The University of Hong Kong

ELEC3848 Integrated Design Project

Group Report: Automated Guided Vehicle (AGV)

Group_E12 Members:

LI Ching Yam Joecelyn (3035688485)

Wang Shuyi (3035951741)

Chen Bozhou (3035820760)

Jiang Feiyu (3035770800)

Contents

1 Abstract

In this project, our group built an automated guided vehicle, which moves automatically towards the maximum light intensity position. This design mainly focuses on the light chasing and obstacle avoidance function of the intelligent cart, followed by a preliminary study on the navigation, OpenCV algorithm and IoT of system. During the process of finding and stopping at the position of maximum light intensity, the camara would collect the images of a car and send them to PC for model inference. The car could reach automatically the object through web page remote control and interrupt using computer vision.

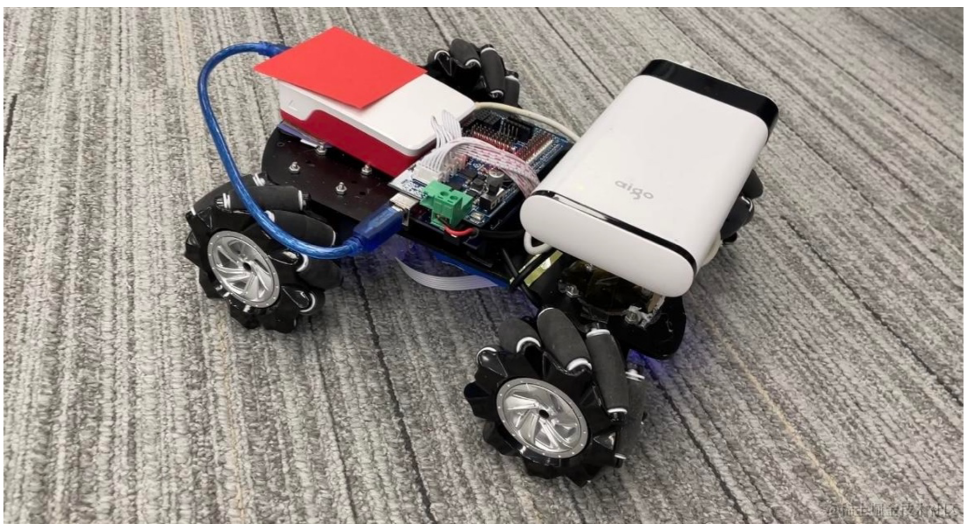

Figure 1. Final product.

2 Required Function

The intelligent trolley is mainly composed of a power supply part, a drive part, a control part, a signal acquisition part and a computer vision analysis part. The power supply part is provided by a battery (12V, 5000mA). The motor drive part uses Arduino Mega2560 and shield board to drive the wheels of the cart to rotate. The control part is provided by Arduino Expansion chip. The signal acquisition part includes light sensor, ultrasonic sensor.

2.1 Hardware

The required function system can be divided into following subsystems:

The intelligent cart is a collection of Arduino Mega, servo and DC motors, and different sensors as one of the intelligent control systems, through a multi-angle light sensor to collect the data of the surrounding sunlight, which is analyzed by the microcontroller and then output to find the direction of the strongest light.

(1) **Light charging system** of the cart, including the data acquisition of the sensors and the acquisition of the light intensity from the solar panel.
(2) **Control system** of the trolley, including the driving part of the trolley and the light chasing.

2.1.1 Robot Body Assembly

Attach the four wheels and SG90 servo with mount set to build the body of the vehicle. The orientation of the wheels should be on the corresponding axises of the motors.

Attach the Arduino expansion board connected with the Arduino board onto the vehicle.

Attach the motor cables between the respective motors' connectors and pinouts of the Arduino expansion board. The left front wheel is connected to M1 motor cable on the Arduino board. The right front wheel is connected to M2. The left rear wheel is connected to M3. The right rear wheel is connected to M4.

Connect the battery and Bluetooth module to the Arduino expansion board using COM3 connectors.

Connect the OLED display to its respective connectors using the IIC connectors.

Connect two photoresistors onto the vehicle body.

Connect the ultrasonic sensors to the vehicle.

Connect the solar panel to the vehicle.

After checking the wiring, OLED and Bluetooth connection, connecting the camera module to the PC.

2.2 Hardware Setup

After the construction of the body and wiring of the car, wiring and signals should be checked. Under this stage, the screen of OLED should be displayed the relative words “AI Robot”. The LED lamp should also be blinked quickly.

Next is to test the movement of the car. Using Bluetooth module to control remotely the movement. The car should be able to go in a direction, checking the light intensity and turning the wheels.

In the required function, two possibilities for turning the direction of wheels are used: upper left and upper right directions.

Figure 2. Wheels of vehicle.

2.3 Experimental Sensors and changes from proposal

During our preparation process, different types of sensors were installed on our experimental vehicle. After comparison, only DC motors and servo were retained for achieving a better performance and results by our car.

2.3.1 Ultrasonic Sensor plus Buzzer

The function of ultrasonic is to detect the distance. The method is to send out the signal by Echo, receiving the signal by Trig; the other pins that should be remembered are Vcc (power supply) and GND (ground). The reflection time would be recorded; thus the distance could be obtained. (Nedelkovski, 2015)

A buzzer produces sound that can serve as warning signals in various applications. There are two pins on a buzzer. One should be connected to the ground and the other should connect to Arduino.

Changes from proposal: Stability is an important component that needs to be considered. For these two sensors, temperature and humidity may affect the sensor's accuracy and stability. Also, apart from distance detection only, our group expected to find a way that not only detects the distance, but also detects the position or color of objects. Thus, the ultrasonic sensor plus buzzer had been eliminated in our final product.

2.3.2 OLED display

When the Arduino board is disconnected from the computer, say on a toy car, the serial monitor is no longer available. It would be convenient to have a small display on the Arduino board to serve as the serial monitor in that case.

Changes from proposal: As the function of OLED screen could be displaced by computer screen, the information detected and calculated would be sent through computer. Our group expected that there may be limited outcome by OLED display.

2.3.3 Photoresistors

Photoresistor is the use of semiconductor photoconductivity effect made of a resistance value with the intensity of incident light and change the resistor, also known as photoconductive detector; photoresistor is generally used for light measurement, light control and photoelectricity conversion (the change of light into electrical changes).

Changes from proposal: The disadvantages of photoresistors could be different levels of individual sensitivity. Some of them may be more sensitive towards the light, some may not, this may cause the result of abnormal turning of experimental car.

2.3.4 Servo and motors

DC motor and servo would be suggested in this project for the function of rotation control. The definition of a servo motor is to control the angular rotation under the status of precise control over motion. By changing the width of pulse width modulation (PWM) signal to reach the goal of controlling the rotation angle of the SG90 servo motor. The red power line of the servo motor connected to the 5V pin; the black or brown ground line connected to the GRD pin; last but not least, yellow signal line connected to a digital pin. (Koumaris)

Construct a DC motor controller with an encoder. To make the experimental car move, DC motor is required to formulate mechanical energy from electrical energy. Wiring of the DC motor would be: Vcc line connected to 5V encoder, ground line connected to ground, M+/M- connected to power supply. A and B connected to the encoder signal lines. During the final experiment, the front part of the car would shift due to the steering gear error. This can be corrected later using pid.

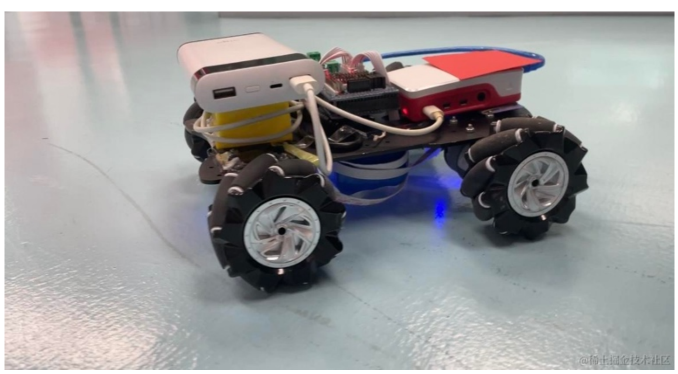

Figure 3. Wiring of the Car.

2.4 Software

The required function of this project is to accomplish an automatic movement goal and find the maximum light intensity. To achieve this, our group designed the logistics of vehicle's wheel movement.

Two photoresistors setting on the vehicle is to detect the light intensity level of surrounding environment of the car. Therefore, the car would determine the turning direction.

```
// Process data from light sensor
int_adc0=analogRead(A0);   // Left sensor at ambient light intensity
int_adc1=analogRead(A1);   // Right sensor at ambient light intensity

// measure the sensors reading at zero light intensity
int_adc0_c=245;   // Left sensor at zero light intensity
int_adc1_c=475;   // Right sensor at zero light intensity
```

Figure 4. Code of Light Intensity Detection and Measurement.

The control of wheel movement is important as the precise turning of the experimental car is crucial for judgement. The decision of whether the wheels would be turning leftwards or rightwards depends on the difference between the left and right sensors in the aspect of light intensity levels. According to our algorithm, the higher level of light intensity detected by the sensor would determine its turning direction. For example, if the data collected by the left sensor is larger than the other side, the reaction of turning at a larger angle of right part would happen, the car consequently turn left.

```
// This function is to detect the direction of the car
void tracker()
{
 // If left sensor is brighter than right sensor, decrease servo angle and turn LEFT
  if (int_left>(int_right+TOL))
  {
      Serial.println("left1t");
      LEFT_1();
  }

 // if right sensor is brighter than left sensor, increase servo angle and turn RIGHT
  if (int_left<(int_right-TOL))
  {
      Serial.println("right1");
      RIGHT_1();
  }

  // if right sensor is nearly the same, stop the car
  if ((int_left>(int_right-TOL))&& ((int_left<(int_right+TOL))))
  {
      Serial.println("stop");
      STOP();
  }
}
```

Figure 5. Code of turning directions of the car.

Solar panel was attached to on the vehicle to determine the direction of car. If the light intensity level detected by the solar panel is increasing, the vehicle is moving towards the desired location, which is moving towards the maximum light intensity point, the direction of the vehicle is to move ahead in the same direction. This is the case that the car is moving towards the light source.

If the light intensity level detected by the solar panel decreases, the car is moving further from the light source. The judgment should be made to move in the opposite direction of the current movement. This is the case that the car moving backwards the light source.

The distance control of our project is to use the function of calculation. Firstly, define the forward and backward movement. The power sensors would collect the data from solar panel setting on the experimental car, the data of light intensity value in the environment would be generated. Thus, the moving direction would be determined. In the next stage, we would also collect the current value and voltage values. The value of power could be calculated from these two data and displayed on the OLED screen, which means the light intensity power detected by the solar panel. If the value of power increased, the car should be moving forward to close to the point of maximum light intensity. The forward function is executed simultaneously with the steering function.

3 Innovative Function

In the innovative function, camera has been connected to PC, combining with fronted web setting up http connection with Raspberry Pi, constructing physical serial connection with Arduino Mega2560 board. Through data transmission, the route navigation of the robot based on the webcam is realized, so that it can reach the intended place, and remote manual start and emergency intervention are realized.

The system consists of following:

Hardware -

(1) **Camera**: Using OpenCV to collect data and send the frames to the PC for model inference.
(2) **PC**: Analyzing data and sending the coordinates of the car and suspicious items to the Raspberry Pi for further analysis.

Software -

(1) **Frontend Web Page**: This serves as the remote control for workers to remotely activate and stop the car (Arduino).
(2) **Image preprocessing programme**: Images would be preprocessing in the programme for further analysis.
(3) **Raspberry Pi calculation programme**: It continuously listens to the client for commands to wake up or stop. Once awakened, it receives data from the PC to calculate the car's route and sends instructions to Arduino through the physical serial port.
(4) **Arduino movement programme**: Receives data from the Raspberry Pi and moves according to the instructions.

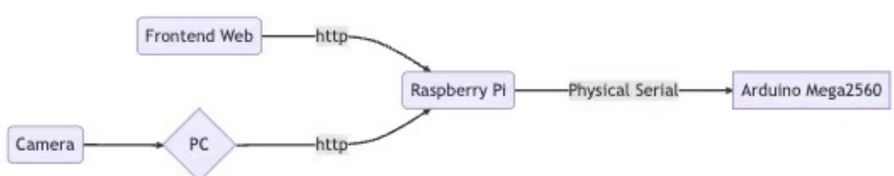

Figure 6. Data flow of project.

3.1 Hardware

3.1.1 Camera and Object Recognition

The equipment our group adopted is a camera connected to the Raspberry Pi board, constituting a tracking system. The tracking system is to monitor and track the transport movement of the experimental car in an indoor environment.

There are three advantages of our innovative design:

(1) Low-cost. Compared with those high-end surveillance cameras, this setting was cheaper and easier to build.
(2) Convenient. The camera would be set on a fixed point in the indoor environment, it could be held in hand or fixed on a stand. As the system's function is to record the tracking paths of the experimental car, the field view of webcam includes the whole setting of experiment – the path and the car.
(3) Tracking movement and object detection function.

Webcam was used for our innovative function – tracking system.

After researching multiple possibilities, we decided to use the Computer Vision - Open-Source Computer Vision (OpenCV) library. It features a wide-ranging selection of the classic and updated versions of computer vision and machine learning algorithms. The OpenCV library is available in various programming languages. Python has been selected for our project. (OpenCV, 2024)

The definition of object detection is to identify and localize the object by inputting a video or image, which belongs to the aspect of computer vision. Image Localization is the process of identifying the correct location of one or multiple objects using bounding boxes, which correspond to rectangular shapes around the objects.

Next, the method of object detection had been found and chosen, which is color detection. Red labels were attached to the front part of our car. This function takes a frame (image) as input and performs object detection using color thresholds. def detect_object(frame):

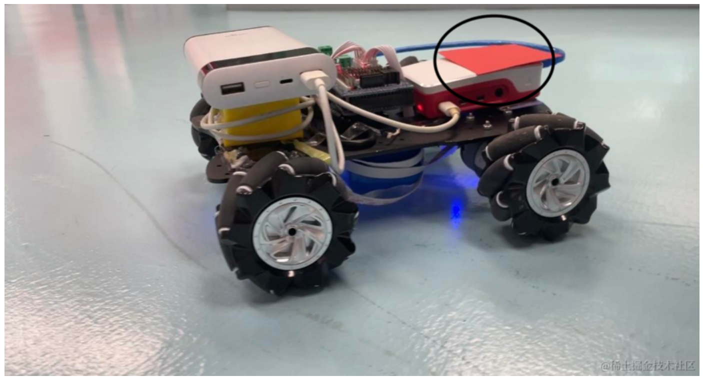

Figure 7. Photo of Colored Labels for Recognition.

Using OpenCV algorithm, define the method of object detection and distinguish the color, which is the yellow object and red car body. The range of red color would be converted from BGR to HSV format, setting to be (200, 255). After detection, using contour to filter the area of red color only.

Find the largest contours in each mask by calculating the maximum area.

A green bounding box would be formulated around the largest area of color points. Centre coordinates in the green frame would be calculated to be the position of the car.

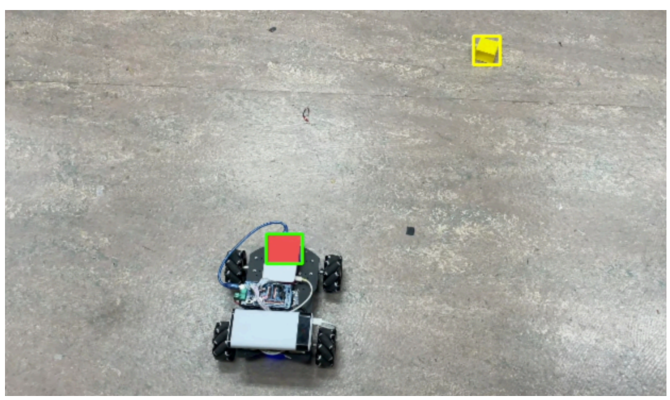

Figure 8. Find the object.

```
# Draw rectangle around the largest red object
if largest_red_contour is not None:
    x, y, w, h = cv2.boundingRect(largest_red_contour)
    cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

    # Calculate the center of the rectangle
    red_center_x = x + w // 2
    red_center_y = frame.shape[0] - (y + h // 2)  # Invert y coc

    print("Red coordinates:", red_center_x, red_center_y)
else:
    red_center_x, red_center_y = 0, 0

# Draw rectangle around the largest yellow object
if largest_yellow_contour is not None:
    x, y, w, h = cv2.boundingRect(largest_yellow_contour)
    cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 255),

    # Calculate the center of the rectangle
    yellow_center_x = x + w // 2
    yellow_center_y = frame.shape[0] - (y + h // 2)  # Invert y

    print("Yellow coordinates:", yellow_center_x, yellow_center_
else:
    yellow_center_x, yellow_center_y = 0, 0
```

Figure 9. Code of Frame Formation and Centre Coordinate.

When the car was moving, the centre coordinates would be detected and recorded in (x-coordinate, y-coordinate). The navigation part would be calculated from the difference of coordinates, showing the changing movement of the car.

Send the coordinate through http request.

```
response_post = requests.post(server_address, data={'message': message})
```

Figure 10. Send coordinates.

Check if the request was successful (status code 200).

```
if response_post.status_code == 200:
    print("Server Response:")
    print(response_post.text)
else:
    print(f"Error: Server returned status code {response_post.status_code}")
```

Figure 11. Check the request.

Optimizer

Detection_interval setting:

```
detection_interval = n  # Run detection every 10 frames

 frame_count = 0

 while cap.isOpened():
     ret, frame = cap.read()

     frame_count += 1

     if frame_count % detection_interval == 0:
         center_x, center_y, yellow_x, yellow_y = detect_object(frame)
```

Figure 12. Send coordinates.

Setting the image pixel:

```
frame = cv2.resize(frame, (854, 480))
```

Figure 13. Send coordinates.

3.2 Software

3.2.1 Communication

Differences between Bluetooth and Wi-fi

To choose between Bluetooth module or Wi-Fi module, our final decision is to use Wi-Fi module after trying both. The decision is to make sure that performance of transmission would reach the best results.

Bluetooth equipment is a wireless protocol technology to connect laptop and phone. The equipment we used includes Bluetooth module and its relevant software. During the process of setup, pair-up between different devices is required for remote control and communication. (Bluetooth, 2023)

Start by connecting the Bluetooth to Arduino. The TX of the Bluetooth connects to the RX of the Arduino and the RX connects to the TX.

Next connect the Bluetooth to the Raspberry Pi and set up the Bluetooth serial port. Scan to find the Bluetooth device and pair it successfully. Raspberry Pi installed Python, install GPIO module for python. Install serial for serial communication and USB communication.

Then connect raspberry pi with Arduino. Connect raspberry pi with Arduino through usb cable. Raspberry Pi and Arduino communicate through GPIO pins.

Connection method:

RX of Raspberry Pi ----- TX of Arduino

TX of Raspberry Pi ----- RX of Arduino

GND of Raspberry Pi ----- GND of Arduino

Thus, Bluetooth connectivity can be controlled and linked with nearby Bluetooth devices.

The role of Bluetooth is to complete the communication between the two parties so that they can transmit messages.

Wi-fi is a wireless technology used to set up connections between computers, laptops and phones etc. In our daily lives, we may use a web browser to search for information, and the network protocol used is the HTTP protocol. In this process, two computers are included: one computer acts as a server, and the server will reply to the requested information to the client after receiving the client request; the other acts as a client, such as a browser or another computer. It is to send a network request to the server as a TCP client. Run an HTTP server program on the server side, and then run an HTTP client program on the client side to communicate with the server.

The essential difference between these two: Bluetooth could only be used in a limited distance, for instance, in a room; Wi-Fi could be used in remote locations, for example, a place where it has connections. Also, Bluetooth is to exchange data between devices with a lower transmission speed while Wi-fi is to connect between devices and Internet with the higher speed.

3.2.2 Serial communication

Serial connection is the communication between Raspberry Pi and PC.

Raspberry Side

(1) Check the usb devices connected to Raspberry Pi: “lsusb” to search for connection information and check whether the port exists.

```
def find_arduino_port():
    ports = serial.tools.list_ports.grep("1a86:7523")
    for port, desc, hwid in ports:
        return port
    return None
```

Figure 14. Arduino port connection.

(2) If the port does exist, messages could be sent. A sample code is shown:

```
try:
    while 1:
        res=ser.readline() #read in data and print
        print(res) #send the data
        time.sleep(1)
        ser.write("Hello! I am Raspberry!".encode("utf-8"))
except:
    ser.close()
```

Figure 15. Send the data.

(3) Arduino board sends message to Raspberry Pi.

```
void waitForPythonMessage() {
  String str = "";
  while (Serial.available()) {
    // sendResponseToPython("aaa");
    char ch = Serial.read();
    str += ch;
    delay(10);
  }
  if (str.length() > 0) {
      // command that received
      processCommand(str);
      // send response to Python
      sendResponseToPython("Message received!");
    }
 }
```

Figure 16. Arduino sends message to Raspberry Pi.

3.2.3 HTTP Communication in the Same WLAN

Both computers are on the same network and that they can reach each other.

Sample HTTP Server

```python
from http.server import SimpleHTTPRequestHandler
from socketserver import TCPServer

# Specify server address and port
host = '0.0.0.0'  # Can be a specific IP address, or 0.0.0.0 to accept any available network interface
port = 8888

# Set up server
server_address = (host, port)
httpd = TCPServer(server_address, SimpleHTTPRequestHandler)

# Print server information
print(f"Serving on {host}:{port}")

# Start the server
httpd.serve_forever()
```

Figure 17. Sample HTTP server.

HTTP Client

```python
import requests

# Server address and port
server_address = 'http://192.168.1.20:8888'

# Send HTTP GET request
response = requests.get(server_address)

# Print server response
print("Server Response:")
print(response.text)
```

Figure 18. HTTP client.

<u>Server Side: Flask</u>

As a flexible and lightweight microwave framework designed based on Python, “Flask” was adopted. All the requests from client should be “POST” request, the “process_client_data” function will read and send the specific message to Arduino.

```python
def process_client_data():
    received_data = request.form.get('message')
    print("Received from client:", received_data)
    # hand the received_data through content
    response_to_client = "Message received by server"
    return response_to_client

# Define routes and handle GET and POST requests
@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'GET':
        # Handle client's GET request
        return "Server is ready to receive messages"
    elif request.method == 'POST':
        # Process the client's POST request and call the encapsulated function
        return process_client_data()
```

Figure 19. Flask function.

<u>Frontend: Remote control</u>

The request sent from the front-end web page is through Javascript:

```javascript
fetch(serverAddress, {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/x-www-form-urlencoded',
                },
                body: 'message=' + message,
            })
                .then(response => response.text())
                .then(data => {
                    console.log('Server Response:', data);
                })
                .catch(error => {
                    console.error('Error:', error);
                });
```

Figure 20. Front-end web page.

The request shall be sent as the button is triggered.

4 Performance measurement

Required function: The testing was set with the 50-cm half cycle, the car should be able to detect the maximum light intensity position, driving and turning automatically in the correct direction. Finally, it would stop at the point. The time and distance were recorded.

Innovative design: The performance of our experimental car was satisfactory. The car was to be placed on the floor. The lamp (which was marked as the point of destination) was to be placed in its front direction, 50-cm away on the ceiling of the indoor room. Using camera and OpenCV algorithm to collect and send the data to PC. PC will then analyze the coordinates calculated of the experimental vehicle and suspicious items on the ground to the Raspberry Pi. People remotely activate and control the car (Arduino). The Arduino Mega2560 board on the car plays the function of receiving data from the Raspberry Pi and moving in the desired direction.

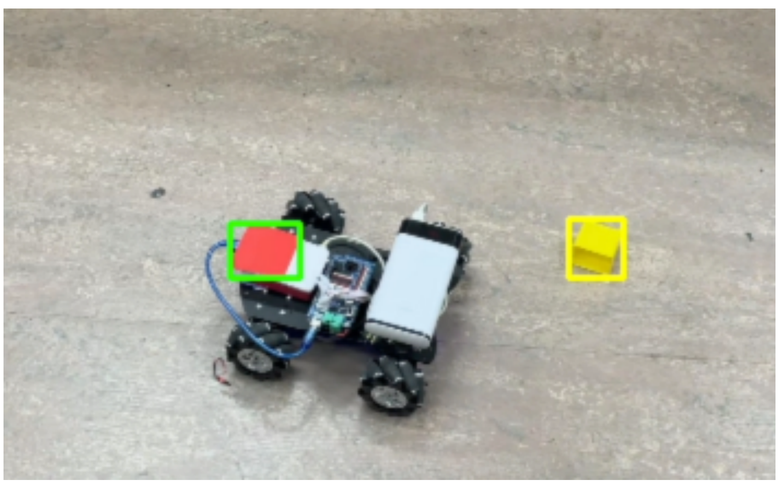

Figure 21. Performance.

5 Reflection

With the development of technology, light detection combining with object detection becomes reality. Our project designed an experimental car combining solar technology with computer vision algorithm, detecting the maximum light intensity and car movement, locating the position of the experimental car, gaining the result of data. The project comprehensively included various latest technologies and applications.

The objective of our project is to analyze the function of solar energy and locate the position of light and pathway of car movement. One example is to use in the industry factory. Our vehicle could be used in the indoor factory environment, workers could remotely control the vehicle in the central control office. Using the computer, the workers could start the programme of it, the vehicle could be able to detect or collect suspicious objects on the ground, their positions could be determined by the camera on the ceiling of the factory.

In this project, our group firstly built up the car chassis using various sensors, servo and motors to construct the hardware part of our car; in the next stage, our group designed the software part to solve the problems of wheels turning and moving directions. In the last stage, Bluetooth module and computer vision algorithm were added to build the function of tracking. We use Raspberry Pi as the main controller and at the same time connect web pages, cameras (mobile phones), and computers to achieve a complete ecosystem of the Internet of Everything.

We successfully achieved our goal despite encountering challenges. In the hardware aspect, issues with non-functional photoresistors, improper wheel speed, and undetected cameras were addressed by replacing hardware components and devising a new Arduino algorithm for wheel control. On the software aspect, connectivity problems and unstable accuracy in the detection algorithm were resolved by opting for color-based detection. We selected a suitable algorithm, conducted training and testing, ultimately reaching our milestones, and gaining a deeper understanding of vehicle and Arduino functionality.

Limitations: The difference in motor rotation causes route errors and lacks monitoring and correction. The image recognition camera is stuck. Currently, methods of compressing pictures, extracting frames, and hardware optimization are used, but the implementation effect is still not ideal. The navigation algorithm is placed at the front of the car. When deviations occur, there are still deficiencies.

Future development: We hope that the deficiencies of the system could be improved, making the vehicle more feasible, convenient and user-friendly as a wish that it could be used in real situations to help worker for control.

Reference

Nedelkovski, D. (2015). Ultrasonic Sensor HC-SR04 and Arduino – Complete Guide. Howtomechatronics.com. Ultrasonic Sensor HC-SR04 and Arduino - Complete Guide (howtomechatronics.com)

Koumaris, N. (2018). Using the SG90 Servo Motor With an Arduino. Electronics Lab.

Using the SG90 Servo Motor With an Arduino - Electronics-Lab.com

Venkatanarayanan, A. and Spain, E. (2014, April 16). 13.03 - Review of Recent Developments in Sensing Materials. In Comprehensive Materials Processing. Academic Press. https://doi.org/10.1016/B978-0-08-096532-1.01303-0

OpenCV. (2024). What is OpenCV? The Complete Guide (2024) https://viso.ai/computer-vision/opencv/

Bluetooth. (2023). Bluetooth Wireless Technology. Bluetooth Technology Overview | Bluetooth® Technology Website