T.C.

ISTANBUL AYDIN UNIVERSITY

INSTITUTE OF GRADUATE STUDIES

REAL-TIME VISUAL TARGET IDENTIFICATION AND TRACKING VIA UNMANNED GROUND VEHICLE (UGV)

MASTER'S THESIS

Nour ZAKARIYA AMMAR

Department of Software Engineering

Artificial Intelligence and Data Sciences Program

JULY, 2022

T.C.

ISTANBUL AYDIN UNIVERSITY

INSTITUTE OF GRADUATE STUDIES

INTELLIGENT MACHINE LEARNING CUSTOMER SEGMENTATION ALGORITHM

MASTER'S THESIS

Nour ZAKARIYA AMMAR

(Y2013.140008)

Department of Software Engineering

Artificial Intelligence and Data Sciences Program

Thesis Advisor: Prof. Dr. Ali OKATAN

JULY, 2022

APPROVAL PAGE

DECLARATION

I hereby declare with respect that the study “REAL-TIME VISUAL TARGET IDENTIFICATION AND TRACKING VIA UNMANNED GROUND VEHICLE (UGV)”, which I submitted as a Master thesis, is written without any assistance in violation of scientific ethics and traditions in all the processes from the project phase to the conclusion of the thesis and that the works I have benefited are from those shown in the Bibliography. (25/08/2022)

Nour ZAKARIYA AMMAR

FORWARD

First, my thanks go to my advisor, Dr.Prof. Ali Okatan for his advising along the study in Istanbul Aydin University.

Then, my sincere thanks go to Dr.Prof. Naim Ajlouni for the scientific and academic advice along the project process time.

Then, my thanks go to network security specialist, my father at the same time, Engineer Zakariya Ammar, for his electronics and networks support.

Also, my grateful and sincere thanks go to my father and my mother for their permanent support, trust, and patience.

Finally, my thanks go to my colleagues Lastly, engineers, professors, and Istanbul Aydin University staff for the support along the duration of the program.

August 2022 Nour ZAKARIYA AMMAR

REAL-TIME VISUAL TARGET IDENTIFICATION AND TRACKING VIA UNMANNED GROUND VEHICLE (UGV)

ABSTRACT

Vision-based Autonomous Robots field is becoming rapidly popular, according to the artificial intelligence revolution. The proposed system is a composed Unmanned Ground Vehicle vision-based target tracking robot prototype. Target tracking is useful in multiple real-life issues such as the assistance and security fields. Acquitted visual input processing is applied through using OpenCV library. The object detection stage of UGV system, is done based on a pre-built deep learning object detection model. YOLOv3-tiny is used to detect objects, which is a light computation-cost version comparing to original YOLO models, and the other complex deep-learning networks. Object to track is specified to be a human only. The target tracking algorithm is based on a sequence of mathematical equations with Region of Interest and stream's frame coefficients. Coefficients refer to the values of locations according to x-axis and y-axis of the frame. A simple mathematical technique is used for the delayed feedback issue. The locomotion of UGV is based on transmitted commands from algorithm to the motors through local network connection. Ultra-sound technique is used for collision avoidance. Robotic eye-based face tracking is a subsystem. In the subsystem, the face detection stage in Robotic eye-based face tracking subsystem is done using Harr-like cascade. The locomotion algorithm is based on various trigonometry rules using pan/tilt and servos means. Innovative methods are followed for performance measurement and comparison. The results show, an autonomous behavior, streamlined and accurate locomotion of tracking.

Keywords: Autonomous Robot, UGV, Object Tracking, Target Tracking, Human Following, Object Detection, YOLOv3, COCO dataset, Face tracking, Robotic eye, pan/tilt.

İNSANSIZ YER ARACI (UGV) ÜZERİNDEN GERÇEK ZAMAN GÖRSEL HEDEF BELİRLEME VE İZLEME

ÖZET

Vizyon tabanlı Otonom Robotlar alanı, yapay zeka geliştirme devrimine göre hızla popüler hale gelmektedir. Önerilen sistem, insansız zemin aracı görüş tabanlı göre hedef takipi robotu prototipinden oluşmamaktadır. Hedef takipi, gerçek hayat sorunlarında yararlıdır, yardım ve güvenlik alanları gibi. UGV sisteminin nesne algılama aşaması, önceden oluşturulmuş bir derin öğrenme nesne algılama modeline dayanarak yapıldı. YOLOv3-tiny, orijinal YOLO modellerine ve diğer karmaşık derin öğrenme ağlarına kıyasla hafif bir hesaplamalı versiyonu olan nesneleri tespit olduğu için kullanıldı. Takip edilecek nesne yalnızca insan olarak belirtilirde. Hedef takipi algoritması, İlgi çeken Bölgesi ve akışın çerçeve katsayılarına sahip bir dizi matematiksel denkleme dayanmaktadır. Katsayılar, çerçevenin x eksenine ve y eksenine göre konumların değerlerini ifade edildi. Gecikmiş geri bildirim sorunu için basit bir matematiksel teknik kullanıldı. UGV'nin lokomotifi, algoritmadan motorlara yerel ağ bağlantısı üzerinden özel komutlar iletilmektedir. Çarpışmayı önlemek için ultrason tekniği kullanıldı. Robotik göz tabanlı yüz takipi bir alt sistemdir. Alt sistemde, Robotik göz tabanlı yüz takipi alt sistemindeki yüz algılama aşaması Harr-like kaskad kullanılarak uygulandıdır. hareket algoritması, pan / tilt ve servos araçlarını kullnan aletli trigonometri kurallarına dayanmaktadır. Performans ölçümü ve karşılaştırma için yenilikçi yöntemler izlenmektedir. Sonuçlar, özerk bir davranış, akıcı ve doğru takipi lokomotifi olduğunu göründü.

Anahtar kelimeler: Otonom Robot, UGV, Nesne takibi, Hedef İzleme, İnsan takibi, Nesne Algılama, YOLOv3, COCO veri kümesi, Yüz takibi, Robotik göz, pan/tilt.

TABLE OF CONTENTS

ABREVIATIONS

RB4 **:** Raspberry Pi 4

SBC **:** Single Board Computer

AI **:** Artificial Intelligence

UAV **:** Unmanned Aircraft Vehicle

AUV **:** Autonomous Under-water Vehicle

UUV **:** Unmanned Under-water Vehicle

UGV **:** Unmanned Ground Vehicle

PTZ **:** Pan Tilt Zoom

RAM **:** Read Access Memory

GPIO **:** General-Purpose Input/Output

GND **:** Ground

5V **:** 5 Volts power supply

3V3 **:** 3 volts power supply

VCC **:** Voltage Common Collector

SCL **:** Serial Clock Line

SDA **:** Serial Data

I2C **:** Inter-Integrated Circuit

SPI **:** Serial Peripheral Interface Bus

UART **:** Universal Asynchronous Receiver/Transmitter

DNC **:** Do Not Connect

PWM **:** Pulse Width Modulation

PCM **:** Pulse-code Modulation

UI **:** User Interface

LED **:** Light Emitting Diode

USB **:** Universal Serial Bus

HDMI **:** High-Definition Multimedia Interface

CSI **:** Camera Serial Interface

DSI **:** Display Serial Interface

ML: **:** Machine Learning

LAN **:** Local Area Network

SD card **:** Secure Digital card

ENA **:** Enable A

ENB **:** Enable B

TTL **:** Transistor-transistor logic

Hz **:** Hertz

GB **:** Gigabytes

DC **:** Direct Current

DIY **:** Do It Yourself

FFC **:** Flat Flexible Cable

YOLO v3 **:** You Only Look Once version 3

SSD **:** Single Shot Detection

CNN **:** Convolutional Neural Network

R-CNN **:** Region based Convolutional Neural Network

COCO **:** Common Objects in Context

KLT **:** Kanade-Lucas-Tomasi

ReLU **:** Rectified Linear Unit ()

AdaBoost **:** Adaptive Boosting

GPU **:** Graphical Processing Unit

R **:** Resistor

IN **:** Input

OUT **:** Output

FPS **:** Frame Per Second

Cm **:** Centimeters

k **:** Kilo -one thousand

l **:** Longitude or latitude

w **:** Width

h **:** Height

$\boldsymbol{d}$ **:** Distance

$\boldsymbol{v}$ **:** Velocity

$\boldsymbol{t}$ **:** Time

SYMBOLS

Δ	**:** Change rate
$\boldsymbol{x}$	**:** Longitude position
$\boldsymbol{y}$	**:** Latitude position
$\boldsymbol{\alpha}$	**:** Alpha
$\boldsymbol{\Omega}$	**:** Ohm

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

I. INTRODUCTION

A. Overview

The rapid evolution of AI coupled with automation control induce Unmanned Vehicle concept. UV refers to the Unmanned control of a Vehicle with either human supervision or semi-supervision through remote networks. Diverse types of UVs are surfaced. Most popular and general types are categorized into UAV, UUV and UGV (Moud, et al., 2018). UAV defined as the Unmanned Aircraft Vehicle, typically named drone, aerodynamics rules are used for the physical motion in air (Chen, et al., 2009). UUV defined as the Unmanned Underwater Vehicle, -named as AUV as well- is an intelligent submersible either human unsupervised or semi-supervised (Yang, et al., 2021). UGV defined as the Unmanned Ground Vehicle that proposed to navigate surface environments with either human-unsupervised or semi-supervised.

Vision-based object tracking plays a significant role in the computer vision field. Vision-based object tracking refers to multiple fields such as video surveillance, vehicle navigation, and robot autonomous control. Object tracking is based on diverse object detection methods. Such as color-based, motion-based, and shape-based methods. The tracking process rely on the chosen optical criteria, such as point-based, kernel-based, and silhouette-based criteria methods (Balaji, et al., 2017).

In these kinds of projects, diverse methods of object detection are followed. Machine learning simple methods are widely applied such color-based detection such as the work in (Kumar, et al., 2016) because of its low-cost computation. More expensive computation method is followed in work (Yosafat, et al., 2017), where the detection is based on the Haar-like cascade for face detection model (Voila, et al., 2004).

Recently, deep learning-based projects are increasing. Various models that are based on COCO dataset are popular, such as SSD (Wei, 2016), Mask R-CNN (He, 2017) and Faster R-CNN (Ren, 2015). SSD is the most famous network for object

detection field and its related applications. SSD model depends on one single shot to detect the objects with GPU usage, followed by YOLO models, which refer to You Only Look Once. YOLOv3-tiny is the best model to be chosen according to the fair computing capabilities in comparison with heavy models. The proposed system depends on deep learning pre-built model: YOLOv3 and YOLOv3-tiny (Redmon, 2018; Adarsh, 2020), which both are based on (Redmon, 2016).

The purpose of the system is to superimpose an autonomous, real-time, object tracking system using a Single Board Controller (SBC). It is usually composed of image acquisition, image processing, automatically detection and allocation the object, followed by mathematical process to keep tracking the object and send orders to SBC. The project is applied using Raspberry Pi 4 model B controller with 8 GB RAM. Raspberry Pi is a small and low-cost SBC. It provides an easy-to-use tool to help learners and beginners to learn robotic essentials and to simulate the factories' big robots in real world based on Python language. Since Raspberry Pi is an SBC, its computing power considered like embedding computers (etechnog.com, 2021).

A simple obstacle avoidance system is followed, by measurement of the distance between the UGV and the opposite object based on ultrasound technique (Everett. 1989).

The basic aim of the study is to find an object and track it. The project depends on deep learning pre-built models: YOLO v3 and YOLO-tiny v3 (Redmon, 2018; Adarsh, 2020) using Raspberry Pi controller, electronic chips, and camera.

Robotic eye applications are important in security field. It is becoming popular in robotic applications. The subsystem in this project is the face tracking using pan/tilt, servos and camera based on frontal face Haar-like cascade (Padilla, 2012).

Haar-like cascade model innovated by Viola & Jones. It is a model based on binary feature selection technique to detect objects for face detection and tracking (Viola, et al., 2004). It results a high-speed rate of object detection. Frontal face Haar-like cascade is one of Haar-like cascade models. In reference (Padilla, 2012), viola-jones cascade is evaluated as a robust framework to detect faces in object location-based systems.

Results are acceptable despite the slowness of the robot since there is no accelerator nor GPU processor. In aim to speed the process up, frame size is decreased, and YOLOv3-tiny is used instead of YOLOv3.

B. Problem Statement

Scientists and researchers need to discover the solutions of different issues in different environments. Some environments are extremely difficult for human to reach, dangerous for human survival, costly to send a human for discovery or navigations as well as the need of disabled people and their needs of help at all the time.

Space discovery is a remarkable sign of the science revolution. There were a big ambiguity about mars sphere until 2004, there when NASA institute sent the famous robots, Discovery rovers, to Mars, thus, the robots keep discover the sphere by sensors via visual connection till 2019 (Nasa.gov, 2010). Discovery rovers are UGV robots with specific functions that satisfy the mission they tasked with.

Wildlife is full of discoveries. Sometimes, it is dangerous for human to deep-inter in it, even rarity, animals would behave in abnormal way, which make extracting information is not accurate. National geographic, used a robotic gorilla to spy on the gorilla's life. It takes photos and behaves like real gorillas. As well as robotic inventions such as Spy Dolphin, Spy Monkey, Spy Sloth, and Spy Macaw (Gregory, "Spy in the Wild", 2020)

UGV robots are extremely capable of navigating safely inside unknown environments but are restricted to a limited field of view (Hood, et al., 2017).

Face tracking by stable robot is important in security field (Seong-Hoon, 2019). As well as other fields like personal interactive assistance (Sanjaya, 2018).

C. Proposed System

The purpose of this study is to superimpose an autonomous, real-time, unmanned, motional object tracking system using SBC. it is usually composed of image acquisition, image processing to automatically detect and allocate the object, followed by mathematical process to keep tracking the object and compute orders to electronic chips. The first stage of system is the object detection, which is based on deep-learning

pre-trained model, YOLOv3-tiny, for wheel vehicle tracking process, whereas a simple usage of frontal face cascade for robotic eye tracking process is applied. Second stage is hardware composition including setting up inputs and outputs. Third stage is to integrate detection algorithms with real-time tracking algorithms.

D. Related Works

(Mir-Nasiri, 2006) followed the vision-based object tracking, where the acquiesced image is binarized and contoured, therewith, the computation cost of tracking process is reduced.

(Chatterjee, et al., 2020) experimented YOLO model, which is a low-resolution process of images -480p stream. In their experiment, YOLOv3 shows the best results. The controller that is used is NVIDIA Jetson Nano controller, which has more computing capabilities than Raspberry Pi controller.

In experiment of (Hussain, 2019), the object detection is based on mask contour, where the difference concept is followed. The background and different group of pixels are subtracted from each other, and the larges masked group of pixels is assessed as the ROI area.

(Tonberg, 2016) experimented RB 2 controller using C++ programming language. he followed two different tracking algorithms using color-based object detection. He utilized pan/tilt servos approach. He compared between two algorithms KLT Feature Tracker and CAMshift, where KLT results records high performance of the system.

(Padreep, 2016) experimented the object detection using contour method, which is a computation low-cost method but cannot be used on a wide range. The tracking method is based on the grids of frame, the number of grids of frame is 9, thus the mathematical algorithm is consisting of nine conditional states.

In (Safran, 2012) experiment, the researchers followed two tracking processes, robotic eye and UGV target tracker. The object detection is based on contours since he used the Arduino controller, which is an SBC with lower computation capabilities than Raspberry Pi SBC's. The main system is the UGV target tracker whereas the robotic eye-based target tracker as the subsystem, which is alike the proposed system in this study.

In (helloworld.co.in,2022) plug, the author used TensorFlow framework to create a UGV that follows a human with the employment of an accelerator device. Tracking algorithm in project (helloworld.co.in,2022) is a novel simple tracking algorithm, which this project is inspired by it

(Palinko, et al., 2015) followed the architecture of tracking eyes gaze, which first meant detecting eyes, then keeping the deviation near to that gaze to track the center of object in the proposed system.

(Seong-Hoon, 2019) experimented the PTZ system, which is a fixed robotic eye that tracks a person and detect his position in real-time. Pan and tilt means are used in the mentioned system, which is alike the proposed system.

II. TOOLS AND METHODS

Miscellaneous tools, means, and methods are utilized in this study. Hardware tools and software packages are detailed in the following subsections.

A. Hardware Tools

Hardware components are based on the convenience with the universal protocols of electrical boards.

1. Raspberry Pi SBC

Figure 1 Raspberry Pi 4 model B SBC controller

The project is based on Raspberry Pi 4 B model controller with 8 gigabyte RAM, as shown in Figure 1Figure 2. It comes with a fan and other small chips. Raspberry Pi is a small, low-cost SBC. It is invented in the United States by the Raspberry Pi foundation. It provides an easy-to-use tool to help learners and beginners to learn robotic essentials and to simulate the factories' big robots in real world based on Python language. Hence, the Pi part of its name came from the focus on using it to code in Python. Since it is a SBC then it is not as powerful as personal computers. Its computing power is considered like the embedding computers (etechnog.com, 2021).

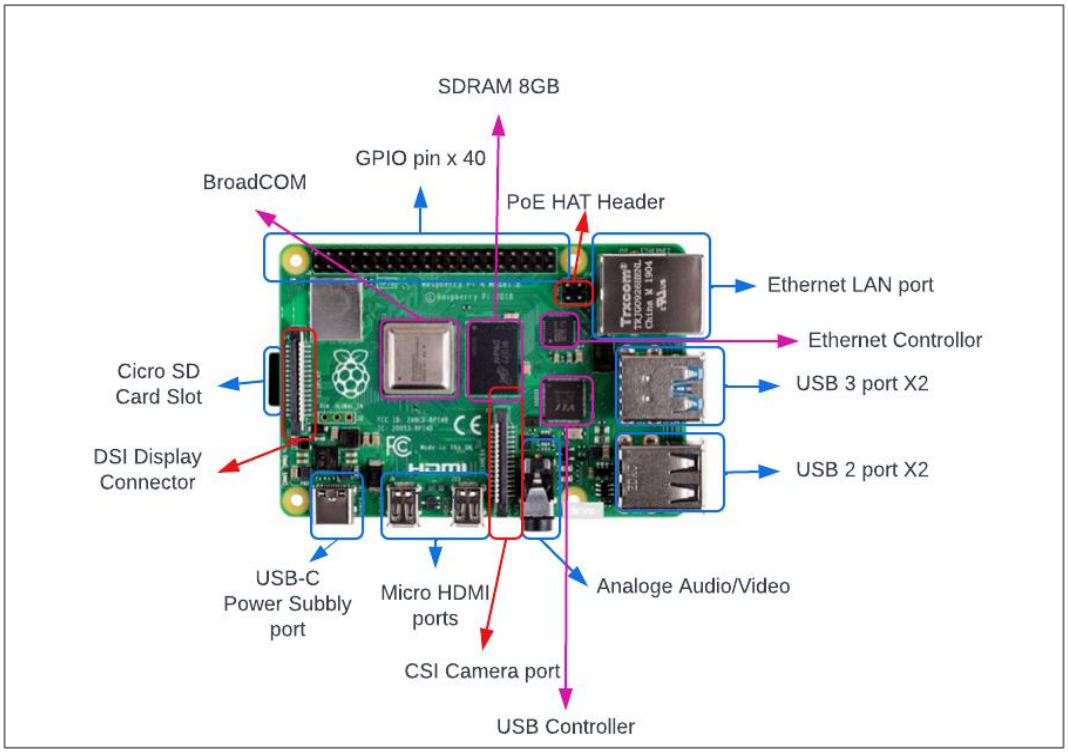


Figure 2 Raspberry Pi 4 B layout diagram

As shown in figure 2, Raspberry Pi parts are declared. In Figure 3 Raspberry Pi's 40 pins are named and Table1 specifies the functionality of each pin (etechnophiles.com, 2022). Noticeably, the GPIO pins may take more than one functionality except for ground and power pins.

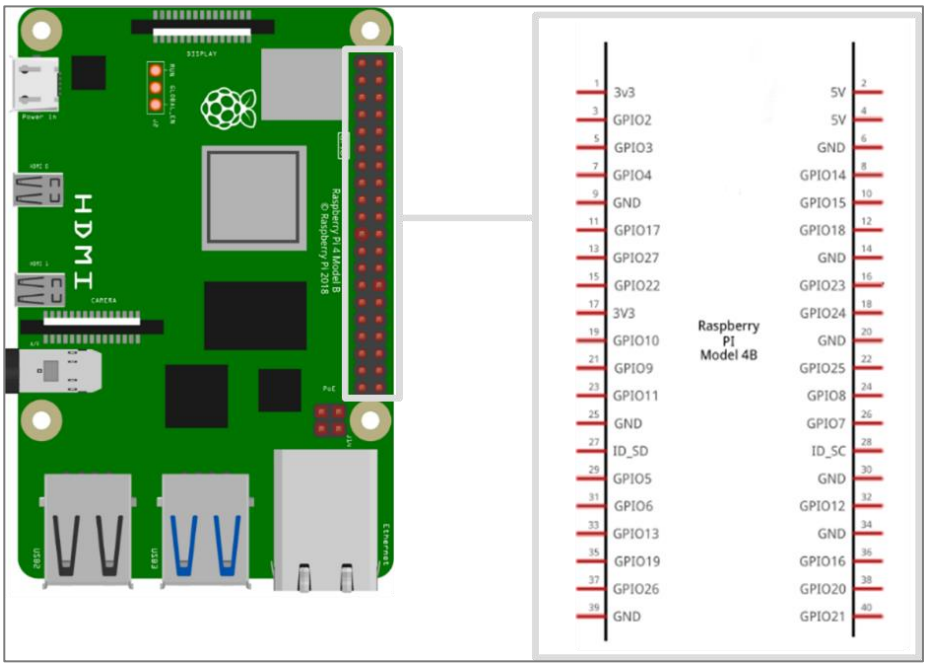


Figure 3 Raspberry Pi GPIO pins diagram

Table 1 Raspberry Pi GPIO pins functionalities declaration

pins (GPIO)		Functionality
GPIO	All pins except Power and GND pins	General-Purpose Input/Output
GND	Pin6, Pin9, Pin14, Pin 20, Pin25, Pin30, Pin34, Pin 39	Ground pins
PWM	Pin32 GPIO12(PWM0), Pin 33 GPIO13(PWM1), Pin12 GPIO18(PWM0), Pin35 GPIO19(PWM1),	PWM (Pulse-width Modulation) pins are for controlling analog voltage of servo and DC motors.
Power	pin1, pin1 (3.3V pins), pin2, pin4 (5V pins).	Power out means taking power from this board to the external devices. 3.3V pins are connected through the regulator circuit, so they can provide a stable 3.3V power output. The 5V pins are also connected to the USB ports.
I2C	Pin3 GPIO2 (SDA), Pin5 GPIO3 (SCL)	I2C pins are working according to I2C protocol, multi-drop bus. multiple devices can be connected to these pins. Each device has its own unique I2C address. GPIO2 (SDA) used for data transaction where GPIO3 (SCL) used for clock transaction.
PCM	Pin12 GPIO18 (CLK), Pin35 GPIO19 (FS), Pin38 GPIO20 (DIN), Pin40 GPIO21 (DOUT)	PCM (Pulse-code Modulation) is a digital representation of sampled analog. On the Raspberry Pi it's a form of digital audio output
SPI	Pin11 GPIO17 (SPI1 CE1), Pin12 GPIO18 (SPI1 CE0), Pin19 GPIO10 (SPI0 MOSI), Pin21 GPIO9 (SPI0 MISO), Pin23 GPIO11 (SPI0 SCLK), Pin24 GPIO8 (SPI0 CE0), Pin26 GPIO7 (SPI0 CE1), Pin35 GPIO19 (SPI1 MISO), Pin36 GPIO16 (SPI1 CE2), Pin38 GPIO20 (SPI1 MOSI), Pin40 GPIO21 (SPI1 SCLK)	SPI refers to the four-wire serial bus. SPI allows to attach several compatible devices to one set of pins along power supply and ground.

Raspberry Pi setup process is done before the system. A user and a password are created, then access and groups of users is allowed to avoid access deny problems. It is recommended to create a virtual environment. After programs execution via terminal or desktop, the ways, and preferences of displaying UI of SBC are shown in **App-1**.

2. H- bridge L298N board

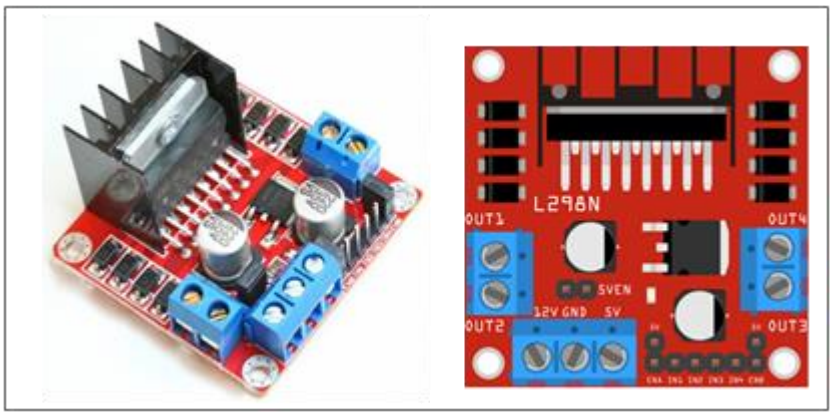


Figure 4 Dual H-bridge L298N electronic board actual (left) and drawn (right)

Table 2 L298N parts' functionalities

Functionality	Pin	Description
Speed control pins	ENA	ENA input function is controlling the speed of Motor A. in case of connecting a jumper is connected to ENA, the ENA pin connected to +5 V power supply pin, which enable the motor A rotates with the maximum speed.
	ENB	ENB input function is controlling the speed of Motor B. in case of connecting a jumper is connected to ENB, the ENB pin connected to +5 V power supply pin, which enable the motor B rotates with the maximum speed.
Control pins	IN1, IN2	The input pins of Motor A. they are responsible of the rotating direction of Motor A. the followed technique is based on the logic gate OR. With commands HIGH and LOW
	IM3, IN4	The input pins of Motor A. they are responsible of the rotating direction of Motor A. the followed technique is based on the logic gate OR. With terminals HIGH and LOW
Output pins	OUT1, OUT2	The output terminals sent to Motor A.
	OUT3, OUT4	The output terminals sent to Motor B.
Power supply pins	5v	The power supply of 5v Electric current for the switching logic circuitry in the board.
	+12v	The extended power supply up to 35 v Electric current for the switching logic circuitry in the board.
	GND	Ground pin negative point. It is required to complete the electrical circuit.

As shown in Figure 4, L298N board contains terminal driver part of the supply area with range 5 – 35 VMS of voltage supply capability. The logical part of the terminal power supply range lies in range of 3.3 -5.5 Vs. H-bridge Board helps to control the power supply to avoid the damage for raspberry Pi. In order that, it is not allowed to connect DC motors to Raspberry Pi GPIO pins directly (computervision.zone, 2022). Table 2 illustrates the functionality and the description of each part and pin in L298N h-bridge board.

3. Ultra-Sonic Sensor HC-SR04 Board

Ultra-sonic sensor HC-SR04 board is a simple board. It consists of two ultra-sonic waves emitters, as shown in Figure 5. The emitted waves lay in range of distance from two cm up to of 400 cm. as illustrated in Table 3, the required voltage of the board is lied between 3.3V and 5V. The frequency of the emitted waves from the board is 40Hz. The angle of triggers is 15° (thepihut.com,2022).

Figure 5 Ultra-sonic sensor HC-SR04 board actual (left) drawn (right)

Table 3 HC-SR04 board parts' functionalities

Pin	Functionality and description
VCC	Power supply pin. Supply voltage range is 3.3V to 5V.
Trig	The generator of pulse that input signal is analog and output signal is binary. Trigger output Signal is measured in micro TTL pulse.
Echo	The generator of ultra-sonic waves. The frequency of waves is 40Hz.
GND	Ground pint to complete the electrical circuit.

4. PWM driver PCA9685 16 Channel I2C Module board.

PCA9685 board is an PWM driver with a built-in clock. Its controlling system is based on I2C protocol, as shown in Figure 6. The basic required power supply of the board lies between 3.3Vs and 5Vs, which means it can be controlled via a 3.3V microcontroller, with range of 3.3V logic pull-ups to 5.5V outputs. It contains a total of 992 outputs – that can be plugged to multiple servos or LEDs (MCWHORTER,2022).

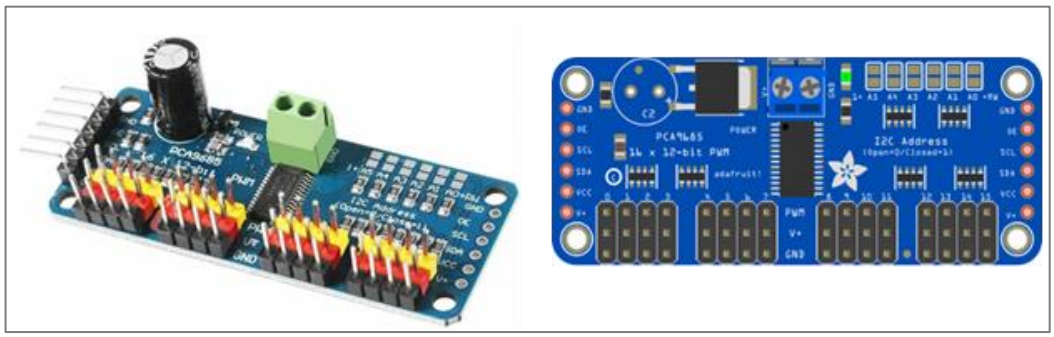


Figure 6 PCA 9685 PWM electronic board actual (left) and drawn (right)

5. Camera board

Two camera types are experimented, webcam camera and Raspberry Pi camera module 2 v1.3. Webcam, as shown in

Figure 7, is a USB portable device. Whence, its functionality is different from Pi camera board. Pi camera board, as shown in Figure 8, is especially factorized for SCB devices such as Jetson nano, Arduino, and Raspberry Pi. Pi camera is a CSI portable device. Figure 2 illustrates the CSI port in Raspberry Pi board.

Figure 7 Webcam with USB port

Figure 8 Raspberry Pi camera module v1.3

Both experimented cameras are alike, but there are remarked strengths and shortcomings of them, which are appreciated to be mentioned, detailed in **App-2**.

Problems with webcam:

- Heaviness: it is heavy on the fragile hardware system
- Its resolution processing is difficult: since the process of centralization is applied the accuracy in webcam is not satisfying

It is not flexible since it depends on USB port and results some problems.

Each execution of program requires unplugging the port then plugging it again. -Error (-215) is occurred, which feeds that there is no detected webcam while it is plugged but not detected (not seen by the SBC)- which make the system not robust and fragile.

Problems with Pi camera:

- Pi camera does not work when motion library is installed. Thereby, motion library must be deleted to make it works for real-time capture stream at each single boot - kill <motion> process is required after each boot.

6. Electronic simple parts

- 2 x micro servo with Pan/Tilt

As shown in Figure 9, Pan/Tilt, is a stand that moves left-right and up-down. The Pan/Tilt are appended with 2× micro servo 9g. Micro servo SG90 is a motor with a small amount of energy and a high capability of controlling by GPIO PWM pins in Raspberry Pi (MCWHORTER,2022).

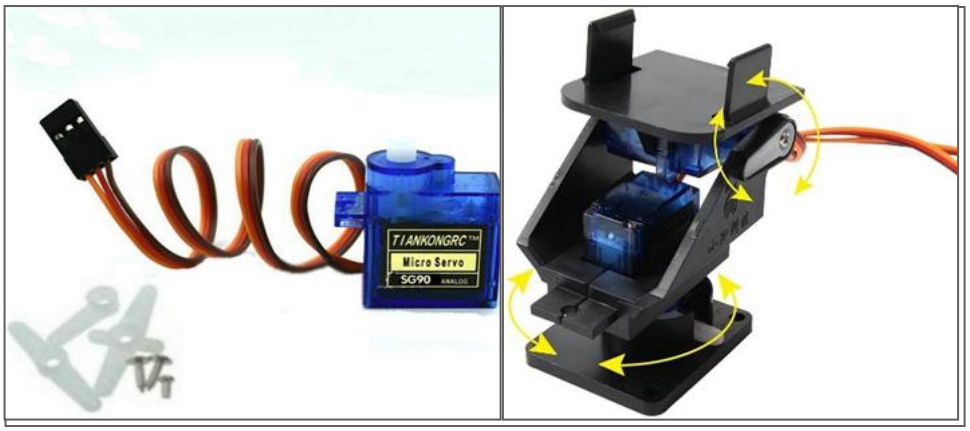


Figure 9 SG90 micro servo (left) and pan/tilt kit (right)

- Robotic vehicle kit with DC motor x 2

A DIY car kit with 4 wheels is used, but only two DC motors are run. The kit comes with wheels, batteries case and a modified chasses to carry the hardware boards and means.

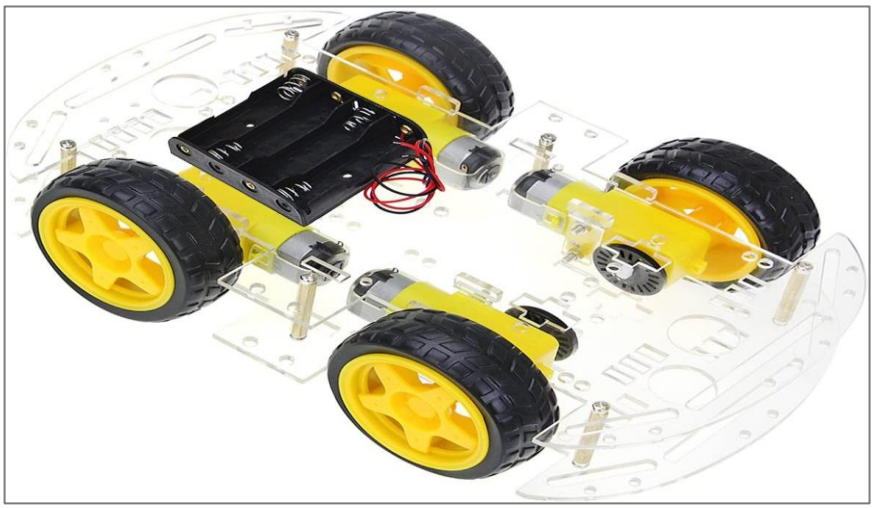


Figure 10 DIY robot smart car with four DC motors kit

- Caster rotating wheel

Caster rotating ball, as shown in Figure 11, is a metallic ball fixed by a holder. The motion technique is the ball rotation in each direction. It makes the movement of car faster and more flexible.

Figure 11 Caster rotating wheel

- Wiring cables

FFC 15 pin cable for pi camera, as shown in Figure 8. Female to Female, Male to Male, Female to Male, colored wires and not classified wires to connect the electrical circuits

- Power supply

5v power supply device -power bank- to supply the Raspberry Pi wirelessly. 8 × rechargeable battery to supply two DC motors that need 12 volts, so that 8 x AA 1.5 v batteries are used.

- Raspberry Pi small fan

To cool up the mother board. It cools up to 30 ˚ Celsius degrees. The power supply for the fan is retrieved directly from 5V pin in the Raspberry Pi.

B. Software Tools and Methods

1. Packages and Libraries:

As shown in Table 1, virous powerful libraries are imported. OpenCV is the most important library in this project, which process the real-time object detection and handle all images pre-processing functions (opencv.org, 2022).

Table 4 Libraries installed on the controller

Module	=>Version	Dependencies	Description
Gcc	-		Compiling configuration software
Libc6	2.2B		Compiling configuration library
Libgccl	1.8B		Compiling configuration library
Libstdc++	8.3		Compiling configuration C++ library
GNUmake	3.13		Compiling configuration software
CMake	3.15.2	Gcc, Libstdc, GNUmake	An extensible system that manages the build process
Build-essentials	12.6	cMake	build-essential is a meta-package that a pre-request to install packages
Wheel	0.37.0	Build-essentials	A built-package format for python
Cython	0.29.24	cMake	A compiler for writing c extensions for python language
Python 3	3.7	cMake, cython	Programing language software
Pip3	21.0.1	Build-essentials	A tool for installing python packages
OpenCv	4.0.1	Pip3	An open-source computer vision and ML library
NumPy	1.16.2	Pip3, python	A smart library for handling numeric arrays operations
Fswebcam	-		Simple software to setup webcam
Pillow	8.4.0	Pip3	A smart library to handle image processing
Time	-	Pip3	To handle delay and accurate real time
Threading	-	Pip3	A framework that allows to run different parts of program concurrently and simplify the code in aim to speed the system up
RPi.GPIO	-	Pip3	A library used to identify and connect Raspberry Pi's GPIO pins with electronic boards
adafruit-circuitpython-servokit	-	Pip3	A special library installed with helpful to deal with Adafruit PCA9685 chip
adafruit-circuitpython-pca9685	-	Pip3	A special library that includes pre-built trigonometric functions to deal with Adafruit PCA9685 chip
I2C	-	Pip3	To access and handle PWM I2C protocols
BCM V4l2	-	Pip3	To handle videos with servos
Tightvncserver	-		To allow control pi camera via VNC
Gpiozero	-	Pip3	To handle operations of GPIO pins
DistanceSensor	-	gpiozero	To measure distance by HC-SR04 board

2. Dataset

The artificial models are based on COCO dataset (Lin, et al., 2014). COCO is a novel dataset that afford four functionalities: image classification, semantic segmentation, object localization, which is used in this project, and segmenting

individual object instances, which is the innovation of Microsoft team. COCO dataset contains 80 labeled object categories and 91 stuff categories. COCO dataset consists of 330k images, 5 captions per image. Only person category consists of 250K person images with key points.

3. Target detection methods

Two methods are followed in the target detection stages. In the main system, YOLOv3-tiny model is used, whereas in the subsystem Haar-like cascade is used.

a. YOLOv3 algorithms

The YOLOv3-tiny model is used for object detection (Adarsh, et al., 2020),which is a light curtailed version of the original YOLO models (Redmon, et al., 2015, 2018). It is the best model to be utilized, according to the fair computing capabilities of Raspberry Pi SBC. In original YOLOv3, as shown in

Figure 12, there are 109 layers in the network three of them are output layers with different dimensionality of each output layer, Whereas in YOLOv3-tiny, the architecture consists of 13 layers only. There are multiple strength points of YOLOv3-tiny over YOLOv3. Such as speed, low process computation -no need for GPU. Weights and cfg files of the pre-built model are imported from the reference (pjreddie.com, 2022) then implemented in the object detection process.

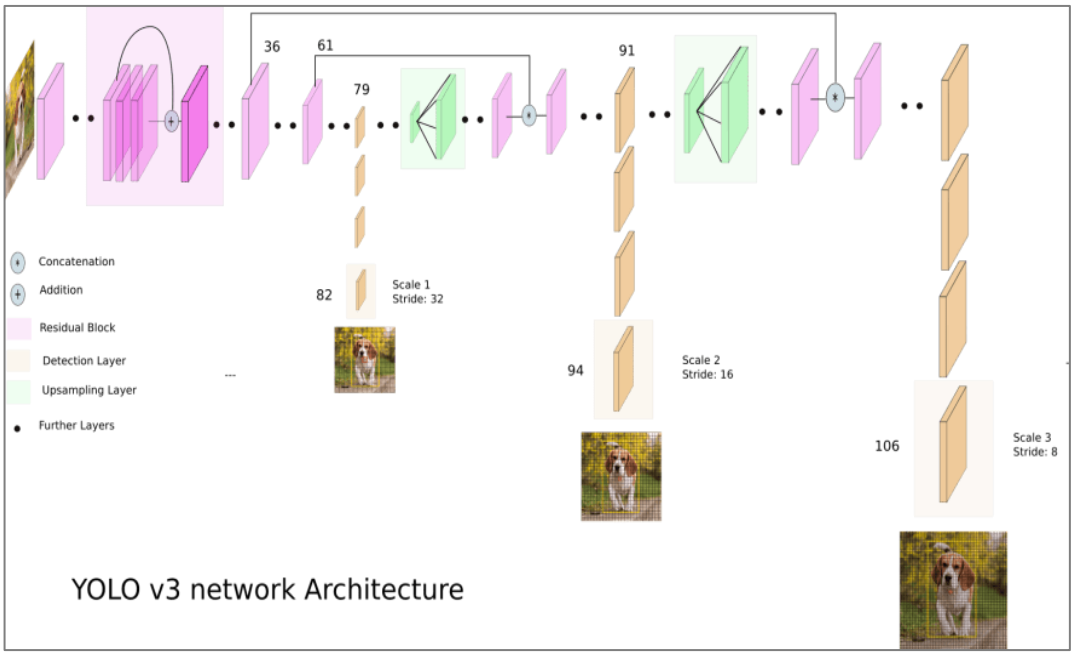


Figure 12 YOLOv3 neural network architecture visualization

The structure of YOLOv3-tiny network consists of two output layers, with different dimensionality. The shape of premier output layer is (300 x 85). The shape of the conclusive output layer is (1200 x 85). Second dimension refers to the indices' values: y-axes center of RoI frame, width of RoI frame, height of RoI frame, confidence value of detection and 80 other indices which are the COCO classes names (Adarsh, et al., 2020).

Table 5 Visualization of output layer with 300x85 dimensions, with person detection

Index no.	1	2	3	4	5	6	7	8	…	85
Box no.	x_o	y_o	w	H	Confidence	1st class	2nd class	3rd class	…	80th class
Class name						Person	Bicycle	Car	…	Toothbrush
1	0.51	0.64	0.58	0.36	0.93	**0.93**	0	0	…	
…									…	
300									…	

The shape of first output layer in YOLOv3-tiny neural network is (300 x 85). The first multi-dimensional output array of the model's neural network is simulated in Table 5 of prediction output array, as shown in Figure 12 visualization, the box's number is predicted as a car since it has the highest confidence value. YOLOv3-320 speed is 22 milliseconds to detect. The difference between YOLOv3-tiny and YOLOv3 is the number of convolution layers, where it decreases the cost of computing and memory, but less detection accuracy (Zhang,2019). The number of layers in YOLOv3 is 53 convolutional layers. Each of them is followed by a batch normalization layer and a Leaky ReLU activation layer with no usage of pooling layers, whereas in YOLOv3-tiny, seven convolutional layers and six max-pooling layers -total of 13 layers. Table 5 declares a tubular visualization of the first output layer of YOLOv3-tiny model. The first five indices refer to the location and the confidence of the object, whereas the rest of indices refer to the COCO classes' names. The confidence indicates the predicted object, which is a human in the simulated example.

b. Frontal Face Cascade

The Frontal Face Cascade is used as a simple model for face detection for the robotic eye sub-system. Haar-like cascade model innovated by Viola & Jones is a

model based on binary feature selection technique to detect objects based on Haar features, as shown in

Figure 13, and AdaBoost classification algorithm (Viola, et al., 2001, 2004). In reference (Viola, et al., 2004), researchers improved the original cascade for face detection. It results an object detection with high-speed rate. Frontal face Haar-like cascade is one of Haar-like cascade models. In reference (Padilla, 2012), viola-jones cascade is evaluated as a robust framework to detect faces in object location-based systems.

Figure 13 shows three types of the Haar-like feature selection. Edge features occurred when a feature is composed of two edges only zero and one -white and black. Line features occurred when the ones -black- are surrounded with zeros -white. Four rectangle features when four corners are opposite to each other. First, the acquiesced image is preprocessed and binarized. The binarized image is full of features, small features such as (00-11) feature, medium features such as a part of the eye and a part of the eyelash and large features such as the entire mouth and the entire eye view. Mouth identification is an example of edge Haar-like feature, whereas nose identification is an example of line Haar-like feature.

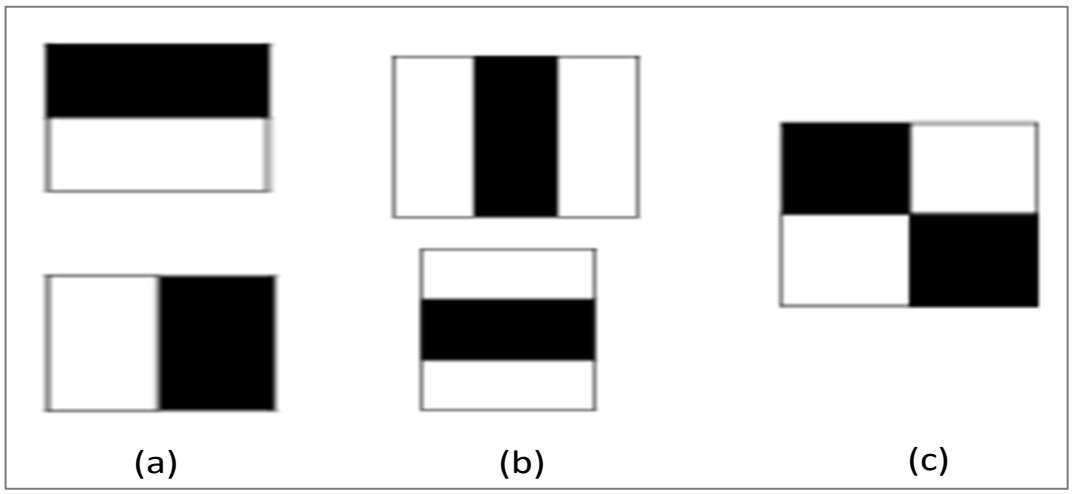


Figure 13 Haar-like feature types. (a) Edge features. (b) Line features. (c) Four rectangle features

III. SYSTEM OVERVIEW AND DESIGN

A. Architecture of Hardware System

Hardware system components are an SBC Raspberry board, L298N H-bridge board, two DC motor drivers, power supply, PCA9687 PWM board, pan/tilt with SG90 servos, camera board, HR-SR ultra-sonic sensor board and a monitor to control and connect with Raspberry Pi via wireless local network.

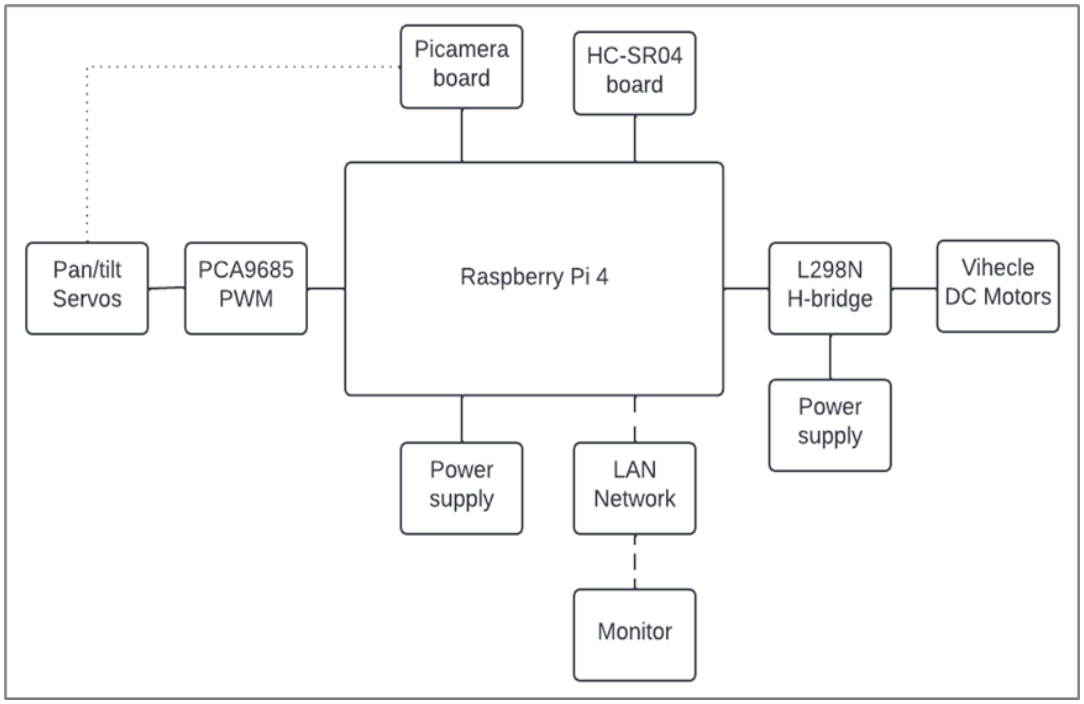


Figure 14 Block diagram of the whole hardware system

The proposed system is segmented into three subsystems. As shown in Figure 18, components of the object locomotion-tracking subsystem are connected serially, the power supply resource for Raspberry Pi, the Raspberry Pi SBC, the camera board, the L298N H-bridge board, the monitor, and the DC motors and power supply resource.

Components of the obstacle avoidance subsystem are connected serially, the Raspberry Pi SBC, the power supply resource for Raspberry Pi, the Raspberry Pi SBC, a bunch of transistors, the HC-SR04 ultra-sonic sensor board.

Components of the robotic eye face tracking subsystem are connected parallelly, the Raspberry Pi SBC, the power supply resource for Raspberry Pi, the Raspberry Pi SBC, the PCA9687 PWM board, Pan/Tilt with 90SG servos and camera. The servos are plugged to the Raspberry Pi indirectly, while Pan/Tilt are concatenated with servos to handle face tracking locomotion process.

B. Design

The composition of the prototype robot is done across multiple stages. First stage: USB port webcam, 4 × 1.5 AA batteries, 4 wheels are used and only 2 DC motors are composed, which results heaviness on the wheels and the power supply was not enough. Second stage: In this stage, two wheels are removed and replaced with a singular caster rolling ball to enhance the movement of car. After comprehensive changes, webcam is heavy and not connected robustly because of USB port webcam connection. Third Stage: In this stage, the webcam is replaced with the Pi camera board, as shown in Figure 8. As well, the direction of movement is changed, where caster became forward and two wheels with DC motors are in backward. Design is completed here as a sophisticated design.

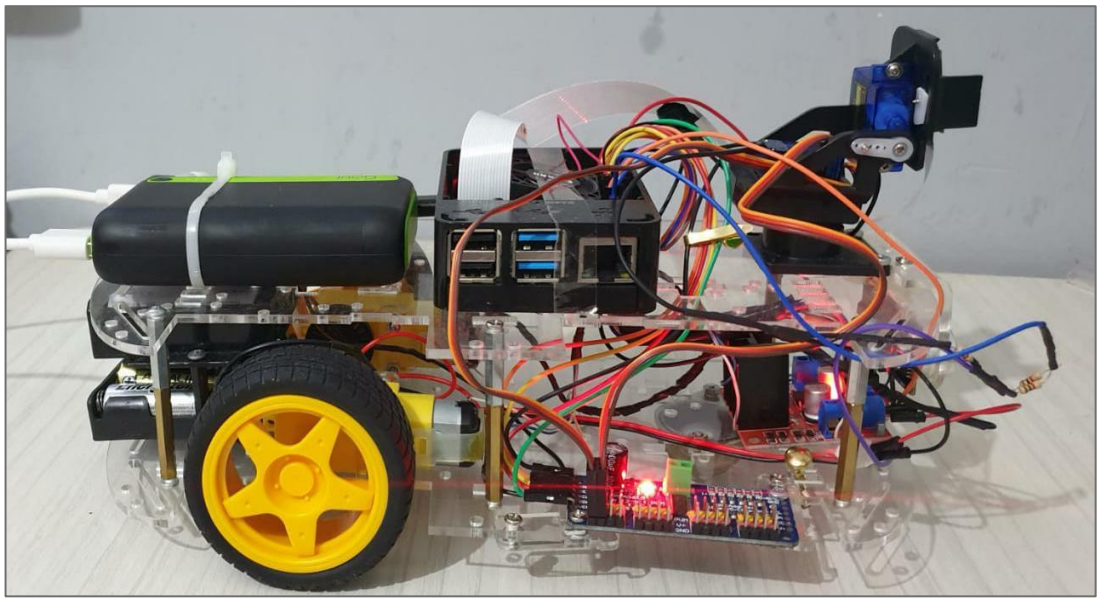


Figure 15 Last design of UGV robot prototype side view

As shown in Figure 15, the raspberry Pi board is posited in the middle of pan tilt and the power bank to reduce the comprehension and complication of the connection's wires. L298N h-bridge and PCA8796 PWM boards are posited on the ground floor of

the vehicle. The batteries are posited rearward the ground floor to be near to the DC motor drivers and the L298N board. Figure 16 shows the prototype from the frontal side of vision. The camera is posited on the surface floor of the vehicle to allow the camera to acquisit the maximum range of visual input. HC-SR04 ultra-sonic sensor board is posited in the middle of surface floor and ground floor to enhance the process workflow of the obstacle avoidance system.

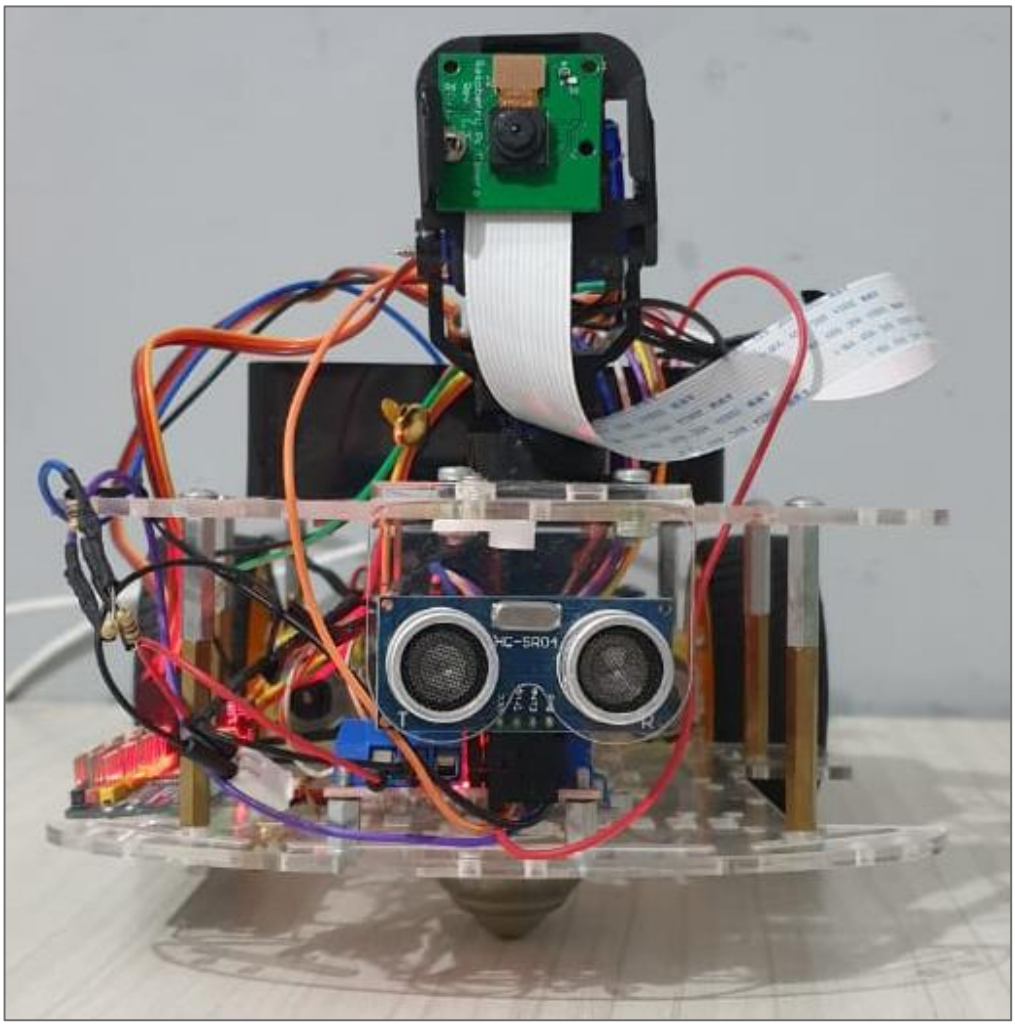


Figure 16 UGV robot prototype frontal view

IV. UNMANNED GROUND VIHECLE SYSTEM

A. Integrated System Overview

The system is divided into two stages, the stage of object detection and the stage of object tracking. Figure 17 shows the flowchart of the entire system. The system executes two processes at the same time. First process is the primary function of the system, which is the human tracking. Second process is the obstacle avoidance function, which is a secondary process that aims to handle collision problem.

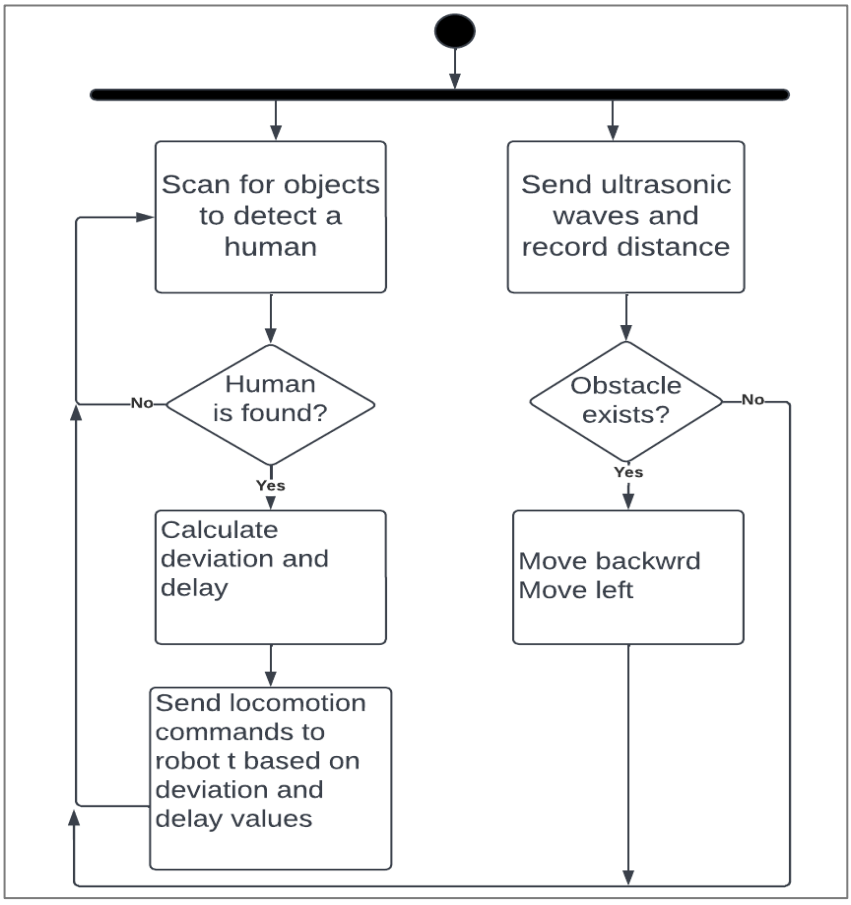


Figure 17 . Flowchart of Object detecting and tracking via UGV system

The system starts with the object detection flow state, which is consists of a loop of capturing images, pre-processing the image, and passing them into object detection model. The target detection is based on the largest area that is covered with a human

object. When Human object is detected, the system moves to the next flow state. Second state consists of mathematical equations that process the tracking state according to frame and bounding box coefficients. x deviation and delay values are induced. Third state contains the motion commands that are transmitted to electronic boards to actuate the vehicle regarding the induced values. In case an obstacle appears, a new locomotion orders transmitted to electronic boards to make vehicle avoids the obstacle. The end of the flowchart is the start again since it is a loop of computations and commands.

The whole process of detection, tracking and movement is done in main function. First, the image pre-processing and object detection process are done, as explained in Table 7 and Table 8. After that, the tolerance value is specified. In line 16 of Algorithm 7, "Object locating" function is called to handle object tracking process. Thenceforth, "Robot locomotion" function is threaded. The main function displays the feedback as illustrated in Table 6.

Table 6 Integrated system algorithm pseudocode

Algorithm 1 Object detection test system pseudo code
1. Function: main
2. Input: images stream
3. Output: Human tracking with visual locomotion
4. Dimensions of input image= 320
5. While True:
6. Call human detection function
7. Target = output of human detection function
8. Call object locating function (thread Robot locomotion function)
9. Send terminals output of Robot locomotion function to motor drivers
10. If distance < 15 cm:
11. Move backward and turn left
12. Display the stream with target to track is box-bounded
13. End

B. Architecture of Software System

Since the model displays indices of output layers of the network, image pre-processing is done by "dnn" module that is provided by OpenCV (opencv.org, 2022).

The blob is a 4-dimensional array function that input dimensions respectively, image, scale factor, 2-D input size width and height, color channels 3-tuple (Red, Green, Blue). The scale factor is the image dividing by 255 to result a flattened binary array. The tuple of color channels is filled with zeros since the image is binarized

Table 7 Object detection algorithm pseudocode

Algorithm 2 Real time object detection system pseudocode
1. Function: object detection
2. Input: image
3. Output: object detected with bounding box
4. Initialize real time stream
5. While True:
6. Read image
7. Convert image to binary scale
8. Resize input image to 320 x 320
9. for Index in unconnected Layers names:
10. Get name of layer
11. Detect objects in the frame
12. Display result on screen
13. if interrupted:
14. Break program
15. End

"Object detection" function, as shown in Table 7, that pass two attributes. Predictions and image, where predictions are the three output layers of YOLO CNN. Image is the acquiesced image from the stream. The algorithm target is to detect the object and displays it bounded with a rectangular box. The process of detection, box bounding and displaying frame are done using OpenCV library.

Table 8 Human detection function algorithm pseudocode

Algorithm 3 Human detection function pseudocode
1. Function: Find object
2. Input: Three predictions of the image, streaming frame
3. Output: A rectangle on the detected objects with labels and detection accuracy
4. For each prediction in predictions:
5. For each detected object in prediction:
6. Occupy the detection of the 6^{th} class "human class" only
7. Bound the detection with a bounding box
8. Display the bounding box of the maximum accuracy-appended detection
9. Thread object locating function
10. End

To track a specific object from the list of COCO dataset classes. Names of classes that the dataset contains are considered during this process. Person class is specified, which refers to the sixth class of coefficients in yolo models, as mentioned in YOLOv3 algorithms sub-section, first five indices are responsible of x, y, width, height, and confidence values, whereas classes start from 6^{th} index. In case the detected object is recorded with the highest confidence value, then it is bounded with a box. The process aims to avoid overlapping boxes, as illustrated in Table 8.

C. Target Tracking System

The tracking process depends on a sequence of processes and multiple factors, as illustrated in the following subsections.

1. Control System Architecture

a. Hardware architecture

The architecture of L298N chip with Raspberry Pi connections are shown in Figure 18 and connections of pins are clarified in Table 9 and IN1 and IN2 pins in L298N chip are responsible for the control power of DC motor A -in this project is the right DC motor- and ENA pin is to control the speed of DC motor A (computervision.zone,2022).

Table 10.

Table 9 Connections of motors, L298N's input pins and Raspberry Pi GPIO pins

L298N chip pins	Device connected to	Pins of device
Out1	DC motor A	Ground
Out2		Power
Out3	DC motor B	Ground
Out4		Power
GND	Raspberry Pi	Pin6 on Raspberry Pi
	Batteries	Ground
V12+	Batteries	Negative wire

IN1 and IN2 pins in L298N chip are responsible for the control power of DC motor A -in this project is the right DC motor- and ENA pin is to control the speed of DC motor A (computervision.zone,2022).

Table 10 Connections of motors, L298N's output pins to connected devices

Motor	L298N chip pins	Raspberry Pi pins (GPIO)
Motor A	ENA	Pin22 (GPIO25)
	In1	Pin18 (GPIO24)
	In2	Pin16 (GPIO23)
Motor B	ENB	Pin11 (GPIO17)
	In3	Pin15 (GPIO22)
	In4	Pin13 (GPIO27)

IN3 and IN4 pins in L298N chip are to control power of DC motor B -in this project is the left DC motor and ENB pin is to control the speed of DC motor B. Figure 18 clarified Outputs of L298N chip to Raspberry Pi -GND, power supply and DC motors (computervision.zone,2022).

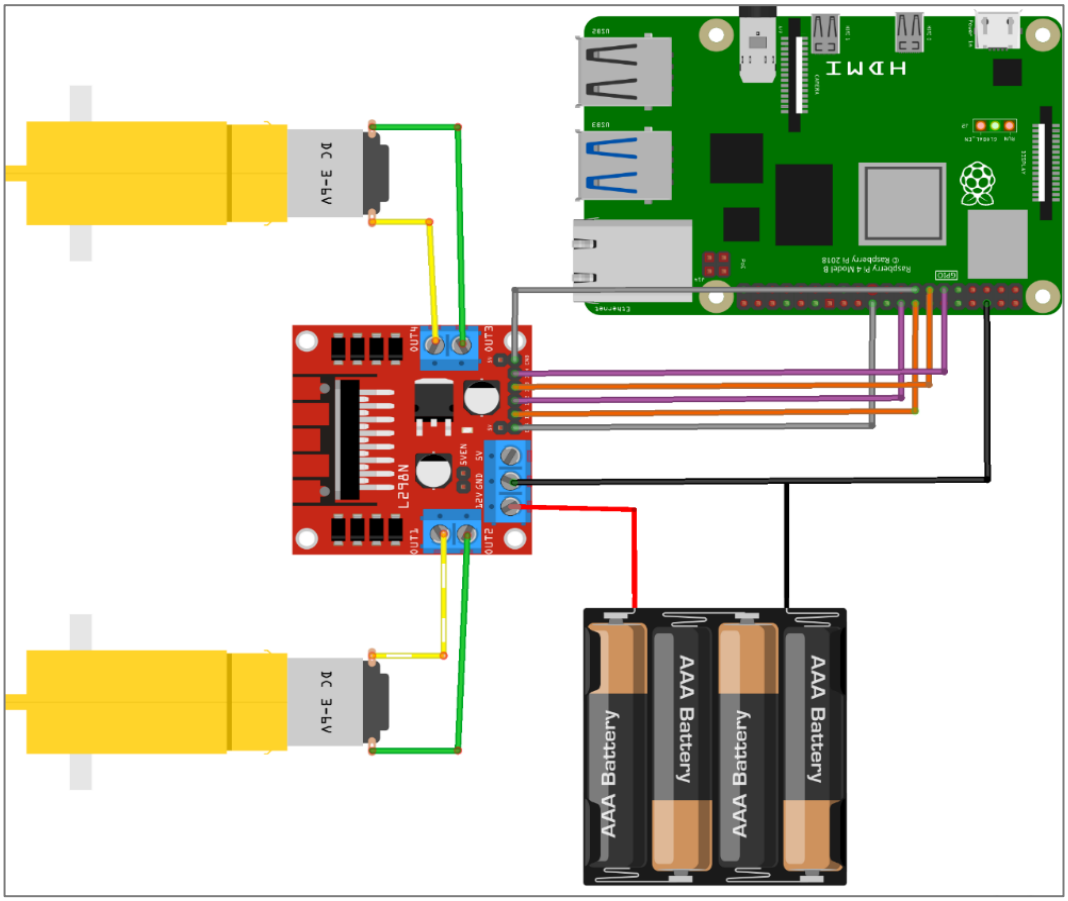


Figure 18 Raspberry Pi, L298N board, DC motor drivers and power supply I/O connections architecture diagram

b. Control commands adjustment

Table 11 Boolean logic terminals of H-bridge adjustment values

	Motor A		Motor B	
Command	Terminal	PWM	Terminal	PWM
Forward	1	100	0	100
Left	1	80	0	100
Right	1	100	0	80
Backward	0	100	1	100
Stop	0	0	0	0

Adjustment of H-bridge regulator is required for setting up DC motors. The process is done through using L298N regulator, that is connected to Raspberry Pi and DC motors, as shown in Table 9. For instance, In1 and in3 are responsible of moving forward (clockwise), whereas In2 and in4 are responsible of moving backward (anti-clockwise), as illustrated in Table 11.

2. Locomotion System

The system of locomotion is based on multiple factors, commands, target location and delay. Each factor is handled according to multiple dependent algorithms and equations.

a. Frame segmentation

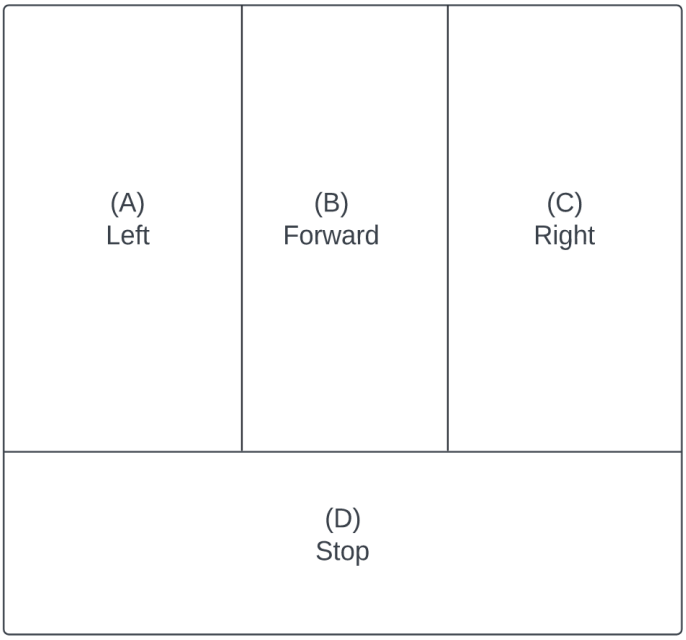


Figure 19 Frame segmentation for four segments. Respectively, (A), (B), (C) and (D) segments with Left, Forward, Right and Stop commands

Researchers in (Bruce, 2001) introduced an annotated, fast, and low-cost frame plane segmentation method for interactive systems. The method states that the plane can be divided into more than one segment in aim to simulate the visual interaction system in humans. As shown in Figure 19, the frame plane is segmented for four segments each segment is appended with a locomotion command. As illustrated in Figure 20, when the system locates the target on the segment (A), the command sent is left etc.

b. Locomotion algorithm

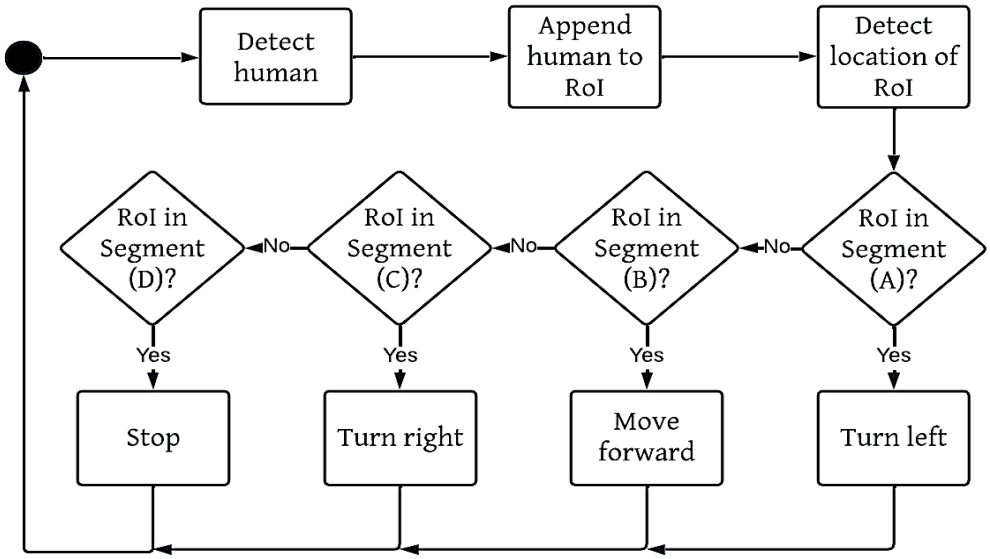


Figure 20 Flowchart of object tracking locomotion commands regarding position of object.

The locomotion algorithm depends on the plane segmentation of the frame, as illustrated in the previous subsection. The location of the target to track is the main factor in this algorithm. Figure 20 indicates the flowchart of the conditions of command based on segment located object to track. In case the object is in segment (A), the control command process state is turning the robot left. In case the object is in segment (B), the control command process state is moving the robot forward. In case the object is in segment (C), control command process state is turning the robot right. In case the object is in segment (D), the control command process state is stopping the robot because the object is near to the UGV. Threshold values of the segmented lines are defined as tolerance for (B) area, Z value for (D) area, positive α_x value for (C) area and negative α_x value for (A) area, as detailed in following subsections.

c. Equations of UGV locomotion

Center of frame is extracted by following coordinate point equation. In coding format: (cxf,cyf)=(img.shape[1]//2,img.shape[0]//2) since shape[0] is height and shape[1] is width.

$$(C_x, C_y) = \left(\frac{h_f}{2}, \frac{w_f}{2}\right)$$

Equation 1

The followed principle of tracking is the object centralization in front of webcam and keep it the center. The center changes regarding the object movement, and consequently the car moves according to the object's center. The object is being tracked must be brought inside this zone while tracking.

The algorithm of tracking is shown below. First, we calculate the difference between max and min of x and y separately where x_{min}, x_{max}, y_{min} and y_{max} refer to the axis's points of the object frame.

$$\Delta x = x_{max} - x_{min}$$

Equation 2

$$\Delta y = y_{max} - y_{min}$$

Equation 3

The changes on movements are employed to extract the center location of RoI frame, where (Rx_o, Ry_o) refer to the coordinates of RoI area on the visual frame.

$$Rx_o = x_{min} + \frac{\Delta x}{2}$$

Equation 4

$$Ry_o = y_{min} + \frac{\Delta y}{2}$$

Equation 5

After extracting the center of the object, we calculate the deviation of that object. l_w refers to the central longitude of the frame. The measured distance between l_w and Rx_o is expressed by α_x.

$$l_w = \frac{w_f}{2}$$

Equation 6

$$\alpha_x = l_w - Rx_o$$

Equation 7

To calculate the area of segment (D) that corresponds to stop command, Z coefficient is computed, which refers to the area underneath the RoI. The threshold is specified to be only 20% of the frame bottom area. It means, in case of the target is in the bottom of the frame, bellow 80% of the frame Stop command must be sent to locomotion process.

$$Z = h_f - y_{max}$$

Equation 8

Z variable refers to the bottom distance between object frame and camera frame, as shown in Figure 21.

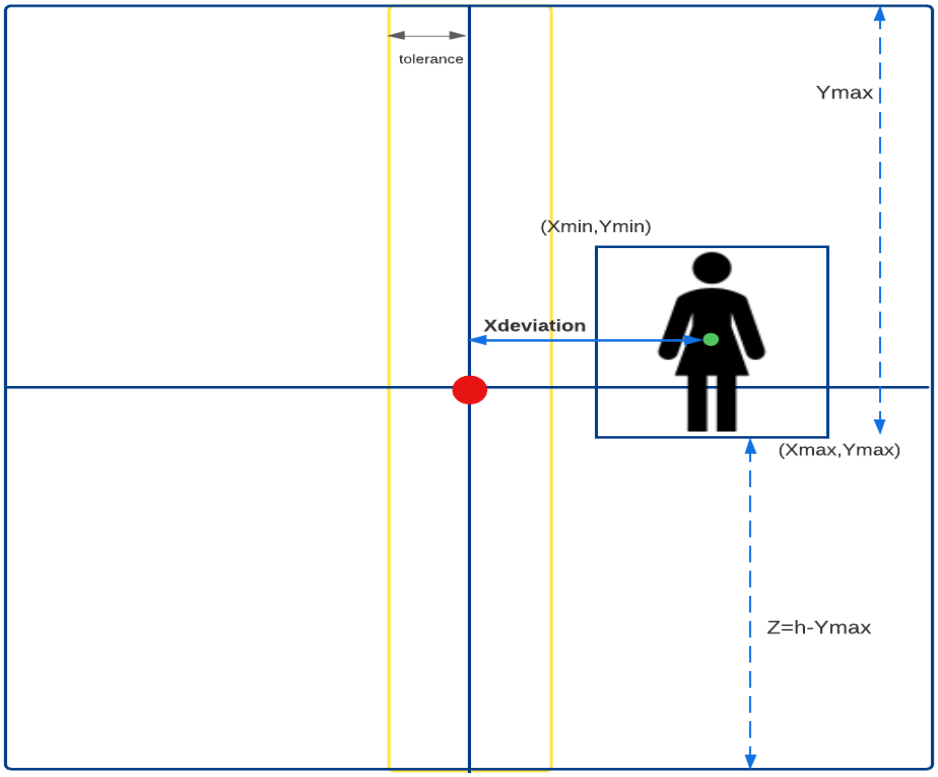


Figure 21 Frame with target and coefficients visualization

Table 12 Algorithm 3 Object locating algorithm pseudocode

Algorithm 4 Object locating pseudocode
1. Function: Object locating
2. Input: delayed time value, deviation value
3. Output: x, y coordinates location of RoI frame on the stream frame
4. Global: $x_{deviation}$, tolerance, y_{max}, delay
5. Locate the center of target to track RoI
6. Calculate x alpha value
7. Tolerance = 30% of the frame
8. Thread Robot locomotion function
9. End

Object tracking process is capsulated in one function named “Object locating”, as shown in Table 12. It has four global variables. The purpose of this function is to extract the location of the RoI center, tolerance, and deviation values. Finally, thread output into “Robot locomotion” function.

d. Delay issue

In Autonomous Robots, delay of response occurs customarily. There are diverse reasons behind the delay. One of the most recurrent reasons is the time consumed during transmission and response over a network. Another reason is the waiting of object-to-track to be detected (K. M. Hasan, 2012). To manage the induced delay, multiple techniques are improved (Slotine, 2004).

In the proposed system, a simple delay handle technique is utilized. The delay time value is executed regarding the threshold of α_x value as detailed in $d(\alpha_x)$ function.

$$d(\alpha_x) = \begin{cases} 0.8\,, & \alpha_x > 0.4w_f \\ 0.6, & 0.35w_f < \alpha_x < 0.4w_f \\ 0.4, & 0.35w_f < \alpha_x < 0.2w_f \\ 0.2\,, & \alpha_x > 0.2w_f \end{cases}$$

Equation 9

Table 13 Delay handle algorithm pseudocode

Algorithm 5 get delay function pseudocode
1. Function: get delay(deviation)
2. Input: deviation: deviation value from Object locating function
3. Output: delayed time value
4. Delayed time =0
5. Deviation = |deviation|
6. if deviation ≥ 4:
7. Delayed time = 80 s
8. else if (deviation ≥ 35 and deviation < 40):
9. Delayed time = 0.60 s
10. else if (deviation≥ 20 and deviation < 35):
11. Delayed time = 0.50 s
12. else:
13. Delayed time = 0.40 s
14. return Delayed time
15. End

The time between object detection and locomotion control commands reception induces observable delayed feedback of movements. "Get delay" function, as illustrated in Table 13, is a dependent function. "Get delay" function depends on the output of "Object locating" function. The function's purpose is to manage the occurred delay. "Get delay" function inserts a short interval of time based on $x_{deviation}$ value. When $x_{deviation}$ is large, delay value equals to a longer time interval. On the other hand, when $x_{deviation}$ is small, delay value equals to a smaller time interval.

e. Locomotion integrated system

Robot locomotion orders process is incapsulated in “Robot locomotion” function, as shown in Table 14. “Robot locomotion” function depends on “object locating” function and “get delay” function.

Table 14 Robot locomotion algorithm pseudocode

Algorithm 6 Robot locomotion function pseudo code

```
Function: locomotion
Input: Object locating output, human detection output.
Output: locomotion commands
Global: x_deviation, tolerance, y_max, delay
    z = 20% # distance from bottom of the frame
    If target does not exist:
        Delay
        print “No object exists”
    else if target exists:
        If | x deviation | < tolerance:
            If z >= 20%:
                Stop command
                print “Stop Object is reached”
            else:
                Delay = Get delay(x deviation)
                Forward command
                Delay
        else if (|x deviation | > tolerance):
            delay = get delay (x deviation)
            Command left
            Delay
        else if x deviation ≤ − tolerance:
            Delay = get delay (x deviation)
            Command right
            Delay
End
```

“Robot locomotion” algorithm is detailed in flow chart in **Error! Not a valid bookmark self-reference.**, which is only a different visualization of the algorithm in Table 14. When the camera views an object then the existence equals to 1 whence the algorithm starts. The tolerance is the area of segment (B) as explained previously. When alpha x value or as named in the code, x deviation, is greater than tolerance area then the left command is activated. When x deviation is smaller than negative tolerance then right command is activated. When x deviation is smaller than tolerance, two conditions are faced. When the area from the bottom of the frame, or as named in the code, z variable, equals or less than 20%, then stop command is activated. Otherwise, forward command is activated.

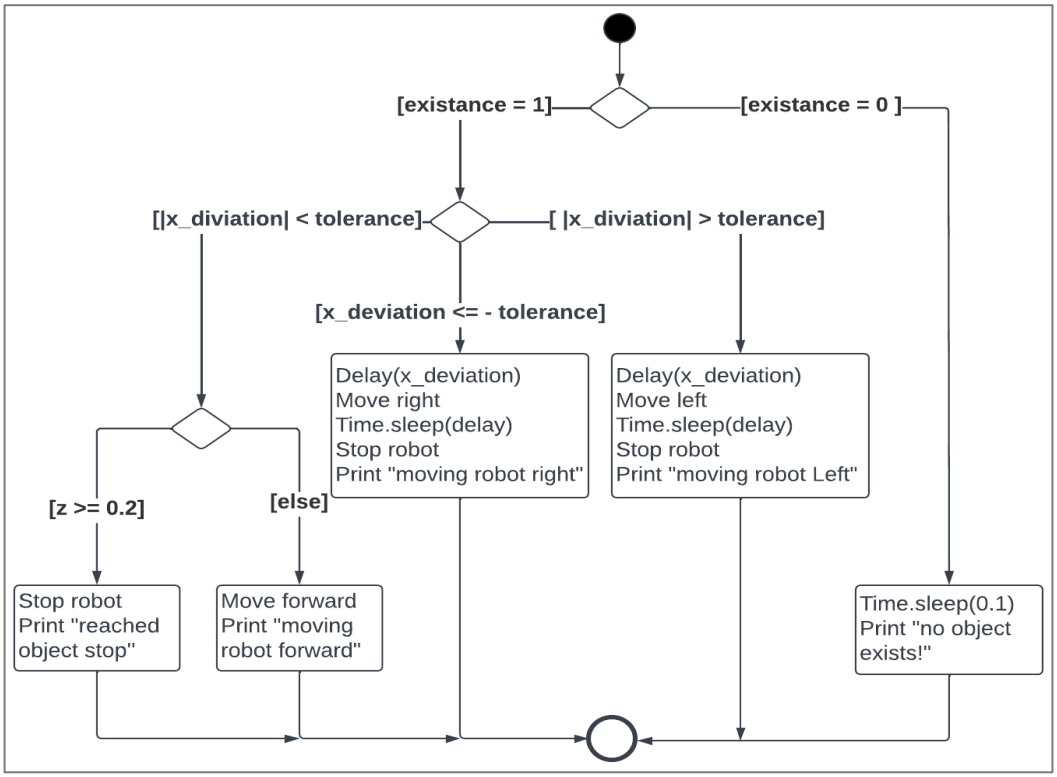


Figure 22 Movement algorithm of UGV locomotion system flowchart diagram

D. Obstacle Avoidance System

Collision problem is potential in unmanned systems. Multiple techniques are applied to avoiding obstacles. Ultrasound emission is an effective, low-cost, rapid, and precise technique (Everett, 1989). The basic concept of ultrasonic sensor is to emit ultrasonic waves and receive it back. Ultra-sonic sensor HC-SR04 board, as shown in Figure 5, is utilized, which is a compliant, simple, and efficient board to be used with Raspberry Pi board (thepihut.com, 2022).

1. Hardware System Architecture

Before implementing the system of hardware boards, voltage regulation is required to avoid the damage of the Raspberry Pi board. The resistance is measured according to simple physical computations based on Ohm's rule, as shown in Equation 10, Equation 11,Equation 12, Equation 13. Three 1k Ω resistors are utilized, R_1, R_2 and R_3 .The desired output voltage is 3.3V thence, V_{out} refers to the target voltage, which is 3.3, where V_{in} refers to the voltage emitted by the Echo pin (circuitdigest.com, 2022).

$$V_{out} = V_{in} \times \frac{R_2+R_3}{R_1+R_2+R_3}$$

Equation 10

$$V_{out} = 5V \times \frac{2k\,\Omega}{3k\,\Omega} = 3.3V$$

Equation 11

The usage of two 1k resistors induces the required resistance value ohm, as illustrated in Equation 12 and

Equation 13.

$$R_2 + R_3 = V_{out} \times \frac{R_1}{v_{in}-v_{out}}$$

Equation 12

$$R_2 + R_3 = 3.3v \times \frac{1k\,\Omega}{5v-3.3v} = 1.94118 \approx 2k$$

Equation 13

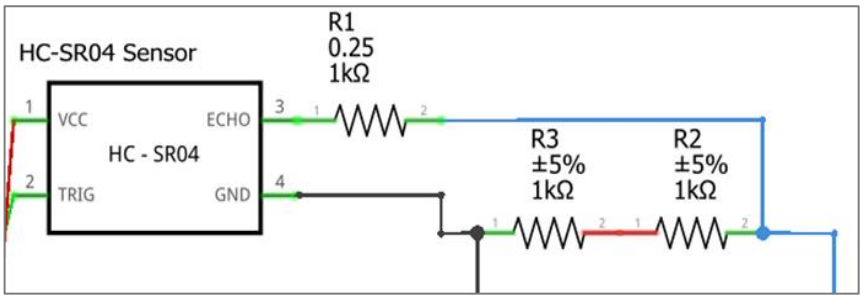


Figure 23 Schematic diagram of the HC-SR04 board, Raspberry pi and resistors

The scheme of resistors connection is illustrated in Figure 23. After the electric circuit is prepared, as shown in Figure 24, Echo pin is connected to GPIO11, whereas trigger pin is connected to GPIO13 on the Raspberry Pi.

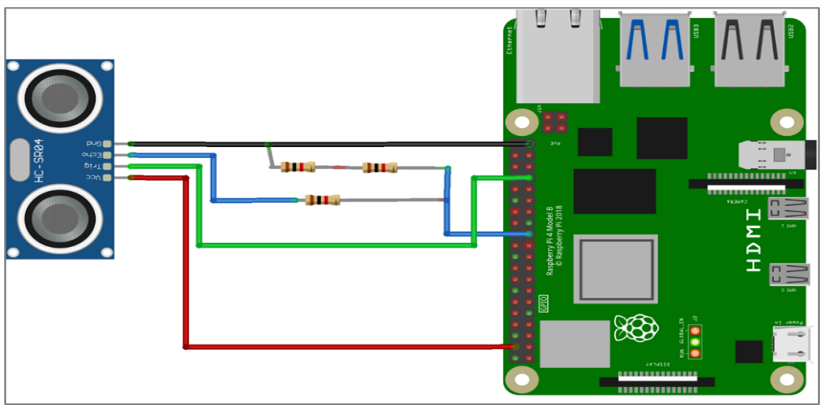


Figure 24 Obstacle avoidance system structure diagram declaring connections between raspberry pi pins and HC-SR04 ultrasonic sensor board's pins.

2. Distance measurement

The distance is measured using the recorded value, as shown in

Equation 14, where v_{sound} refers to the speed of sound in air, which is a constant that equals to $343\, m/s$. t refers to the time of ultrasonic wave broadcasting and reception travel.

$$d_O = v_{sound} \times \frac{t}{2}$$

Equation 14

V. ROBOTIC EYE-BASED FACE TRACKING SYSTEM

In reference (Seong-Hoon,2019), the Pan Tilt Zoom system is experimented, which is a fixed robotic eye that track a person and detect his position in real-time. In the proposed system, the face locating, and tracking is implemented. First, the hardware architecture is composed using the following means, respectively, 2 x servos, Pan/Tilt, PWM driver board, and Pi camera board, as shown in Figure 8. However, two algorithms are used. A simple algorithm of face locating using pre-built model and an algorithm for locomotion commands control based on trigonometry rules.

Required libraries to interact with board, as illustrated in Table 4, “RPI.GPIO” library, “time” library and “adafruit_servokit” library, which is specifically provided for PCA9687 board. “RPI.GPIO” library is required for the integration with GPIOs pins in raspberry pi. “time” library that is used to control the to handle latency of locomotion. The proposed system is influenced by (pyimagesearch.com,2019).

A. Pan/tilt Micro Servos Hardware Architecture

The connection of the system is shown in Table 15 and Table 16 and visualized in

Figure 25. The input pins are the first part pins of PCA9685 board, which are responsible of transmitting electronic signals across the connected wires to Raspberry pi SBC.

Table 15 GPIOs pins from Raspberry Pi for PCA9687 input pins clarification

Raspberry Pi GPIOs pins	PCA9687 chip input pins
Pin1 3v3	Vcc pin
Pin2 5v	5v pin
Pin3 SDA GPIO	SDA pin
Pin5 SCL GPIO	SCL pin
Pin6 GND	GND pin

Table 15 illustrates the connections between PCA9685 board and Raspberry Pi SBC (core-electronics.com.au, 2022).

The output terminals are executed on servos. Output pins are VCC, PWM and GND pins. PCA9685 board includes 16 channels, which means, there are 16 output pins that can be plugged to 16 servos. Two channels are occupied, channel 0 for the pan servo and channel 1 for the tilt servo as illustrated in Table 16.

Table 16 Output pins from PCA9687 chip for servo motors

PCA9687 chip channel and pins		Servo Motor
Channel 0	GND PWM Power	Pan servo (Bottom)
Channel 1	GND PWM Power	Tilt servo (Upper)

The complete connections architecture of robotic eye system, as shown in

Figure 25. Pi camera is attached with tilt but not connected, whereas the rest of connections are electrically connected with PCA9687 board.

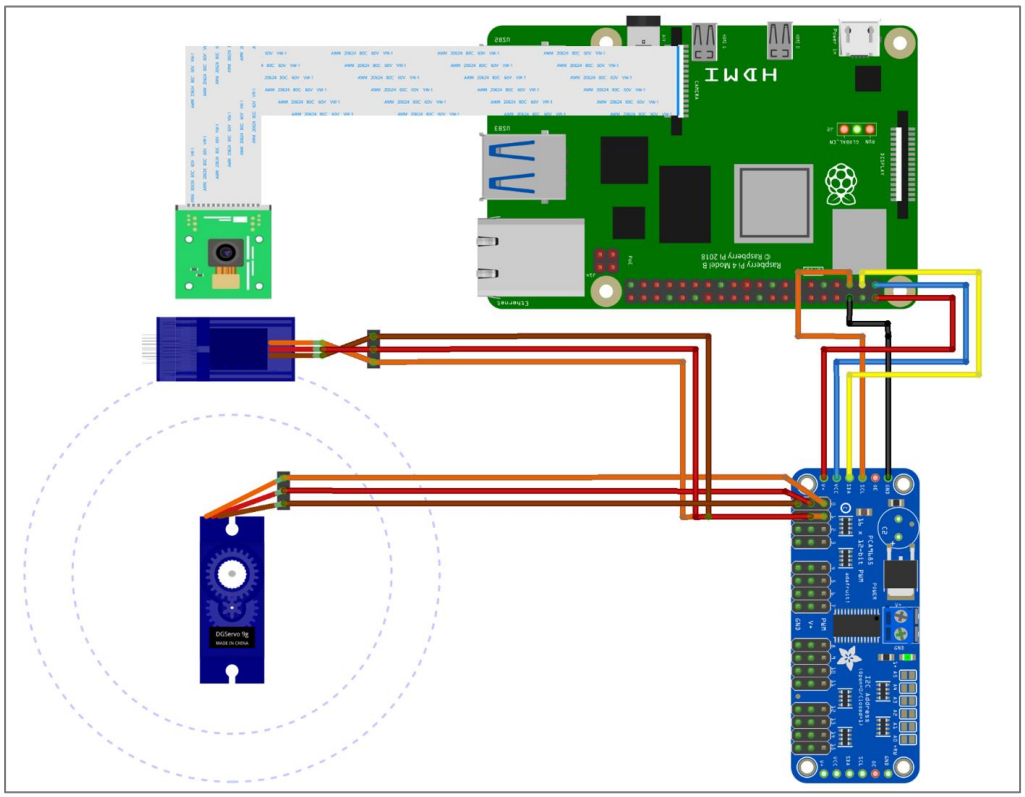


Figure 25 Raspberry Pi, PCA 9687 PWM chip, SG90 servos and camera diagram connections architecture diagram

The PCA9687 board is used to avoid the complication and to receive more accurate movement feedback. A prototype system is experimented without the usage of PCA9687, as shown in **App-3**.

B. Face Tracking System

The system of robotic eye is divided for two stages. The stage of face detection and the stage of face tracking. As shown in figure 22. First, the image is acquiesced, then a consequence of pre-processing processes is done. The image is resized to a smaller frame, converted to greyscale to be ready to be fitted into the cascade. The detection process starts from the fifth state in the flowchart, as shown in Figure 26, whereas the pre-processed image is inserted into the cascade, the largest face is detected and bounded with a rectangular box.

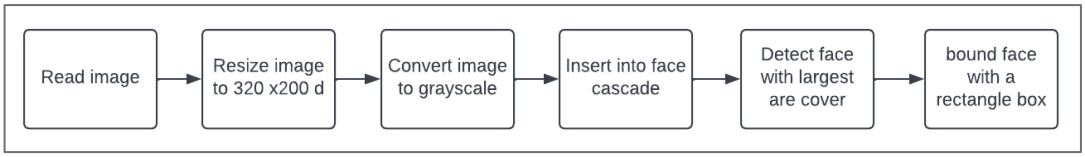


Figure 26 Flowchart of face detection stage in robotic eye-based face tracking system

The process of tracking the face depends on half cycle rotation in two servos. Tilt is the base whereas pan is the upper holder that holds the camera. The tracking algorithm, as shown in Algorithm 6, after defining variables and calling face cascade model, pan is the servo that is connected on 0 channel and tilt is the servo that is connected on 1 channel. Pan is based on the x-axis coordinate system, whereas tilt is based on the y-axis coordinate system. As shown in Equation 16 and Equation 17, x-center RoI and y-center of RoI are extracted. Latitude l_f and longitude l_h of the frame are expressed in Equation 15, Equation 16 and Equation 17.

$$l_h = \frac{w_f}{2}$$

Equation 15

$$\alpha_{pan} = Rx_o - l_f$$

Equation 16

$$\alpha_{tilt} = Ry_o - l_h$$

Equation 17

The flowchart of robotic eye-based face tracking system, as shown in Figure 27, starts with setting up the pan and tilt angles followed by detection the face to track, followed by the defining the error values, as illustrated in Equation 16 and Equation 17 then resetting the angles of pan and tilt according to a long series of arithmetical conditions. The conditions are processed to preserve the range of movement of the servos, where the range is between 1 to 180 angles for each servo, either pan or tilt. In case of lack of the conditions, the system would be fragile and less interactive since the miss of feedback to user and the disability of the robot to execute the received commands. For example, in case the face moved toward one direction on latitude line, in one point, the value of pan would be 181, hence the system would hang out, the servo would burn off and the user would not know what to behave. However, the sequence of arithmetical condition is the simplest technique to avoid any overlapping situation and to preserve the simple computation cost.

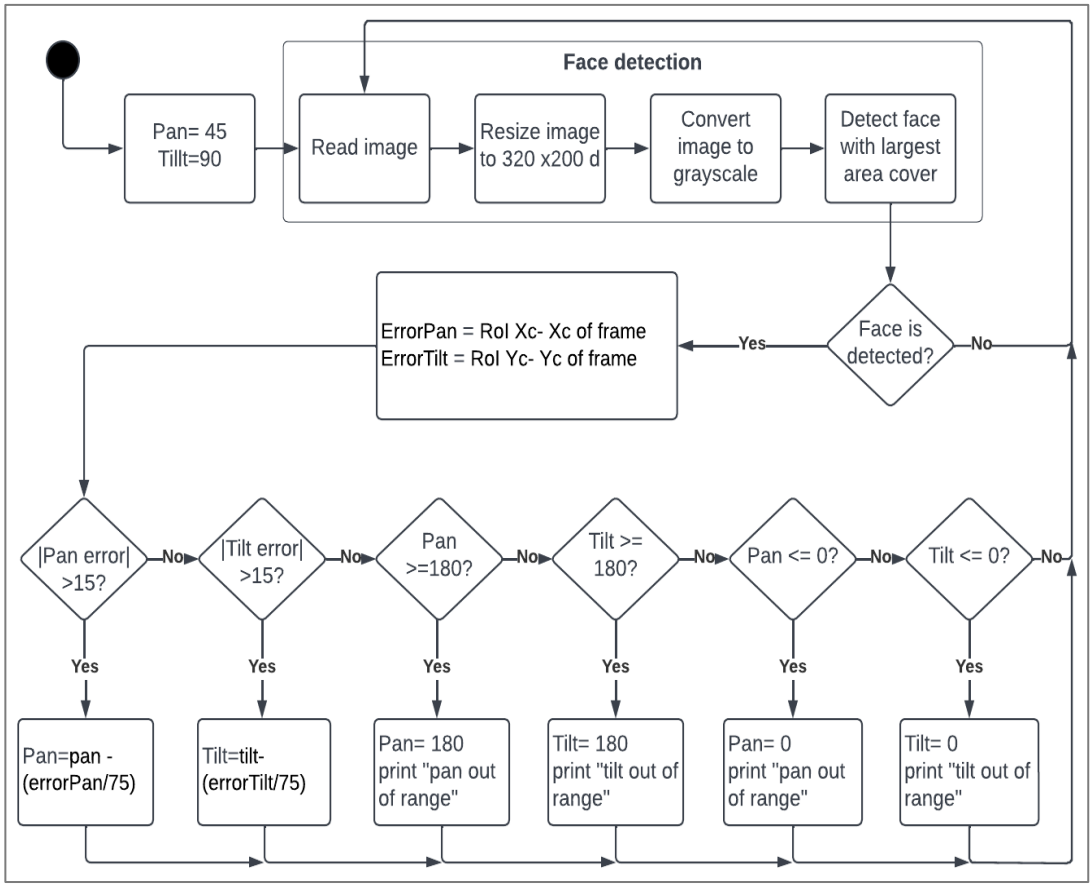


Figure 27 Flowchart of robotic eye-based face tracking system

The algorithm of the system is incapsulated in one code, as shown in pseudocode in Table 17, the input for the algorithm is the stream frame that is displayed using

OpenCV library features. The output of the algorithm is the detected face to be tracked and the locomotion terminals that directly sent to PWM circuit to be executed. The feedback of the system is observed on the actual robot prototype, as detailed in RESULTS section.

Table 17 Robotic eye-based Face detection and tracking pseudocode Algorithm

Algorithm 7 Face detection and tracking pseudocode
1. Input: Real-time video stream.
2. Output: Face to track detected and pan/tilt locomotion updated terminals.
3. Read image
4. Resize image to 320 x 200 d
5. Convert image to grayscale
6. Fit image into face cascade
7. Detect face with threshold 1.3
8. Display the detected face with a bounding box
9. Locate the center of detected face (x, y)
10. Pan error= $X_{\text{object center}}$ -width of frame/2
11. Tilt error= $Y_{\text{object center}}$ -height of frame/2
12. if | Pan error | > 15:
13. Pan = pan - Pan error /75
14. if | Tilt error | > 15:
15. Tilt = tilt - Tilt error /75
16. if pan > 179:
17. Pan = 179
18. print “Tilt Out of Range”
19. if pan < 1:
20. Pan = 1
21. print “Tilt Out of Range”
22. if tilt > 179:
23. Tilt = 179
24. print “Tilt Out of Range”
25. if tilt < 1:
26. Tilt = 1
27. print “Tilt Out of Range”
28. In Servo [0] angle = pan
29. In Servo [1] angle = tilt
30. Resize stream frame to larger dimensions
31. Show image with face bounded with a box
32. End

VI. RESULTS

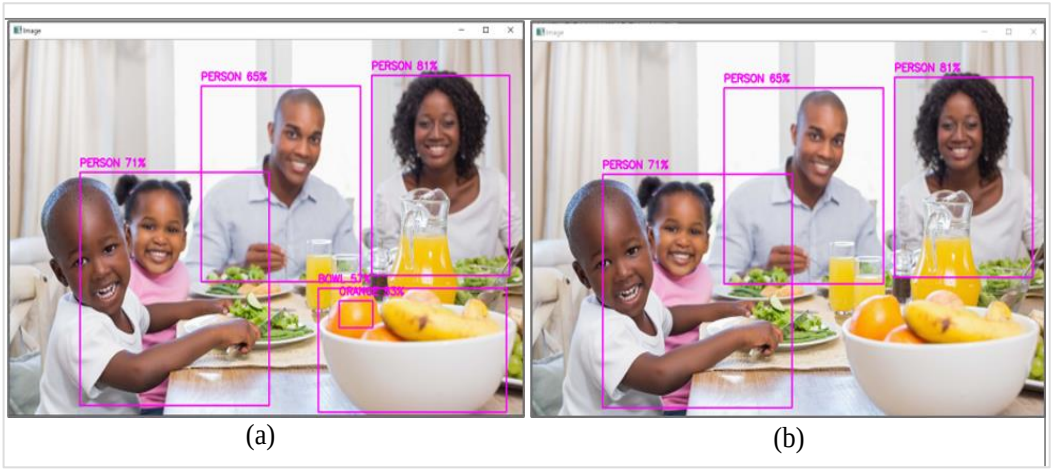
Innovative measurement methods are followed in aim to display the feedback in rational, arithmetical, and accurate view. The results, measurements and comparisons are divided into multiple subsections to display a better view of the study feedbacks.

A. Deep learning-based Object Detection

YOLOv3-tiny model computation low-cost, that makes it convenient and affordable in this project. As shown in Figure 28, accuracy of the detected objects using YOLOv3-tiny model equals to 74% in real time stream. The center of the frame is the yellow point, which is the construction of latitude and longitude of the frame, that are observed by the blue lines, whereas the tolerance, that equals to 30% of the frame, is observed by the two green lines.

Figure 28 Clear execution of software that shows boundaries of tolerance, center of frame and center of object

In the target to track class specification stage, the human class is chosen. An image of a family while they having a meal is tested by YOLOv3-tiny model. Before and after of specifying human to be tracked comparison is constructed. The results are observed in Figure 29. As shown in Figure 29 (a), the number of objects detected before the class specification is five objects. It is remarkable to mention, even small objects are detected such as the orange with accuracy of 55% and the bowl of fruits with accuracy of 53%. After the object-to-detect to be human only is specified, as observed in Figure 29 (b), only human class objects are detected. Though only three humans out of four are detected, it is considered as an acceptable detection ratio since the usage of the light model.



(a) (b)

Figure 29 Object detection result. (a) The frame of OpenCV before the definite of class specifying. (b) The frame of OpenCV after specifying the person class for detection.

B. UGV-based Human Tracking

Performance of UGV is remarked better above a flattened roof ground compared with a roof covered by a rug. The friction force value is higher while moving on a rug comparing to moving on a smoothed surface.

Three tests of UGV robot are experimented, as shown in Table 18. Three factors are recorded from the experiments. and c values are recorded: α_x value, distance between UGV and object as well as the direction of the route that is followed by UGV robot. The records are taken based on an interval of time in range of three seconds. As in illustrated in Table 18, the standard deviation values of the three tests are, respectively, 9.0, 9.1 and 11.4, which evince that the best recorded test is test 2. by the

same method, distance values, that are recorded from the ultrasonic sensor, indicates the stability of route and the fast feedback of commands.

Table 18 Three tests of UGV tracking with recorded values

t (s)	Test 1			Test 2			Test 3		
	α_x	S_d	c	α_x	S_d	c	α_x	S_d	c
0.0	68.0	30.3	F	101.5	83.5	S	20.0	47.7	F
0.5	55.5	27.3	L	107.0	83.5	F	41.0	42.4	R
1.0	68.5	34.1	F	120.5	76.0	F	43.0	37.5	R
1.5	69.5	31.5	F	123.5	72.0	F	49.0	28.0	R
2.0	54.0	20.1	L	124.5	67.0	F	52.5	22.7	R
2.5	55.0	29.5	L	122.5	61.0	F	50.0	02.3	B
3.0	77.0	38.0	S	121.0	64.0	F	33.0	45.0	L
$\bar{x}$	**63.9**	**30.1**	-	**117.2**	**72.4**	-	**41.2**	**32.2**	-
σ	**9.0**	**5.6**	-	**09.1**	**09.0**	-	**11.4**	**16.0**	-

For the test of obstacle avoidance technique, as remarked in test 3 in time interval, 2-3 seconds, the distance was 2.3 cm then it becomes 45 cm, and the direction of route is toward back until the object is avoided from the route of the obstacle. S_d refers to the distance from the opposite obstacle whence c refers to the real direction movement of the UGV.

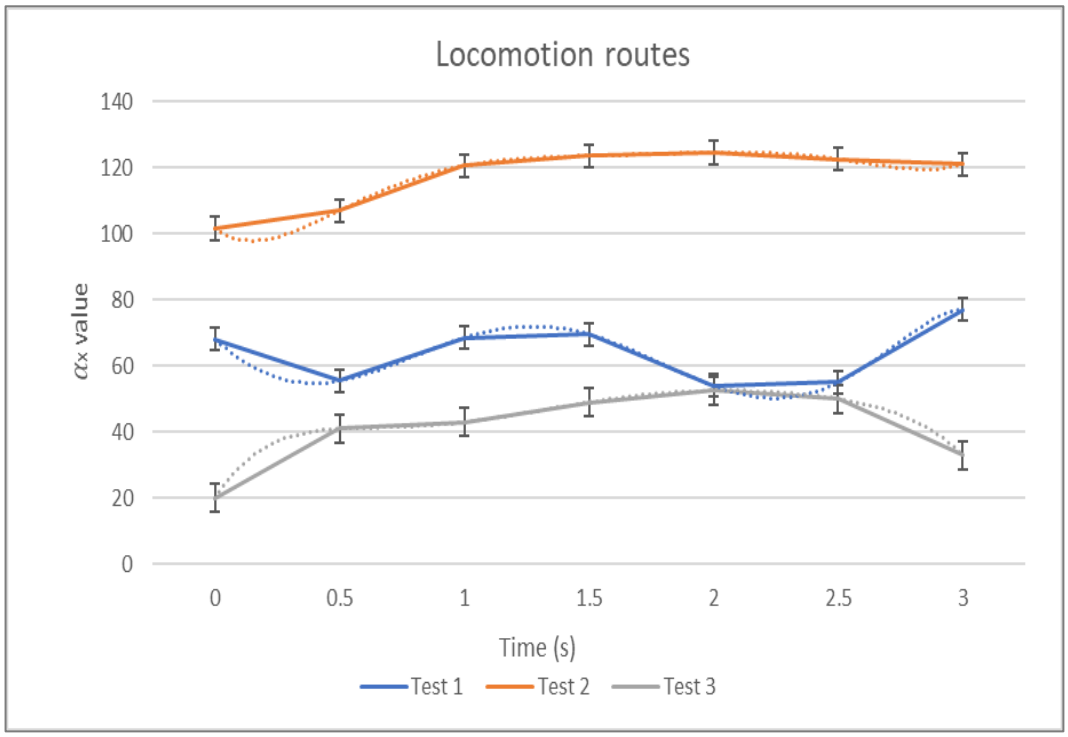


Figure 30 Scatter chart of test 1, test 2 and test 3 of tracking process values over an interval of time with error rate and trend line observation

Figure 30 indicates α_x values of the three tests over six periods of time. according to records in Table 18, the routes are measured with polynomial trendlines to indicate

the neat route of the UGV. Error bars measurement is utilized as well. According to Table 18, the best route is clear in test 2 (the orange series) since the standard deviation that belongs to test 2 is the smallest value, followed by the route of test 1 followed by the route of test 3.

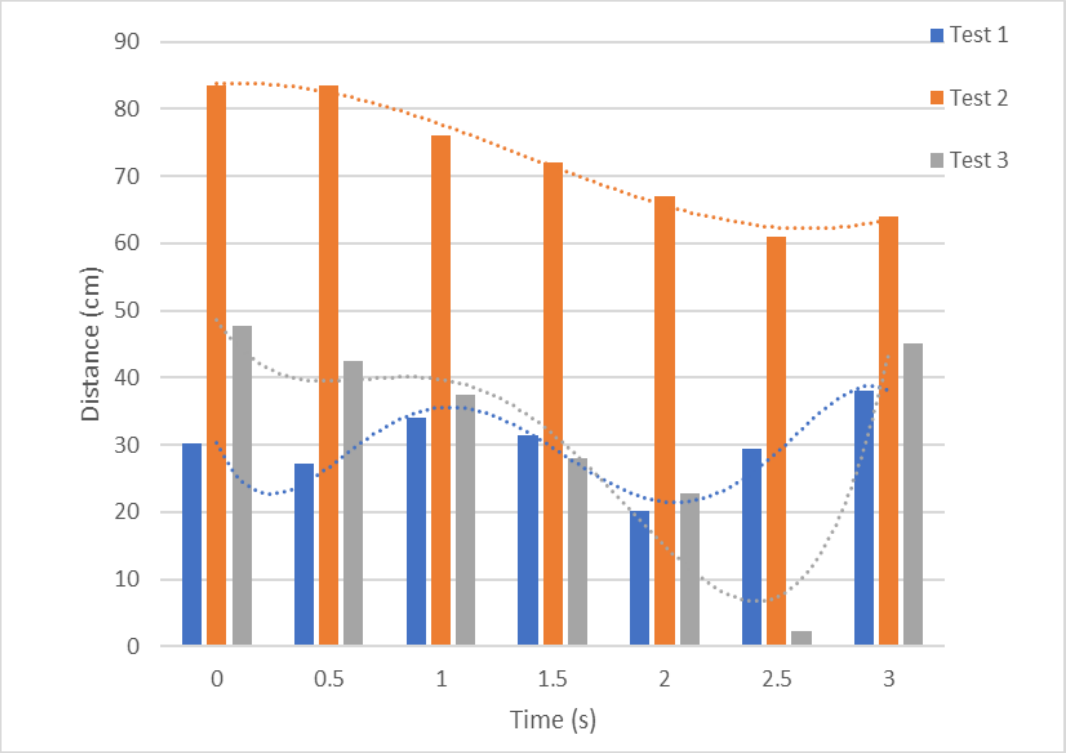


Figure 31 Bar chart of test 1, test 2 and test 3 of obstacle farness distance values over an interval of time.

Figure 31 shows the bar chart of recorded distance values from the ultrasonic sensor. The chart indicates the stability of distance farness from the object along the tracking process. The best emulated route is clear in test 1 followed by test 2 followed by test 3. It is valuable to mention, at the last two time periods, a noticeable outlier is occurred. The outliers refer to the obstacle confrontation and the feedback to take another route in aim to avoid that obstacle

C. Robotic Eye-Based Face Tracking

The results of robotic eye-based face tracking are exquisite, accurate and rapidly feeding back. FPS rate is approximately equals to 20 FPS. Two tests are recorded of the system execution. As shown in Table 19, the first test refers to a young man, whereas the second test refers to an old man.

Table 19 Recorded values of two tests of robotic eye-based face tracking

t (s)	Test 1 (young man)			Test 2 (old man)		
	α_{pan}	α_{tilt}	(Pan, tilt) coordinates	α_{pan}	α_{tilt}	(Pan, tilt) coordinates
0.25	31.5	-32.5	76.713, 145.353	-7	-48	20.413, 105.533
0.50	33	-27	75.386, 144.186	-8.5	-52.5	20.413, 104.113
0.75	40.5	-28.5	74.339, 143.433	-7.5	-51.5	20.413, 101.879
1.0	51.5	-35.5	72.419, 142.059	-7	-54	20.413, 100.486
1.25	57.5	-40.5	70.933, 140.999	-6	-58	20.413, 98.219
1.50	52	-26	68.953, 139.966	-4	-61	20.413, 95.806
1.75	47.5	-20.5	67.653, 139.359	-3	-70	20.413, 94.006
2.00	65.5	-22.5	65.246,138.486	-15	-72	20.413, 92.039
2.25	70.5	-19.5	63.386, 137.999	-25	-76	20.979, 90.033
2.50	78	-13	60.466, 137.999	-37.5	-57.5	22.346, 87.413
2.75	77.5	-18.5	58.413, 137.546	-45	-53	23.486, 86.006
3.00	90.5	-25.5	54.966, 136.579	-57	-39	24.913, 84.739
3.25	75.5	-18.5	52.799, 136.079	-65	-41	26.626, 83.666
3.50	76.5	-11.5	49.753, 135.626	-79	-37	29.573, 82.119
3.75	76	-11	47.713, 135.626	-86.5	-32.5	31.779, 81.259
4.00	75.5	12.5	46.639, 135.626	-86	-31	35.193, 79.979
$\bar{x}$	**62.43**	**-21.13**	-	**-33.69**	**-52.13**	-
σ	**17.38**	**11.87**	-	**30.62**	**13.24**	-

Figure 32 shows the deviation values α_x and α_y, where α_x refers to the x deviation and α_y refers to y deviation. The deviation values express the face detection and movement tracing process. As observed in

Figure 32, the face in test 1 was located down and left, whereas the face in test 2 it located down at the first, then it went toward the bottom and right.

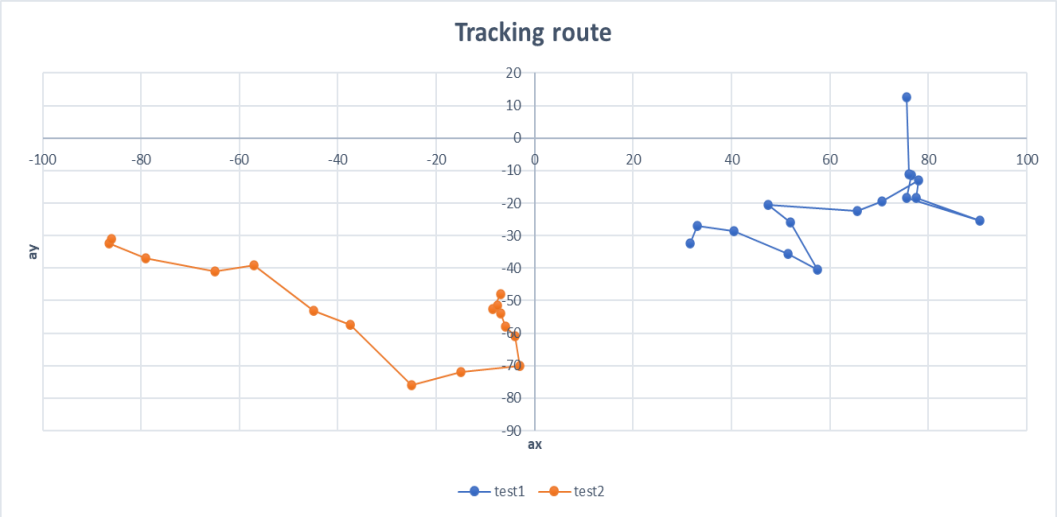


Figure 32 Tracking route chart of robotic eye-based face tracking according to pan and tilt alpha values of two tests

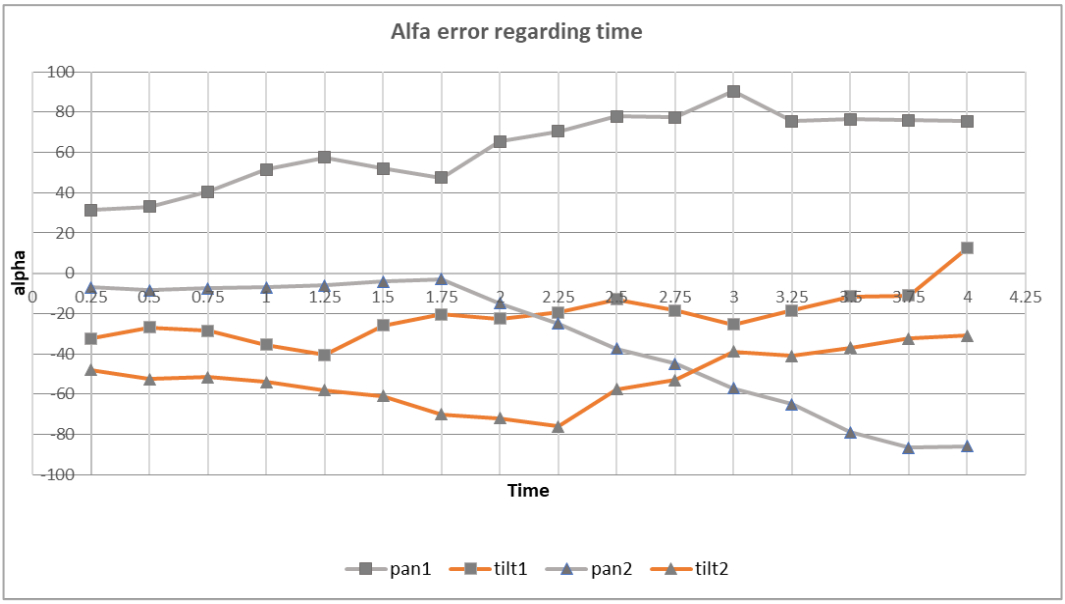


Figure 33 Alpha error of pan and tilt chart of robotic eye-based face tracking in two tests

The standard deviations of first test records are better comparing to records of test 2. In test 2, standard deviation values are high since the occurred outliers in pan error rate. The lines of error rates are illustrated in Figure 33. The smooth of lines of error rates is observed, which is proved by the standard deviation values.

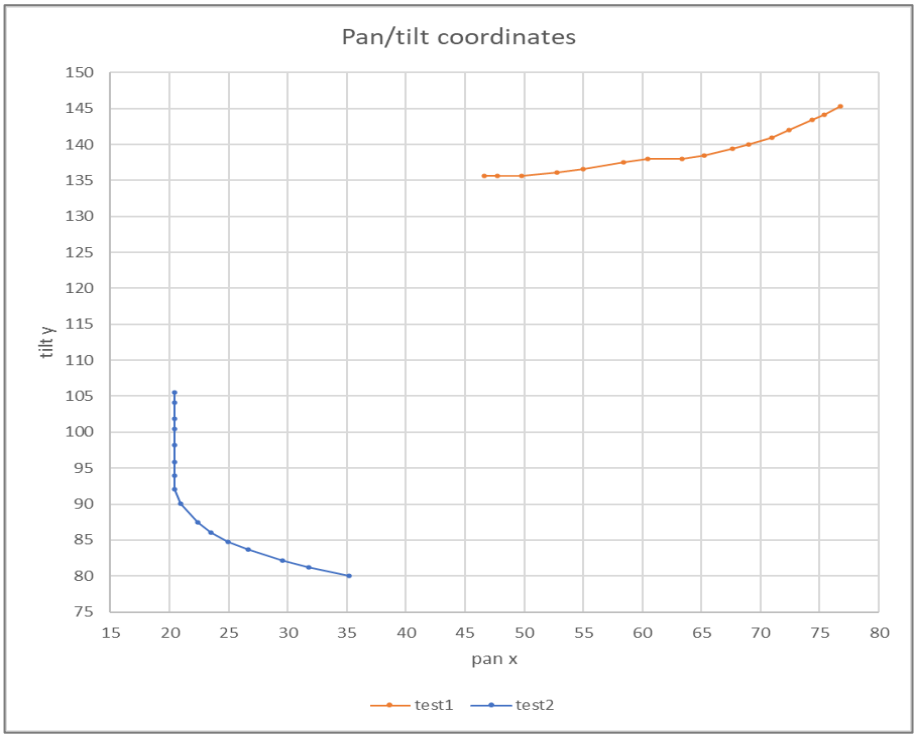


Figure 34 Locomotion coordinates chart of robotic eye-based face tracking regarding two tests

Figure 34 emulates the coordinates of 3-Dimentional locomotion route of both test 1 and test 2. The route of test 1 deduces that the robotic eye first was looking up then decreasingly looking down, whereas in test 2 the route deduces the robotic was looking down then increasingly looking up. The pan coordinate location is not accurate since the values are absolute.

D. Challenges

There are multiple challenges faced during this study. More than one limitation is remarked, regarding the computation cost of the algorithm. The lack of some electronic boards such as accelerator, which makes the Robot's speed higher by multiple times. Pre-built model YOLOv3-tiny heavy load on the SCB, which induces a slowness in FPS rate and late response of locomotion commands. The system detects more than one target at the same time since the target-to-track is not authorized, which produces overlapping, thence it deduces system hanging.

VII. CONCLUSION

In this study, a prototype of object tracking via UGV system is experimented. The system is vision-based target tracking. Image processing stage performed using OpenCV library. YOLOv3-tiny model is employed in object detection stage. pre-trained model based on COCO dataset. Object to track is specified to be a human only. Accuracy of objects detected are between 65% and 99%. Various techniques are used over the stages of the system. Keyword processes are acknowledged in this study are obstacle avoidance, frame segmentation, autonomous control commands and response-delay problem handling. Algorithm of locomotion is based on plane segmentation method and a sequence of equations and arithmetical conditions. Measurements of locomotion values are based on recorded real experimented tests on the UGV during tracking a person. Over multiple tests, an acceptable standard deviation is observed. Frames per second rate is floored to 2 FPS only. The results of UGV tests shows flexibility and robustness. In robotic eye-based face tracking system, pan/tilt means are used. Haar-like cascade is used for face detection phase. Frame Per Second rate is approximately 18 FPS. The results of Robotic eye show accuracy, speed, and robustness.

Multiple challenges are faced in this project, the lack of some electronic boards. Difference of versions considered as a challenge in creating these kinds of projects, thence each update may ruin the entire software system. Results are acceptable since the fair technical possibilities. In future work, the SCB board is going to be improved to handle more complex algorithms and to manage the delay more accurately. A different camera type is going to be attached with the ability to view in the dark view. Better-performed power supply is going to be utilized rather than simple AA batteries. The system and the subsystem could be improved to be an integrated singular complex system. The target-to-track is going to be authorized by training the images of users through the pre-built model to avoid the overlap, to increase the proficiency of tracking and to increase the speed of process.

VIII. BIBIOLGRAPHY

ARTICLES

ADARSH P., Rathi P., et al. (2020). "YOLO v3-Tiny: Object Detection and Recognition using one stage improved model", **6th International Conference on Advanced Computing and Communication Systems (ICACCS)**. pp. . 687-694. Coimbatore.

BALAJI S. R., Karthikeyan, S. (2017). "A survey on moving object tracking using image processing", **11th International Conference on Intelligent Systems and Control (ISCO)**. pp. 469-474.

BRUCE J. (2001). "Fast and Inexpensive Color Image Segmentation for Interactive Robots", **IEEE 2001 International Conference on Intelligent Robots and Systems**, pp. 2061-2066.

CHATTERJEE S., ZUNJANI F., et al. (2020). "Real-time object detection and recognition on low-compute humanoid robots using deep learning", **International Conference on Control, Automation and Robotics**. volume 6, pp. 202-208. Prayagraj

CHEN H., WANG X., et al. (2009). "A survey of Autonomous Control for UAV", **2009 International Conference on Artificial Intelligence and Computational Intelligence.** pp. 267-271. Shanghai.

EVERETT C. H., "Survey of collision avoidance and ranging sensors for mobile robots, Robotics and Autonomous Systems", **Robotics and Autonomous Systems**, volume 5, number 1, pp. 5 – 67.

HASAN K. M., AL-NAHID A., et al. (2012) "Implementation of vision-based object tracking robot", **2012 International Conference on Informatics, Electronics & Vision (ICIEV)**, pp. 860-864.

HOOD S., BONSAN K., et al. (2017). “Bird's eye view: Cooperative exploration by UGV and UAV”, **2017 International Conference on Unmanned Aircraft Systems (ICUAS)**, pp. 247-255. Cairo.

HUSSAIN S., Hussain S. et al, “Mini Rover-Object Detecting Ground Vehicle (UGV)”, **University of Sindh Journal of Information and Communication Technology (USJICT)**, volume 3, number 2, 2019, pp. 104-108.

KUMAR G.H, P., Gurram, R. et al, “Object Tracking Robot on Raspberry Pi using Opencv”, **Journal of Engineering Trends and Technology (IJETT)**, volume 35, number 4, 2016, pp.160-163.

MOUD H., SHOJAEI A. (2018). “Current and Future Applications of Unmanned Surface, Underwater and Ground Vehicles in Construction” **2018 Construction Research Congress**. Volume 14, New Orleans, LA.

NASIRI M. (2006). “Camera-based 3D Object Tracking and Following Mobile Robot”, **2006 IEEE Conference on Robotics, Automation and Mechatronic.**

PRADEEP K. GURRAM H. et al. (2016). “Object Tracking Robot on Raspberry Pi using Opencv”, **International Journal of Engineering Trends and Technology (IJETT)**, volume 35, problem 4, pp.160-164.

PADILLA R. C. (2012) “Evaluation of Haar Cascade Classifiers for Face Detection”, **International Conference on Digital Image Processing ICDIP**, volume 6, number 4.

PALINKO O., REA F.,’et al. (2015). “Eye gaze tracking for a humanoid robot”, **IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids),** pp. 318-324, Seoul.

SAFRAN M., HAAR S., “Arduino and Android Powered Object Tracking Robot”, Department of Computer Science, Southern Illinois University Carbondale, 2012.

SANJAYA W.S. Mada, ANGGRAENI Dyah, “Design of Real Time Facial Tracking and Expression Recognition for Human-Robot Interaction”, **Journal of physics: Conference series**, volume 1090, p. 012044.

SEONG-HOON K., & CHOI J. (2019). “Face Recognition Method Based on Fixed and PTZ Camera Control for Moving Humans”, **2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN)**, pp. 561-563.

SLOTINE G. ,“Telemanipulation with time delays”, **Int. J. Robot. Res**, volume 23. Number 5, 2004, pp. 873-890.

VIOLA J, PAUL M. (2001). “Rapid object detection using a boosted cascade of simple features”, **IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)**, pp. 1063-6919. Kauai, HI.

VIOLA Jhon, Paul Michel, “Robust Real-Time Face Detection”, **International Journal of Computer Vision**, volume 57, number 2, 2004, pp.137-154.

YANG Y., LI M., et al., “Vision Target Tracker Based on Incremental Dictionary Learning and Global and Local Classification”, **Abstract and Applied Analysis**, 2013, p. 323072.

YANG Yue, XIAO Yang, et al., “A Survey of Autonomous Underwater Vehicle Formation: Performance, Formation Control, and Communication Capability”, **IEEE Communications Surveys & Tutorials**, volume 23, number 2, pp.815 - 841.

YOSAFAT S. R., MACHBUB C., et al., (2017). “Design and implementation of Pan-Tilt control for face tracking”, **2017 7th IEEE International Conference on System Engineering and Technology (ICSET)** , pp. 217-222.

WEI L., ANGUELOV D., et al. (2016). “SSD: Single Shot MultiBox Detector”, **European Conference on Computer Vision 2016**, pp. 21-37.

ZHANG Yi, SHEN Yongliang, et al., “An improved tiny-yolov3 pedestrian detection algorithm”, **Optik**, volume 183, pp.17-23.

ELECRONIC RESOURCES

HE K., GKIOXARI G., et al. (2017), “Mask R-CNN”, **Proceedings of the IEEE International Conference on Computer Vision (ICCV)**, pp. 2961-2969, https://doi.org/10.48550/arXiv.1703.06870 , (Access Date: 20 June 2022).

HSAINI J., “AI Robot - Human Following Robot using TensorFlow Lite on Raspberry Pi”, https://helloworld.co.in/article/ai-robot-human-following-robot-using-tensorflow-lite-raspberry-pi , (Access Date: 9 May 2022).

LIN T., MAIRE M., et al. (2014). “Microsoft COCO: Common Objects in Context”. https://doi.org/10.48550/arXiv.1405.0312, (Access Date: 11 May 2022).

MCWHORTER Paul, “AI ON THE JETSON NANO LESSON 30: Building a Servo Pan Tilt Camera Controller”, https://youtu.be/5WeTAA8Unqo , (Access Date: 21 June 2022).

REDMON J., & FARHADI A. (2018). “YOLOv 3 : An Incremental Improvement”, **University of Washington**, https://arxiv.org/abs/1804.02767 , (Access Date: 30 April 2022).

REDMON J., DIVVALA S., et al. (2016). “You Only Look Once: Unified, Real-Time Object Detection”, https://doi.org/10.48550/arXiv.1506.02640 , (Access Date: 20 May 2022).

SHAOQING R., HE K., et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, **Advances in Neural Information Processing Systems 28 (NIPS 2015)**. NeurIPS Proceedings. Retrieved from https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf , (Access Date: 19 June 2022).

URL-1 “Mars Exploration Rovers Overview”, NASA, https://mars.nasa.gov/mer/mission/overview/ , (Access Date: 19 June 2022).

URL-2 “Raspberry Pi Layout”, https://www.etechnophiles.com/raspberry-pi-4-gpio-pinout- specifications-and-schematic/ , (Access Date: 5 April 2022).

URL-3 “L298N chip with Raspberry Pi and DC motors configuration”, https://www.computervision.zone/topic/motor-module/, (Access Date: 17 April 2022).

URL-4 “hc-sr04-ultrasonic sensor on raspberry pi tutorial”, https://thepihut.com/blogs/raspberry-pi-tutorials/hc-sr04-ultrasonic-range-sensor-on-the-raspberry-pi , (Access Date: 10 June 2022).

URL-5 “ Open cv functions”, https://docs.opencv.org/3.4.16/ , (Access Date: 20 March 2022).

URL-7 “Artificial models’ weights and cfg files”, https://pjreddie.com/darknet/yolo/, (Access Date: 10 May 2022).

URL-8 “L298N chip parts”, https://www.electroduino.com/introduction-to-l298n-motor-driver-how-its-work/, (Access Date: 10 April 2022).

URL-9 “Object detection using YOLO model course”, https://www.computervision.zone/courses/object-detection-yolo/, (Access Date: 18 May 2022).

URL-10 “Self-driving car using Raspberry Pi course”, https://www.computervision.zone/courses/self-driving-car-using-raspberry-pi/, (Access Date: 5 May 2022).

URL-11 “Learning Artificial Intelligence on the Jetson Nano course”, https://www.youtube.com/watch?v=5INy0FvaWLw&list=PLGs0VKk2DiYxP-ElZ7-QXIERFFPkOuP4_ , (Access Date: 29 April 2022).

URL-12 “Face and Movement Tracking Pan-Tilt System with Raspberry Pi and OpenCV”, https://core-electronics.com.au/guides/Face-Tracking-Raspberry-Pi/ , (Access Date: 9 March 2022).

URL-13 “Obstacle avoidance circuit connection project”, https://circuitdigest.com/microcontroller-projects/raspberry-pi-obstacle-avoiding-robot , (Access Date: 9 June 2022).

URL-14 “Pan/tilt face tracking with a Raspberry Pi and OpenCV”, 2019, https://pyimagesearch.com/2019/04/01/pan-tilt-face-tracking-with-a-raspberry-pi-and-opencv/, (Access Date: 15 May 2022).

URL-15 “VNC viewer”, https://www.pitunnel.com/doc/access-vnc-remote-desktop-raspberry-pi-over-internet , (Access Date: 1 March 2022).

DISSERTATIONS

TÖRNBERG Isac, “Real time object tracking A comparison between two tracking algorithms”, department of Industrial Engineering and Management, KTH Royal Institute of Technology, 2016.

TV SHOWS

W. Gregory, “Spy in the Wild”, TV Series, London, BBC, 2020.

APPENDICES

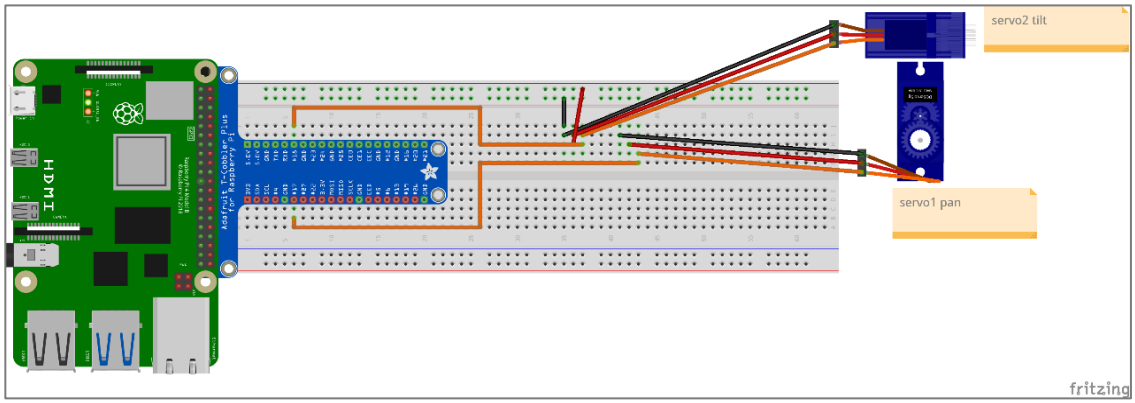
Appendix-1: Difference between VNC, SSH and TCP

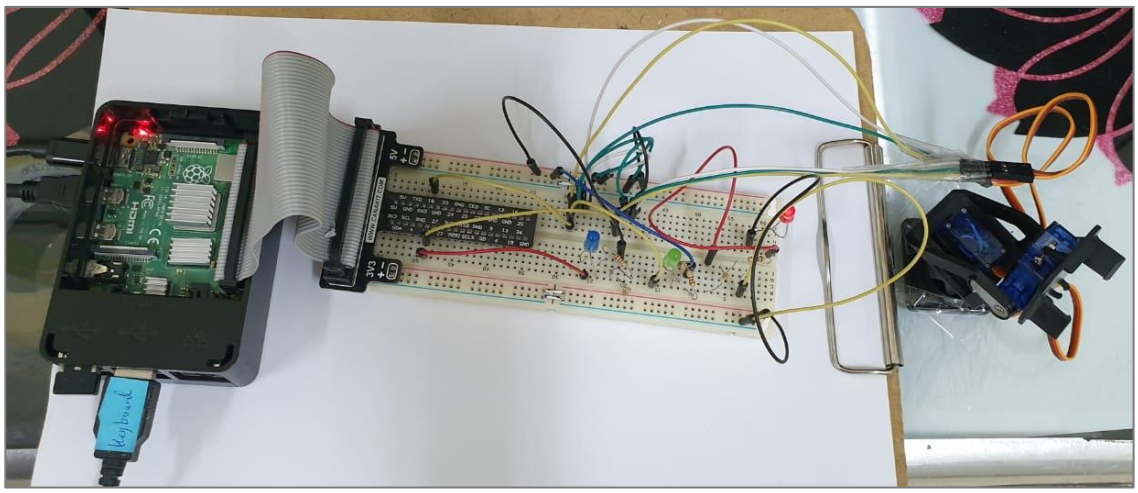
	HDMI Monitor	LCD monitor	SCP	SSH	VNC
Multiple of running apps at a time	Multiple apps	Multiple apps	Single app	Single app	Multiple apps
Files transferring	No	No	Transfer files between remote device and controller	No	No
Security UI	Highest full-fledged graphical UI desktop	Highest full-fledged graphical UI desktop	High Files and terminal	Lowest Terminal	Lower full-fledged graphical UI desktop
Connection requirements	HDMI wire and monitor	LCD screen	Network, an IP address, and WinSCP software	Network an IP address and PuTTY software	Network, an IP address and VNC viewer software
Accessibility	HDMI wire via HDMI	Wiley via GPIOs pins	Remotely access via SFTP protocol	Remotely access via SSH protocol using port 22	Remotely access via VNC protocol

Appendix-2 Differences between webcam and pi camera devices

	Webcam	Pi camera
Port	USB port	CSI camera port
Libraries	Motion fswebcam	Raspistill
FPS rate	Low	Higher
Robustness	Weak	Strong
Flexibility	Low	High

Appendix-3 Pan tilt basic electric circuit





RESUMEE

Name Surname: Nour Ammar

EDUCATION:

- **Bachelor:** 2019, Arap Open University, Faculty of Computer Studies, Information Technology and Computing program, Riyadh, Saudi Arabia.
- **M.S:** 2022, Istanbul Aydin University, Faculty of Engineering, Department of Software Engineering, Artificial Intelligence and Data Science program.

PUBLICATIONS FROM DISSERTATION, PRESENTATIONS AND PATENTS:

- AMMAR. Nour, OKATAN. Ali. (2022, July 15-16). *Real-Time Visual Target Detection and Tracking Via Unmanned Ground Vehicle* [Paper presentation]. The 2nd International Conference on Computing and Machine Intelligence (ICMI-22). Istanbul, Turkey. DOI:10.1109/ICMI55296.2022.9873647. Retrieved from https://ieeexplore.ieee.org/document/9873647.
- AMMAR. Nour, OKATAN. Ali. (2021, July 9-10). *Movie Reviews Text Sentiment Analysis Based on Hybrid LSTM and GloVe* [Paper presentation]. "International Conference on Advanced Engineering, Technology and Applications (ICAETA-2021)". 260-264. Istanbul, Turkey. Retrieved from https://icaeta.aiplustech.org/assets/docs/Proceedings_Book.pdf.