APPLICATIONS OF
MODELLING AND SIMULATION
http://arqiipubl.com/ams eISSN 2600-8084 VOL 8, 2024, 101-109

Autonomous Person-Following Telepresence Robot Using Monocular Camera and Deep Learning YOLO

Ahmad Amin Firdaus Sakri[1], Izzuddin Mat Lazim[1*], Suffian At-Tsauri Mauzi[1], Musab Sahrim[1], Liyana Ramli[1] and Aminurrashid Noordin[2]

[1]Faculty of Engineering and Built Environment, Universiti Sains Islam Malaysia, Bandar Baru Nilai, Negeri Sembilan, Malaysia
[2]Faculty of Electrical Technology and Engineering, Universiti Teknikal Malaysia Melaka, 76100 Durian Tunggal, Melaka, Malaysia

[*]Corresponding author: nh.izzuddin@usim.edu.my

Submitted 11 January 2024, Revised 27 March 2024, Accepted 10 April 2024, Available online 25 April 2024.

Abstract: Telepresence robots (TRs) are increasingly important for remote communication and collaboration, particularly in situations where physical presence is not possible. One key feature of TRs is person-following, which relies on the detection and distance estimation of individuals. This study proposes an autonomous person-following TR using a monocular camera and deep-learning YOLO for person detection and distance estimation. To compensate for the monocular camera's inability to provide depth information, a novel distance estimation algorithm based on focal length and person width is introduced. The estimated width information of the detected person is extracted from the bounding box generated by YOLO. A pre-trained model using the MS COCO dataset is employed with YOLO for the person detection task. For robot movement control, a region-based controller is proposed to enable the robot to move based on the detected person's location in the image captured by the camera. Finally, integration and deployment of the proposed method in the TR is carried out using the Robot Operating System (ROS). Experimental results demonstrate that the TR can successfully follow a person using the proposed algorithm, thus highlighting its effectiveness for person-following tasks.

Keywords: Person following; Service robot; Telepresence robot; You Only Look Once (YOLO).

1. INTRODUCTION

Telepresence robots (TRs) are becoming increasingly popular due to their ability to function as avatars for remote users, allowing them to feel physically present in a local environment while being able to move around and interact with local individuals [1]. TRs offer numerous advantages, including cost and time savings, increased safety, and improved accessibility. For example, a physician can remotely monitor and interact with COVID-19-positive patients in a hospital safely via a TR, reducing the risk of exposure to infectious diseases [2]. TRs have also been used in various other settings, such as offices to facilitate remote collaboration and communication [3], shopping malls for customer service and assistance [4], and elderly homes to provide companionship and assistance to residents [5]. With their versatility and potential to provide immersive telepresence experiences, TRs are a promising technology for a wide range of applications.

One of the important features of the TR is person-following. The person-following feature is essential for TR as it allows the robot to autonomously follow a person while maintaining visual contact with them, providing an immersive and interactive experience for the remote individual [6]. In addition, the person-following feature can enhance the robot's safety by allowing it to navigate autonomously in crowded environments while avoiding obstacles and people. Several autonomous person-following methods for mobile robots have been proposed in the past. Most of these works exploit sensors such as laser range finders (LRFs) [7]–[10], RGBD cameras [6], [11]–[13], monocular cameras [14], [15], and stereo camera [16]–[18] for the target person detection and distance estimation.

LRF-based TRs utilize laser sensors to detect and estimate distances to the target person. However, they may not be suitable for certain environments, such as hospitals, due to restrictions on laser usage. On the other hand, TR using an RGBD camera may not perform well outdoors since infrared sensors perform poorly in the presence of sunlight [15], [19]. In comparison to stereo cameras, monocular cameras typically offer a more cost-effective solution. Nevertheless, accurately estimating a person's distance using this sensor is non-trivial. This is because, unlike stereo cameras which rely on at least two cameras for depth perception, a monocular camera necessitates an additional technique, such as computer vision, for distance estimation. In [14], a person-following robot using two monocular cameras positioned facing forward and downward was successfully employed to estimate a person's distance without the need for ranging sensors, as demonstrated by experimental

findings.

In the field of computer vision, object detection is widely recognized as one of the most challenging tasks, as it involves both classifying and locating objects within a scene. Numerous approaches to object detection have been proposed, employing both neural and non-neural techniques. R-CNN stands out as a popular neural approach, utilizing region proposal methods to initially generate potential bounding boxes within an image, followed by the classification of these proposed boxes. However, this approach is complex, slow, and hard to optimize because each component must be trained separately [20]. In contrast to R-CNN, You Only Look Once (YOLO) proposed in [21] treats object detection as a single regression problem, thereby reducing computational complexity and offering a significant speed advantage [22]. Several studies have compared the performance of YOLO with other detection methods, consistently demonstrating YOLO's superior detection speed [20], [23], [24]. This makes the YOLO approach suitable for real-time applications such as the detection of persons for TR.

Motivated by the above studies, a person following TR adapting a monocular camera and object detection based on YOLO is proposed. Unlike previous works that focus on using sensors that provide depth information directly, e.g., stereo camera, this study derives the distance information of the target person image from a monocular camera using a novel algorithm that manipulates the focal length and person width information. The width of a person is obtained by using a bounding box from YOLO. The proposed algorithm is deployed on a TR, and several experiments are conducted to test the effectiveness of following a person in real-time.

The paper follows the structure outlined below. In Section 2, the robot's hardware and software components, including its architecture and design, are outlined. Section 3 describes the experiments conducted. Lastly, Section 4 provides closing remarks to conclude the paper.

2. DESIGN OF THE TELEPRESENCE ROBOT

2.1 Overall Design

The overall framework of the person-following TR is illustrated in Figure 1. This paper focuses on a novel framework for a person-following algorithm using a cost-effective monocular camera and the YOLO algorithm. The robot platform is developed to test the proposed algorithm, comprising upper and lower body parts. The upper parts include the head, hands, and monocular camera, while the lower body features a differential drive system for robot locomotion. In this research, the monocular camera captures 2-dimensional images of the robot's front region, which are then processed by an embedded computing board (NVIDIA Jetson Nano) acting as a high-level controller. The proposed algorithm for person detection and distance estimation is executed on the embedded computing board, a detailed discussion of which follows in subsequent sections. Based on the algorithm's outputs, control signals are transmitted to the lower-level controller (Arduino Mega), which subsequently adjusts the motors to track the detected person. The following sections provide further elaboration on the person detection, distance estimation of humans, and controller design of the TR.

2.2 Person-Detection Using Deep Learning YOLO

There are two main approaches to implementing YOLO for object detection, using custom models, or pre-trained models. When using a custom model, a dataset containing images with annotated bounding boxes around objects of interest is collected. This dataset is then used to train the YOLO model to recognize specific objects, such as people. Alternatively, pre-trained YOLO models that have already been trained on large datasets are available. These pre-trained models can be directly used for object detection tasks, saving time and computational resources required for training from scratch. In this study, the YOLOv4 pre-trained model with the MS COCO dataset is utilized to detect, and subsequently estimate the distance of a person. The YOLO model used for human detection and distance approximation is executed on the GPU with the CUDA backend to achieve faster image processing speeds and improved FPS. It should be noted that the pre-trained YOLO model using the MS COCO dataset will detect any human in the image without specifically identifying a person of interest. Therefore, for this study, the proposed person-following feature is restricted to tracking one person in front of the TR.

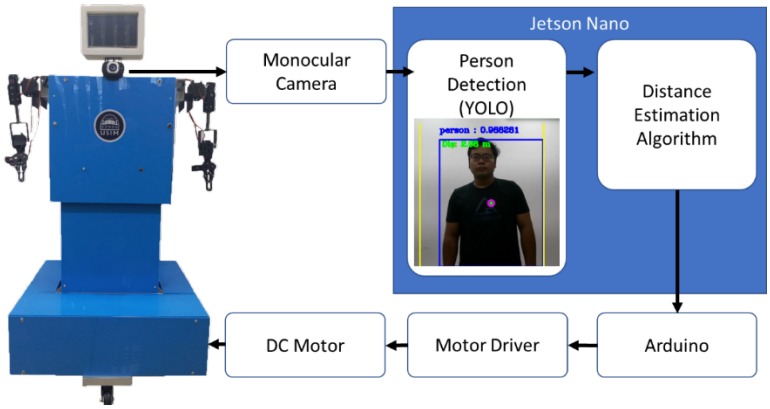

Figure 1. Framework of the person-following TR.

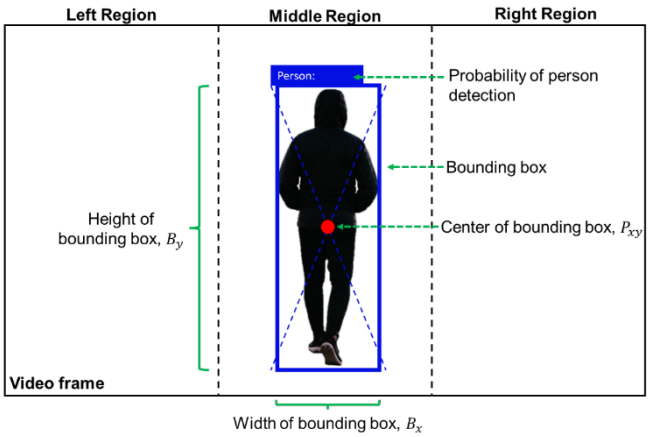

Figure 2. Components of video frame.

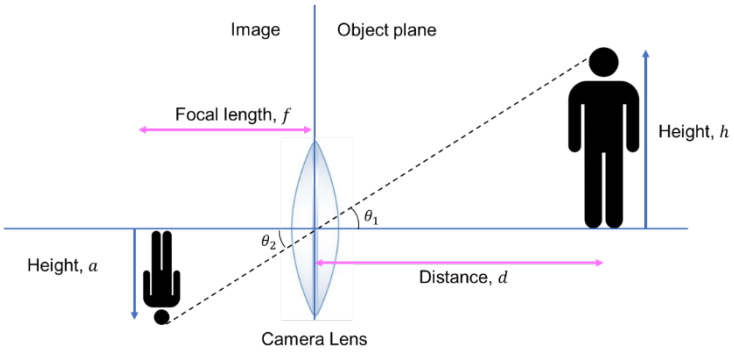

Figure 3. Method for obtaining the focal length of the camera.

The image of the person detected by YOLO is encapsulated by a bounding box drawn using the OpenCV libraries method. The live streaming output from the camera is displayed on the Jetson Nano and it can be accessed by the control station using a desktop sharing software (VNC Viewer). The size of the output frame is set to 640 x 480 pixels. Figure 2 shows the output video frame of the camera when detecting a person. It consists of the image, bounding box of the person, center of the bounding box P_{xy}, probability of person detection, and regions of detection. The bounding box information is utilized for the estimation of the distance between the TR and the person which is discussed in the next section. For the region of detection, the image is divided into three detection regions: the left region, the middle region, and the right region. This is important for the TR to determine the location of the person with respect to itself to follow the person, i.e., the robot needs to turn left if the person is detected at the left camera image, the robot stay or move forward if the person is detected at the middle of the camera, and the robot needs to turn right if the person is detected at the right of the camera image. This will be discussed further in Section 2.4.

2.3 Distance Estimation of The Detected Person

To enable the person-following feature, the TR needs to estimate the distance between itself and the detected person. The distance estimation can be carried out using several approaches such as stereo cameras and monocular cameras. In this study, a low-cost monocular camera is used to detect and estimate the distance between TR and the person. An algorithm is proposed to calculate the estimated distance based on the focal length information, and the width of the person using the bounding box produced by YOLOv4. Figure 3 illustrates the method used for finding the focal length information of the monocular camera where $h, d, a,$ and f are the object's height, the distance between the object and lens, image height, and focal length, respectively. On the left side of the plane, the image of the object (person) captured by the camera lens is inverted.

The angle θ_1 and θ_2 between the actual person and the image with the camera lens are given as:

$$\tan\theta_1 = \frac{h}{d} \tag{1}$$

$$\tan\theta_2 = \frac{a}{f} \tag{2}$$

Since the angles θ_1 and θ_2 are equal, Equations (1) and (2) can be equated as:

$$\frac{h}{d} = \frac{a}{f} \tag{3}$$

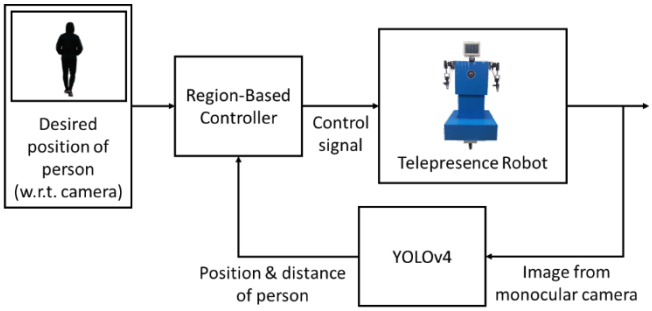

Figure 4. Control block diagram of the TR.

$$f = \frac{ad}{h} \tag{4}$$

Once the focal length of the lens f is obtained, the unknown distance of the targeted person can be measured by deriving the previous focal length equation. However, rather than relying on height, the width of a person is utilized for estimating distance. This is because the width of a person exhibits less variability across different individuals compared to height. Additionally, the camera employed typically captures only a portion of a person's full height due to the proximity between the robot and the individual. The estimated distance based on the width of the person is given as follows:

$$d^* = \frac{wf}{B_x} \tag{5}$$

where d^* is the estimated distance, w is the actual width of a person, and B_x is the width of the bounding box obtained from YOLO.

2.4 Controller Design: Region-Based Controller

In this paper, a region-based controller is proposed that enables the robot to move based on the position of the detected person in the image captured by the camera. Figure 4 shows the closed-loop control system that consists of the TR as a plant, a region-based controller, visual feedback from a monocular camera, and the desired position of the person with respect to the robot. The visual feedback from the monocular camera is used by YOLOv4 for person detection and distance estimation before entering the region-based controller. The inputs to the controller are the position (actual and desired) and distance of the detected person, as described in Sections 2.2 and 2.3, respectively. The output of the controller is the control command sent to the actuators (motors).

The region-based controller is a control strategy used in this person-following application, where the robot's movement is determined based on the person's position within different regions of the image captured by a monocular camera, as well as the distance between the person and the robot. This controller divides the image into three distinct regions: left, right, and middle. When the person is detected in the left region of the image, the controller instructs the robot to turn left. This is achieved by activating the robot's left-turning mechanism, causing it to change its orientation towards the left direction. Similarly, when the person is detected in the right region of the image, the controller commands the robot to turn right. The robot's right-turning mechanism is activated to adjust its orientation towards the right direction. The corresponding control input-output relation is tabulated in Table 1.

The region-based controller provides an intuitive method for controlling the robot's movements based on the person's position within different regions of the captured image. By dividing the image into distinct regions, the controller enables the robot to respond appropriately and navigate toward the person more naturally and responsively. If the person is detected in the middle region of the image, the controller determines the appropriate movement based on the person's distance within that region. For example, if the person is too close to the robot, i.e., less than 1 m, the controller instructs the robot to move backward. If the distance between the robot and the person is between 1 m to 5 m, the controller may instruct the robot to move forward. Lastly, if the distance between the robot and the person is too far, i.e., more than 5 m, the controller may instruct the robot to stop. The algorithm for the controller used in this study is shown in Algorithm 1.

The algorithm begins by initializing ROS publishers to facilitate communication between the robot and its environment. Constants are then set to establish parameters for subsequent operations. Following this, the YOLOv4 model is loaded into OpenCV's DNN module, enabling the robot to perform object detection tasks. The algorithm then enters a frame processing loop where it captures frames from the camera. For each frame, it employs the object detection function to identify objects, particularly focusing on detecting people. Upon detecting a person, the algorithm calculates their distance from the camera using the focal length and object width as discussed in the previous section. Based on this distance and the person's position within the frame, the algorithm determines the appropriate movement direction for the robot. Twist messages are then published to control the robot's movement accordingly. Once all frames have been processed, the camera capture is released, and any open OpenCV windows are closed, concluding the algorithm's execution.

Table 1. Input-output relation.

Position of the Person in the Image	Estimated Distance, d^*	Robot Movement
Middle Region	$d^* \leq 1$	Backward
Middle Region	$1 < d^* \leq 5$	Forward
Middle Region	$d^* > 5$	Stop
Right Region	d^*	Right
Left Region	d^*	Left

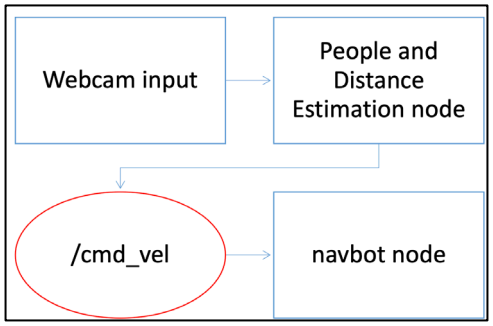

Figure 5. ROS RQT Graph of the TR.

```
Algorithm 1: Person-following using a monocular camera and YOLOv4.
Initialize ROS Publishers
Set Constants
Load YOLOv4 Model in OpenCV's DNN module

Start Frame Processing Loop:
    Read the frame from the camera
    data = object_detector(frame)
    For each detected object in the data:
        if the object is a person:
            Calculate distance of person from camera using focal length and object width
            Determine robot's movement direction based on person's position and distance
            Publish twist messages to control robot's movement
    Display bounding boxes, labels, and distance information on the frame
Release Camera Capture
Close OpenCV Windows
```

For the implementation of the algorithm, ROS is used for hardware abstraction and low-level device control. Figure 5 shows the communication between nodes and topics involved in the implementation of TR using ROS. The input images captured from the webcam are fed into the People and Distance Estimation node from Algorithm 1. In this node, the YOLOv4 model is loaded using OpenCV's DNN module. The algorithm continuously captures frames from the camera, detects persons within each frame, and calculates the distance of the detected person from the camera. Then, it determines the appropriate movement direction for the robot based on the position and distance of the person and subsequently publishes twist messages to control the robot's movement accordingly. This twist message is sent through the /cmd_vel topic, which is then subscribed by the navbot node that controls the movement of the TR. The source code of algorithms used in the TR is available at: https://github.com/minfirdaus/PERSON-FOLLOWING-ROBOT-USING-YOLOV4/.

3. RESULTS AND DISCUSSION

In this section, experimental results of the proposed person-following TR are presented. First, the accuracy of the person detection algorithm with respect to distance is discussed, followed by the performance of the proposed distance estimation. Finally, results for the deployment of the proposed algorithm are presented. In this study, the reference distance and the reference width of the person are set to 1.143 m and 0.406 m, respectively.

3.1 Person Detection Accuracy with Respect to Distance

The TR needs to accurately detect a person within a defined operating range to safely follow the target person. To test the accuracy of the YOLOv4 in detecting a person with respect to distance, an experiment was conducted by recording the probability of the person being detected. In this case, the probability values were recorded for every 0.2 m, starting from 0.9 m to 3.3 m distance between the camera and the person in an indoor environment with optimal lighting. Figure 6 illustrates the output of the experiment conducted. A bounding box highlights the detected person and indicates the location of the person in the image in terms of height, width, and center of the bounding box. Meanwhile, the probability of a person being detected is indicated above the bounding box with a value between 0 to 1.

Figure 6. Experiment setup for person detection and distance estimation.

Table 2. Comparison between the actual distance and estimation distance.

Actual Distance (m)	Estimated Distance (m)
0.90	0.91
1.10	1.13
1.30	1.35
1.50	1.54
1.70	1.73
1.90	1.94
2.10	2.20
2.30	2.30
2.50	2.57
2.70	2.76
2.90	2.93
3.10	3.20

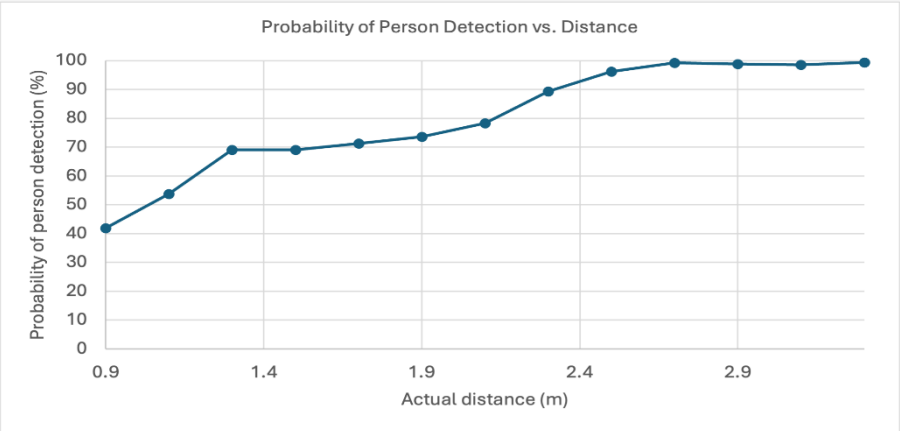

Figure 7. Accuracy of the person detection with respect to distance.

Figure 7 shows the graph for person detection accuracy with respect to the actual distance between the monocular camera and the person. It shows that the accuracy in detecting a person increases as the distance between the camera and the person increases. The accuracy begins to increase gradually at 2.3 m and remains constant above 98 percent starting at 2.7 m. This is because as the person gets closer to the camera, only a small part of their body is captured in the bounding box, lowering the detection's accuracy. At 2.3 m, the bounding box from YOLOv4 detects two-thirds of the person's body and at 2.7 m, the bounding box detects almost the entire body. The bounding box is responsible for detecting the coordinates and probability of the object in each cell produced by image separation. It suppressed the bounding box with the low probability score and remained the bounding box with the highest probability score only. Furthermore, the accuracy of the detection for the range 1 m to 3 m is all above 50% percent which is good for our application since the detection and following action happens in this range. The accuracy of person detection is important to ensure that the robot follows the real person rather than other objects, especially in areas with poor lighting which can reduce accuracy. The output bounding box is also important for distance estimation as discussed in the next section.

In this study, experiments were conducted solely with a single individual, primarily because the implemented pre-trained YOLO using the MS COCO dataset emphasizes person detection rather than individual identification. However, to enhance YOLO's capabilities for person identification, future research may propose the integration of supplementary algorithms.

3.2 Distance Estimation using Monocular Camera

The experiment is carried out in a similar way as in section 3.1 to test the accuracy of the distance estimation algorithm using a monocular camera. Table 2 and Figure 8 both show the comparison between the actual distance and estimation distance. The estimation distance produced by the algorithm is quite accurate with only slight differences. The greatest difference between the estimation and actual distance is 0.10 m for distances 3.10 m and 2.10 m. The accuracy of the algorithm is defined by the width of the boundary box detecting the person. The position of the hand is critical in determining its accuracy. The estimation distance becomes less accurate as the hand is raised wider.

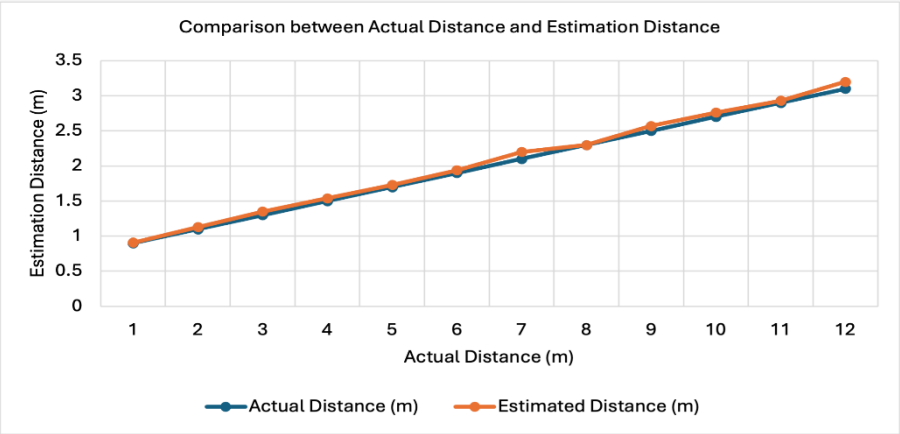

Figure 8. The comparison between the actual distance and estimated distance.

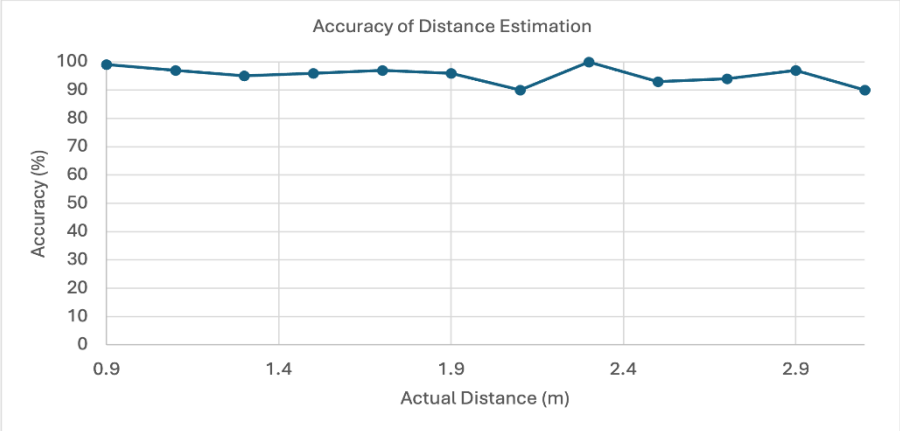

Figure 9. The accuracy of the distance estimation algorithm.

(a) (b) (c)

Figure 10. The output of person detection using YOLO in bounding boxes (a) person at the left camera image; (b) person at the middle of the camera image; (c) person at the right side of the camera image.

Figure 9 illustrates the accuracy of the distance estimation performed by the algorithm. All of the readings achieved an accuracy above 95%. Thus, this proves the algorithm is reliable and accurate enough to be used in the project. The framerate has also been increased by utilizing the CUDA backend and Nvidia GPU computation. The algorithm takes 6 seconds to estimate distance using CPU-only computation. By enabling and compiling the CUDA backend together with OpenCV, the delay time was reduced to one second. This gave enough time for the TR to follow the targeted person in a real-time application.

3.3 Validation of Region-Based Controller

After the detection and distance estimation of the person are successfully validated as discussed in the previous sections, the proposed region-based controller is deployed on the TR to enable the autonomous person-following feature for the TR. First, the capability of the robot is tested to detect a person based on regions left, middle, or right regions. Then, the controller prints the output for the motors whether to move left, move forward or move right. The results of the experiment are shown in Figure 10 which shows the capability of the robot to detect the person and give a correct command to the actuator.

Figure 11. The TR successfully detects and follows the targeted person with the correct measurement.

Consequently, the TR's movement is tested to ensure the correct command is produced. The TR is programmed to follow a person in a range of 1 to 5 m and to move backward when the distance is less than 0.5 m when the person stands or walks in front of the robot. From the experiment, it is found that the TR managed to autonomously follow the detected person as shown in Figure 11. Video of the demonstration is available at: https://youtu.be/Z5JlozXGDvc.

4. CONCLUSION

In conclusion, this study presents an autonomous person-following TR utilizing a monocular camera and deep learning YOLO for person detection and distance estimation. A novel distance estimation algorithm based on focal length and person width is introduced to compensate for the monocular camera's lack of depth information. The proposed method incorporates a region-based controller for robot movement control based on the detected person's location in the camera image. Integration and deployment of the method in the TR are achieved using the Robot Operating System (ROS). Experimental results validate the effectiveness of the proposed algorithm, demonstrating the TR's successful person-following capability. This research contributes to advancing telepresence technology, enhancing remote communication, and facilitating collaboration in situations where physical presence is not feasible. Note that the pre-trained YOLO using the MS COCO dataset will detect any human in the image without identifying which person should be followed. Thus, future works involve person identification which allows only one person to be followed at one time. Also, a performance comparison between the current version of YOLO and others will be made.

ACKNOWLEDGEMENT AND FUNDING

The authors would like to thank Universiti Sains Islam Malaysia (USIM), Malaysia for the financial support through the research grant PPPI/FKAB/0122/USIM/12522.

DECLARATION OF CONFLICTING INTERESTS

The authors declare no potential conflicts of interest with respect to the research and publication of this article.

REFERENCES

[1] A. U. Batmaz, J. Maiero, E. Kruijff, B. E. Riecke, C. Neustaedter and W. Stuerzlinger, How automatic speed control based on distance affects user behaviours in telepresence robot navigation within dense conference-like environments, *PLoS One*, 15(11), 2020, e0242078.

[2] S. D. Sierra Marín *et al.*, Expectations and perceptions of healthcare professionals for robot deployment in hospital environments during the COVID-19 pandemic, *Frontiers in Robotics and AI*, 8, 2021, 102.

[3] L. Riano, C. Burbridge and T. M. McGinnity, A study of enhanced robot autonomy in telepresence, *Proceedings of Artificial Intelligence and Cognitive Systems,* 2011, 271–283.

[4] L. Yang, B. Jones, C. Neustaedter and S. Singhal, Shopping over distance through a telepresence robot, *Proceedings of the ACM on Human-Computer Interaction*, 2 , 2018, 1-18.

[5] A. Di Nuovo *et al.*, The multi-modal interface of Robot-Era multi-robot services tailored for the elderly, *Intelligent Service Robotics*, 11(1), 2018, 109-126.

[6] A. Cosgun, D. A. Florencio and H. I. Christensen, Autonomous person following for telepresence robots, *IEEE International Conference on Robotics and Automation*, Macau, China, 2013, 4335-4342.

[7] A. Leigh, J. Pineau, N. Olmedo and H. Zhang, Person tracking and following with 2D laser scanners, *IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, USA, 2015, 726-733.

[8] M. Cen, Y. Huang, X. Zhong, X. Peng and C. Zou, Real-time obstacle avoidance and person following based on adaptive window approach, *IEEE International Conference on Mechatronics and Automation (ICMA)*, Tianjin, China, 2019, 64-69.

[9] J. Cai and T. Matsumaru, Human detecting and following mobile robot using a laser range sensor, *Journal of Robotics and Mechatronics*, 26(6), 2014, 718-734.

[10] A. H. Adiwahono *et al.*, Human tracking and following in dynamic environment for service robots, *IEEE Region 10*

Conference, Penang, Malaysia, 2017, 3068-3073.
[11] K. Koide and J. Miura, Identification of a specific person using color, height, and gait features for a person following robot, *Robotics and Autonomous Systems*, 84, 2016, 76-87.
[12] G. Doisy, A. Jevtić and S. Bodiroža, Spatially unconstrained, gesture-based human-robot interaction, *8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Tokyo, Japan, 2013, 117-118.
[13] F. Basso, M. Munaro, S. Michieletto, E. Pagello and E. Menegatti, Fast and robust multi-people tracking from RGB-D data for a mobile robot, *Intelligent Autonomous Systems*, 12, 2013, 265-276.
[14] X. Cheng, Y. Jia, J. Su and Y. Wu, Person-following for telepresence robots using web cameras, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, 2019, 2096-2101.
[15] K. Koide, J. Miura and E. Menegatti, Monocular person tracking and identification with on-line deep feature selection for person following robots, *Robotics and Autonomous Systems*, 124, 2020, 103348.
[16] M. Bajracharya, B. Moghaddam, A. Howard, S. Brennan and L. H. Matthies, A fast stereo-based system for detecting and tracking pedestrians from a moving vehicle, *The International Journal of Robotics Research*, 28(11-12), 2009, 1466-1485.
[17] B. X. Chen, R. Sahdev and J. K. Tsotsos, Integrating stereo vision with a CNN tracker for a person-following robot, *International Conference on Computer Vision Systems (Lecture Notes in Computer Science)*, 2017, 300-313.
[18] J. Satake and J. Miura, Robust stereo-based person detection and tracking for a person following robot, *ICRA Workshop on People Detection and Tracking*, Kobe, Japan, 2009, 1-10.
[19] M. J. Islam, J. Hong and J. Sattar, Person-following by autonomous robots: A categorical overview, *The International Journal of Robotics Research*, 38(14), 2019, 1581-1618.
[20] B. Bhavya Sree, V. Yashwanth Bharadwaj and N. Neelima, An inter-comparative survey on state-of-the-art detectors-R-CNN, YOLO, and SSD, *Intelligent Manufacturing and Energy Sustainability*, 2021, 475-483.
[21] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, You only look once: Unified, real-time object detection, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, 779-788.
[22] P. Bharati and A. Pramanik, Deep Learning techniques R-CNN to mask R-CNN: A survey, *Computational Intelligence in Pattern Recognition*, 2020, 657-668.
[23] H. Deshpande, A. Singh and H. Herunde, Comparative analysis on YOLO object detection with OpenCV, *International Journal of Research in Industrial Engineering*, 9(1), 2020, 46-64.
[24] M. Li, Z. Zhang, L. Lei, X. Wang and X. Guo, Agricultural greenhouses detection in high-resolution satellite images based on convolutional neural networks: comparison of faster R-CNN, YOLO v3 and SSD, *Sensors*, 20(17), 2020, 4938.

.