ECE-3332-302

Project Lab at Texas Tech University

Khisa-Lee Lebrun, Hope Regaldo, Boyd Paxton & Nicholas Woodward.

Group 5

R1180898

Mark Haustein, Ph.D., P.E., IEEE

Texas Tech University

Electrical and Computer Engineering Department

December 2024

Outline

Hardware

- Raspberry Pi 5 (Main controller)
- LCD screen (GUI display)
- M.2 HAILO AI hat (To run the artificial intelligence) Raspberry Pi Camera Module V2-8 Megapixel (To get a live view)
- LM 2596 buck Converter (Lower the battery voltage) PCA9685 Servo Driver (to control the servo pointer) SG90 servos (Pointer control)
- Adafruit 105 Laser Pointer (Pointer itself)
- Mouse & keyboard (To program the PI 5)
- Monitor (To program the PI 5)
- Battery 12V, 7AH (Powering the whole system)

Materials

- 3D printing material (Casing)
- Solderable Board, Wires, solder & resistors (To build the powering circuit)

Software

- Roboflow (Database creation)
- Yolov9 (AI training)
- Fusion 360 (3D modeling for the casing)
- Raspberry Pi Connect (Remote Access to the Pi)
- Google slides (presentation of the project)
- Microsoft Excel (Sharing and keeping track of information)
- Linux OS (Terminal used to activate programs)
- Raspian OS (Terminal used to activate programs)

Python Libraries

- Adafruit_servokit (servo control)
- Pytorch (Ai environment)
- Open CV (Picture treatment)
- Tkinter (Graphical User Interface building)

Abstract

` This paper describes the design and implementation of all the progress made towards the implementation of an AI in the task of playing the popular American game "I-Spy with my little eye". All the elements necessary for the result of the project will be discussed, including the software and hardware aspects. Additionally, the requirements of this project will be decomposed and analyzed to ensure the demands are met. This document's purpose will mainly be found in the revelation of the definitive results that came out of this project accompanied by explications of the approach taken.

Acknowledgment

The results obtained these past seven weeks and displayed in this report would not have been possible without the help and guidance of friends, classmates, and roommates, who provided help, mentorship, moral support, and advice. Regarding classmates, an extra thank you should be given to team partners: Boyd Paxton, Nicholas Woodward, and Hope Regaldo. Additionally, gratitude is granted to all classmates who participated in the training of the AI through pictures of themselves or their belongings. Also, a special recognition is extended to all ECE professors and staff who took the time to point the project in the right direction and lend out advice while issues were faced during the making of this project. It is crucial to remark that without all this assistance the project would not be where it is currently.

Table of Contents

List of Figures

1. Introduction

This document describes the thought process, conception, and realization of artificial intelligence built to play the well-renowned game “I-Spy with my little eye” to meet the directions of the Microcontroller Laboratory course at Texas Tech University. In this modernized version of the game, the user will be able on a touch screen to access the graphical user interface (GUI) where the user is prompted to choose the game mode. In the color mode, the user picks a color he wants the little eye to see then the AI will display on the screen its guess of what the user “spied” while the laser pointer will point out the object in the room. Then in the letter mode, the little eye detects an object that starts with the letter selected by the user, and the screen and the pointer will display its guess the same way it did for the color mode. To do so, the AI will detect with extreme precision objects, and their colors, no matter if they are static or moving, all that while ensuring the project respects certain security criteria and while being durable. To attain this goal, using a Raspberry Pi 5, its M.2 AI module with the Hailo 8 processor, and the Roboflow and YOLO programs is mandatory to detect objects. These components and a camera allow the game to interact live with its environment, with the use of a laser pointer. To make the experience more pleasant for the user, a graphical user interface (GUI) was built.

2. Body of Technical Report

2.1 – Powering circuit

For this project, a Raspberry Pi 5, as seen in Figure 1 accompanied by its AI-hat was the best way to make artificial intelligence (AI), Figure 2, be able to play a game of "I-Spy with my little eye". The Raspberry Pi 5 is rated at 5V, 5A. The AI hat and the camera are then connected to the Pi 5 to get power.

Figure 1: Raspberry Pi 5 [1].

Figure 2: Hailo AI hat [2].

The battery is rated at 12V, 7AH making its power too high for the Raspberry Pi 5, but ensuring over an hour of usage. To ensure the safety of all components and to reach the 5V needed for the Pi 5 and Servo hat the use of a buck converter is needed to ensure the wellbeing of all components as you can see in Figure 3 below.

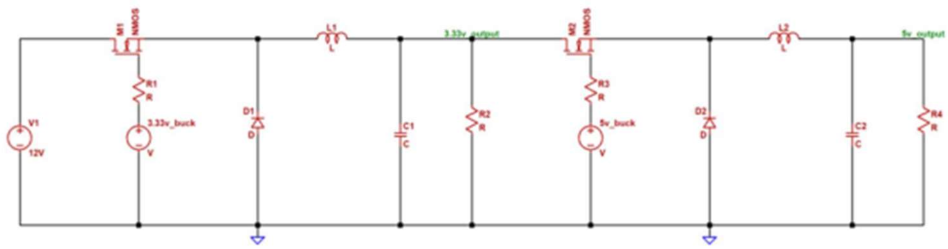

Figure 3: Powering circuit [3].

A USB-C connector is connected from the 5v output to be connected to the Pi 5

2.2 – Laser Pointer

2.2.a – Casing

To ensure the good flow of the game, the camera will have a wide view of a classroom but to be able to know what object the AI is thinking about, a laser pointer is implemented. The laser pointer is included in a large casing that holds all the components so that the project fits on one big plane to make it neater. In Figure 4 below you can see an overall connection of all the components.

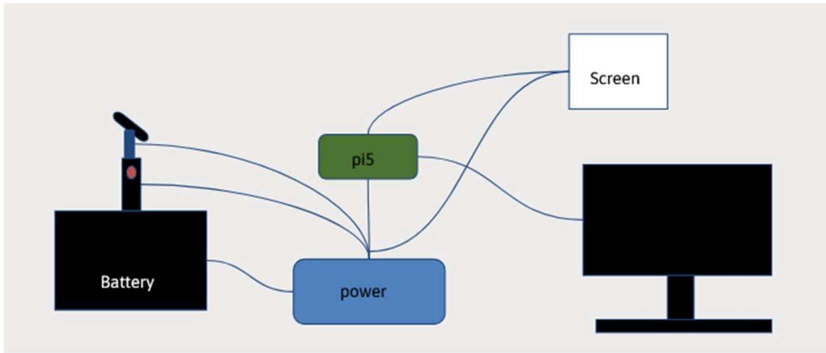

Figure 4: Schematic of the connection of all the components.

Where the pointer and the camera were stabilized on a battery to give it some height to enhance the camera's view. They were both held by a 3D printed mount as can be seen in Figure 5 below where the pointer is right above the camera.

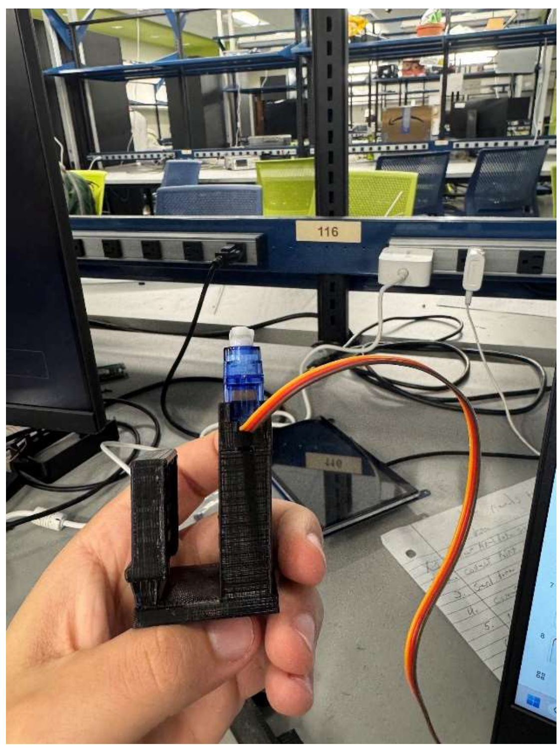

Figure 5: 3D printed casing for the camera and pointer system [4]

2.2.b – Pointer

The pointer itself is composed of two servos, one for the X-axis and the other for the Y-axis. So that it points with the coordinates the PI 5 gives it for the object the AI thinks the user says with his little eye. The laser used is the Adafruit 105[5], rated 2.8-5V at 25mA, controlled by 2SG90 servos [6] rated at 5V. The servos will only be turned on when needed by the Pi 5 to save the battery. The pointer will also be a way to look at the pixels around the one it's pointing at to determine the color of the object. The servos are controlled by a PCA9685 Servo Driver [7]. That servos driver uses the Adafruit_servokit python library to control the servos by assigning them angles and controlling the x and y-axis. From a pixel location given by the Pi 5, the servo angle is calculated by dividing the x or y coordinates by the width or height of the picture and then multiplying it by the angular range of the servo. This returns the angle the servo needs to take to be proportional to the picture. To ensure the pointing destination of the laser is accurate, a dataset had to be built where the camera and the pointer worked hand in hand. In a dark room the pointer would point at a wide range of angles while the camera would detect where the laser landed in the room, associating the pixel location and the angles of the servo. A few lines of code to control the pointer as described can be found in Appendix A.

2.3 – GUI

The aesthetic aspect of the project is just as important as the technical part of it because the graphical user interface (GUI) is what ensures the right flow of the project. In Figure 6 below you can see how the user will go through a game of AI generated I-Spy.

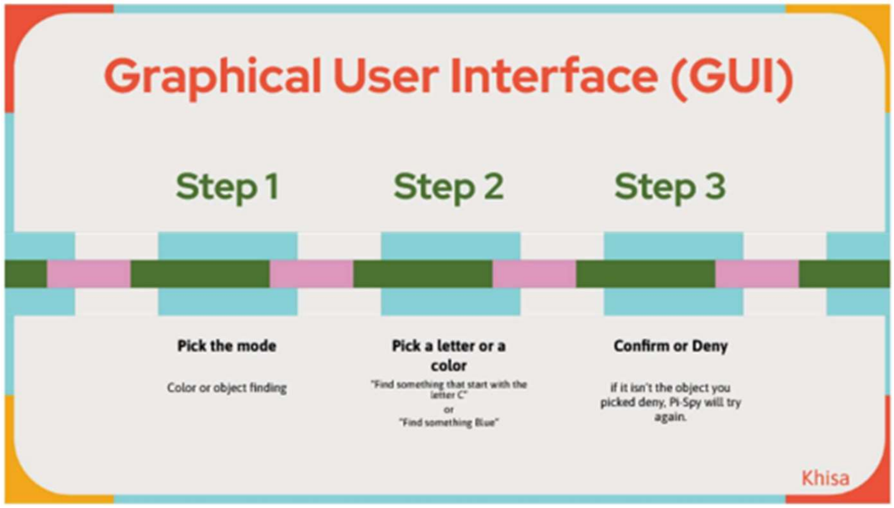

Figure 6: Description of the GUI's flow.

It starts with a page prompting the user to pick the mode, whether he would like to give a color or a letter to the AI as you can see in Figure 7 below.

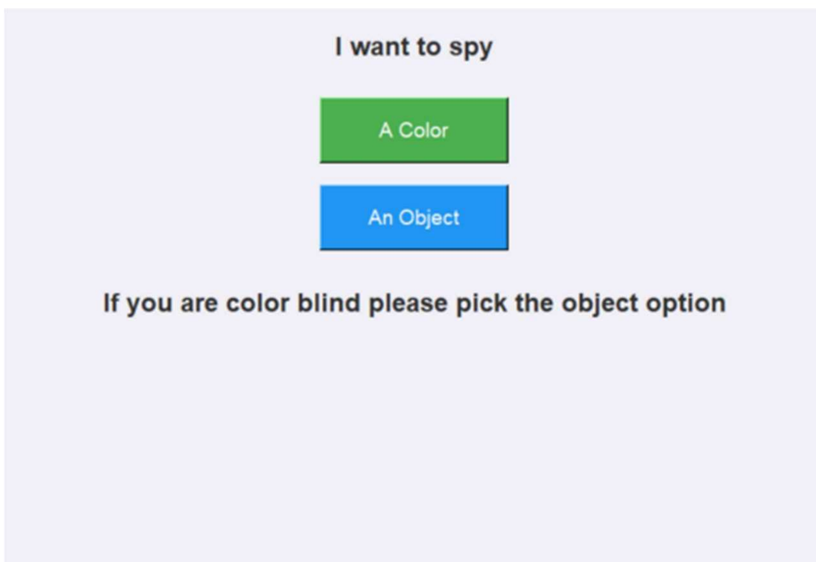

Figure 7: Home page of the GUI.

Then depending on the user's choice, a page asking to pick a color or type a letter will appear. If the first option is picked a lift of colors will appear, asking the user to select one as seen in Figure 8 below. Then a pop-up, as seen below in Figure 9, will appear displaying the camera's view of the object the AI thinks the user saw with his little eye, while the laser pointer will point it out in the room.

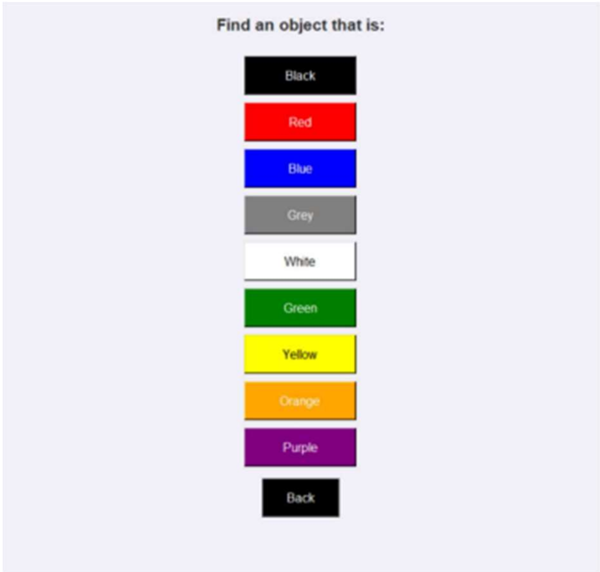

Figure 8: Color choice page of the GUI.

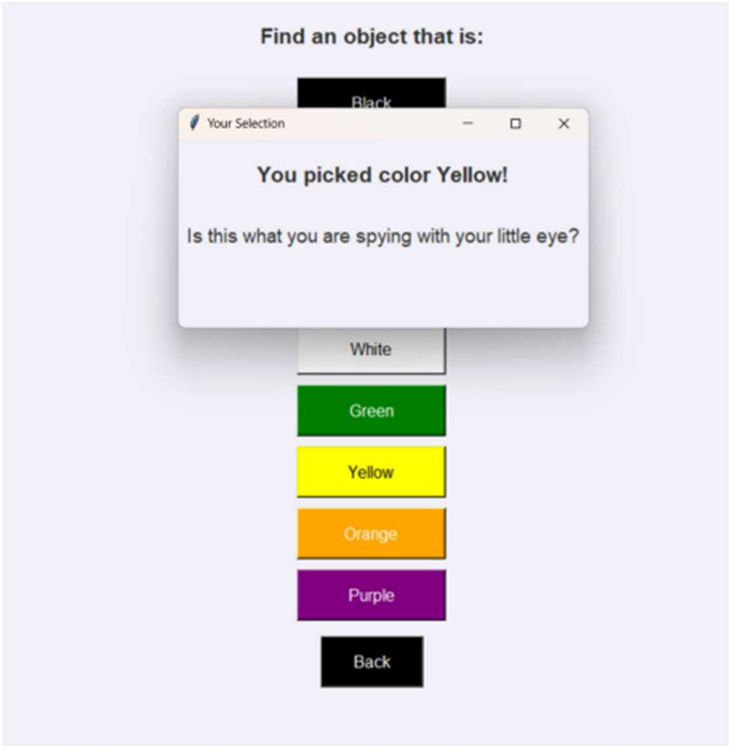

Figure 9: Result of Color Picking.

On the other hand, if the letter option is chosen, a page with a text box will appear. Displaying a live view the same way as for the color once a valid letter is inputted as seen below in Figure 10.

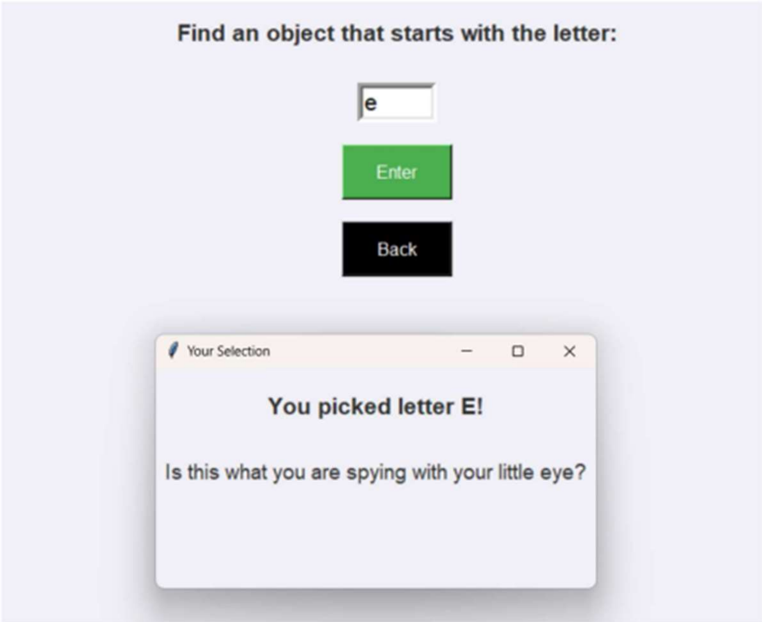

Figure 10: Letter Picking Page with Result.

All of this is displayed on a 4-inch 800*480 resolution display screen as seen on Figure 11 below.

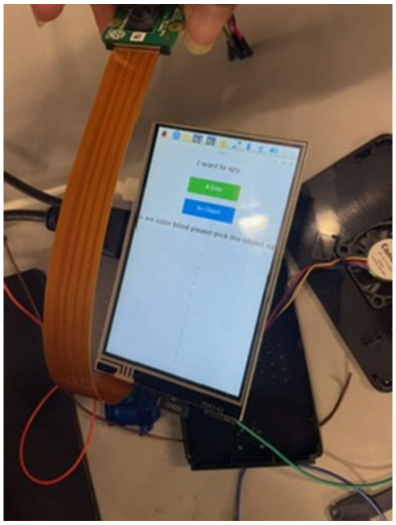

Figure 11: Display of the GUI on the LCD screen.

2.4 – Artificial Intelligence Training

Training with the Hailo AI Hat on a Raspberry Pi 5 aimed to optimize object detection and analysis for the real-time application of the game of I-Spy. In this case, the YOLO (You Only Look Once) model is employed to detect and classify objects efficiently as you can see in Figure 12 below. During inference, the YOLO model generates bounding boxes around detected objects, and critical data is extracted from these boxes. Specifically, the center point of each box is calculated and saved, alongside the corresponding pixel color in RGB. This data is used to control a laser pointer, which is guided to the object's center, while the color data is sent to a color analyzer for further classification or feedback.

```
data = "ClassRoomObj.v4i.yolov8\data.yaml" #data.yaml location
epochs = 300 # number of iterations
imgsz = 669 # number of images
# Load a YOLO model
model = YOLO("yolov8s.pt")
# Train the model
results = model.train(data=data, epochs=epochs, imgsz=imgsz, save=True,patience=0,freeze=10)
```

Figure 12: Training code for the AI model [8].

The YOLO model is trained with a specified dataset for 3000 epochs, which provides the model with ample opportunities to learn patterns and features from the data. Images used in training are resized to a standard size, ensuring that the model processes inputs consistently and avoids complications arising from varying dimensions. To maintain the integrity of the low-level features critical for object detection, the first 10 layers of the model are frozen during training. This step prevents these foundational layers from being overwritten, which is especially important when fine-tuning the model on specific datasets. Additionally, a "patience" parameter is used to control training termination. Setting patience to 0 or infinite ensures that the model does not stop prematurely before reaching optimal performance. Model saving is also enabled, allowing the trained weights to be stored and reused for further development, testing, or deployment, ensuring the process is efficient. By leveraging the Hailo AI Hat, this setup maximizes the potential of the Raspberry Pi 5.

2.5 – Color Detection

To make the AI able to detect a certain color of HSV, hue saturation value, it's a combination of values used to describe each color. It could be represented as a circle with depth, as shown in Figure 13 below.

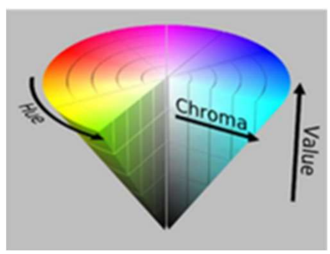

Figure 13: HSV wheel [9].

The pointer points to the center pixel of the bounding box and the RGB value of that pixel will be returned to the code where it will be translated to HSV to determine the color of that center pixel. The range of HSV can be found in Figure 14 below.

```python
def get_color_name(h, s, v):

    if v <= 30 and s >= 125:
        return "Black"
    elif s <= 25 and v > 65:
        return "White"
    elif s <= 36 and 30 <= v <= 65:
        return "Gray"

    if 8 <= h < 40 and 50 <= s <= 135 and 20 <= v <= 85:
        return "Brown"
    elif (0 <= h < 15 or h >= 345) and s >= 50 and v >= 30:
        return "Red"
    elif 15 <= h <= 26 and s >= 50 and v > 60:
        return "Orange"
    elif 26 < h < 70 and s >= 20 and v > 60:
        return "Yellow"
    elif 35 <= h < 160 and s >= 20 and v > 40:
        return "Green"
    elif 170 <= h < 255 and s >= 20 and v > 50:
        return "Blue"
    elif 255 <= h < 290 and s >= 30 and v > 40:
        return "Purple"
    elif 160 <= h < 335 and s >= 30 and v > 60:
        return "Pink"
    else:
        return "Unknown"
```

Figure 14: Code to decide the color name from the HSV value.

For the detection of the objects, a custom-made database was built to fit the objects found in a classroom. However, the premade database that came with the Hailo AI hat was more precise in detection and bounding box delimitations. It also included every object from the homemade database, therefore the premade database was chosen.

3. Engineering Standards, Specifications, and Intellectual Property Considerations

In this project, various standards and specifications were followed to ensure quality, compatibility, and avoid plagiarism. For programming and debugging, tools such as multimeters, function generators, oscilloscopes, and graphical user interface simulations were used. Specifically, the use of Python3.12.6 with the Tkinter3.8.20 library, alongside Roboflow and Yolov9. This ensured compliance with industry standards and compatibility with existing products using the same protocols. The exclusive use of Python minimized the risk of conflicts with other proprietary technologies.

4. Safety, Public Health, and Welfare Considerations

Safety, health, and well-being are key considerations of the project. Great importance is attached to observing safety rules in the laboratory, particularly the wear of appropriate personal protective equipment (PPE) such as goggles and gloves. Additionally, protective, and appropriate outfits are just as important in the laboratory. During testing and while demonstrating our project, special care was taken to make sure nobody was in the range of the pointer and the blinds were shut to avoid eye injury for people in and outside the room. During soldering, extra steps are taken to make sure to work in a well-ventilated environment to avoid inhaling toxic fumes. In addition to the use of appropriate tools and equipment to minimize the risk of injury and burns. After each use, a careful cleanup of work areas is done to ensure the safe and proper handling of waste to prevent environmental contamination. By implementing these rigorous safety procedures, a safe working environment is ensured, while minimizing health risks and preserving general well-being.

5. Global, Cultural, Social, Environmental, and Economic Factor Considerations

In this project for the Microcontroller Laboratory course at Texas Tech University, a strong emphasis is put on sustainability and efficiency. Integrating an overcurrent protection circuit into the design not only protects the Raspberry Pi 5 from potential damage due to excessive currents but also contributes to the project's ecological profile. Reducing the risk of Raspberry Pi 5 and its AI hat and camera aims to extend the material's lifespan, minimizing electronic waste and promoting resource conservation. The use of fuses protects the more expensive and hard-to-obtain parts of the project such as the Raspberry PI 5 and its components at a lower cost. These are important to protect as they total to 185.76 USD, see Appendix B. This mindful approach underlines a strong commitment to responsible engineering practices.

6. Conclusion

In conclusion, this project represents a significant advance not only in the microcontroller field but also in the field of artificial intelligence, demonstrating the ability to design and implement an advanced system to meet the requirements of a game of AI-generated I-Spy. It is the fruit of 15 weeks of planned work, see Appendix C, it proves that the use of a Raspberry Pi 5 with an M.2 AI hat and a Hailo 8 processor creates efficient real-time object detection and interaction, while the use of the Roboflow and YOLO software makes its object recognition precise. Priority was given to the project's sustainability and safety by incorporating overcurrent protection circuits and fuses, making sure the hardware is safe and will last. This contributes to environmental protection by extending the life of the components, therefore it minimizes waste. More were put in place around the use of the laser pointer to maintain a safe laboratory environment. This project is proof of the power that comes out of combining microcontroller technology with artificial intelligence. Artificial intelligence is the future and this project displays it, by creating new opportunities. The successful completion of the project reflects forward-thinking engineering while supporting academic development and technological progress.

7. References

[1] Raspberry Pi 5 https://www.raspberrypi.com/products/raspberry-pi-5/

[2] Raspberry Pi AI-hat https://www.raspberrypi.com/products/m2-hat-plus/

[3] Powering circuit diagram made by Hope Regaldo

[4] 3D print camera and servo holder made by Boyd Paxton

[5] Adafruit 105 Laser Pointer https://www.adafruit.com/product/1054

[6]2SG90 Servos

https://www.adafruit.com/product/2307?gad_source=1&gclid=Cj0KCQjwmt24BhDPARIsAJFYKk2jxzHBvUbG3GZr7vmO1IFbvKrw94K4095N82DBUexRwgTwcJsZKycaAoaeEALw_wcB

[7] PCA9685 Servo Driver

https://www.amazon.com/s?k=pca9685+servo+driver&hvadid=694341352309&hvdev=c&hvexpln=67&hvlocphy=9028540&hvnetw=g&hvocijid=469342962393375019--&hvqmt=e&hvrand=469342962393375019&hvtargid=kwd720117165261&hydadcr=24357_13533826&tag=googhydr20&ref=pd_sl_68h06s2t7j_e_p67

[8] Training code for the AI hailo model written by Nicholas Woodward

[9] HSV representation https://en.wikipedia.org/wiki/HSL_and_HSV

[10] Servo Controlling code written by Boyd Paxton

8. Appendices

Appendix A

```python
contours, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
for contour in contours:
    area = cv2.contourArea(contour)
    # Filter out small areas that might be noise
    if area > .2:

        # Fit a minimum enclosing circle to the contour
        ((x, y), radius) = cv2.minEnclosingCircle(contour)
        circle_area = np.pi * (radius ** 2)
        circularity = area / circle_area

        # Check if the contour is roughly circular
        if circularity > 0.65 and radius > 1:  # Circularity and size thresholds
            cx, cy = int(x), int(y)
            hsv_value = img_hsv[cy,cx]
            print(f"Red laser dot found at position: ({cx}, {cy}) with radius: {radius} and HSV values of {hsv_value}")

            ## Draw the detected laser dot
            cv2.circle(output_hsv, (cx, cy), int(radius), (255, 0, 0), 2)  # Blue circle
            cv2.circle(output_hsv, (cx, cy), 5, (0, 255, 0), -1)  # Green dot at center
            cv2.circle(mask, (cx, cy), int(radius), (255, 0, 0), 2)  # Blue circle
            cv2.circle(mask, (cx, cy), 5, (0, 255, 0), -1)  # Green dot at center
```

Figure 15: Pointer detection code [10].

This code detects a red laser dot from a camera feed by processing contours in a binary mask image. It filters out noise by excluding small shapes and checks the circularity of contours, focusing only on those resembling a circle with a radius above 1. Once a valid dot is identified, its HSV color values are extracted to verify it matches a red laser's profile. The position, radius, and HSV values of the dot are printed for reference. For visualization, the code overlays blue and green circles on the detected dot. This system is likely part of a laser pointer-controlled interface for a Raspberry Pi 5.

Appendix B

		I-Spy Budget				GRP 5		
	Running Total			Total Estimate			Start Date:	08/27/2024
Direct Labor:	rate per hour	hrs	total	rate per hour	hrs	total	Today:	12/03/2024
Hope	18	151	$2 718,00	15	220	$3,300.00		
Nick	18	169	$3 042,00	15	220	$3,300.00	End Date:	11/23/2024
Boyd	18	169	$3 042,00	15	220	$3,300.00		
Khisa	18	161	$2 898,00	15	220	$3,300.00		
DL subtotal:			$11 700,00			$13,200.00		
Material Cost:								
Bought			Price		Budget : $150			
Rasberry Pi 5			$90,76					
AI kit - unit only			$70,00					
Camera			$25,00					
Camera adapter			$1,00					
HDMI adapter			$8,99					
Laser			$6,00					
MC subtotal:			$201,75					
Total			$11 901,75					

Figure 16: Overall Budget.

The budget sheet displays the cost of all the materials used and bought, as well as the number of hours worked by each member for the project as well as the cost of labor.

Appendix C

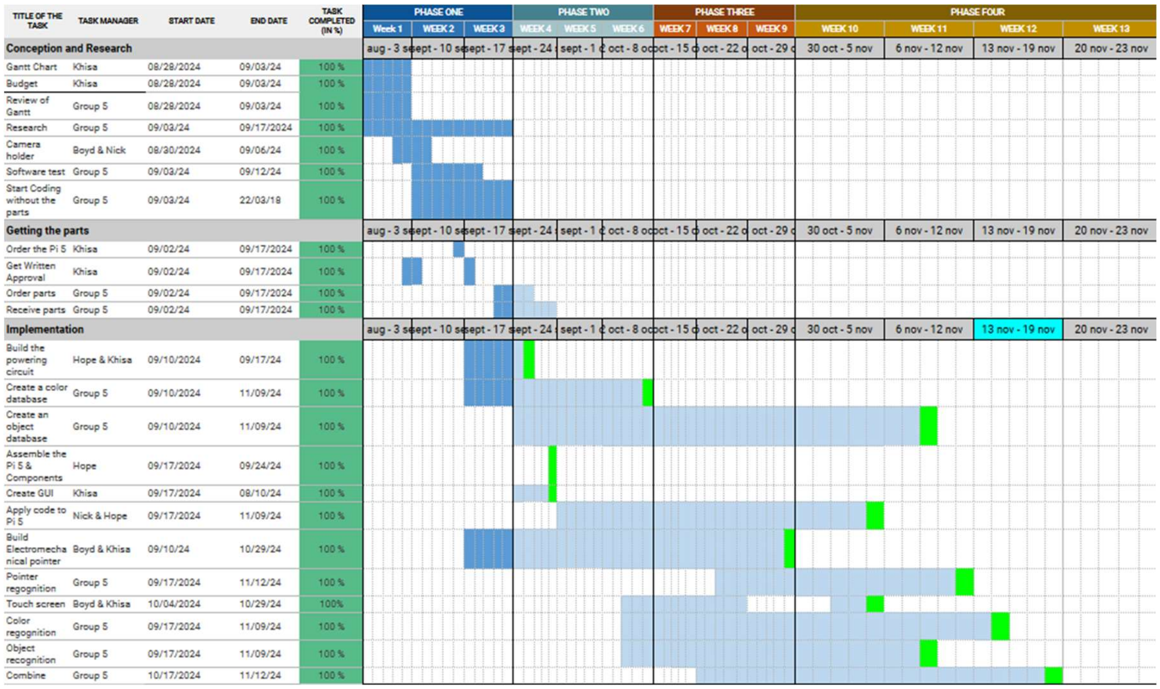

Figure 17: Gantt chart part 1.

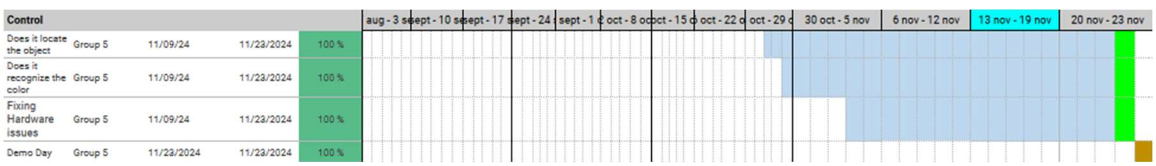

Figure 18: Gantt chart part 2.

The 15-week Gantt chart for the AI I-Spy game project outlines the phases of planning, development, testing, and refinement. The early weeks focus on defining objectives and setting up the foundation. This is followed by an extended period for building core functionalities and integrating key components. Midway through, the emphasis shifts to testing and ensuring everything works as intended. Later weeks involve refining the system, incorporating feedback, and enhancing the user experience. The final stage is dedicated to wrapping up the project, completing documentation, and preparing for deployment.