Real-Time Visual Target Detection and Tracking Via Unmanned Ground Vehicle

Nour Ammar and Ali Okatan

August 29, 2022

Real-Time Visual Target Detection and Tracking Via Unmanned Ground Vehicle

Nour Ammar
Software engineering faculty
Istanbul Aydin University
Istanbul, Turkey
nourammar@stu.aydin.edu.tr

Ali Okatan
Software engineering faculty
Istanbul Aydin University
Istanbul, Turkey
aliokatan@aydin.edu.tr

***Abstract*— Vision-based autonomous robot field is becoming rapidly popular, according to the Artificial Intelligence revolution. The proposed system is a composition of Unmanned Ground Vehicle vision-based target tracking robot. 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 process 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 the original YOLOv3 model 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. Ultra-sound technique is used for collision avoidance. The locomotion of UGV is based on transmitted commands from algorithm to the motors through local network connection. The results show an autonomous behavior, streamlined and accurate locomotion of tracking.**

Keywords—Autonomous robot, UGV, Object tracking, Target tracking, Obstacle avoidance, Human following, Object detection, YOLOv3, COCO dataset.

I. INTRODUCTION

There are multiple applications in autonomous control robotics field. For instance, Video surveillance, vehicle navigation and object tracking. Computer vision is a powerful sector of Artificial Intelligence. The computer vision concept focuses on digital replication of complex human vision system. The conjunction of autonomous control robotics and computer vision induces a powerful artificial simulation of human decisions and procession of that decision dynamically. Vision-based object tracking plays a significant role comprehensively in both computer vision and robotics fields [1]. Unmanned Ground Vehicle (UGV) robots are extremely capable of navigating safely inside an unknown environment [2].

In related works, there are various heterogeneous methods followed. Chatterjee et al. [3] experimented YOLO model, which is a low-resolution process of images (480p stream). In their experiment, YOLOv3 shows the best results. The utilized controller in study of [3] is a controller with high qualifications comparing to the utilized controller in this study. Mir-Nasiri. [4] followed the vision-based object tracking, where the acquiesced image is binarized and contoured. Therefore, the computation cost of tracking process is reduced. Palinko et al. [5] followed the architecture of eyes tracking, which first meant detecting eyes, then keeping the deviation values near to that gaze to tracking the center of object in the proposed system. In [6] study, the author used TensorFlow framework to create a UGV robot that follows a human with the employment of an accelerator device. Tracking algorithm in project [6] is a novel simple tracking algorithm, which the proposed project is inspired by it.

In these kinds of projects, diverse methods of object detection are followed. Machine learning simple methods are widely applied such color-based detection. For instance, the work [7] because of its low-cost computation. More computation-expensive method is followed by author in work [8], which is a face tracking system, where the detection process is based on the Haar-like cascade model [9, 10] for frontal face detection. It results high speed even though it is not highly accurate.

Recently, deep learning-based projects are increasing. Various models are popular based on COCO dataset [11], such as Single Shot multi-box Detection (SSD) model [12], YOLO models [13 - 15], as experimented in work [3], Mask R-CNN [16] and Faster R-CNN [17]. SSD is a powerful model, but it is a GPU based model. YOLO model is a powerful model with low-cost computation. Therefore, YOLOv3-tiny [15], which is a light curtailed version of the original YOLO models, is the best model to be used, according to the fair computing capabilities of Raspberry Pi Single Board Controller (SBC). The object to track is specified to be human class only, which covers the largest area of the camera frame.

The purpose of this study is to superimpose an autonomous, real-time, vision-based object tracking prototype using an SBC [18]. The prototype system is composed of image acquisition and processing to detect, locate, and track the targeted object autonomously. A simple obstacle avoidance system is followed, which is based on the ultrasound technique [20]. Locomotion control process is based on frame segmentation technique. Delayed feedback issue is handled following a simple mathematical conditional equation. Motors are controlled through the auto-transmitted commands regarding state of UGV, which are detailed in this paper.

Results of the proposed study are acceptable, despite the slowness of the robot's feedback since neither an accelerator nor GPU processor is utilized. In aim to increase the speed of system, the frame size is decreased, and YOLOv3-tiny is applied rather than the original YOLO models. Accuracy of object detection is high on average. Three tests of UGV routes are recorded. Standard deviation of routes over an interval of time feedback with positive values.

II. SYSTEM ARCHITECTURE AND METHODOLOGY

A. *System Components Architecture*

Hardware system components are an SBC Raspberry Pi board, L298N H-bridge board, two DC motor drivers, power supply, pi camera board, camera holder, HR-SR ultra-sonic sensor board and a monitor to control to connect with Raspberry Pi via wireless local network (LAN). As shown in "Fig. 1", components of the object locomotion-tracking subsystem are connected serially, the power supply source for Raspberry Pi, the Raspberry Pi SBC, the camera board, the L298N H-bridge board, the monitor, the DC motors and a power supply resource. Components of the obstacle avoidance subsystem are connected serially, the Raspberry Pi SBC, a power supply resource for Raspberry Pi, the Raspberry Pi SBC, a bunch of transistors, the HC-SR04 ultra-sonic sensor board.

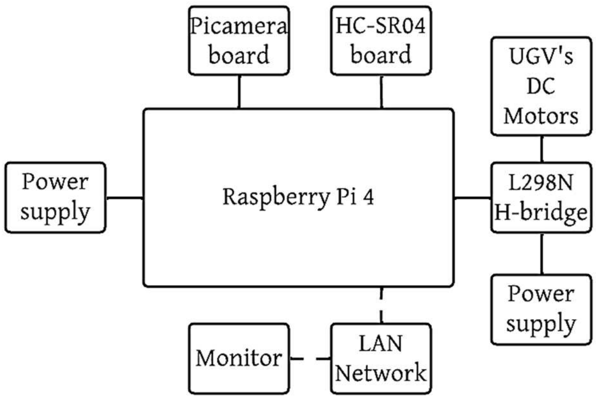


Fig. 1. Component diagram of the entire hardware system.

B. *Flowchart of UGV System*

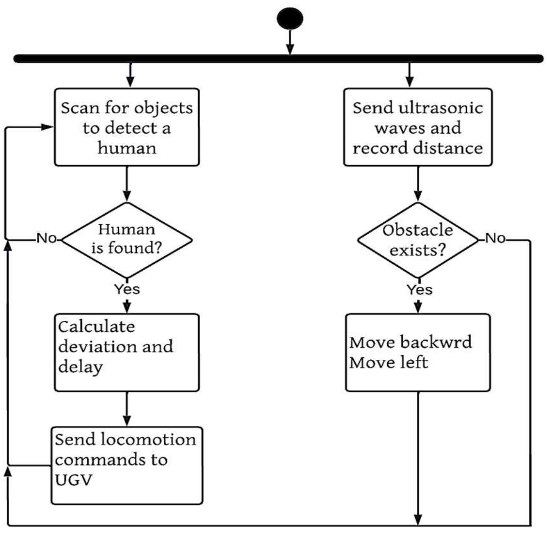


Fig. 2. Flowchart of software object detecting and tracking with obstacle avoidance via UGV system.

"Fig. 2" shows the flowchart of the entire system. The system executes two processes at the same time. First process is the primary function of the proposed system, which is the human tracking. Second process is the obstacle avoidance function, which is a secondary process in aim to handle collision issue. The system starts with the object detection flow state, which 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 a human is detected, the system moves to the next flow state. Second state consists of mathematical equations to execute the tracking state according to frame and bounding box value. Deviation of x-axes and delay values are induced. Third state is the motion commands that are sent to electronic boards to make vehicle moves regarding the induced values. In case of an obstacle appearance, a new locomotion orders transmit to electronic boards to make vehicle avoids the obstacle. The flowchart of the proposed system is a loop of computations and commands.

C. *Robot Assembly*

UGV prototype is assembled over multiple stages and multiple experiments. As shown in "Fig. 3", a flexible camera holder is utilized to allow the camera to be flexible in manual guidance. Only two DC motor drivers are composed in aim to reduce the power consumption. Eight batteries are utilized to supply the motors with power, which are located rearward of the Raspberry Pi. Castle rotation ball is utilized in aim to decrease friction force affection and increase the flexibility of movement, which makes the impetus power is pushed from the rearward motor drivers.

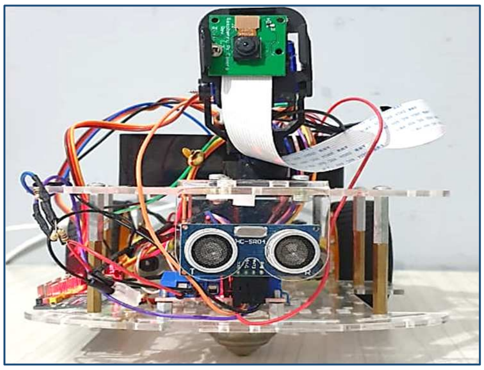


Fig. 3. Assembly of UGV robot prototype.

III. OBJECT DETECTION SYSTEM

A. *Methods and Software*

1) *COCO dataset*

The artificial object detection model is based on Common Objects in Context (COCO) dataset [11]. COCO dataset is a novel dataset system innovation that solves complex real-life image-related issues. It is divided for four tasks: image classification, semantic segmentation, segmenting individual object instances and object localization. COCO dataset consists of 80 labeled object categories and 91 stuff categories. It contains 330k images, 5 captions per image. Person category consists of 250K people with key points, which is a significant number for only individual class. The categories are indexed in range of 1 to 85 indices. First five indices are ignored since they are referring to axe's location and confidence values [11]. Sixth index and beyond, are considered. Sixth index refers to human class, which is specified in the proposed system.

2) *OpenCV library*

OpenCV is the most efficacious library that is used in this study, which is utilized for processing the real-time object detection and visualization as well as handling images pre-processing functions [19]. Other powerful libraries are utilized, such as NumPy to handle mathematical computation, pandas to handle data frames manipulation, and RPi.GPIO library is used to handle connections of electronic boards.

3) YOLOv3-tiny model

YOLOv3-tiny model is implemented in this study. YOLO v3 refers to You Only Looks Once third version with light computation cost. There are multiple advantages of YOLOv3-tiny. Such as speed, low-cost computation -no need for GPU, which makes it convenient for SBC-based projects. Weights and cfg files of the pre-built model are imported and implemented in the object detection process.

In original YOLOv3, there are 109 layers in the network three of them are output layers [10, 13], with different dimensionality of each output layer, whereas YOLOv3-tiny consists of only 13 layers [15] adding to two output layers. The structure of YOLOv3-tiny network is seven convolutional layers, six max-pooling layers and two output layers, with different dimensionality. The shape of premier output layer is (300,85), whereas the shape of the conclusive output layer is (1200,85). Second dimension refers to the indices' values: y-axes center of Region of Interest (RoI) frame, width of RoI frame, height of RoI frame, confidence value of detection and 80 other indices which are the COCO classes names [15].

B. Object Detection Process

1) Object-to-track specification

The object to track is specified to be a human. The index of human class in COCO dataset structure is the sixth index.

2) Implementation

Input image is resized to 320× 320 resolutions in aim to reduce computation cost. "Fig. 4" shows the flowchart of human detection system. Acquiesced image is normalized to be one-dimensional array instead of two-dimensional array through dividing acquiesced image by 255.

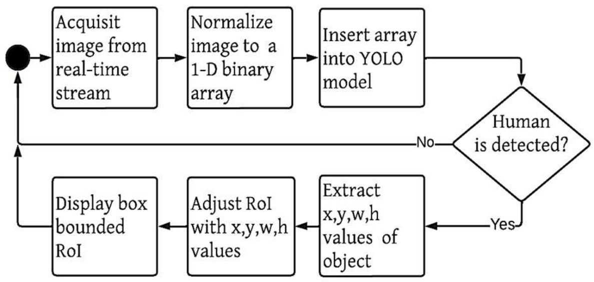


Fig. 4. Flowchart of human detection process

IV. Unmanned Ground Vehicle

A. Components Archtecture

Hardware system components are Raspberry Pi SBC, L298N H-bridge board, DC motors, power supply, pi camera, HC-SR04 ultrasonic sensor board and monitor to control and connect with Raspberry Pi via wireless network.

- Raspberry Pi SBC.
- L298N H-bridge driver.
- Camera board.
- HC-SR04 ultrasonic sensor board.
- Two 5v DC motor.
- Monitor for system execution.
- 5v power supply for Raspberry Pi
- Eight 1.5v batteries for motors power supply.

a) Raspberry Pi 4 controller

The project is based on Raspberry Pi 4 B model controller with 8 gigabyte RAM, as shown in "Fig. 5", Raspberry Pi is a small, 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, it belongs to embedded devices regarding the limited computation cost [18].

Fig. 5. Raspberry Pi 4 model B SBC board.

b) Electronic boards

As shown in "Fig. 6", three of basic electronic boards are utilized. (a) shows L298N Dual H-bridge driver board. The board is utilized to regulate both input volts and Pulse-Width Modulation (PWM) values. The SCB controller transmit orders of PWM to the H-bridge driver. Therefore, the H-bridge driver regulate the energy duty for motors. (b) shows Pi camera v1.3 5 MP board. The board is utilized for visual input acquisition in object detection stage. (c) shows the HC-SR04 ultrasonic sensor board. The board is utilized for collision avoidance technique.

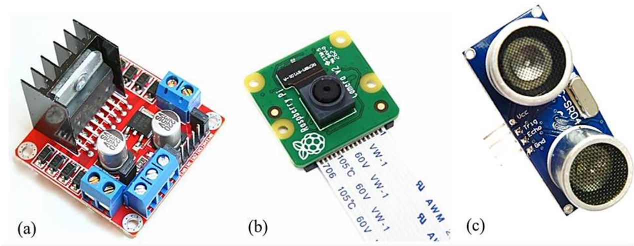


Fig. 6. Boards are used in the project. (a) L298N H-bridge board. (b) pi camera v 1.3 board. (c) Ultra-sonic sensor HC-SR04 board.

B. Obstacle Avoidance Technique

Collision is a potential issue in the unmanned systems. Multiple techniques are applied to avoiding obstacles. Ultrasound emission is an effective, low-cost, rapid, and precise technique [20]. The basic concept of ultrasonic sensor is to emit ultrasonic waves and receive it back. The distance is measured using the recorded value through the equation of distance, 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 waves broadcasting and reception travel. The sensor board emits ultrasonic waves with frequency of 40 kHz/s. Resistor with 1.5k ohm is used to avoid SCB damage.

$$d_O = v_{sound} \times {}^{t}/_{2} \quad (1)$$

C. Robot Autonomous Tracking Algorithm

1) Segmentation of frame plane

Camera-based frame plane is divided into segments. Locomotion control commands are based on the location of target to track regarding the plane-segments area. The division is based on the frame of visual input through camera [21]. As shown in "Fig.7", the frame plane is simply divided into four segments: (A), (B), (C) and (D). "Fig. 8" indicates the conditions of command based on segment target to track. In case of the target is in segment (A), the control command process state is turning the robot left. In case of the target is in segment (B), the control command process state is moving the robot forward. In case of the target is in segment (C), the control command process state is turning the robot right. Finally, in case of the target is in segment (D), the control command process state is forcing the robot to stop since the target to track is near to the vision of robot.

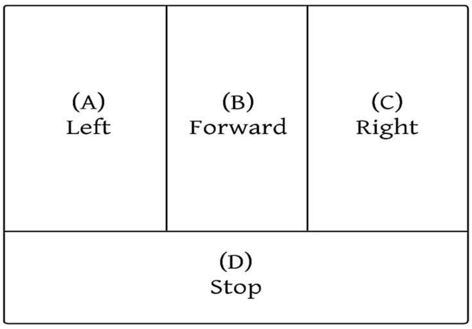


Fig. 7. Frame segmentation for four segments. Respectively, (A), (B), (C) and (D) segments with Left, Forward, Right and Stop commands.

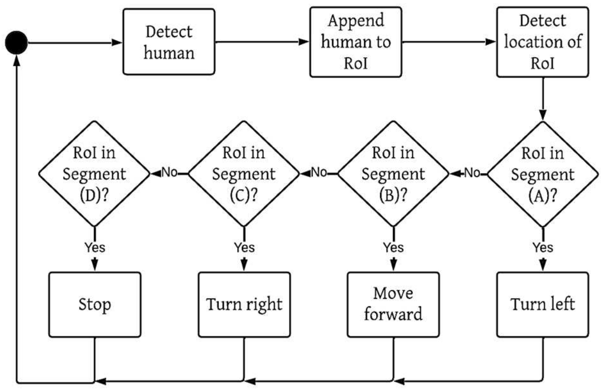


Fig. 8. Flowchart of object tracking locomotion commands regarding position of object.

2) Equations of locomotion

The followed principle of tracking the target is based on coefficients of the relation between RoI frame and the entire visual input to the vision of robot frame. The basic technique is to keep the RoI in the forefront area of the frame. The coefficients of RoI are changing regarding the object locomotion, and vice versa.

$$\Delta x = x_{max} - x_{min} \quad (2)$$

$$\Delta y = y_{max} - y_{min} \quad (3)$$

$$Rx_o = x_{min} + {}^{\Delta x}/_{2} \quad (4)$$

$$Ry_o = y_{min} + {}^{\Delta y}/_{2} \quad (5)$$

However, the technique of tracking starts with the calculation of the changes in RoI location, Δx and Δy. Coefficients, x_{min}, x_{max}, y_{min} and y_{max} refer to the axe's points of the RoI frame. The changes on movements are employed to extract the center coordinates of RoI frame, which are expressed by (Rx_o, Ry_o) point.

$$l_w = {}^{w_f}/_{2} \quad (6)$$

l_w refers to the central longitude of the frame. the measured distance between l_w and Rx_o is expressed by α_x, which refers to the distance between center of target and center of frame, named x deviation in this study.

$$\alpha_x = l_w - Rx_o \quad (7)$$

To calculate the area of segment (D) that corresponds to stop command, Z coefficient is computed, which refers to the area underneath the RoI.

$$Z = h_f - y_{max} \quad (8)$$

a) Delay issue

In autonomous robots' execution, delay of response occurs customarily. There are diverse reasons behind the delay. Most recurrent reasons are the time consumed of transmission and response over a network and the waiting of target-to-track to be detected [21]. To manage the induced delay, multiplicity of techniques is improved [22]. In the proposed system, a simple delay handling technique is utilized. the delay time value is executed regarding the threshold of $\alpha_x$value as detailed in $d(\alpha_x)$ function. Outputs of the delay function refer to part of second values.

$$d(\alpha_x) = \begin{cases} 0.8, & \alpha_x > 0.4w_f \\ 0.6, & 0.35w_f < \alpha_x < 0.4w_f \\ 0.4, & 0.35w_f < \alpha_x < 0.2w_f \\ 0.2, & \alpha_x > 0.2w_f \end{cases} \quad (9)$$

3) Motor drivers circuit

Adjustment of H-bridge regulator is required for DC motors setting up. The process is done by using L298N regulator connected with Raspberry Pi and DC motors, as shown in "Table I".

TABLE I. 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

V. RESULTS AND DISCUSSIONS

Various methods of comparison are followed. Multiple factors are put in consideration about the execution of the prototype, the speed of person that is being tracked, the power supply of the motor drivers beside the environment of testing such as the flat of roof, the lightness of room and the heat of SCB's mother board. The induced heat is handled by a small cooling fan based above the Raspberry Pi board. The cooling fan decreases the temperature up to 35 Celsius degrees.

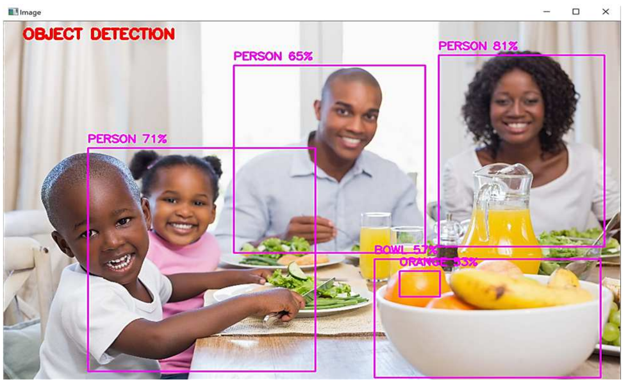
A. Object Detection

Lightness degree of room affects the performance of object detection process. The performance is observed better in a room with a high light degree. Since the utilized camera works only in exposed to light environment, the system failed to detect any object in the dark, whereas it works in the exposed to light environment accurately, as shown in “Fig. 9”.

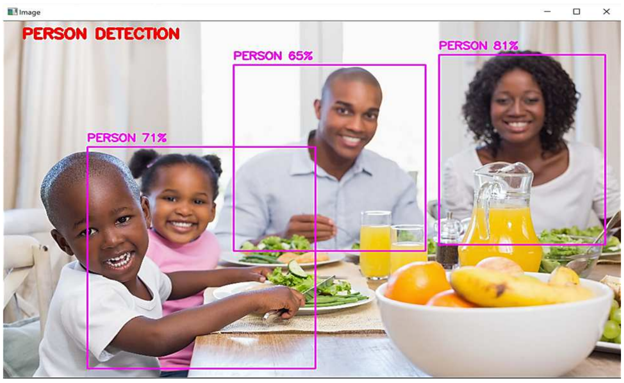
(a) (b)

Fig. 9. Two shots of the real-time object detection test based on diffirent light degrees. (a) The failure of object detection in darkness. (b) A person is detected in the lightness.

“Fig. 10” shows the number of objects detected before the specification of the object-to-detect to be human only human and after the specification. The number of objects in (a) figure is five objects, which it is remarkable to be mentioned, even small objects are detected such as an orange. (b) figure shows the number of objects detected is only three, which all objects are humans. The accuracy is acceptable since the usage of YOLOv3-tiny, which is an abridged version of the original YOLO models.



(a)



(b)

Fig. 10. Object detection specifying. (a) The frame of OpenCV without a definite of a specific class to be detected, where five objects are detected. (b) The frame of OpenCV with specifying the person class for detection, where three people out of four are detected.

B. Locomotion Tracking

Performance of UGV robot is remarked better above a flattened roof ground compared to a roof covered by a rug, where the friction force value is higher in the second case.

“Table II” shows three tests of UGV robot. Three factors are considered in these experiments. Distance between UGV and object, deviation value (α_x) and the direction that is followed by UGV robot. The records are taken based on an interval of time in range of three seconds. As in indicated in “Table II”, 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 records, which output from the ultrasonic sensor, indicate the stability of route and the fast feedback of commands.

TABLE II. Three tests of UGV tracking with recorded values.

t (s)	Test 1			Test 2			Test 3		
	α_x	S_d[a]	c[b]	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**	-

[a] Distance between UGV and object

[b] Direction of UGV route

“Fig. 11” indicates α_x values of the three tests over six periods of time, according to “Table II”. The routes are measured with polynomial trendlines to indicate the neat route of the UGV. Error bars measurement is utilized as well. The best route is clear in test 2 (the orange series) since the standard deviation that belongs to test 2 is the smallest value, according to “Table II”, followed by the route of test 1 followed by the route of test 3.

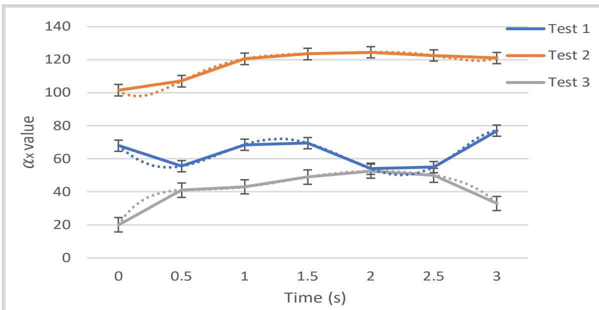


Fig. 11. Scatter chart of tests 1, 2 and 3 of tracking process values over an interval of time with error rate and trend line observation.

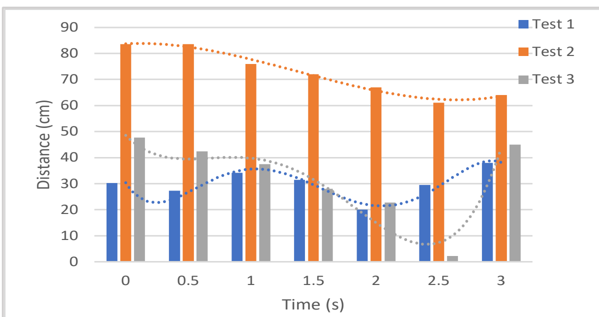


Fig. 12. Bar chart of test 1, 2 and 3 of obstacle farness distance values over an interval of time.

"Fig. 12" 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 occurred. The outliers refer to the obstacle confrontation and the feedback to take another route in aim to avoid that obstacle. 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.

VI. Limitations

More than one limitation is remarked in this study, regarding the computation cost of the algorithm. Pre-built model YOLOv3-tiny is computation-expensive on the SCB, which induces a slowness in FPS rate and late response of locomotion commands.

VII. Conclusion

In this study, a prototype robot of human tracking via UGV system is experimented. The system is vision-based input. Image processing stage performed using OpenCV library. YOLOv3-tiny deep learning model is employed in object detection stage. Object to track is specified to be a human only. Various techniques are used over the stages of the system. Obstacle avoidance, frame segmentation, autonomous control commands and response-delay problem handling. Performance measurements are innovated. Performance of object detection model is indicated by two methods of comparison. First method is to compare the accuracy values of a detected object in a dark vision versus a shiny vision. Second method is to compare the number of objects detected before specifying the class human to be detected. Measurements of locomotion values are based on recorded real experimented tests on the UGV during tracking a person. Over multiple tests, the results show acceptable standard deviations. an autonomous execution behavior, streamlined and accurate locomotion tracking.

In future work, the SCB board is going to be improved to handle more complex algorithms and to manage the delay more accurately. A pan/tilt-based robotic eye is going to be appended as a sub-system to scan for new targets. An infrared camera type is going to be attached to add the ability of the dark view. Better-performed power supply is going to be utilized rather than simple AA batteries.

References

[1] Jafri, R., Ali, S.A., Arabnia, H.R. *et al.* Computer vision-based object recognition for the visually impaired in an indoors environment: a survey. *Vis Comput* 30, 1197–1222 (2014). [Online]. Available: https://doi.org/10.1007/s00371-013-0886-1.

[2] Hood, S., Benson, K., Hamod, P., Madison, D., O'Kane, J. M., & Rekleitis, I. (2017). Bird's eye view: Cooperative exploration by UGV and UAV. 2017 International Conference on Unmanned Aircraft Systems (ICUAS) (pp. 247-255). Cairo: IEEE. doi:10.1109/ICUAS.2017.7991513.

[3] Chatterjee, S., Zunjani, F. H., & Nandi, G. C. (2020). Real-time object detection and recognition on low-compute humanoid robots using deep learning. *International Conference on Control, Automation and Robotics. 6*, pp. 202-208. Prayagraj: ResearchGate. [Online]. Available:.

[4] N. Mir-Nasiri, "Camera-based 3D Object Tracking and Following Mobile Robot," 2006 IEEE Conference on Robotics, Automation and Mechatronics, 2006, pp. 1-6, doi: 10.1109/RAMECH.2006.252655.

[5] Palinko, O., Rea, F., Sandini, G., & Sciutti, A. (2015). Eye gaze tracking for a humanoid robot. *IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)* (pp. 318-324). Seoul: IEEE. doi:10.1109/HUMANOIDS.2015.7363561.

[6] Hsaini, J. (2022, May 13). *AI Robot - Human Following Robot using TensorFlow Lite on Raspberry Pi*. [Online]. Available:Hello world: https://helloworld.co.in/article/ai-robot-human-following-robot-using-tensorflow-lite-raspberry-pi

[7] Pradeep Kumar.G.H, R. G. (2016). Object Tracking Robot on Raspberry Pi using Opencv. *International Journal of Engineering Trends and Technology (IJETT), 35*(4). [Online]. Available: http://ijettjournal.org/2016/volume-35/number-4/IJETT-V35P235.pdf.

[8] S. R. Yosafat, C. Machbub and E. M. I. Hidayat, "Design and implementation of Pan-Tilt control for face tracking," 2017 7th IEEE International Conference on System Engineering and Technology (ICSET), 2017, pp. 217-222, doi: 10.1109/ICSEngT.2017.8123449.

[9] Viola, Paul & Jones, Michael. (2004). Robust Real-Time Face Detection. International Journal of Computer Vision. 57. 137-154. [Online]. Available: https://www.researchgate.net/publication/220660094_Robust_Real-Time_Face_Detection.

[10] Rafael Padilla, C. F. (2012). Evaluation of Haar Cascade Classifiers for Face Detection. International Conference on Digital Image Processing ICDIP (p. vol6. no4). Venice: World Academy of Science.

[11] Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., . . . Dollar, P. (2014). Microsoft COCO: Common Objects in Context. *ArXiv*, arXiv:1405.0312.

[12] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Springer, Cham. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-46448-0_2.

[13] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788). [Online]. Available: https://ieeexplore.ieee.org/document/7780460.

[14] Redmon, Jo. & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *arXiv:1804.02767*

[15] Adarsh, P., Rathi, P., & Kumar, M. (2020). YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)* (pp. 687-694). Coimbatore: IEEE. doi:10.1109/ICACCS48705.2020.9074315.

[16] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961-2969).

[17] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*. Retrieved from Redmon, J., & Farhadi, A. (2018). *YOLOv 3 : An Incremental Improvement.* Washington: Arxiv. doi:10.48550/arXiv.1804.02767.

[18] Ghael, Hirak. (2020). A Review Paper on Raspberry Pi and its Applications. 10.35629/5252-0212225227. [Online]. Available: https://www.researchgate.net/publication/348993428_A_Review_Paper_on_Raspberry_Pi_and_its_Applications.

[19] A. B. K. K. V. E. Kari Pulli, "Realtime Computer Vision with OpenCV: Mobile computer-vision technology will soon become as ubiquitous as touch interfaces," *Qeue,* vol. 10, no. 4, pp. 40 -56, 2012. [Online]. Available: https://doi.org/10.1145/2181796.2206309.

[20] Survey of collision avoidance and ranging sensors for mobile robots, Robotics and Autonomous Systems, Volume 5, Issue 1, 1989, Pages 5-67, ISSN 0921-8890. [Online]. Available: https://doi.org/10.1016/0921-8890(89)90041-9.

[21] K. M. Hasan, Abdullah-Al-Nahid and A. Al Mamun, "Implementation of vision-based object tracking robot," *2012 International Conference on Informatics, Electronics & Vision (ICIEV)*, 2012, pp. 860-864, doi: 10.1109/ICIEV.2012.6317485.

[22] G. Niemeyer and J. J. E. Slotine, "Telemanipulation with time delays", Int. J. Robot. Res., vol. 23, no. 9, pp. 873-890, 2004.