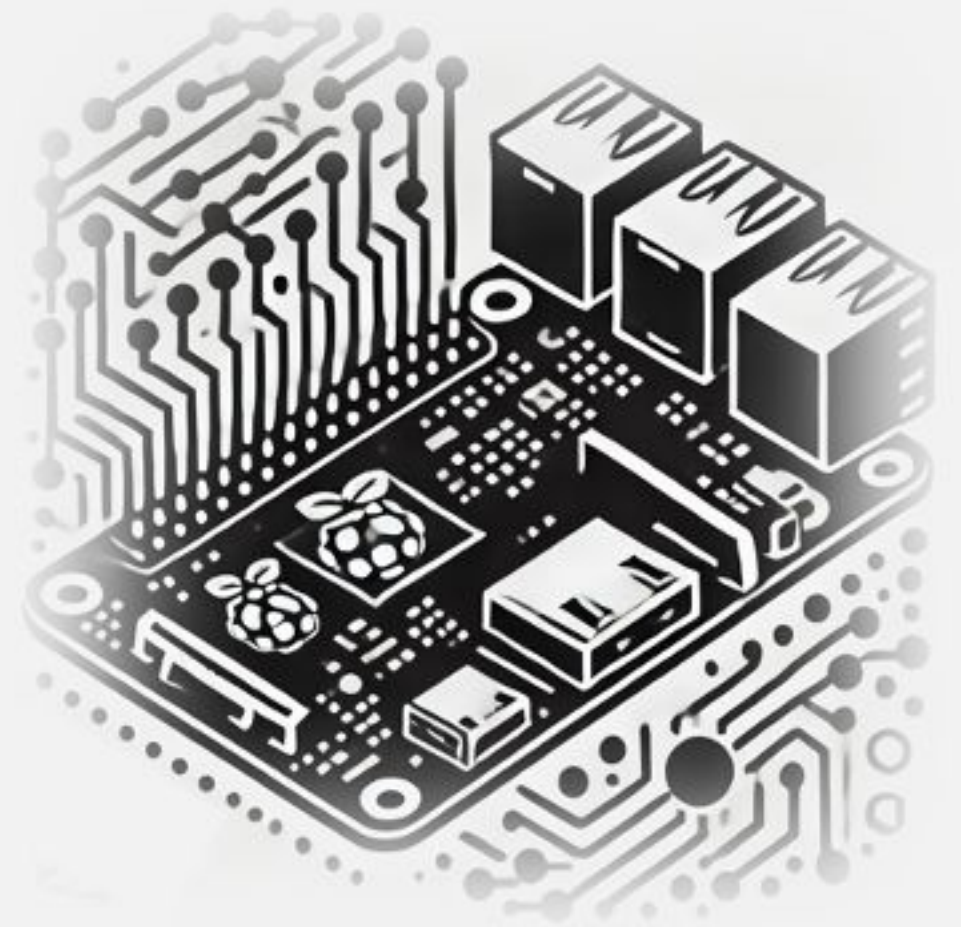


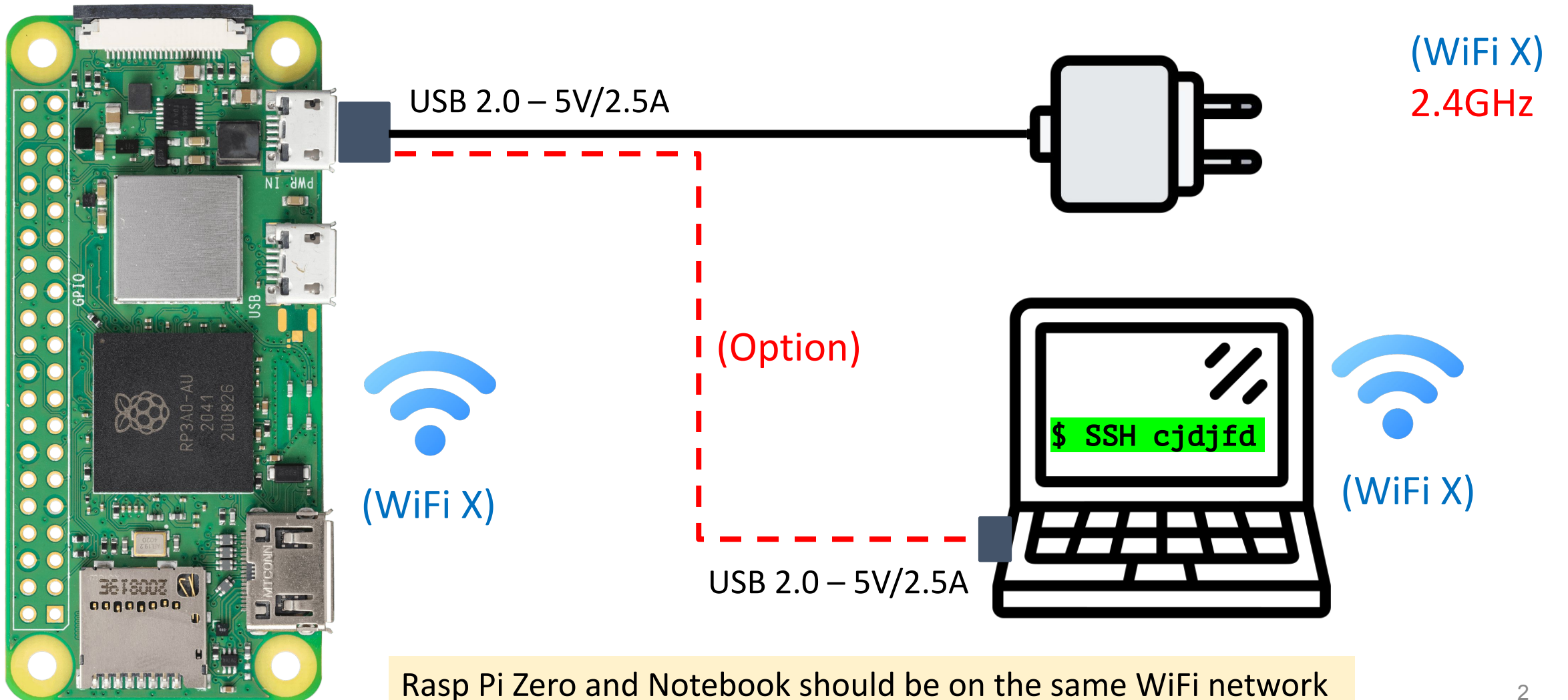
IESTI05 – Edge AI

Machine Learning System Engineering

4. Python Setup & Camera Handling on Embedded Linux



Power Supply and WiFi Network



Installing Required **System Level** Libraries

1. Ensure your Raspberry Pi is **up to date**:

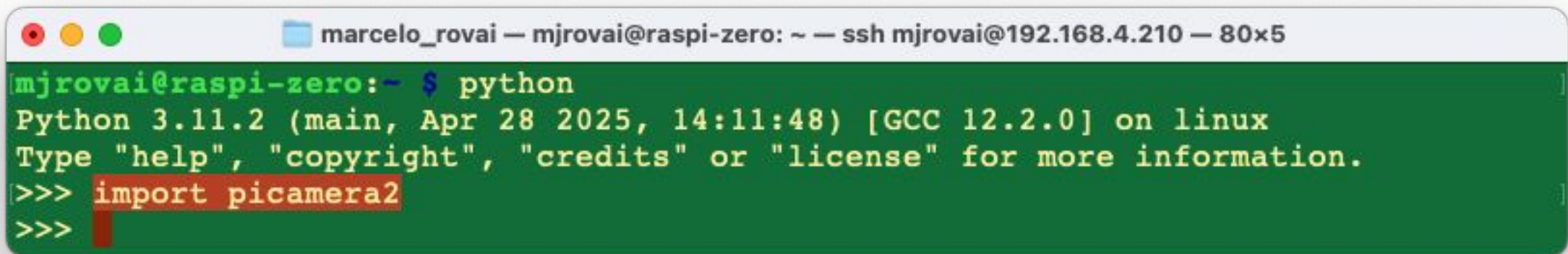
```
sudo apt update  
sudo apt upgrade -y  
sudo reboot # Reboot to ensure all updates take effect
```

Root Installation

2. Install **Python tools** and **camera libraries** (Picamera2)

```
sudo apt install -y python3-pip python3-venv python3-picamera2  
sudo apt install -y libcamera-dev libcamera-tools libcamera-apps
```

3. Testing **picamera2** installation



```
marcelo_rovai — mjrovai@raspi-zero: ~ — ssh mjrovai@192.168.4.210 — 80x5  
[mjrovai@raspi-zero:~ $ python  
Python 3.11.2 (main, Apr 28 2025, 14:11:48) [GCC 12.2.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> import picamera2  
>>>
```

Setting up a Virtual Environment

1. Create a **virtual environment** with access to system packages to manage dependencies:

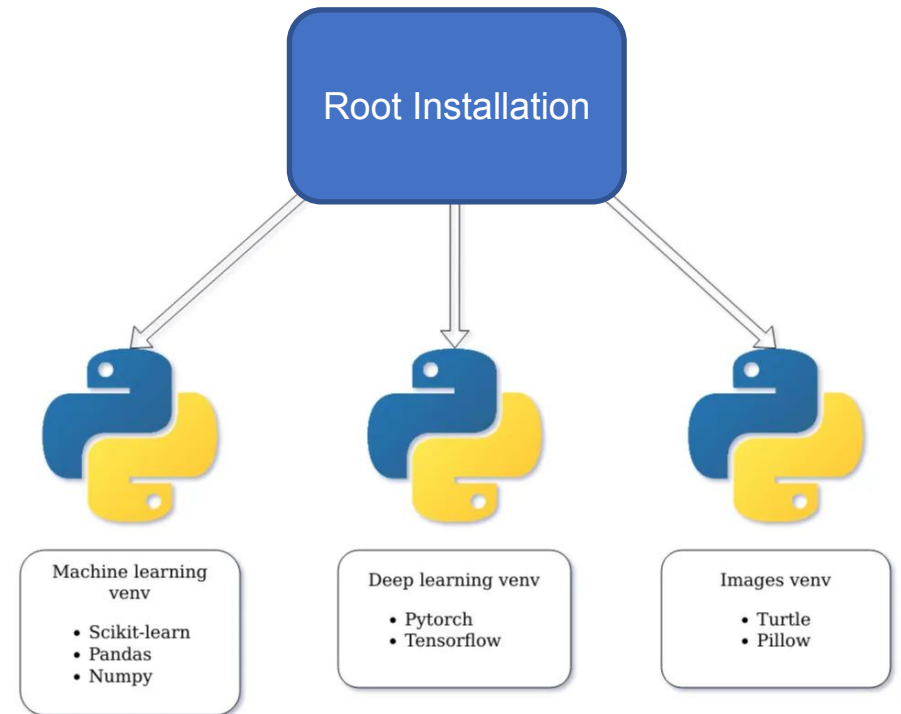
```
python3 -m venv ~/tflite_env --system-site-packages
```

2. **Activate** the environment:

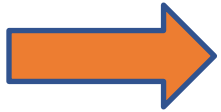
```
source ~/tflite_env/bin/activate
```

To **exit** the virtual environment, use:

```
deactivate
```



Install Python Packages (inside Virtual Environment)



Ensure you're in the virtual environment (venv)

1. Install Python Packages

```
pip install numpy pillow # Image processing
pip install matplotlib # For displaying images
pip install opencv-python # Computer vision
```

2. Verify installation

```
pip list | grep -E "(numpy|pillow|opencv|picamera)"
```

If the Numpy version is not 1.24.2

```
pip uninstall numpy
pip install numpy==1.24.2
```

A terminal window titled 'marcelo_rovai — mjrovai@raspi-zero: ~ — ssh mjrovai@192.168.4.210 — 95x7'. The prompt is '(tflite_env) mjrovai@raspi-zero:~\$'. The command 'pip list | grep -E "(numpy|pillow|opencv|picamera)"' has been executed. The output is as follows:

Package	Version
numpy	1.24.2
opencv-python	4.12.0.88
picamera2	0.3.30
pillow	11.3.0

The prompt is now '(tflite_env) mjrovai@raspi-zero:~\$'. A blue arrow points from the '1.24.2' version number in the output to the 'If the Numpy version is not 1.24.2' text box.

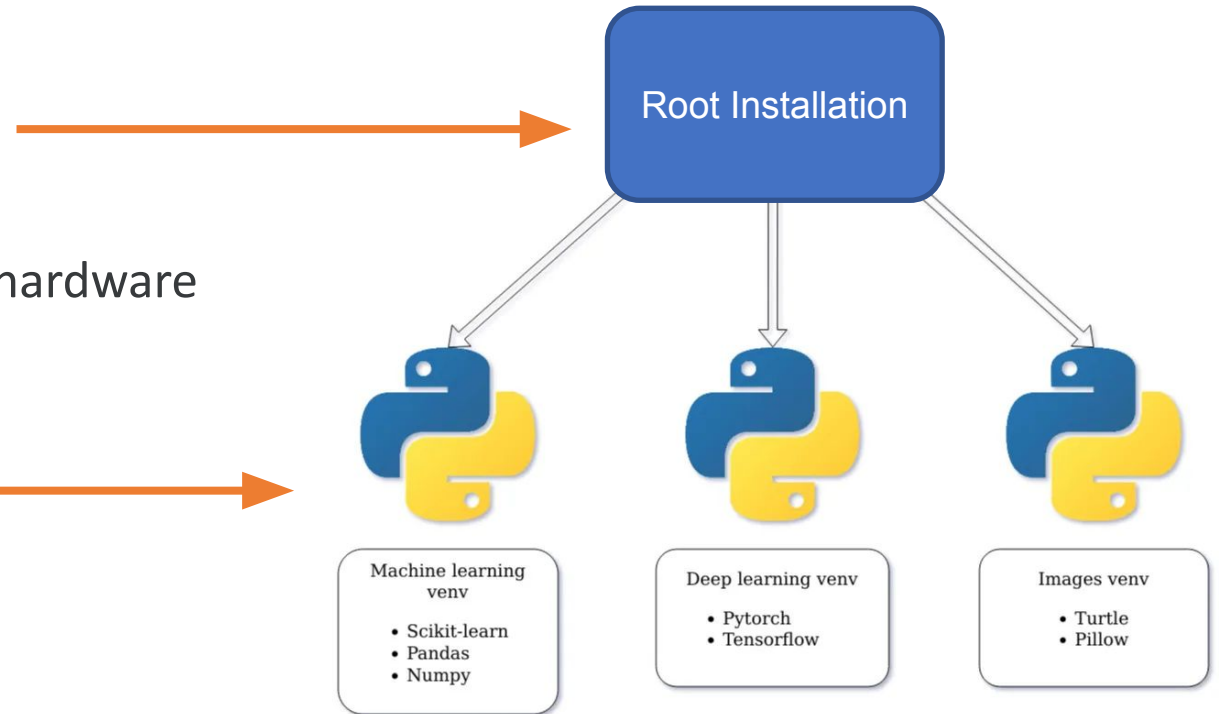
System vs pip Package Installation Rule

Use sudo apt install (outside of venv) for:

- System-level dependencies and libraries
- Hardware interface libraries (as a camera)
- Development headers and build tools
- Anything that needs to interface directly with hardware

Use pip install (inside venv) for:

- Pure Python packages
- Application-specific libraries
- Packages that don't need system-level access



Rule of thumb: Use sudo apt install only for system dependencies and hardware interfaces. Use pip install (without sudo) inside an activated virtual environment for everything else. Inside the vent, PIP or PIP3 are the same.

Setting up Jupyter Notebook



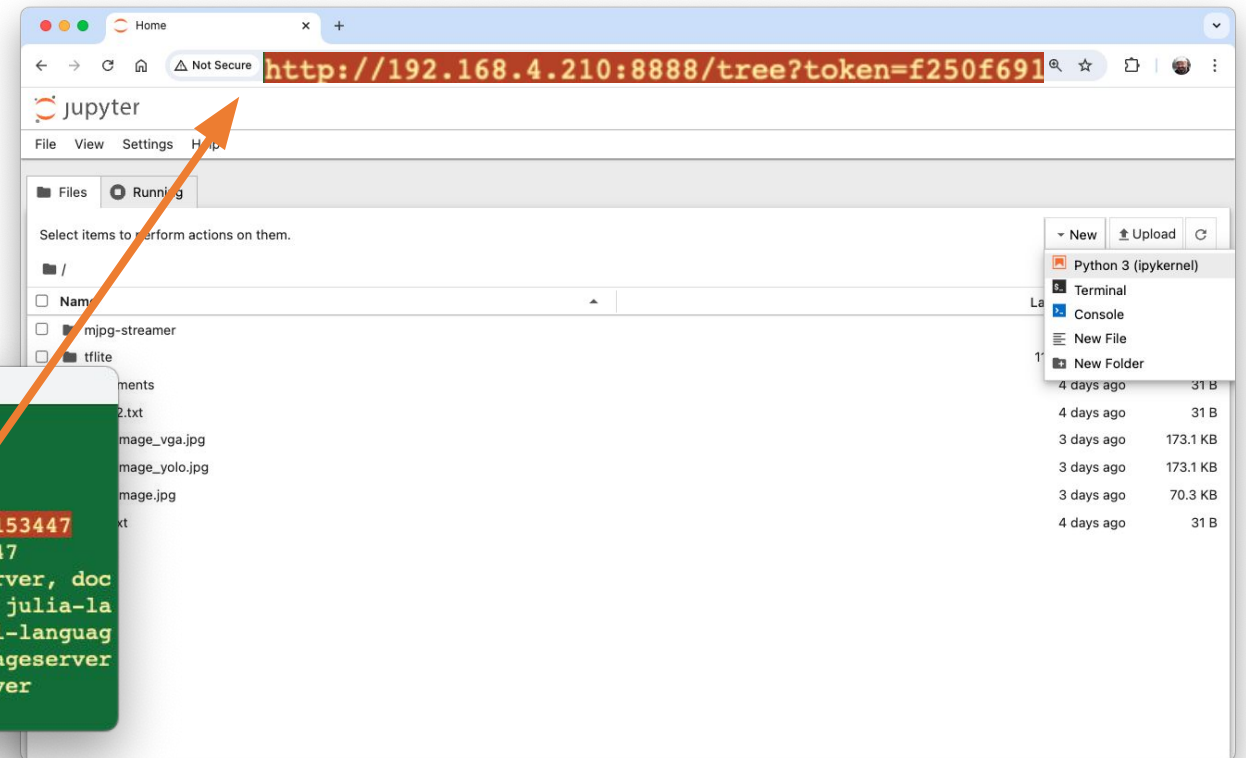
```
pip install jupyter jupyterlab notebook
```

```
jupyter notebook --ip=[YOUR IP ADDRESSES] --no-browser
```

Copy the **Raspberry Pi's IP address** and the provided **token** in a web browser.

```
marcelo_rovai — mjrovai@raspi-zero: ~ — ssh mjrovai@192.168.4.210 — 96x12

To access the server, open this file in a browser:
  file:///home/mjrovai/.local/share/jupyter/runtime/jpserver-7399-open.html
Or copy and paste one of these URLs:
  http://192.168.4.210:8888/tree?token=f250f69117df5adf9d1aff01dd3ede657cb47b0ba8153447
  http://127.0.0.1:8888/tree?token=f250f69117df5adf9d1aff01dd3ede657cb47b0ba8153447
[I 2024-08-23 19:40:59.979 ServerApp] Skipped non-installed server(s): bash-language-server, doc
kerfile-language-server-nodejs, javascript-typescript-langserver, jedi-language-server, julia-la
nguage-server, pyright, python-language-server, python-lsp-server, r-languageserver, sql-languag
e-server, texlab, typescript-language-server, unified-language-server, vscode-css-languageserver
-bin, vscode-html-languageserver-bin, vscode-json-languageserver-bin, yaml-language-server
```



Creating a working directory

```
cd Documents
```

```
mkdir Python
```

```
cd Python
```

Load an image from the internet

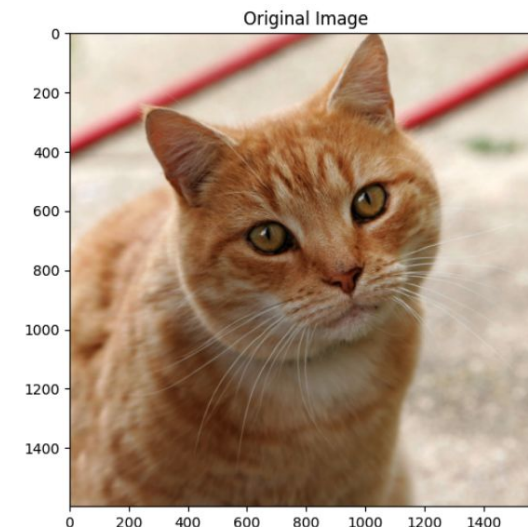
```
wget https://upload.wikimedia.org/wikipedia/commons/3/3a/Cat03.jpg
```



```
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image

img_path = "Cat03.jpg"
img = Image.open(img_path)

# Display the image
plt.figure(figsize=(6, 6))
plt.imshow(img)
plt.title("Original Image")
plt.show()
```



Capturing an Image with the Camera



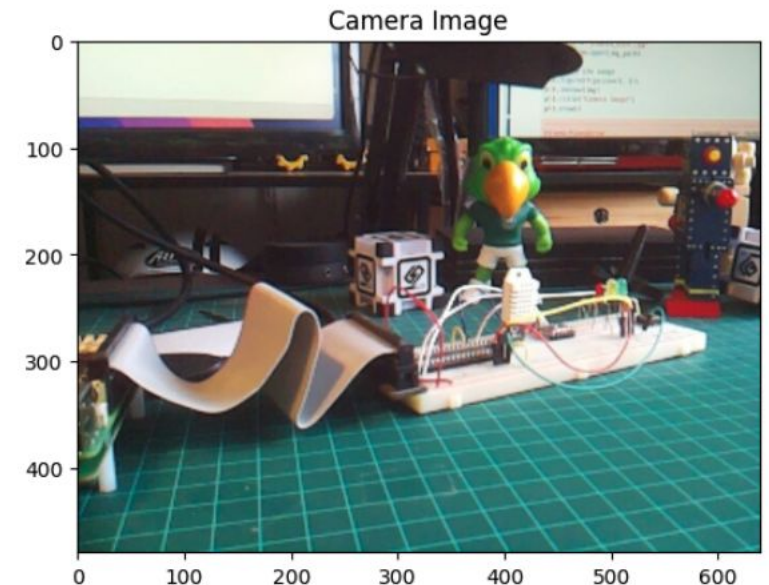
```
from picamera2 import Picamera2
import time

# Initialize camera
picam2 = Picamera2()
picam2.start()

# Wait for camera to warm up
time.sleep(2)

# Capture image
picam2.capture_file("class3_test.jpg")
print("Image captured: class3_test.jpg")

# Stop camera
picam2.stop()
picam2.close()
```



Questions?



Prof. Marcelo J. Rovai

rovai@unifei.edu.br



UNIFEI