Lab Report for

CE6023 – Computer Vision Systems

Student Name : **Dylan Rodrigues**

Student ID : **24121479**

Revision Timestamp: 08/10/2024 13:46:05

Contents

Introduction

In this lab, we explored various aspects of setting up and using the Raspberry Pi 4 (RPi4) for camera-based applications, integrating both hardware and software components to capture images and video. Starting with the physical assembly of the RPi4 kit and connecting the Camera Serial Interface (CSI) to the camera module, we ensured that the system was properly configured to function with the PiCam module. Using Linux-based commands like `sudo raspistill` and `sudo raspivid`, we captured still images and videos, including student ID verification, and transferred these media files to a laptop using tools such as the `scp` command and FileZilla. We also delved into the specifications of the camera module, specifically comparing the **Raspberry Pi Camera Module 2** with industrial-grade cameras used in fields like automotive, medical imaging, and manufacturing. These comparisons highlighted the limitations of the Raspberry Pi Camera in terms of resolution, speed, and sensor quality when contrasted with advanced cameras required for high-performance industrial tasks. Throughout the lab, a focus was placed on understanding the practical applications and challenges associated with the Raspberry Pi camera, setting the stage for deeper investigations into its potential uses and how it compares to other similar modules.

Setting up Wi-Fi connection on University of Limerick IOT network

To set up a Wi-Fi connection on the University of Limerick IoT network using the Raspberry Pi 4 (RPi4), I followed the lab manual instructions with a few modifications suited to my macOS environment. After connecting a keyboard, mouse, and display to the RPi4, I powered on both my laptop and the RPi4. Instead of using puTTy, I used the terminal's SSH command to connect to the RPi4. I successfully connected to the university's "ul iot" network and confirmed that the RPi4 was not connected to the internet by attempting to access Google. Using the `ifconfig` command, I retrieved the RPi4's WLAN IP address and noted it for remote connection. I then created a new user with administrative privileges to ensure security when accessing the device remotely. Finally, I used Windows App as a remote desktop client to access the RPi4, confirming a stable connection before closing the SSH session and powering down the RPi4 as instructed. All relevant details, including the IP address and new user credentials as illustrated in Figure 1: Ip & mac address, are recorded in my lab report.

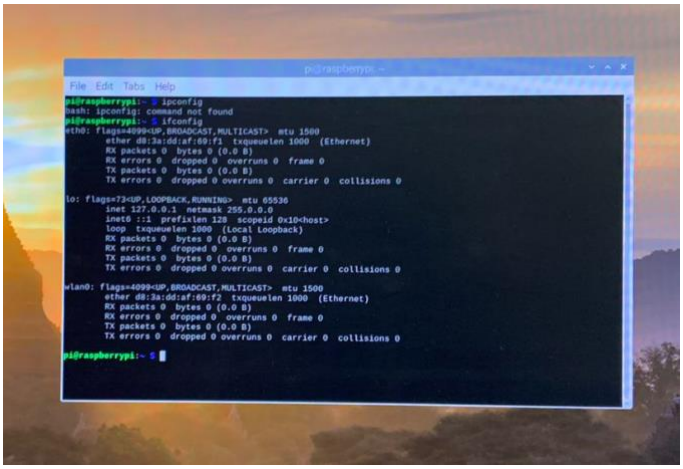

Figure 1: Ip & mac address

Setting up a Camera

To set up the camera module on the Raspberry Pi 4 (RPi4), I began by opening the casing of the RPi4 kit to expose the Raspberry Pi board, identifying the Camera Serial Interface (CSI) connector as shown in the lab manual. Following the instructions, I carefully connected the flexible camera cable to the CSI connector, ensuring the blue side of the cable faced to the right as indicated in Figure 7 and Figure 8. I then took a photo of my setup, including my student ID card as illustrated in Figure 2: Camera Setup. After reconnecting the RPi4 kit and threading the camera cable out of the casing, I attached the other end of the cable to the camera module, confirming that the blue side of the cable faced the correct direction towards the board. This step completed the physical setup of the camera module on my RPi4.

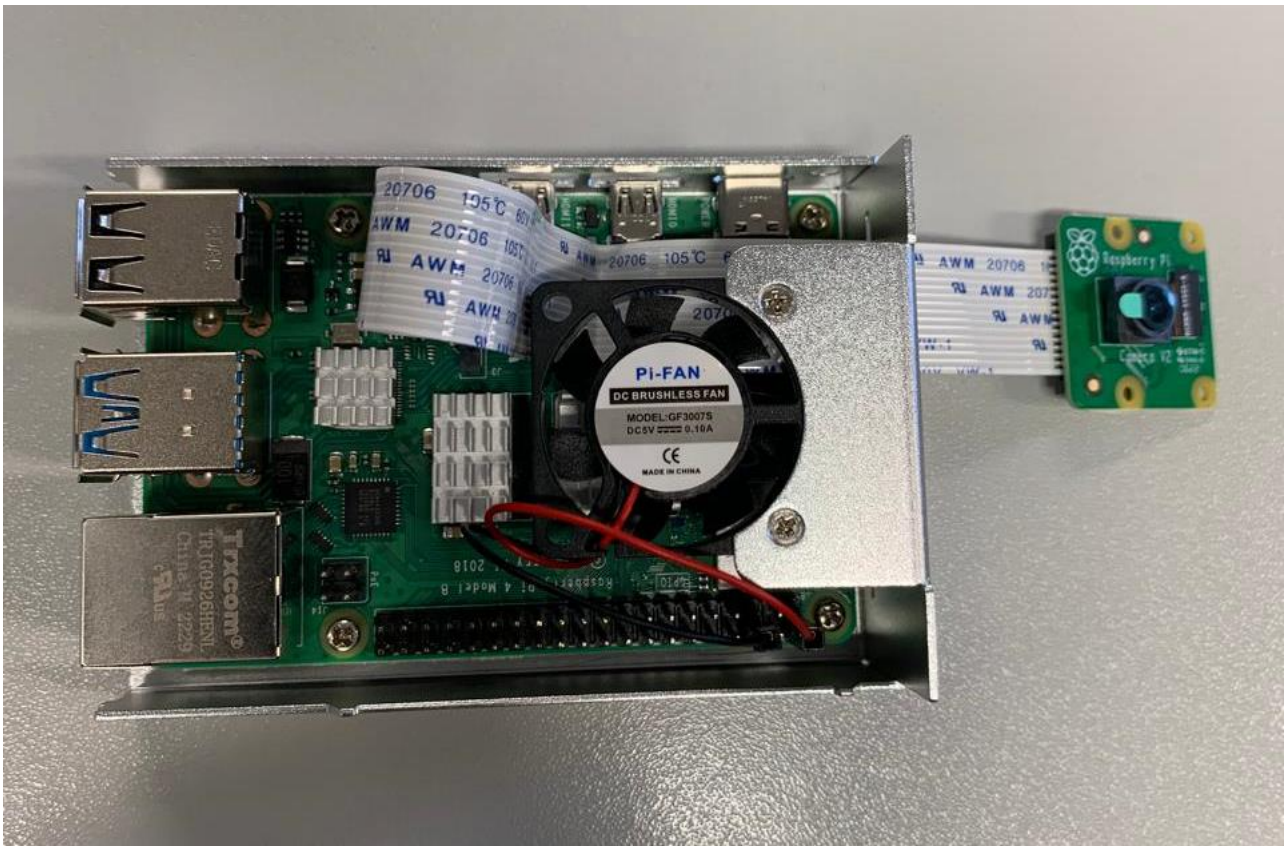

Figure 2: Camera Setup

Connect to RPi4 using remote desktop

To connect to the Raspberry Pi 4 (RPi4) using a remote desktop, I followed the instructions specific to my macOS environment. First, I ensured the RPi4 was powered on and connected to the university's Wi-Fi network, using the previously configured Wi-Fi IP address. I then opened the terminal on my Mac and used the `ssh` command to establish a secure connection to the RPi4, logging in with the new user credentials I had created earlier. After successfully connecting via SSH, I proceeded to set up a remote desktop connection using the Microsoft Remote Desktop app, entering the RPi4's IP address, username, and password to gain access to the RPi4's graphical interface. I avoided running the `sudo apt update && sudo apt upgrade -y` command as instructed in the lab manual, maintaining the current system configuration. Once connected, I could access and control the RPi4 remotely from my laptop.

Take a picture or video

To set up the PiCam module on the Raspberry Pi 4 (RPi4), I first watched the instructional video on how to properly connect the camera module to the board. Once the camera was securely in place, I used the `sudo raspistill -o output.jpg` command to capture a still picture of myself (I did not have my student ID card; hence as discussed with Prof. Patrick, I am uploading a picture of myself), verifying that the camera was functioning as expected. For further details about the camera functionality, I

explored additional options using the `sudo raspistill --help` command. Next, I captured a short video by running the `sudo raspivid -o video.h264` command, ensuring my student ID card was included in the video. After the video was saved in `.h264` format, I converted it to `.mp4` format using the `ffmpeg` package with the command `ffmpeg -framerate 24 -i video.h264 -c copy video.mp4`, making it easier to view on my laptop. The final step involved transferring the video to my laptop, where I confirmed that it played without any issues.

Moving files between RPi4 and Laptop

To transfer the image and video files from the Raspberry Pi 4 (RPi4) to my laptop, I used the `scp` command as suggested in the tutorial. This allowed me to securely copy the still image and video directly from the RPi4 to my laptop. Alternatively, for macOS users like myself, the FileZilla Client can be used to connect via SFTP using the RPi4's IP address, logging in with the newly created user credentials. Once connected, I was able to navigate through the RPi4's home directory and download the necessary files, including the image and video.

Upon reviewing the files, the still image captures my student ID card clearly as illustrated in myself as depicted in Figure 3: An Image demonstrating a functional RPi4 Camera, displaying the static scene with crisp details. In contrast, the video captures a dynamic scene, showing movement as I reposition the card, which adds depth and motion to the footage. The video is smoother but less sharp compared to the still image. The subjective quality of both files was decent, but the still image appeared clearer. The quality of the image could potentially be improved by adjusting the camera settings, such as increasing the resolution, adjusting exposure, or tweaking the focus. However, limitations of the PiCam's sensor and lighting conditions might still affect overall quality.

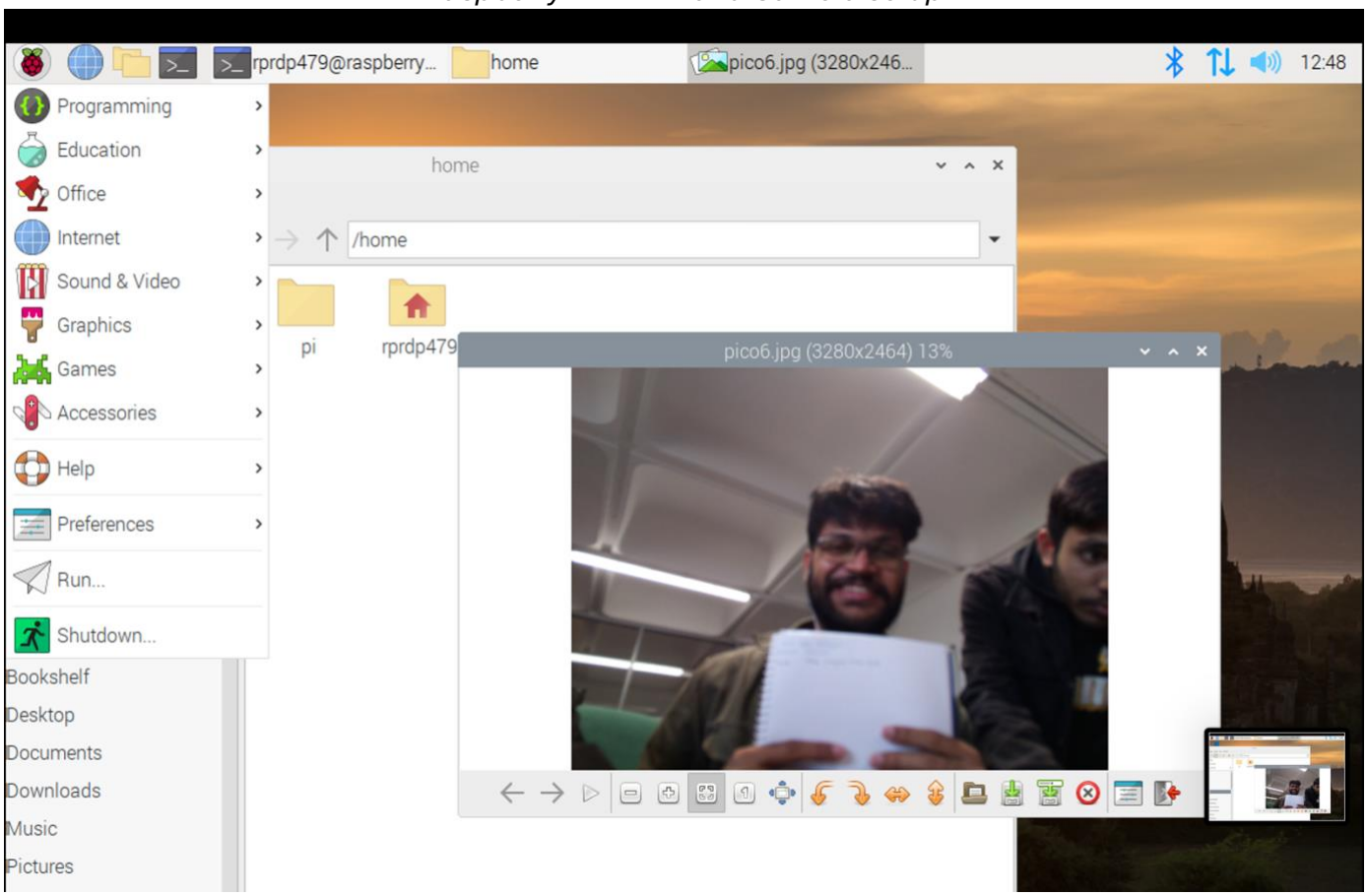

Figure 3: An Image demonstrating a functional RPi4 Camera

Comparison with other cameras

The camera module I used is the **Raspberry Pi Camera Module 2**, featuring an 8MP Sony IMX219 sensor (Raspberry Pi Co-op, 2024). It captures static images at 3280 x 2464 pixels and supports video recording at resolutions of 1080p at 30 fps, 720p at 60 fps, and 640x480p at 60/90 fps. This camera, while suitable for general-purpose applications like hobbyist projects and educational use, differs significantly from industrial cameras used in fields such as automotive, medical imaging, and manufacturing. Industrial cameras, such as those used in automotive systems, often offer higher frame rates, more advanced sensors for low-light conditions, and better durability to withstand harsh environments. For example, automotive cameras typically feature enhanced resolution, dynamic range, and precise calibration systems for tasks like autonomous driving, which the Pi Camera lacks.

In comparison to medical imaging, cameras used in this field must offer ultra-high resolution, exceptional color accuracy, and sometimes 3D imaging capabilities. While the IMX219 sensor can capture reasonably sharp static images and decent video for general use, it lacks the advanced imaging capabilities required for medical applications like MRI scans or endoscopy, where clarity, depth perception, and real-time data are critical. Similarly, in manufacturing, cameras are often part of robot vision systems that rely on fast image processing and high accuracy for detecting defects or sorting objects. The Pi Camera falls short in terms of durability, precision, and processing speed compared to specialized industrial cameras designed for high-speed production lines.

Compared to other students' cameras, such as the **OKDo 5MP Camera** with the OV5647 sensor, the Raspberry Pi Camera Module 2 offers better image resolution and video capabilities. However, both cameras share limitations when compared to high-performance industrial cameras. The OKDo camera provides wide-angle lens options and slightly larger lens features, which could be more useful for broader scenes but still lacks the performance and sensor quality required for professional-grade applications. Engaging with class discussions, I found that the differences between these modules lie primarily in resolution, lens quality, and intended use cases, all of which limit their application to simple tasks rather than sophisticated industrial systems.

Conclusion

In conclusion, this lab successfully demonstrated the integration of the Raspberry Pi 4 with the PiCam module, showcasing the process of capturing high-quality images and videos while also emphasizing the importance of proper configuration and connection techniques. By utilizing Linux commands and file transfer protocols, we efficiently managed to move our media files to a personal laptop, reinforcing our understanding of the Raspberry Pi's capabilities. The comparative analysis of the Raspberry Pi Camera Module 2 with industrial cameras revealed critical insights into the performance limitations and suitability of each for specific applications, such as automotive and medical imaging. This exploration highlighted the Raspberry Pi camera's strengths in accessibility and ease of use while acknowledging its constraints in professional environments. Overall, the lab not only enhanced our practical skills but also provided a solid foundation for future projects involving computer vision and image processing, paving the way for further exploration in this rapidly evolving field.

References

Raspberry Pi Foundation. (2024). *Raspberry Pi Camera Module 2 Specifications*. Retrieved from Raspberry Pi Documentation: https://www.raspberrypi.com/documentation/accessories/camera.html

OKDo. (2023). *5MP Camera with OV5647 Sensor for Raspberry Pi*. Retrieved from OKDo Product Datasheets: https://www.okdo.com/p/5mp-camera-module-with-ov5647-sensor

Sony Semiconductor Solutions Corporation. (2023). *IMX219PQ CMOS Image Sensor*. Retrieved from Sony Product Information: https://www.sony-semicon.co.jp/products/en/imx219