Design and Development of a Human Following Robot meant to assist the Elderly

Kelly Atieno Diang'a, Dr Evon Lim
Swinburne University of Technology,
Sarawak Campus
Faculty of Engineering, Computing and Science
Jalan Simpang Tiga,93350 Kuching,
Sarawak, Malaysia
kellydianga@gmail.com,
ewlim@swinburne.edu.my

***Abstract*— This paper provides a comprehensive examination of existing human-following robots, focusing on their human detection and tracking methods. It proposes the development of a Human Following Robot designed to autonomously assist the elderly in their daily activities, including carrying their belongings. The project integrates advanced technologies such as TensorFlow Lite and OpenCV for marker detection and human tracking, along with cameras, and sophisticated algorithms to ensure seamless interaction with the locomotion system and ensure efficient transportation of goods. The paper also includes an in-depth overview of the development process.**

Keywords— Human-following robot, human detection, human tracking, elderly care, TensorFlow Lite, OpenCV, cameras, algorithms, locomotion system.

I. INTRODUCTION

Human Following Robots represent a significant advancement in robotics technology. They possess the capability to autonomously detect and track humans using various sensors such as cameras, infrared devices, and lasers, particularly in environments prioritizing safety and security, such as crowded areas or public events. However, despite their recognized potential in aiding the elderly population, there remains a notable gap in leveraging this technology to enhance the quality of life for older individuals [1].

This research project aims to investigate the current applications of human-following robots and their effectiveness in meeting the specific needs of the elderly population. Key research questions include exploring the current applications of human-following robots and identifying the challenges in utilizing this technology for elderly care. Additionally, the project aims to design and develop a prototype tailored to assist older individuals in their daily activities.

II. LITERATURE REVIEW

A. Background Study

Human following robots have found valuable applications in various fields like healthcare, retail, and entertainment. For instance, in hospitals, these robots can assist nurses and doctors by carrying medical supplies, ensuring that the staff can focus more on patient care. Aethon's TUGw is an autonomous robotic delivery system designed for hospitals. This mobile robot navigates hospitals independently and can use elevators, open doors, avoid obstacles, and alert people to move aside if necessary [2]. In transportation, they can assist in guiding people and goods safely. An example is the Smart Luggage Robot, designed to assist airport passengers with luggage. Equipped with GPS, the robot autonomously navigates to users, following them to check-in or exit while avoiding obstacles [3].

When it comes to elderly care, previous research has explored robots designed to assist older individuals in daily tasks. Reference [4] studied older people's needs for mobile service robots, focusing on enhancing independence, social interaction, and recreational activities. In [5] a robotic arm is incorporated into a care facility's mobile service robot, finding high acceptance among older individuals. The study also evaluated older people's interactions with an autonomous mobile robot in tasks like water delivery and walking encouragement, indicating the robot's usefulness. Reference [6] investigated older adults' acceptance of a humanoid robot as a walking partner, with participants showing a preference for walking with the robot. Later research confirmed the positive impact of a similar robot [7], especially among cognitively impaired adults [1].

B. Human detection and Tracking using CCD cameras and Image Processing Algorithm

The designed robot, ApriAttenda^{TM}, uses two CCD cameras to capture the target person's image. Higher resolution images are obtained through stereo vision, ensuring accurate recognition of the target. An image processing algorithm extracts and recognizes the individual using feature points from the input image. Stereo vision calculates motion velocities and measures the distance between the robot and the target. By evaluating feature parameters, the robot selects the optimal region for target detection. Once identified, the system recognizes the target based on pre-registered clothing information such as color and texture. Processed data, including distance and direction, is sent to the Motion Control Module. The robot adjusts its movements to maintain a constant separation distance, moving forward, stopping, or backing off as needed. [8]

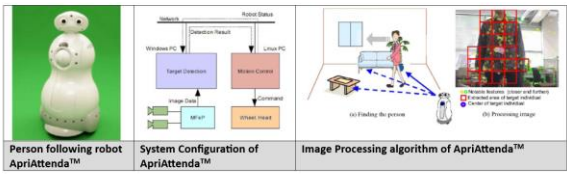


Fig. 1. Appritenda^{TM} system configuration

C. Using an IR active marker tracking system and a depth camera

Active markers are descibed as wearable designs or patterns equipped with electronically powered Infra-Red Light Emitting Diodes. One major advantage they have over passive markers is their ability to provide a brightly lit pattern less affected by surrounding light. They can also be tuned to specific frequencies, to further minimize false readings. However, the additional electronics required to operate them might be intrusive for the wearer. In [9], an IR camera such as the Sony PlayStation Eye, is used to track IR beacons through a special lens which removes visible light. Picking up only IR in its input eliminates the need for extensive image processing. The active marker, made of IR LEDs, emits light invisible to the human eye in a specific pattern and can be embedded into a vest worn by a tracked person. This approach minimizes distractions to the user and the surroundings and ensures the robot tracks the correct target.

D. Using Infra-Red Camera on human and LED markers on the Robot

In a study by Quoc Khanh and Young Soo, a WII camera is attached to the target human to capture four groups of IR-LEDs mounted on the robot. When the camera detects all of the four IR-LEDs on the mobile robot, the camera's position and orientation relative to the robot's coordinate system and vice versa is calculated. After getting camera's position and orientation information, a virtual link between human and robot, such as the distance between them, is created. To maintain this virtual link, the robot thus moves the same distance as the target human.
This method is advantageous in that high accuracy of relative position of a human and a robot is still obtained despite gentle lighting. The Nintendo WII camera is also affordable making the project easily applicable in real life scenarios[10].

III. METHODOLOGY

The methodology of this project is divided into three main components: Human Detection, Human Tracking, and Programming the Robot. Human Detection involves using TensorFlow for marker detection, which is worn by the target human. Human Tracking utilizes template matching for distance calculation and object location. Based on the results from human detection and human tracking, the robot is then programmed to follow the human or marker.

A. Human Detection

A belt containing a specific pattern is worn by the target human. The pattern consists of a pink rectangular box attached to the belt, which serves as a visual marker for the robot to detect and follow.

Fig. 2. Belt with the marker

To enable the robot to recognize this pattern, a custom TensorFlow Lite model, EfficientDet Lite0, is used due to its efficiency and accuracy, making it ideal for deployment on resource-constrained devices such as the Raspberry Pi 4.

Model Training Process:

1. *Data Preparation:*

Over 350 images of the target pattern are collected from the deployment environment. The images include various angles, lighting conditions, and backgrounds to ensure a comprehensive training dataset.
The images are annotated with bounding boxes labeled as 'marker' to indicate the location of the pink rectangular box. This annotation process creates a labeled dataset saved as XML files, essential for supervised learning.

2. *Model Specification:*

TensorFlow Lite Model Maker tool for creating TensorFlow Lite models, is employed to simplify the process and reduce the amount of code required.
The model specification used is EfficientDet Lite0, known for its lightweight architecture suitable for mobile and edge devices, ensuring the final model can operate efficiently on the Raspberry Pi 4 utilized in the robot.

3. *Training Setup:*

The training process was conducted in a Jupyter Notebook environment using Google Colab. Google Colab provided the necessary computational resources and GPU support.
Python 3.9 environment was also used for its compatibility with TensorFlow and other dependencies.
The training parameters are set with a batch size of 4 and epoch count of 20, balancing the training time and the model's performance.

4. *Code Implementation:*

A training script, train.py, was used to load the training and validation data using the Pascal VOC format, common for object detection tasks.
The object detector is created and trained using the specified model (EfficientDet Lite0) and parameters (batch size and epochs).
After training, the model is evaluated on validation data to assess its accuracy and performance.
The final trained model was exported as a TensorFlow Lite model (best.tflite), which was then deployed on the robot for real-time detection.

B. Human Tracking

This method combines template matching and real-time object detection to effectively track and measure the distance of the marker from the camera and its x coordinate position within the camera frame. The following processing were applied to achieve the two parameters:

1. *Template Creation*

A reference image of a marker is captured by the webcam at a predefined distance of 60 cm. The actual width of the marker, which is 3 cm, serves as a basis for subsequent calculations.

2. *Image Processing Using OpenCV*

Identify applicable funding agency here. If none, delete this text box.

To isolate and measure the marker in the reference image, several image processing techniques are applied. First, the reference image is converted to the HSV color space, and a color mask is applied to isolate the marker based on its color range. Then, a binary threshold is applied to the masked image, creating a binary representation that highlights the marker. Next, contours, which represent the boundaries of the marker, are detected within the binary image. Finally, bounding boxes are calculated around each detected contour to determine the width of the marker.

3. Focal Length Calculation

Using the known distance (60 cm), the real width of the marker (3 cm), and the width of the marker in the reference image, the focal length of the camera is calculated with the formula:

$$Focal\_legth = \frac{width\_in\_rf\_image \times known\_distance}{actual\_width}$$

4. Real-Time Distance Calculation

With the focal length known, the actual distance of the marker can be calculated in real-time. The width and the x coordinate of the marker in the current image frame is obtained from MediaPipe's object detection properties. The distance is computed using the formula:

$$Obj\_distance = \frac{actual\_width \times Focal\_length}{obj\_width\_in\_frame}$$

5. Real-Time Tracking Implementation

The calculated distance and x-coordinate are transmitted via UART (serial communication) to the ESP32 for motor control purposes.

C. Robot Programming

An Arduino code was written to control the two 12V motors by an ESP32 microcontroller, utilizing an L298N motor driver. The purpose was to enable precise motor control to maneuver a robot effectively. A serial communication was set up between ESP32 and Raspberry Pi 4. During operation, the code continuously receives x coordinate and the actual distance of the marker from the robot, via serial communication. It then processes this data to adjust the motor speeds accordingly, ensuring the robot maintains a desired distance from the object. Additionally, mapping is performed to adjust the motor speeds, allowing the robot to move left or right to ensure the marker remains at the center position within the frame. This dynamic adjustment enhances the robot's ability to follow the marker accurately and effectively within its environment.

D. Developed Mechanical System

The mechanical setup of the robot consists of a webcam attached to an aluminum rod for support. An acrylic sheet serves as the chassis, to which four wheels are attached for movement. A basket with two compartments is also mounted on the chassis: the lower compartment stores the electronic components, while the upper compartment is designated for the target person's belongings.

Fig. 3. Mechanical Design of the robot

E. Electronic System Design

The electronic system comprises of Raspberry Pi 4 which is powered by a power bank. An ESP32 is connected to the Raspberry Pi 4, facilitating communication and control of the motor driver. The L298N motor driver, connected to the ESP32, controls the two 12V DC motors. A LiPo battery is also used to power the motors, supplying a maximum voltage of 12.6V.

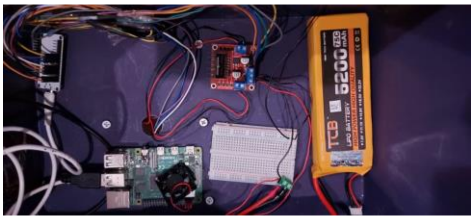


Fig. 4. Electronics System

IV. RESULTS AND DISCUSSION

A. Tflite Model Training and Testing

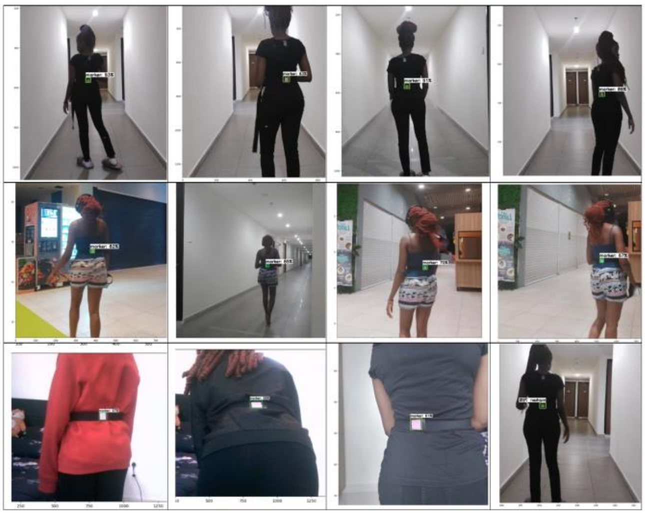
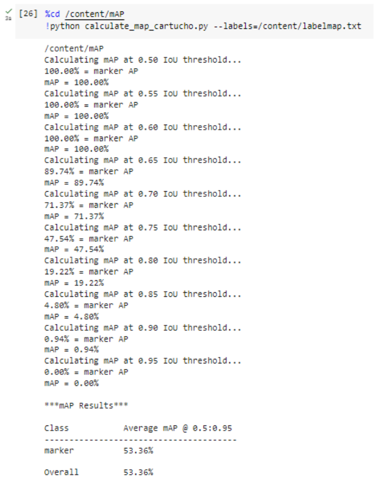


Table 1. Model Testing results after training on test images

The table above showcases the outcomes of EfficientDet Lite0 performance when subjected to testing using a dataset comprising images never encountered during the training phase. This evaluation allowed for an objective assessment of the model's ability to generalize to unseen data, mimicking real-world conditions. By analyzing the model's behavior in an unfamiliar environment, I was able to gain valuable insights into its adaptability and reliability, crucial for its practical deployment. The results obtained from this testing phase served as a benchmark, offering confidence in the model's potential to deliver consistent performance when deployed in the intended environment.


```
[26] %cd /content/mAP
     !python calculate_map_cartucho.py --labels=/content/labelmap.txt

/content/mAP
Calculating mAP at 0.50 IoU threshold...
100.00% = marker AP
mAP = 100.00%
Calculating mAP at 0.55 IoU threshold...
100.00% = marker AP
mAP = 100.00%
Calculating mAP at 0.60 IoU threshold...
100.00% = marker AP
mAP = 100.00%
Calculating mAP at 0.65 IoU threshold...
89.74% = marker AP
mAP = 89.74%
Calculating mAP at 0.70 IoU threshold...
71.37% = marker AP
mAP = 71.37%
Calculating mAP at 0.75 IoU threshold...
47.54% = marker AP
mAP = 47.54%
Calculating mAP at 0.80 IoU threshold...
19.22% = marker AP
mAP = 19.22%
Calculating mAP at 0.85 IoU threshold...
4.80% = marker AP
mAP = 4.80%
Calculating mAP at 0.90 IoU threshold...
0.94% = marker AP
mAP = 0.94%
Calculating mAP at 0.95 IoU threshold...
0.00% = marker AP
mAP = 0.00%

***mAP Results***

Class           Average mAP @ 0.5:0.95
---------------------------------------
marker          53.36%

Overall         53.36%
```

Fig. 5. Mean Precision results after training

The moderate results observed in the mean Average Precision (mAP) can be attributed to several factors. Firstly, the limited training resources available, particularly the constrained training duration of 20 epochs due to resource limitations in Google Colab, might have hindered the model's ability to fully converge and optimize its parameters. This could have led to suboptimal performance in terms of object detection accuracy. Additionally, the dataset used for training lacked diversity in terms of backgrounds, scenarios, and object variations. This limited variation in the dataset could have also contributed to below optimal generalization of the model to unseen environments, resulting in lower mAP scores. Furthermore, the mAP metric's sensitivity to small variations in detection performance could have also influenced the observed results, especially given the model's constrained training conditions. Overall, these factors collectively contributed to the moderate performance observed in the mAP metric, emphasizing the importance of adequate training resources and diverse datasets in achieving higher object detection accuracy.

B. Template Matching Accuracy

In assessing the accuracy of the template matching method, the actual distance between the marker and the camera was measured at 30 cm. Subsequently, the calculated distance from the Raspberry Pi to the marker was determined to be 29.189 cm which yielded a discrepancy of 0.811 cm between the calculated and actual distances. Despite this minor error, the template matching method demonstrated high precision in estimating spatial relationships, making it a viable option for human tracking applications within the developed system. The observed error suggests that while the template matching method provides relatively accurate distance estimations, there is room for improvement to achieve precise results. Factors such as image quality, lighting conditions, and the template's alignment with the actual object may have contributed to the discrepancies in distance calculations. Thus, further refinement and optimization of the template matching algorithm is needed to enhance its accuracy and reliability in real-world applications.

Fig. 6. Actual distance of robot from marker measured at 30cm

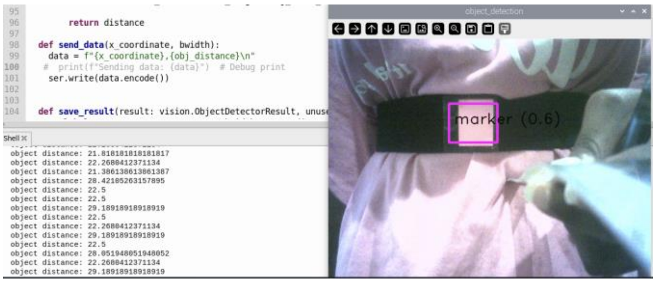


Fig. 7. Calculated distance in Raspberry Pi 4

C. Robot Movement and Performance

In assessing the movement and performance of the robot, several challenges were encountered. One notable issue was the configuration of the back wheels, which shared a common shaft, making it challenging for the robot to execute turns effectively. Furthermore, the absence of additional support on the shafts caused the wheels to point outwards, particularly when the robot bore additional weight. Despite these mechanical challenges, the robot exhibited satisfactory functionality in terms of its electrical components. However, the detection process was hindered by the camera's frames per second (FPS), resulting in slower detection speeds. This limitation impeded the robot's smooth movement and tracking performance.

Nevertheless, despite these obstacles, the robot showcased resilience by maintaining its ability to move and effectively follow the marker.

Conclusion

In conclusion, this project has successfully achieved its objectives of designing and developing a human-following robot prototype tailored to assist the elderly in their daily activities. Through the integration of advanced technologies such as OpenCV and Tensorflow, the robot demonstrated promising potential in enhancing the quality of life for older individuals. However, while the project marks a significant milestone, there are avenues for future improvements and enhancements.

Moving forward, several areas present opportunities for refinement. Firstly, attaching the camera to a stepper motor would enable rotational movement, thereby increasing the robot's field of view and enhancing its ability to detect and track the target individual. Additionally, further training of the TensorFlow Lite model is needed to improve its accuracy in marker detection, thereby enhancing the robot's responsiveness and precision.

Furthermore, upgrading the motors to support more weight would increase the load capacity of the robot, enabling it to carry heavier items and provide more substantial assistance to the elderly. Incorporating ultrasonic sensors could enhance the robot's obstacle avoidance techniques, enabling it to navigate complex environments more effectively and ensuring the safety of both the robot and its user.

Moreover, incorporating a Bluetooth tracking method on top of the camera would increase the robustness of tracking the human, even when not in the camera's view. Additionally, introducing a mobile application to control the robot and adjust various parameters, such as the distance between the robot and the human, as well as receiving alerts when the robot cannot detect or track the human, would enhance user interaction and customization.

By addressing these areas of improvement, the human-following robot can be refined into a more robust and efficient solution for elderly care, paving the way for its successful deployment and widespread adoption in real-world settings.

ACKNOWLEDGMENT

I am deeply grateful to my supervisor, Dr. Evon, for her patience, guidance, and insightful feedback, which was instrumental in achieving the objectives of this research. I am thankful to Swinburne University of Technology, Sarawak campus, and the lab technicians for their invaluable assistance during the robot construction. Special appreciation goes to Mr. Nazir for his technical expertise and practical suggestions, crucial to the project's success. Lastly, I thank my friends and family for their support and encouragement, which helped me stay focused and committed throughout the project.

REFERENCES

[1] Li, S, Milligan, K, Blythe, P, Zhang, Y, Edwards, S, Palmarini, N, Corner, L, Ji, Y, Zhang, F & Namdeo, A 2023, 'Exploring the role of human-following robots in supporting the mobility and wellbeing of older people', *Sci Rep*, vol. 13, no. 1, p. 6512.

[2] Bloss, R 2011, 'Mobile hospital robots cure numerous logistic needs', *Industrial Robot: An International Journal*, vol. 38, no. 6, pp. 567-571.

[3] Shashank, MS, Saikrishna, P, Acharya, GP, Reddy, S & Lavanya, P 2022, 'Design and Development of Human Following Autonomous Airport Baggage Transportation System,' *2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, 1-6.

[4] Johnson, MJ, Johnson, MA, Sefcik, JS, Cacchione, PZ, Mucchiani, C, Lau, T & Yim, M 2020, 'Task and design requirements for an affordable mobile service robot for elder care in an all-inclusive care for elders assisted-living setting', *International Journal of Social Robotics*, vol. 12, pp. 989-1008.

[5] Mucchiani, C, Sharma, S, Johnson, M, Sefcik, J, Vivio, N, Huang, J, Cacchione, P, Johnson, M, Rai, R, Canoso, A, Lau, T & Yim, M 2017, 'Evaluating older adults' interaction with a mobile assistive robot,' *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 840-847.

[6] Karunarathne, D, Morales, Y, Nomura, T, Kanda, T & Ishiguro, H 2019, 'Will older adults accept a humanoid robot as a walking partner?', *International Journal of Social Robotics*, vol. 11, pp. 343-358.

[7] Nomura, T, Kanda, T, Yamada, S & Suzuki, T 2021, 'The effects of assistive walking robots for health care support on older persons: a preliminary field experiment in an elder care facility', *Intelligent Service Robotics*, vol. 14, pp. 25-32.

[8] Yoshimi, T, Nishiyama, M, Sonoura, T, Nakamoto, H, Tokura, S, Sato, H, Ozaki, F, Matsuhira, N & Mizoguchi, H 2006, 'Development of a Person Following Robot with Vision Based Target Detection,' *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5286-5291.

[9] Tsun, MTK, Lau, BT, Jo, HS & Lau, SL 2015, 'A human orientation tracking system using Template Matching and active Infrared marker,' *2015 International Conference on Smart Sensors and Application (ICSSA)*, 116-121.

[10] Quoc Khanh, D & Young Soo, S 2011, in 2011 11th International Conference on Control, Automation and Systems, Control, Automation and Systems (ICCAS), 2011 11th International Conference on, pp. 1054-1058.