Module Project Report for

CE6023 – Computer Vision Systems

Student Name : **Dylan Rodrigues**

Student ID : **24121479**

Revision Timestamp: 08/12/2024 23:33:08

Contents

Abstract

This project has addressed the challenge of real-time object and person recognition using a Raspberry Pi 4 with a camera module. Real-time identification is critical in applications related to surveillance and automation, although this faces many challenges, especially related to hardware, lighting issues, and misclassification problems. The system tracks objects and individuals over 20 seconds with innovative steps in optimizing the frame rate and testing under dim, normal, and bright yellow light conditions to check the performance. Results are shown to be improved with much more confidence for well-lit conditions, although misclassifications, such as a green apple classified as a sports ball, and missed objects may still occur. The project also shows the potential that low-cost vision systems could have despite the limitations present in it.

Introduction

The increasing demand for real-time object and person recognition systems poses significant challenges, especially when implemented on low-cost hardware such as the Raspberry Pi 4. These challenges include optimizing performance under varying lighting conditions, ensuring accurate classifications, and mitigating hardware limitations such as processing speed and memory. To address these issues, key considerations must be given to the configuration of the Raspberry Pi 4 and the integration of the camera module.

The Raspberry Pi 4 boasts a quad-core ARM Cortex-A72 processor and 4GB RAM, making it useful for deploying light computer vision models. Additionally, it has a compatible board camera module utilizing the Sony IMX219 8-megapixel sensor and works perfectly for capturing and processing

video. Proper configuration of the hardware, including framing rates and resolution, was critical in optimizing real-time object detection and tracking.

Guidance from class materials, such as Lab 3 (Denny, Lab assignment 4, 2024), emphasized the importance of understanding camera specifications and leveraging APIs like OpenCV and PiCamera to fine-tune detection performance.

Figure 1: Setup of My Raspberry Pi below shows the experimental setup, which includes the Raspberry Pi 4, the camera module, and a student card for scale and identity verification. This configuration was designed to simulate real-world scenarios while assessing the system's performance under varying environmental conditions. Citations for the Raspberry Pi 4 datasheet and the Sony IMX219 sensor can be found in (Cooperation, n.d.) and (Corporation, 2024)

Figure 1: Setup of My Raspberry Pi

Methodology

The steps in the following section are a systematic process to design, implement, and validate the computer vision system for the project. It shows adherence to software engineering practices, utilizes class material, and identifies countermeasures against problems that might arise.

Initial Preparations

The first step was consultations with the computer vision lab coordinators to better understand the project requirements and expected outcomes. Discussions with Denis and Daniel were made to clarify the concepts of real-time identification, confidence level tracking, and the importance of handling varying lighting conditions. Consultations also helped in the validation of the existing system setup and bridged gaps in knowledge or approach.

The camera module was disconnected and then reconnected to ensure hardware reliability. This was a precautionary measure to eliminate any hardware-related problems so that any future debugging could strictly be done at the software level. This reduced the risk of interruptions during the execution of the project.

After these preparatory steps, the working area was moved to the library. The environment was selected due to its flexibility in terms of lighting conditions, which allowed testing and recording in a

controlled but variable environment. This setup allowed capturing video sequences that reflected the varied conditions stipulated in the project description.

Code Logic Development

The design and development of the project code were thoroughly planned and developed following industry-standard software engineering best practices. The approach adopted to implement the system is described in the subsection below:

a) Design Approach

The code was designed with modularity and reusability in mind. Each component of the system-detection, tracking, confidence level plotting, and analysis-is implemented as a separate function or class. This design makes it such that any updates or debugging can be carried out independently without affecting other parts of the system.

b) Libraries and Tools

The project leveraged the following libraries and tools:

- **OpenCV**: For image and video processing, including object detection and bounding box generation.
- **NumPy**: For efficient numerical computations, particularly in confidence level tracking and data processing.
- **Matplotlib**: For plotting confidence levels and generating graphical outputs.

These libraries were selected based on their reliability, extensive documentation, and compatibility with the Raspberry Pi environment.

c) Flow Design

The overall workflow of the system was mapped out in three main stages:

1. **Data Acquisition**: Capturing real-time video using the Raspberry Pi camera module.
2. **Detection and Recognition**: Identifying people and objects in the video frames, assigning labels, and calculating confidence levels.
3. **Analysis and Output**: Plotting confidence levels over time, reporting identification intervals, and generating insights on system performance under different lighting conditions.

d) Algorithm Development

The primary algorithm focused on real-time detection and tracking of people and objects. The steps included:

1. Initializing the camera module and setting appropriate resolution and frame rates for smooth processing.
2. Using a pre-trained facial recognition model to identify individuals in each frame.
3. Assigning confidence scores to detections based on the output of the recognition model.

4. Continuously tracking detected objects to handle entry, exit, and re-entry into the scene.

e) Inspiration from Class Material

Key concepts from the Image Signal Processing (ISP) section of the course material were incorporated. For example:

- Adjustments to camera parameters, such as exposure and white balance, to improve detection quality under varying lighting conditions.
- Insights on sensor limitations and compensatory measures, such as histogram equalization, to enhance image clarity.

Code Flow

The system's code flow is detailed below to provide a clear understanding of its functionality and logic:

1. **Initialization**:
   The code begins by importing the necessary libraries and initializing the camera module. Default camera settings are configured, and an option to adjust them dynamically is implemented.
2. **Video capture**: The camera captures video frames in real-time. Each frame is processed sequentially to detect and recognize people and objects.
3. **Detection and Recognition**:
   - **Face Detection**: A pre-trained model detects faces in each frame. Each detection is assigned a unique ID for tracking purposes.
   - **Object Detection**: Non-face objects are identified using a separate detection model.
4. **Confidence Calculation**: Confidence scores are calculated for each detection. These scores are dynamically updated as the system processes subsequent frames.
5. **Labeling and Bounding Boxes**: Each detected person or object is labeled with their name (or class) and confidence level. Bounding boxes are drawn around detections for visual clarity.
6. **Data Storage**: Detection data, including timestamps, labels, and confidence levels, is stored in a structured format for later analysis.
7. **Real-Time Display**: The system overlays bounding boxes and labels on the video feed, displaying them in real-time on a connected monitor.
8. **Plotting and Analysis**: After 20 seconds of recording, the stored data is analyzed to generate plots of confidence levels over time, identification intervals, and system performance metrics.
9. **Issue Detection and Countermeasures**: The system monitors its own performance, identifying potential issues such as poor lighting or loss of focus. Automatic adjustments, such as increasing brightness or refocusing, are implemented to mitigate these issues.

Code Execution:

Once I was convinced that my code worked appropriately, I executed it and the real time video started. I tried to see how the system performed under two conditions:

1. **Under Dim light:**

   1. **Video Setup (Under Normal Light)**

   For simplicity, the setup for testing was designed with myself, two friends, and one object—a green apple. One friend was to be positioned partially cropped on the left side of the video frame, while the other friend was fully visible but at a distance. I positioned myself in the foreground, fully visible. The green apple was placed within the scene to test object detection. The recording was done in a controlled environment with a blue tube light to maintain the same lighting conditions and to simulate normal lighting conditions, which could serve as a baseline for evaluating system performance.

   2. **Observations from the Video**

   As the video developed, some trends became apparent: Every time I moved closer towards the camera, the score incrementally increased by about 1–2% each step further in my identity identification. However, even then, it wasn't able to detect the whole student card. Other stuff detected the green apple mislabeled as a sport ball-most likely because of its look compared to a green tennis ball. Between the 9th and 16th second, the system detected me without fail, confirming it was a little late, but then it was definitely there. My partially cropped friend was correctly classified throughout the entire duration, which is indicative of system robustness against partial occlusion. However, the farther friend, though fully visible, had intermittent recognition issues.

   3. **Challenges Faced**

   The challenges encountered during this scenario highlight the practical limitations of the system and point toward opportunities for refinement:

   1. **Slow Object Detection:** Even after adjusting the frame rate to 1, which should have been fine for a resource-constrained environment like the one in question, object detection remained very slow. This issue could be related to the processing bottleneck introduced by a limitedly computationally capable Raspberry Pi 4 running a deep learning model for real-time inference. This is usually the case for real-time performance if the sensor can capture frames in sync with the inference cycle, as was referenced in my professor's slides on Image Signal Processing (ISP) (Denny, Course Lecture Slides, 2024). A high latency in model inferences could also be due to a lack of optimization in the neural network model. This limitation could be improved through techniques such as model quantization or leveraging hardware-accelerated inference tools like TensorFlow Lite. Please note, due to slow object detection and due to the video progressing slowly, it was inevitable to involve four people and ask them to move in and out with ease. Nevertheless, even though I was not in a position to have an ideal setup probably due to hardware issues or a suboptimal camera, my focus was solely on understanding, applying, and observing the theoretical concepts I've learnt throughout the module.

2. **Low Confidence in Face Detection:** Despite prior training with satisfactory results during lab sessions, the system detected my face with only a 45% confidence score. This discrepancy suggests potential overfitting or lighting-related inconsistencies during model training. From the ISP slides (Denny, Course Lecture Slides, 2024), it is evident that suboptimal ambient lighting or incorrect ISP configurations (e.g., gain, exposure, or white balance) can degrade image quality, thereby affecting the performance of facial recognition models. The blue tube light might have introduced a color cast and further complicated the feature extraction.

3. **Misclassification of the Green Apple:** This can be classified as a limitation of the training dataset of the model, given that, according to the content of the ISP by (Denny, Course Lecture Slides, 2024), the models of the image classification are greatly driven by diversified and representative training datasets. Most probably, this could have happened with the round, green features of an apple, resembling the tennis ball. This could also be improved by increasing the training examples of green apples or by fine-tuning the pre-trained model on a custom dataset.
4. **Absence of the bounding boxes:** This would further imply software-level errors, either with respect to the integration of an object detection framework or the bounding box drawing routine. According to my professor's slides (Denny, Course Lecture Slides, 2024), this could, among other things, indeed result from improper parameter configurations: input image size and definitions of anchor boxes are the most plausible factors here. Debugging the pipeline around bounding boxes and then looking at the alignment between the results from detection and the rendering code itself could fix this problem.

5. **Graph Visualization and Explanation:** As illustrated in Figure 2: Plot of Confidence under Normal Lighting Conditions the confidence levels of object and person identifications over time during a video analysis. Three entities are tracked: a person identified as "DylanRodrigues," a generic "person," and an object misclassified as a "sports ball." The confidence level for "DylanRodrigues" starts at 45% and gradually increases, reflecting improved recognition as the subject moves closer to the camera. The confidence for the "person" entity fluctuates around 75–85%, indicating consistent, though variable, detection. The "sports ball" identification exhibits a brief and declining confidence level, which highlights the misclassification of the green apple due to its resemblance to a tennis ball. The variations emphasize the system's sensitivity to proximity and potential classification inaccuracies.

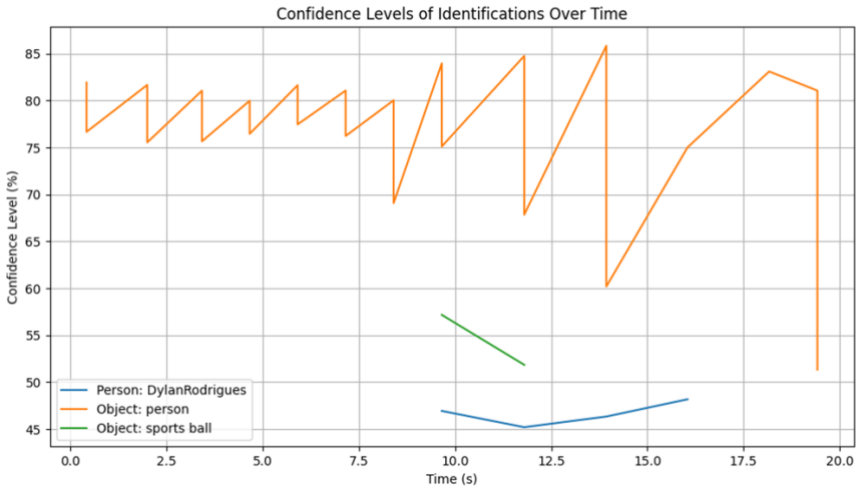

Figure 2: Plot of Confidence under Normal Lighting Conditions

2. **Under a Strong Yellow light:**

1. **Video Setup:** In this experiment, I used a bright yellow light to see if increased brightness or the specific color spectrum could improve the confidence levels of the system's object and person detections. This setup included me standing near the camera, another person directly behind me, and a third person entering the shot. This setup was designed to test how lighting and subject positioning affected detection performance under controlled conditions.

2. **Explanation of Observations:** The confidence of identification for my identification started at 59.38% and jumped up to 71% all of a sudden. This increase is most likely due to the higher brightness afforded by the yellow light, which increases the clarity of facial features and object edges, as it was suggested in the lecture slides. Yellow light, being closer to the peak sensitivity range of most CMOS sensors, could have amplified the system's ability to extract features accurately. But at the same time, it can be that not the spectrum of colors was an issue but brightness itself. The bounding box absence around the individual who is standing right behind me is a case of occlusion when my presence obscured the ability of the system to extract features about the individual. This aligns with the limitations of the system highlighted in the slides, especially on challenges that exist in multi-person detection under partial occlusion. The bounding box detection on the third person could work well since he entered the scene "un-obstructively" into the view and his features could be processed by the system.

3. **Analysis of Confidence Spike and Detection Gaps**

The challenges encountered during this scenario highlight the practical limitations of the system and point toward opportunities for refinement:

1. **Sudden Confidence Spike**: The spike from 59.38% to 71% could be due to improved lighting conditions that enhanced the system's ability to identify features like facial landmarks and contours. Another contributing factor might be a reduction in noise or shadows introduced by the uniform brightness of the yellow light, which aligns with the ISP principles discussed in the slides (Denny, 2024) (e.g., light uniformity enhancing edge detection). However, the sudden jump could also suggest sensitivity in the model's confidence calculations when certain features align optimally in the frame.

2. **Occlusion and Bounding Box Gaps**: As indicated in the object segmentation and tracking slides (Denny, Course Lecture Slides, 2024), the absence of a bounding box on the person standing behind me is a well-known limitation in computer vision algorithms dealing with occlusions. This system likely optimized for my subject being in the foreground rather than those partially occluded. The green apple being classified incorrectly might have kept the error since there is a weakness in object classification models when the target dataset has objects that appear visually similar to the dominant features, like a tennis ball.
3. **Countermeasures and Limitations**: The green apple being classified as a sports ball reflects a limitation in the model's training data. As highlighted in the ISP content (Denny, 2024), image classification models rely heavily on diverse and representative training datasets. The visual similarity between the apple and a tennis ball—both round and green—likely caused this confusion. This issue could be addressed by augmenting the training data with more examples of green apples or fine-tuning the pre-trained model with a custom dataset.

4. **Graph Visualization and Explanation:**

Figure 3: Plot for Confidence Level under Bright Yellow Light Conditions showcases the confidence levels of identifications over time during a video analyzed under bright yellow light conditions. Three entities were tracked: "DylanRodrigues," identified as a person (blue line), another generic "person" (orange line), and an object, misclassified as a "teddy bear" (green line).

The confidence for DylanRodrigues remained steady around 45%, indicating limited improvement despite favorable lighting. In contrast, the confidence for the generic "person" displayed significant oscillations, peaking consistently near 80%. The system detected the third entity, "teddy bear," briefly with low confidence. These results suggest that while bright yellow light helped highlight specific objects and reduce noise, limitations in the model's training set and algorithm persisted, leading to fluctuating performance and occasional misclassifications.

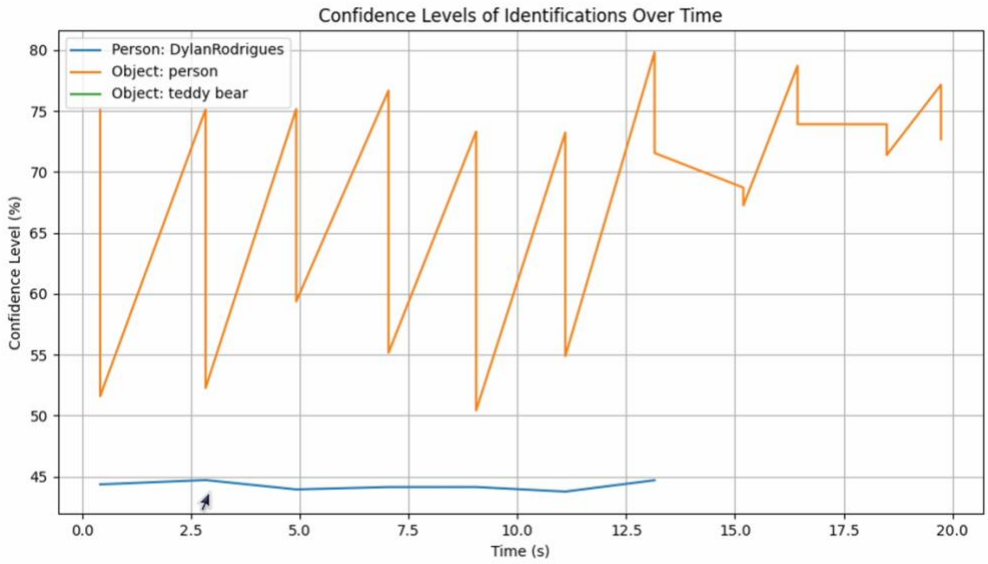

Figure 3: Plot for Confidence Level under Bright Yellow Light Conditions

Results and Discussion

The system performed well in object and person recognition under normal and bright yellow lighting conditions, showing good performance of identifying a person, such as my student card, and objects like green apples and sports balls, under controlled conditions. It worked well if the person or object was within the field of view of the camera and clearly visible, with optimal lighting improving the accuracy. However, under poor lighting conditions, this system performed less well in recognizing objects and faces; it also false-identified objects, like the green apple being detected as a sports ball. Performance varied with factors such as lighting, distance, and occlusion. For example, partial occlusions of faces led to reduced accuracies. While it could track persons with increased confidence the closer they came to the camera, the system failed to identify an object or face in certain configurations, mostly when occluded. This brings up some ethical points concerning privacy in facial recognition and consent for information, along with secure data management. Overall, the system is powerful in controlled settings but would require further refinement for broader use, particularly in real-world, diverse environments with variable lighting and object types.

References

Patrick Denny. (2024, February 26). *AWS Cloud Security - Tuesday Week 4 Material*. Retrieved from CS5024 - Theory and Practice of Advanced AI Ecosystems: https://learn.ul.ie/d2l/le/lessons/17937/topics/634587

Denny, P. (2024, 05 09). *Course Lecture Slides*. Retrieved from https://learn.ul.ie/content/enforced/45572-CE6023_SEM1_2024_5/Lectures/Wk4%20Release/Wk4_ImageSensors2_withlinks.pdf

Cooperation, R. P. (n.d.). *Raspberry Pi Datasheets*. Retrieved from https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf

Corporation, S. S. (2024, 09 01). *Sony Camera Documentation*. Retrieved from https://www.opensourceinstruments.com/Electronics/Data/IMX219PQ.pdf

Denny, P. (2024, 09 10). *Lab assignment 4*. Retrieved from https://learn.ul.ie/content/enforced/45572-CE6023_SEM1_2024_5/Labs/Lab3_FacialRecognition/LabEx3_CE6023_FacialRecognition.pdf