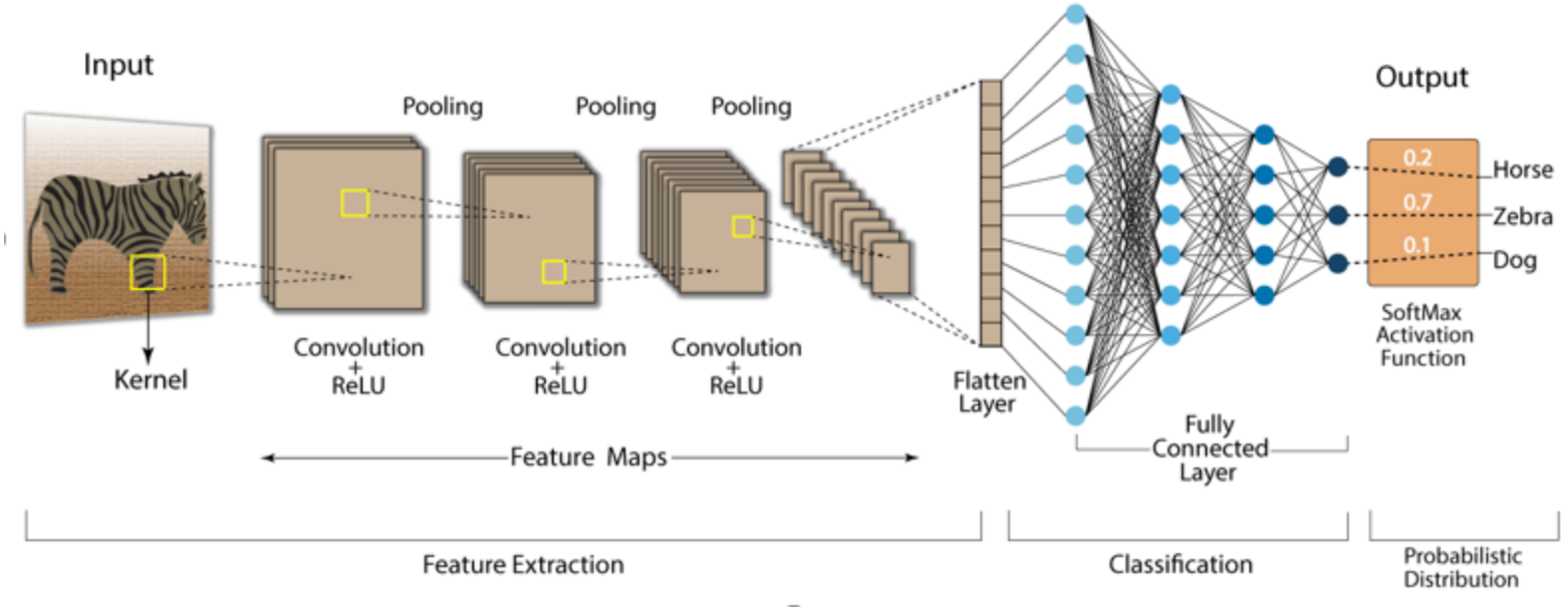


[Source] https://www.qnap.com/solution/openvino/en/

Background: Supervised ML (cont.)

> Architecture of **convolutional neural networks (CNN)**

- **Convolutional (Conv) layers**: detect the presence of specific features
- **Pooling layers**: reduce the resolution while remaining features
- **Fully connected (FC) layers**: calculate probabilities of classes



[Source] Developers Breach

Installing and Initializing the tensorflow library

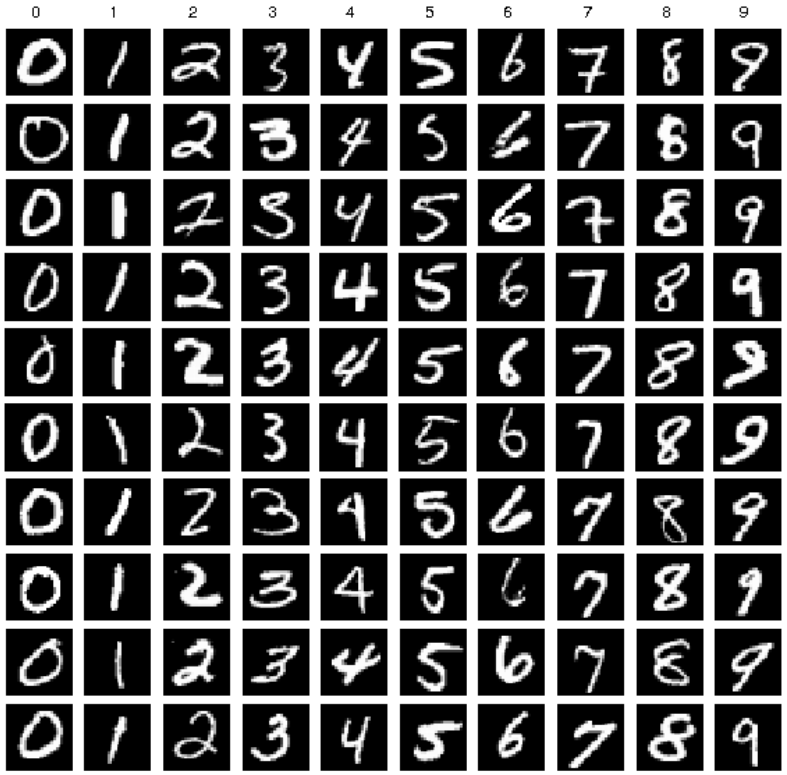
- Create a virtual environment in python to install the tensorflow libraries
  - `python3 –m venv tf-env`
- Activate the environment to install the tensorflow library
  - `source tf-env/bin/activate`
- The environment is activated, and you can see the change in the command prompt **(tf-env)** added in front

```
(tf-env) raspberrypi@raspberrypi:~/Desktop/Labs/Final_Project $
```

- To install the tensorflow libraries
  - `pip install tensorflow`
- To import the tensorflow library just write the following command in the python script
  - `import tensorflow as tf`

Background: Our Data Used for Training

- **MNIST** dataset (handwritten digit dataset)
  - Monochrome images (background color = black)
  - Image size = 28x28



[Source] Opengenus

Demonstration

- An image is taken **4 seconds** after running **capture_image.py**
- The prediction result depends on how the image is taken
- Tips
  - Place the number in the **middle of the image**
  - Make the number **stand upright**
  - Fill the image with a **black background**

Good example
(prediction result = 2)

Bad example
(prediction result = 7)

2

5

8

1

9

Final Deadline

- Submission deadline: **Mar. 16th at 11:59PM**
  - Same as the Lab 1 report and Video submission
  - Upload your report and Video to Canvas
  - Capture the Video while predicting each image
- Page limit: **4 pages**