

# Marked Object-Following System Using Deep Learning and Metaheuristics

Ken Gorro<sup>1</sup>, Elmo Ranolo<sup>2</sup>, Lawrence Roble<sup>3</sup>, Rue Nicole Santillan<sup>4</sup>, Anthony Ilano<sup>5</sup>, Joseph Pepito<sup>6</sup>,  
Emma Sacan<sup>7</sup>, Deofel Balijon<sup>8</sup>

College of Technology, Cebu Technological University, Carmen, 6000, Cebu, Philippines<sup>1,2,3,4,5</sup>

College of Technology, Cebu Technological University, Cebu, Philippines<sup>6,7</sup>

Center for Cloud Computing, Big Data, and Artificial Intelligence<sup>1,2,3,4</sup>

College of Computing, Artificial Intelligence, and Sciences, Cebu Normal University, Cebu, Philippines<sup>8</sup>

**Abstract**—This paper presents a deep learning methodology for a marked object-following system that incorporates the YOLOv8 (You Only Look Once version 8) object identification model and an inversely proportional distance estimation algorithm. The primary aim of this study is to develop a marked object-following algorithm capable of autonomously tracking a designated marker while maintaining a suitable distance through advanced computer vision techniques. In this study, a marked object is defined as an object that is explicitly labeled, tagged, or physically marked for identification, typically using visible markers such as QR codes, stickers, or distinct added features. Central to the system's functionality is the YOLOv8 model, which detects objects and generates bounding boxes around identified target classes in real-time. The proposed marked object-following algorithm utilizes the distance estimation method, which leverages fluctuations in the bounding box width to determine the relative distance between the observed user and the camera. A pathfinding algorithm was created using tabu search and a-star to avoid obstacle and generate a path to continue following the marker object. Furthermore, the system's efficacy was assessed using critical performance metrics, including the F1-score and Precision-Recall. The YOLOv8 model attained an F1-score of 0.95 at a confidence threshold of 0.461 and a mean Average Precision (mAP) of 0.961 at an IoU threshold of 0.5 for all target classes. These results indicate a high level of accuracy in object detection and tracking. However, it is important to note that this algorithm has close door and controlled environments.

**Keywords**—Object detection; YOLOv8; distance estimation; A-star; tabu search

## I. INTRODUCTION

In today's technology-driven world, artificial intelligence (AI) and robotics are revolutionizing various domains, including human interaction and navigation. Autonomous systems capable of tracking and following individuals are highly beneficial in settings such as crowded environments, warehouses, and other dynamic areas. These systems have the potential to enhance efficiency and safety by providing precise and adaptive navigation in real-time. Recent studies have highlighted the advancements in deep learning techniques, particularly in object detection, which significantly improve the capabilities of these systems in complex environments [1][2][3]. The motivation behind this research arises from challenges faced in environments where autonomous systems must reliably track a marked object, particularly in dynamic and crowded areas. Traditional systems have frequently struggled to effectively identify and focus on the correct item in such situations

due to occlusions, competing visual elements, and contextual complications [4][5]. These constraints underscore the need for a stronger and more efficient system.

This study introduces a marked object-following algorithm that integrates deep learning and metaheuristic techniques to address these challenges. Leveraging the YOLOv8 (You Only Look Once version 8) object detection framework, the system ensures robust object recognition and tracking. YOLO has been recognized for its ability to perform real-time object detection with high accuracy, making it suitable for dynamic environments [6] [7]. A distance estimation algorithm based on fluctuations in bounding box width enables the system to maintain an optimal distance from the marked object, thereby preventing collisions. Furthermore, the inclusion of A-star pathfinding and the Tabu Search metaheuristic algorithm enhances the system's ability to navigate around obstacles and generate efficient paths in real-time scenarios [8] [9].

The development of a reliable and effective marked object-following system that can track a designated marker on its own while adjusting to environmental changes is the main goal of this research. A marked object is defined as an object that is explicitly labeled, tagged, or physically marked for identification, typically using visible markers such as QR codes, stickers, or distinct added features. This project aims to use YOLOv8 to create a reliable marked object-following algorithm for object tracking and detection in real-time. Additionally, the design and implementation of a distance estimation method that dynamically calculates the relative distance between the observed user and the camera are crucial components of this research. The integration of pathfinding algorithms, such as A-star and Tabu Search, enables obstacle avoidance and efficient navigation [8] [10].

Furthermore, this research is significant because it addresses the growing need for intelligent and adaptive tracking systems in real-world applications. By combining advanced deep learning models with metaheuristic algorithms, the proposed system offers a novel solution that ensures accuracy, reliability, and adaptability. The findings of this study aim to contribute to the advancement of autonomous tracking technologies, paving the way for their deployment in diverse practical scenarios [11] [2]. The integration of deep learning techniques in object detection has shown promising results, enhancing the performance of tracking systems in complex environments [2] [12].

## II. RELATED STUDIES

In the past few years, LiDAR technology has become a key way of detecting people in robotic systems. Some researchers have been employing LiDARs for precise individual detection and tracking using the ability to measure the distance and location of the target person versus the robot. LiDAR sensors provide 3D data with high resolution, which allows robots to identify and follow a specific human target who is even moving in a dynamic environment [13], [14], [15]. Using LiDAR as the only source of environmental information is a highly unique task, and there has been little study in this area. Some human detection and tracking research has relied only on LiDAR technology. Human detection using LiDAR has been performed on both stationary robots [16], [17], employing several stationary LiDAR sensors [18], and mobile robots [19], [20].

On the other hand, researchers use machine learning for human detection and finding the robot. Machine learning has turned out to be key in providing a human-tracking robot's potential. The presented procedures use enormous amounts of data for training models that would notice the human features and movements, thus finding the correct identification in different scenes. Besides, the machine learning methods are used to help with robot localization by interpreting sensor data, thus finding the position of the robot in relation to the discovered human. Furthermore, this part will elaborate on the different machine-learning methods utilized for human detection and robot localization in these systems along with their advantages, and show the elements of integration [21], [22], [23]. Suet Peng Yong et al. [24] demonstrate human object recognition using deep learning algorithms with the use of a 3DR solo drone equipped with a GoPro camera for real-time surveillance and coverage of forest areas. Suet Peng Yong et al. provide knowledge of video processing using convolution neural networks and how to select the perfect dataset for a specific project.

Ashish U. Bokade et al. [25] discuss video surveillance utilizing a smartphone and Raspberry Pi. This allows you to watch and control the mobility of the robot using Raspberry Pi. The detection procedure may be completed successfully, and the findings can be viewed on the user's smartphone. Jun Zhang et al. [26] provide leaping robot standards, which are superior to traditional robots that cannot walk on rough surfaces or jump to a greater distance. It describes how a PIR sensor and a jumping robot build a zig-bee WSN that allows them to communicate with one another while also allowing the freedom to leap on stairs to reach higher surfaces from the ground up to a range of 105 cm.

Additionally, other researchers use OpenCV as the solution for human-tracking robots, which are used to track the human target during different movements while ensuring a constant distance between the human target and the robot. OpenCV (Open Source Computer Vision Library) is a set of versatile tools for real-time computer vision and it is actually a quite nice alternative for implementing human tracking in robotics. Researchers can write algorithms to detect and follow the humans by OpenCV that direct the robot to be safe and as close as possible to the optimal. In this section, OpenCV implementation in human-following robotic systems such as a detailed overview of its strengths, challenges, and its function

in giving the robots better agility and navigation precision will be explored [25], [26], [27], [28], [29], [30].

Meanwhile, color-based detections for target-following robots have been used by certain researchers since they are one of the possible good approaches to identifying a target, as demonstrated by researchers in Sefat S. et al. and MNA Bakar et al. S. Sefat et al. employed red color and 3D circular shape (red ball) detection, along with a Kalman filter, to predict the position of the individual to be followed. Although MNA Bakar et al. employed color-based detection, it used a special marker with a distinct form to help the robot recognize its target. The yellow hue was tested and had an 80% detection rate. However, MNA Bakar et al. had no obstructions in its route, as opposed to S. Sefat et al., which avoided obstacles while employing sonar sensors [31], [32].

Moreover, the investigation of the human body temperature through a thermal image in real-time is a well-known application of infrared technology by other researchers. The safety and security of a particular place, such as a train station, can be increased by the technology of human presence detection. As a result, the detector is a passable of sensors and a microcontroller. The detector can know the distance between the human and the reference point by using a camera. Sensor equipment is used in the way automated systems of various kinds are employed for people monitoring and various other applications. Infrared sensors have also been used to determine the human walking path. Thanks to such a device, robots can easily generate an exact following motion toward the human that they accompany. By defining the person's thermal footprint, robots can undertake good and continuous surveillance, which serves as a sufficient mechanism for follow-up. A detailed examination of the infrared technology application in robotic systems will be given in this section, including its advantages in human detection, movement predictions, and the overall enhancement of human-following robotic behaviors will be outlined [33], [34], [35], [36].

According to Montiel-Ross *et al.* [37], world perception, path planning and generation, and path tracking make up the robot navigation challenge. By choosing adequate sensor suites that can give the robot controller acceptable environmental feedback, world perception is achieved. Simultaneous Localization and Mapping (SLAM), a well-established technique that allows vision-based imaging equipment to visualize the surrounding depth map, is one of the best solutions for this purpose. Processing of this data can yield the locations of obstacles, targets, and the robot itself. The disadvantage is that in environments with unclear structures, SLAM performs less well [38]. In addition to being computationally demanding, SLAM has significant processing expenses [39]. According to Nowicki *et al.* [40], sudden and erratic motions of its sensing devices also cause SLAM to malfunction. Given that all of the CARMi sensors are installed on the same mobile platform, an incomplete mapping approach might be more appropriate. Al Arabi *et al.* [41] showed that partial mapping may be accomplished with just one rotating rangefinder by converting the data into a relative depth image of the surrounding area. The revolving depth camera configuration on CARMi may make this method useful.

The study of path planning and generation has a long history, and both heuristic and classical methods are still

widely used to prevent collisions and arrive at target locations. The robot that follows a human adds another level of complexity by needing to approach a moving target while keeping a set distance behind it. The path planning system is either “passive” or “anticipative,” according to Ziyou Wang *et al.* [42]. An “anticipative” system uses a velocity model [42] or a dynamically updated version of the Monte Carlo algorithm [39], [43] to predict the possible movements of a human target. Kalman filters, neural networks, fuzzy logic, and similar combinations [44], [45]. Since these techniques also come with high processing costs, it could be preferable for the CARMI navigation model to be “passive,” in which the robot reacts to changes in its surroundings or landmarks in a reactive manner [46].

Additionally, some other researchers utilize depth cameras as well as a selective set of limited proximity sensors. The overall approach of intelligent systems like robots consists of exact, desired-oriented human-tracking algorithms, which keep the robot on the right path and in the right direction minimizing the motion needed. The depth camera will be another evolutionary change in enabling the precise tracking of human targets in a 3D environment, whereas lasers give humans feedback on how far they are. This technology offers robots conventional control functions, like endpoint settings which provide machine-to-human interfacing inside factories or assembly facilities. This part will go deep in the analysis of the use of depth cameras and sensor fusion in these robotic systems for following humans, mainly by the help of them in improving target tracking and obstacle navigation [47], [48], [49].

### III. METHODOLOGY

#### A. Methodology Overview

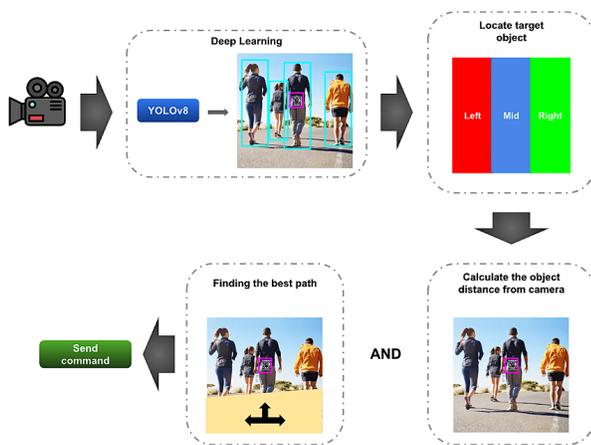


Fig. 1. Conceptual framework.

Fig. 1, titled “Conceptual Framework,” depicts the general methodology of the Marked Object-Following System. The system starts with a camera that captures real-world scenes, and the YOLOv8 deep learning model detects objects and defines “notable symbols” for tracking. A distance estimate technique calculates the target’s vicinity using variations in bounding box width, allowing for accurate distance inference.

The video frame is divided into three zones—left, middle, and right—to direct the robot’s movement. The system directs the robot to move left, forward, or right based on the zone in which the target appears. Pathfinding algorithms like as A\* and Tabu Search are used for obstacle avoidance and optimal navigation, resulting in efficient and precise target tracking in complicated situations. This framework offers a structured way to integrating deep learning and metaheuristics to create strong object-following apps.

#### B. Preparation for Model Training

##### 1) Dataset

A large dataset of photos is critical for this research since it serves as the foundation for training the YOLO object detection model, allowing the marked object-following algorithm with collision prevention to function properly. To provide reliable real-time detection and tracking of persons, the dataset should comprise a wide range of human poses, orientations, clothing kinds, and environmental circumstances such as lighting, weather, and busy locations. This diversity allows the model to generalize well to real-world events and consistently distinguish humans from other items. Fig. 2 are the sample dataset that shows individual is wearing the markers that we can detect and monitor.



Fig. 2. Dataset of images.

Additionally, the quality and diversity of the dataset are critical for training the YOLO model because they improve its capacity to recognize persons in difficult surroundings, reduce false positives, and increase detection accuracy. An extensive dataset also prepares the model to face obstacles such as occlusions, overlapping objects, and complicated backdrops, resulting in robust performance. Without a well-curated dataset, the

model's performance will suffer, potentially leading to errors in the system's marked object-following and collision prevention functions.

### 2) Data Annotation

Rectangular markers known as bounding boxes are used in object detection tasks to show the location and size of items in an image. They are employed in this study to label and annotate the dataset, designating the people that the algorithm must recognize and obey. The YOLOv8 model needs the precise coordinates of the target objects—humans—in each training image in order to learn how to distinguish them from other things in the environment. This is why this step is so important. The model is trained on bounding box-annotated photos, which enables it to anticipate comparable boxes surrounding persons in real-time during deployment, guaranteeing precise tracking and detection. The effectiveness of the marked object-following algorithm with collision prevention depends on the YOLOv8 model's ability to recognize and marked object. This is made possible through the process of building these bounding boxes. An example of an image with bounding boxes is shown in the Fig. 3.



Fig. 3. Examples of bounding box images.

A key challenge in developing an object-following algorithm with collision prevention is the potential for confusion when multiple similar objects are present, as the system might mistakenly track any detected object without a distinguishing feature. In environments where objects lack unique visual characteristics, relying solely on generic detection could result in tracking the wrong target. To address this, the research incorporates distinct logos or markers placed on the intended object, which the YOLOv8 model is specifically trained to recognize as the target class. These markers act as notable symbols, which is shown in Fig. 4, enabling the system to distinguish the designated object from others in the vicinity. By focusing on these specific classes, the algorithm reliably

tracks the intended object, reducing errors and enhancing performance in crowded or dynamic environments. This approach ensures accurate and safe object-following behavior, even in complex settings, by preventing the system from mistakenly tracking unintended objects.



Fig. 4. The four experimental notable symbols.

### 3) Model Selection

The YOLOv8-nano (YOLOv8n) model is designed to operate at fast speeds on embedded systems and other devices with constrained processing power. Take a look at the Table I below.

TABLE I. PERFORMANCE COMPARISON OF YOLOV8 MODELS [50]

Model	mAP <sup>val</sup> 50-95	Speed CPU ONNX (ms)	Params (M)
YOLOv8n	37.3	80.4	3.2
YOLOv8s	44.9	128.4	11.2
YOLOv8m	50.2	234.7	25.9
YOLOv8l	52.9	375.2	43.7
YOLOv8x	53.9	479.1	68.2

With a mean Average Precision (mAP) of 37.3% at the 50-95 Intersection over Union (IoU) threshold, it offers an effective balance between speed and accuracy, achieving an inference time of 80.4 milliseconds when running on a CPU using the ONNX runtime. Its compact architecture, consisting of only 3.2 million parameters, makes it highly suitable for real-time object detection, particularly on low-power processors like the Raspberry Pi. Although larger models, such as YOLOv8-s and YOLOv8-m, provide greater accuracy, their slower inference times (128.4 ms and 234.7 ms, respectively) make them less practical for resource-constrained environments. Therefore, YOLOv8-nano is selected for its efficient performance, ensuring that the marked object-following algorithm can accurately detect and track individuals that are wearing markers in real time while minimizing delay, which is critical for collision prevention and overall system reliability.

### C. Proposed Distance Estimation Algorithm

The distance estimation algorithm for this study leverages the width of the detected target class (such as notable symbols) to estimate the relative distance between the object and the camera in a robotic system. The fundamental concept is that the bounding box width, dynamically generated by the YOLOv8 model, provides a reliable reference for gauging distance. As the bounding box width decreases, the target is inferred to be moving further away, while an increase in the width suggests that the object is getting closer to the camera.

The Fig. 5 illustrates relationship between the bounding box width and the distance is inversely proportional. The equation for calculating the distance is as follows.

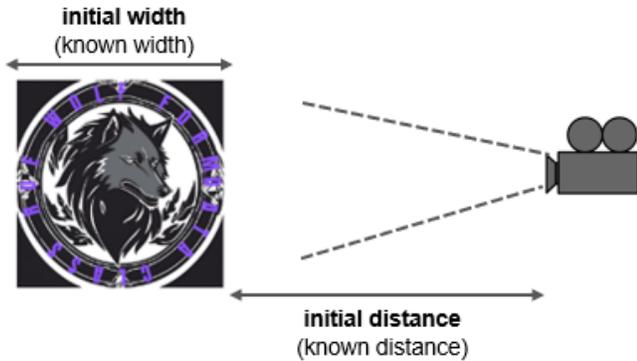


Fig. 5. Relationship between bounding box width and the camera.

$$\text{distance} = \left( \frac{\text{current\_width}}{\text{initial\_width}} \right) \times \text{initial\_distance}$$

Where:

- *current\_width* refers to the detected bounding box width of the target class at a particular moment.
- *initial\_width* is the reference bounding box width at a known distance.
- *initial\_distance* is the known distance from the camera when the object has the initial width.

This formula assumes that the camera and the object are in fixed, calibrated positions, and the size of the object remains constant. YOLOv8 dynamically detects the bounding box width, enabling real-time and accurate distance estimation based on width variations.

#### D. Proposed Object-Following Algorithm

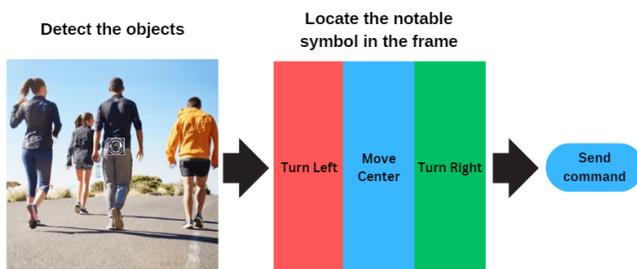


Fig. 6. Object-following algorithm.

Fig. 6 illustrates the proposed marked object-following algorithm. The process begins by detecting objects within the frame, particularly focusing on identifying the notable symbol worn by the designated person or individual. Once the symbol is detected, the algorithm evaluates its position within the frame. Depending on the location of the symbol—whether it is in the left, center, or right of the frame—the system will

send a corresponding command. If the symbol is positioned on the left, the algorithm issues a “Turn Left” command; if it is centered, a “Move Center” command is sent, and if on the right, a “Turn Right” command is executed. This step-by-step analysis ensures precise tracking and directional adjustments, enabling the system to follow the intended target efficiently.

#### E. Integrated Pathfinding and Distance Estimation System

**Algorithm 1** Integrated System Algorithm for Marked Object Following

**Inputs:** Camera feed, YOLOv8 model, grid dimensions (*max\_rows*, *max\_cols*), start position *S*, target position *T*, calibration constant *k*, and serial communication interface.

**Steps:**

1) **System Initialization:**

- Load the YOLOv8 model.
- Set up camera input using *cv2*.
- Define grid dimensions and initialize parameters (*tabu\_list*, *distance\_threshold*).
- Establish serial communication for robot control.

2) **Distance Estimation and Immediate Movement:**

- Detect objects in each frame using YOLOv8.
- Divide the frame into regions: left, center, and right.
- For each detected object:
  - Identify the class and bounding box width  $w_b$ .
  - Calculate distance  $d = \frac{k}{w_b}$ .
  - Send movement commands based on position and distance:
    - Turn Right: If object is on the right.
    - Turn Left: If object is on the left.
    - Move Forward: If object is in the center and  $d > 14$  inches.
    - Stop: If  $d \leq 14$  inches.

3) **Pathfinding with A-Star and Tabu Search:**

- Generate the global path using A-Star by computing  $f(n) = g(n) + h(n)$  for each node.
- Refine the path with Tabu Search:
  - Evaluate neighbors  $N(P)$  of the current path *P* and compute the cost  $\text{Cost}(P) = \sum f(n)$ .
  - Update the *tabu\_list* to avoid revisiting sub-optimal paths.
- Adjust the global path based on the marker’s position from the distance estimation module.

4) **Execute Navigation:**

- Follow the refined path while continuously updating the robot’s position using YOLOv8 detections.
- Use real-time corrections to handle dynamic obstacles and deviations.

**Output:** Efficient navigation to the target position *T* with consistent tracking of the marked object.

In order to effectively locate the best path in complicated surroundings, the pathfinding method integrates the advantages of both A-Star (A\*) and Tabu Search. A heuristic-based technique called A\* is used to determine the shortest path in a grid or graph between a start point and a target. The pathfinding algorithm is almost similar to the study of Gorro

et al. [51]. It strikes a balance between an estimate of the remaining distance to the target and the cost of the road already taken. Because of these two factors, A\* is a popular and effective solution for pathfinding issues. However, because it may revisit less-than-ideal solutions, A\* may lose effectiveness in areas with a high density of barriers or recurrent paths.

Tabu Search is used into this process to overcome this restriction. The “tabu list,” a memory component added by Tabu Search, keeps account of recently traveled routes or moves that have been judged to be less-than-ideal. Tabu Search compels the algorithm to investigate alternate options, even if they seem less promising at first, by forbidding the re-exploration of these routes. This investigation promotes the finding of globally optimal pathways and aids in avoiding local minima.

A\*, the first step in the combined algorithm, creates an initial path from the start to the destination. After that, Tabu Search takes control and iteratively refines this path. As long as it is not in the tabu list, the best neighbor is chosen as the current path after adjacent paths are assessed according to their costs at each stage. Because the tabu list is dynamically updated, recent errors or less-than-ideal routes are kept in mind and avoided in subsequent cycles. Until a certain number of iterations is reached or no more advancements can be made, the process keeps going.

In complex grids or maps, where the existence of barriers or constraints could cause traditional algorithms to be misguided, this hybrid approach works very well. The method produces a stable and adaptable solution by utilizing the advantages of A\* for initial pathfinding and Tabu Search for iterative refining. This makes it appropriate for applications like robotics, navigation, and logistical planning.

## F. Evaluation Metrics

These metrics help in determining how well the model is able to identify humans in various scenarios, ensuring that the robotic system can perform its tasks accurately and reliably. The research can efficiently analyze the model’s strengths and weaknesses, direct the tuning of hyperparameters, and make well-informed decisions about model optimization to achieve the desired performance in real-world environments by utilizing specific evaluation metrics like Precision, Recall, Mean Average Precision (mAP), and F1-score.

### 1) Precision

The ratio of true positive detections to the total of both true positive and false positive detections is known as precision. It assesses how well the model distinguishes, among all the detected things, only the pertinent objects—in this example, humans. High precision reduces false positives by increasing the likelihood that the YOLOv8-nano model’s predictions of humans are accurate. High precision is necessary to prevent the robot from unintentionally following non-human things in the setting of a marked object-following robotic system. This is important for both efficiency and safety.

$$\text{Precision} = \frac{\text{True Positive}}{(\text{True Positive} + \text{False Positive})}$$

### 2) Recall

The ratio of real positive detections to the total of false negatives and true positives is known as recall. It illustrates how well the model was able to identify every pertinent object (people) in the dataset. High recall means that the model is capable of detecting most of the humans present in the environment, minimizing false negatives. In this study, a high recall is important because, in the event that a human is not detected, the robotic system may not follow its intended path, which could be harmful in scenarios where it is used for public space guidance or healthcare assistance.

$$\text{Recall} = \frac{\text{True Positive}}{(\text{True Positive} + \text{False Positive})}$$

### 3) Mean Average Precision (mAP)

A comprehensive statistic called Mean Average Precision (mAP) provides an overall measure of accuracy by assessing the model’s performance across various Intersections over Union (IoU) thresholds. The model’s ability to balance precision and recall is indicated by a single performance score that is obtained by combining the two criteria. Because it aids in understanding the trade-offs between minimizing false positives (precision) and detecting as many humans as feasible (recall), mAP is particularly significant in this research. A greater mAP is a useful parameter for optimizing human detection in the YOLOv8-nano model since it shows that the model performs well in both areas.

$$\text{mAP} = \frac{1}{k} \sum_{i=1}^k \text{AP}_i$$

### 4) F1-score

The F1-score is a measure that provides a balance between Precision and Recall, calculated as the harmonic mean of the two. When it comes to striking a balance between false positives and false negatives, it is especially helpful. In this research, the F1-score is essential because it gives a more holistic view of the model’s performance. A high F1-score shows that the model is successful in capturing all important detections and is accurate in identifying humans. This is especially important in applications like robotic assistance and navigation where misidentifying a non-human object (false positive) or missing a human (false negative) can have serious repercussions.

$$\text{F1-score} = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}$$

IV. RESULT

A. YOLOv8 Performance

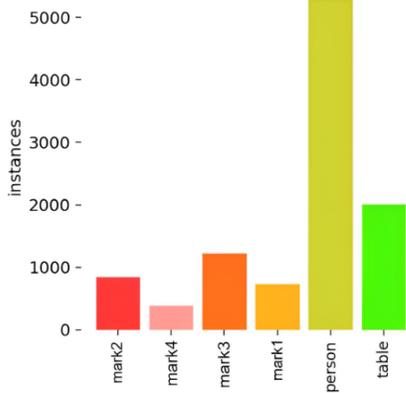


Fig. 7. Class distribution.

Monitoring the class distribution (Fig. 7) reveals significant class imbalance, with the “person” class dominating the dataset. This imbalance is a critical issue in object detection tasks [52], [53], as it can lead to over-optimization for frequent classes and under-performance for rarer ones, such as “mark4”. Techniques like data augmentation, oversampling, or loss re-weighting [54], [56] could address this and enhance performance across all classes.

In particular, the model may become too optimized for detecting the more frequent class (“person”) while struggling to reliably recognize less frequent ones (“mark4”). Regular analysis of this distribution allows researchers to take corrective action, such as boosting underrepresented classes or employing advanced approaches to reduce class imbalance. By addressing these issues, the model’s overall accuracy and generalization capabilities across all classes can be significantly improved, enhancing its robustness in real-world applications.

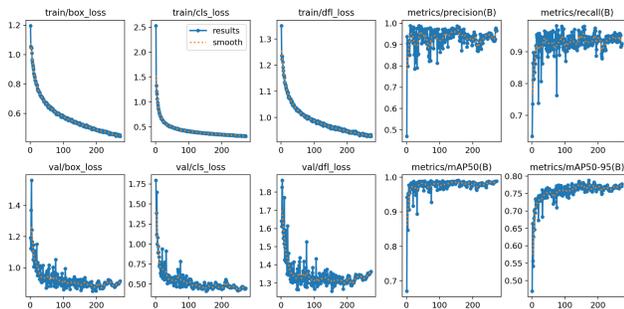


Fig. 8. Result graph of YOLOv8-nano model.

Fig. 8 illustrates the training and validation graphs for key metrics and loss functions in the proposed marked object-following algorithm with collision prevention, based on YOLOv8 object detection. The training loss curves,

which include bounding box regression (train/box\_loss), classification loss (train/cls\_loss), and distributional focal loss (train/df\_l\_loss), show a consistent decline as training progresses, indicating that the model is effectively minimizing prediction errors. This steady reduction in training losses suggests that the model is becoming more accurate in identifying and classifying objects while refining the predicted bounding box coordinates.

The training and validation metrics (Fig. 8) show consistent declines in losses, indicating effective learning and generalization. Compared to YOLOv4-tiny [55], the YOLOv8-based model achieves higher mAP values (0.961 at IoU@0.5), demonstrating competitive detection performance. However, slight precision-recall drops for the “person” class align with findings in [53], suggesting the need for improved handling of dominant classes in imbalanced datasets. Incorporating techniques like focal loss or semi-supervised learning [56], [57] could mitigate this challenge.

The validation losses (val/box\_loss, val/cls\_loss, val/df\_l\_loss) exhibit a similar downward trend, though with natural fluctuations, indicating the model’s generalization to unseen data. Precision and recall metrics remain high and stable, which demonstrates the model’s ability to maintain a balance between correctly identifying true positives and minimizing false positives. The mAP@0.5 and mAP@0.5-0.95 values show continuous improvement, signaling enhanced detection performance across various Intersection over Union (IoU) thresholds.

These graphs provide insight into the model’s training dynamics, showcasing a well-balanced process where the algorithm is consistently improving in both training and validation phases. The steady convergence of losses and strong performance metrics suggest the model is learning effectively without overfitting, ensuring reliable detection and tracking in real-time marked object-following scenarios.

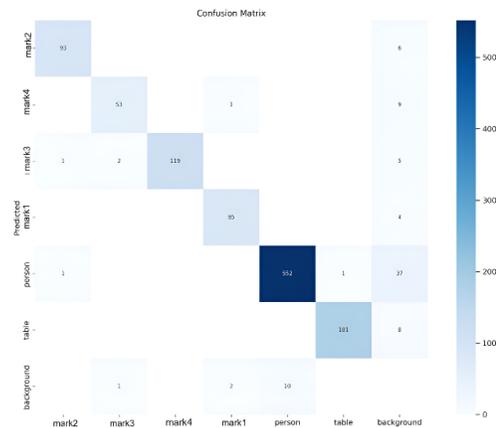


Fig. 9. Confusion matrix for YOLOv8-nano model.

The Confusion Matrix serves as an effective tool for assessing the performance of the YOLOv8-nano model by illustrating its accuracy in predicting various classes. This table presents the frequency of actual versus predicted classes, allowing for an evaluation of the alignment between the

model's predictions and the true labels. Analyzing the confusion matrix in the context of YOLOv8 helps identify specific cases where the model successfully classifies an object or incorrectly identifies it as another class. This insight is crucial for recognizing the model's strengths and weaknesses, enabling targeted improvements to enhance accuracy. By examining the matrix, researchers can identify which classes are frequently confused and make necessary modifications to training data, model architecture, or hyperparameters to rectify these issues.

Fig. 9 is a detailed review of the confusion matrix for the YOLOv8-nano model reveals that the "mark2" class is accurately predicted 93 times, with misclassifications occurring once each as "mark3" and "person." The "mark5" class achieves 53 correct predictions, with minor misclassifications as "mark3" twice and as "background" once. The "mark3" class exhibits perfect performance, yielding 119 correct predictions without any misclassifications. The "mark1" class is accurately predicted 85 times but is mistaken for "mark5" once and "background" twice. The "person" class demonstrates high accuracy with 552 correct predictions, though it is confused with the "background" class on 10 occasions. Finally, the "table" class is correctly identified 181 times, with one misclassification as "person". The model's overall performance is not greatly affected by these minor misclassifications. The YOLOv8-nano model has good prediction ability and little confusion between various object classes in spite of these small inaccuracies.

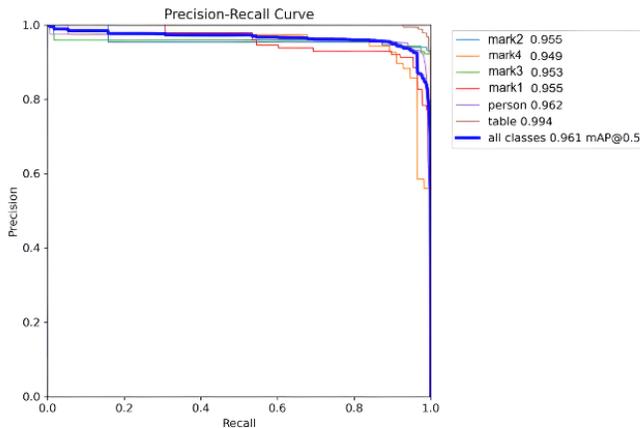


Fig. 10. Precision-recall curve.

Fig. 10 displays the Precision-Recall (PR) Curve for the various classes within the dataset. This curve visually illustrates the balance between precision and recall for each class, highlighting the model's effectiveness in correctly identifying true positives while reducing false positives. The results indicate that most classes attain very high precision and recall values, with both metrics nearing 1.0, which demonstrates the model's strong capability in detecting these objects with minimal errors. However, the "person" class exhibits slightly lower precision and recall values than the other classes, suggesting potential challenges in accurately detecting and distinguishing humans within the dataset. The overall mean Average Precision (mAP) at an Intersection over Union (IoU) threshold of 0.5 across all classes stands at 0.961, signifying an excellent

balance of high precision and recall. This elevated mAP value indicates that the model is well-optimized for precise object detection, ensuring reliable performance in identifying and classifying the various objects analyzed in this study.

The Precision-Recall Curve (Fig. 10) illustrates the model's strong detection capabilities. However, lower precision for the "person" class highlights challenges in distinguishing humans in cluttered environments. Fine-tuning anchor box sizes or using hybrid feature extractors, as shown in [55], could enhance performance.

Overall, the results validate the proposed marked object-following algorithm, achieving reliable detection and collision prevention in real-time scenarios. The high F1-scores across classes (Fig. 11) and stable precision-recall metrics ensure robust tracking. These findings demonstrate the algorithm's potential for deployment in assistive robotics and autonomous systems. Future work could integrate multi-sensor fusion or explore adaptive learning strategies [54], [56] to further improve robustness.

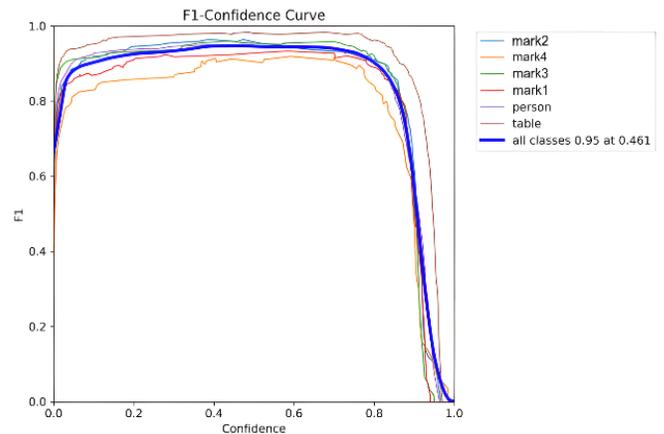


Fig. 11. F1-confidence curve.

As shown in the Fig. 11, the F1-scores for the different classes maintain high values at moderate confidence levels, with an overall peak of 0.95 for all classes at a confidence threshold of 0.461. This indicates a strong balance between precision (correctly identifying the object) and recall (detecting most of the relevant objects) for each class. The curves for each class follow a similar trend, with a sharp drop-off beyond the optimal confidence threshold, suggesting that the model is highly accurate up to a certain point, after which false positives start to increase.

This curve illustrates how well the model can detect different types of objects, meaning that the marked object-following algorithm can track the designated person (represented by the "person" class) and distinguish it from the other target classes, which include the table and various markers. Accurate object identification is ensured by maintaining a high F1-score across these classes, which helps to prevent collisions and ensures reliable human following.

## B. Implementation of Experimental Algorithms

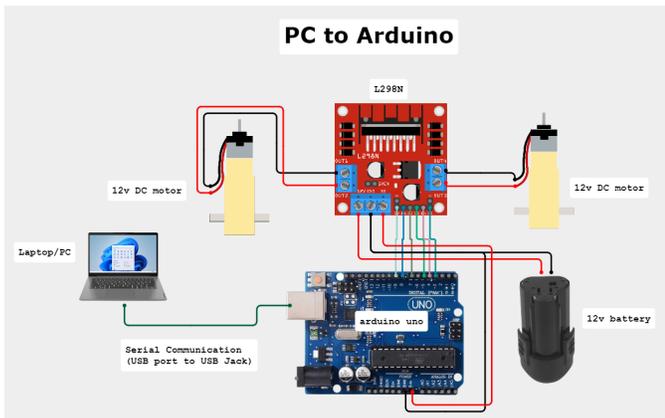


Fig. 12. The circuit diagram for the basic robot.

Fig. 12 illustrates the experimental prototype configuration used to test the marked object-following system. The circuit diagram details the connections between a laptop or PC, an Arduino Uno board, an L298N motor driver module, two 12V DC motors, and a 12V battery. The laptop/PC establishes a USB connection with the Arduino Uno, facilitating serial communication for data exchange and control commands. The Arduino Uno is linked to the L298N motor driver module, which regulates the two 12V DC motors by adjusting their speed and direction based on the signals received. The motors receive power from the 12V battery, which is directly connected to the L298N module, supplying the required voltage for operation. This configuration enables precise motor control through the Arduino, allowing commands from the PC to direct the motors via the L298N driver, effectively simulating navigation and following behaviors in response to target detection and distance estimation algorithms.

During the Test 1 the robot's ability to follow a sample image held by a human in a simple environment. The robot efficiently detects the target image and maintains a consistent following distance, showcasing its tracking accuracy and responsiveness. The straightforward setup allows the robot to smoothly follow the human, effectively illustrating its basic operational capability and fundamental functionality in a controlled, uncomplicated scenario.

Finally, during Test 2 the robot's capability to follow a human in a complex environment, effectively navigating without causing disruptions. When the human passes near the right green line, the robot seamlessly turns right, demonstrating a prompt and accurate response without any delay or difficulty in executing the turn. Similarly, when approaching the left green line, the robot exhibits the same level of efficiency, turning left without encountering any issues. This demonstrates the robot's robust decision-making and adaptability, ensuring reliable marked object-following behavior even in challenging environments.

## V. CONCLUSION

The primary aim of this study was to develop algorithms capable of accurately detecting and following a designated

marked object while estimating the distance between the user and the system in real-time, utilizing the YOLOv8 model for object detection. An obstacle avoidance was created using a distance estimation algorithm with the pathfinding A\* and Tabu search algorithm. The model's performance was evaluated through key metrics, including the F1-score and Precision-Recall. The F1-Confidence curve indicated a robust F1-score of 0.95 for all classes at a confidence threshold of 0.461, reflecting a well-balanced performance between precision and recall, effectively minimizing false positives and false negatives in detecting the target classes. Additionally, the Precision-Recall curve showcased the effectiveness of the YOLOv8 model, achieving an overall mean Average Precision (mAP) of 0.961 at an Intersection over Union (IoU) threshold of 0.5 for all classes. This high mAP value demonstrates the model's reliability in accurately identifying and tracking the target classes while maintaining consistent detection performance.

The successful integration of a YOLO-based detection model with a distance estimation, path finding algorithms (A\*) and Tabu Search highlights the system's potential for real-world applications. Although the system faces limitations in handling visual disturbances and detecting objects from side angles, it has produced promising results under controlled conditions. The achieved F1-score and Precision-Recall values underscore the model's effectiveness, providing a solid foundation for further enhancements and potential applications in various environments. The distance estimation algorithm and the path finding A\* and Tabu search are crucial for detecting potential collisions and obstacle avoidance with marked objects, and the inclusion of an obstacle detection feature could further mitigate collision risks.

## ACKNOWLEDGMENT

We extend our profound gratitude to Cebu Technological University and Cebu Normal University for their unwavering support throughout the course of this research endeavor.

## REFERENCES

- [1] A. Sangha and M. Rizvi, "Detection of acne by deep learning object detection", 2021. <https://doi.org/10.1101/2021.12.05.21267310>
- [2] Y. Fu, "Recent deep learning approaches for object detection", *Highlights in Science Engineering and Technology*, vol. 31, p. 64-70, 2023. <https://doi.org/10.54097/hset.v31i.4814>
- [3] K. Sharada, "Deep learning techniques for image recognition and object detection", *E3s Web of Conferences*, vol. 399, p. 04032, 2023. <https://doi.org/10.1051/e3sconf/202339904032>
- [4] J. García-González, I. García-Aguilar, D. Medina, R. Luque-Baena, E. López-Rubio, & E. Domínguez, "Vehicle overtaking hazard detection over onboard cameras using deep convolutional networks", p. 330-339, 2022. [https://doi.org/10.1007/978-3-031-18050-7\\_32](https://doi.org/10.1007/978-3-031-18050-7_32)
- [5] S. Primakov, A. Ibrahim, J. Timmeren, G. Wu, S. Keek, M. Beuque et al., "Automated detection and segmentation of non-small cell lung cancer computed tomography images", *Nature Communications*, vol. 13, no. 1, 2022. <https://doi.org/10.1038/s41467-022-30841-3>
- [6] J. Redmon, S. Divvala, R. Girshick, & A. Farhadi, "You only look once: unified, real-time object detection", p. 779-788, 2016. <https://doi.org/10.1109/cvpr.2016.91>
- [7] J. Schmidhuber, "Deep learning in neural networks: an overview", *Neural Networks*, vol. 61, p. 85-117, 2015. <https://doi.org/10.1016/j.neunet.2014.09.003>
- [8] K. Wang, S. Dang, F. He, & C. Peng, "A path planning method for indoor robots based on partial a global a-star algorithm", 2017. <https://doi.org/10.2991/fmsmt-17.2017.83>

- [9] O. Vural, K. Çelik, Y. Yurdagül, & M. Sağlam, "A new automation system for equipment status and efficiency detection with machine learning based image processing", *Orclever Proceedings of Research and Development*, vol. 1, no. 1, p. 38-44, 2022. <https://doi.org/10.56038/oprd.v1i1.206>
- [10] N. Chinthamu, "Iot-based secure data transmission prediction using deep learning model in cloud computing", *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 4s, p. 68-76, 2023. <https://doi.org/10.17762/ijritcc.v11i4s.6308>
- [11] P. Sun, G. Chen, & Y. Shang, "Adaptive saliency biased loss for object detection in aerial images", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 10, p. 7154-7165, 2020. <https://doi.org/10.1109/tgrs.2020.2980023>
- [12] Z. Naik and M. Gandhi, "A review: object detection using deep learning", *International Journal of Computer Applications*, vol. 180, no. 29, p. 46-48, 2018. <https://doi.org/10.5120/ijca2018916708>
- [13] M. M. Islam, A. Lam, H. Fukuda, Y. Kobayashi, and Y. Kuno, "A person-following shopping support robot based on human pose skeleton data and lidar sensor," in *\*Intelligent Computing Methodologies: 15th International Conference, ICIC 2019, Nanchang, China, August 3-6, 2019, Proceedings, Part III 15\**, Springer International Publishing, 2019, pp. 9-19.
- [14] Z. Gao, Z. Wang, L. Saint-Bauzel, and F. Ben Amar, "2D lidar-based large workspace frontal human following for a mobile robot," \*Available at SSRN 4538601\*.
- [15] D. Cha and W. Chung, "Human-leg detection in 3D feature space for a person-following mobile robot using 2D LiDARs," *\*International Journal of Precision Engineering and Manufacturing\**, vol. 21, no. 7, pp. 1299-1307, 2020.
- [16] D. Z. Wang, I. Posner, and P. Newman, "Model-free detection and tracking of dynamic objects with 2D lidar," *\*The International Journal of Robotics Research\**, vol. 34, no. 7, pp. 1039-1063, 2015.
- [17] J. Shackleton, B. VanVoorst, and J. Hesch, "Tracking people with a 360-degree lidar," in *\*2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance\**, 2010, pp. 420-426.
- [18] T. Nowak, K. Ćwian, and P. Skrzypczyński, "Real-time detection of non-stationary objects using intensity data in automotive LiDAR SLAM," *\*Sensors\**, vol. 21, no. 20, p. 6781, 2021.
- [19] W. Chung, H. Kim, Y. Yoo, C. B. Moon, and J. Park, "The detection and following of human legs through inductive approaches for a mobile robot with a single laser range finder," *\*IEEE Transactions on Industrial Electronics\**, vol. 59, no. 8, pp. 3156-3166, 2011.
- [20] E. J. Jung, J. H. Lee, B. J. Yi, J. Park, and S. T. Noh, "Development of a laser-range-finder-based human tracking and control algorithm for a marathoner service robot," *\*IEEE/ASME Transactions on Mechatronics\**, vol. 19, no. 6, pp. 1963-1976, 2013.
- [21] R. Mark2bri and M. T. Choi, "Deep-learning-based indoor human following of mobile robot using color feature," *\*Sensors\**, vol. 20, no. 9, p. 2699, 2020.
- [22] S. O. Adebola, *\*A Human Following Robot for Fall Detection\**, Master's thesis, Middle Tennessee State University, 2019.
- [23] M. Padhen, K. Shimpi, R. Thakur, and P. V. Sontakke, "Human detecting robot based on computer vision-machine learning," *\*International Journal for Research in Applied Science and Engineering Technology\**, vol. 8, no. IX, 2020.
- [24] S. P. Yong and Y. C. Yeong, "Human object detection in forest with deep learning based on drone's vision," in *\*2018 4th International Conference on Computer and Information Sciences (ICCOINS)\**, 2018, pp. 1-5.
- [25] A. U. Bokade and V. R. Ratnaparkhe, "Video surveillance robot control using smartphone and Raspberry Pi," in *\*2016 International Conference on Communication and Signal Processing (ICCSP)\**, 2016, pp. 2094-2097.
- [26] J. Zhang, G. Song, G. Qiao, T. Meng, and H. Sun, "An indoor security system with a jumping robot as the surveillance terminal," *\*IEEE Transactions on Consumer Electronics\**, vol. 57, no. 4, pp. 1774-1781, 2011.
- [27] A. Imteaj, M. I. J. Chowdhury, M. Farshid, and A. R. Shahid, "RoboFI: Autonomous path follower robot for human body detection and geolocalization for search and rescue missions using computer vision and IoT," in *\*2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)\**, 2019, pp. 1-6.
- [28] J. Jommuangbut and K. Sritrakulchai, "Development of the human following robot control system using HD webcam," in *\*2018 International Electrical Engineering Congress (iEECON)\**, 2018, pp. 1-4.
- [29] G. R. Poornima, J. L. Avinash, S. Palle, S. S. Kumar, K. S. Kumar, and P. R. Prasad, "Image processing based human pursuing robot," in *\*2020 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)\**, 2020, pp. 408-412.
- [30] M. Sharikmaslat, R. Sidhaye, and A. Narkar, "Image processing based human pursuing robot," in *\*2019 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA)\**, 2019, pp. 702-704.
- [31] M. S. Sefat, D. K. Sarker, and M. Shahjahan, "Design and implementation of a vision based intelligent object follower robot," in *\*2014 9th International Forum on Strategic Technology (IFOST)\**, 2014, pp. 425-428.
- [32] M. N. A. Bakar and A. R. M. Saad, "A monocular vision-based specific person detection system for mobile robot applications," *\*Procedia Engineering\**, vol. 41, pp. 22-31, 2012.
- [33] T. Inoue, Y. Okazaki, and K. Itoya, "Person following algorithm with pixel-area addition method of thermal sensors for autonomous mobile robots," in *\*2024 10th International Conference on Control, Automation and Robotics (ICCAR)\**, 2024, pp. 77-82.
- [34] G. Feng, X. Guo, and G. Wang, "Infrared motion sensing system for human-following robots," *\*Sensors and Actuators A: Physical\**, vol. 185, pp. 1-7, 2012.
- [35] I. T. Ćirić, Ž. M. Ćojbašić, D. D. Ristić-Durrant, V. D. Nikolić, M. V. Ćirić, M. B. Simonović, and I. R. Pavlović, "Thermal vision based intelligent system for human detection and tracking in mobile robot control system," *\*Thermal Science\**, vol. 20, suppl. 5, pp. 1553-1559, 2016.
- [36] C. Filippini, D. Perpetuini, D. Cardone, A. M. Chiarelli, and A. Merla, "Thermal infrared imaging-based affective computing and its application to facilitate human-robot interaction: A review," *\*Applied Sciences\**, vol. 10, no. 8, p. 2924, 2020.
- [37] O. Montiel-Ross, R. Sepúlveda, O. Castillo, and P. Melin, "Ant colony test center for planning autonomous mobile robot navigation," *\*Computer Applications in Engineering Education\**, vol. 21, no. 2, pp. 214-229, 2013.
- [38] Q. H. Nguyen, H. Vu, T. H. Tran, and Q. H. Nguyen, "Developing a way-finding system on mobile robot assisting visually impaired people in an indoor environment," *\*Multimedia Tools and Applications\**, vol. 76, pp. 2645-2669, 2017.
- [39] F. J. Perez-Grau, F. Caballero, A. Viguria, and A. Ollero, "Multi-sensor three-dimensional Monte Carlo localization for long-term aerial robot navigation," *\*International Journal of Advanced Robotic Systems\**, vol. 14, no. 5, p. 1729881417732757, 2017.
- [40] M. R. Nowicki, D. Belter, A. Kostusiak, P. Cížek, J. Faigl, and P. Skrzypczyński, "An experimental study on feature-based SLAM for multi-legged robots with RGB-D sensors," *\*Industrial Robot: An International Journal\**, vol. 44, no. 4, pp. 428-441, 2017.
- [41] A. Al Arabi, P. Sarkar, F. Ahmed, W. R. Rafie, M. Hannan, and M. A. Amin, "2D mapping and vertex finding method for path planning in autonomous obstacle avoidance robotic system," in *\*2017 2nd International Conference on Control and Robotics Engineering (ICCRE)\**, 2017, pp. 39-42.
- [42] Z. Wang, J. Kinugawa, H. Wang, and K. Kazuhiro, "The simulation of nonlinear model predictive control for a human-following mobile robot," in *\*2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)\**, 2015, pp. 415-422.
- [43] W. Mi, X. Wang, P. Ren, and C. Hou, "A system for an anticipative front human following robot," in *\*Proceedings of the International Conference on Artificial Intelligence and Robotics and the International Conference on Automation, Control and Robotics Engineering\**, 2016, pp. 1-6.
- [44] A. Pandey, S. Kumar, K. K. Pandey, and D. R. Parhi, "Mobile robot navigation in unknown static environments using ANFIS controller," *\*Perspectives in Science\**, vol. 8, pp. 421-423, 2016.
- [45] M. Almasri, K. Elleithy, and A. Alajlan, "Sensor fusion based model for collision free mobile robot navigation," *\*Sensors\**, vol. 16, no. 1, p. 24, 2015.

- [46] C. Gomez, A. C. Hernandez, J. Crespo, and R. Barber, "A topological navigation system for indoor environments based on perception events," \*International Journal of Advanced Robotic Systems\*, vol. 14, no. 1, p. 1729881416678134, 2016.
- [47] M. Tee Kit Tsun, B. T. Lau, and H. Siswoyo Jo, "An improved indoor robot human-following navigation model using depth camera, active IR marker, and proximity sensors fusion," \*Robotics\*, vol. 7, no. 1, p. 4, 2018.
- [48] P. Janousek, Z. Slanina, and W. Walendziuk, "Target-following robotic platform based on UWB localization and depth camera," \*IFAC-PapersOnLine\*, vol. 58, no. 9, pp. 247–252, 2024.
- [49] M. Q. Do and C. H. Lin, "Embedded human-following mobile-robot with an RGB-D camera," in \*2015 14th IAPR International Conference on Machine Vision Applications (MVA)\*, 2015, pp. 555–558.
- [50] G. Jocher, A. Chaurasia, and J. Qiu, Ultralytics YOLOv8, version 8.0.0, 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>.
- [51] K. Gorro, L. Roble, M. A. Magana, and R. P. Buot, "Prototype of an Indoor Pathfinding Application with Obstacle Detection for the Visually Impaired," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 15, no. 9, 2024. doi: <https://dx.doi.org/10.14569/IJACSA.2024.01509106>.
- [52] N. Crasto, "Class Imbalance in Object Detection: An Experimental Diagnosis and Study of Mitigation Strategies," *arXiv preprint arXiv:2403.07113*, 2024. Available: <https://arxiv.org/abs/2403.07113>.
- [53] Y. Li, B. Wang, Z. Kang, S. Tang, L. Wu, and J. Li, "Overcoming Classifier Imbalance for Long-tail Object Detection with Balanced Group Softmax," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10991–11000. Available: [https://openaccess.thecvf.com/content\\_CVPR\\_2020/papers/Li\\_Overcoming\\_Classifier\\_Imbalance\\_for\\_LongTail\\_Object\\_Detection\\_With\\_Balanced\\_Group\\_CVPR\\_2020\\_paper.pdf](https://openaccess.thecvf.com/content_CVPR_2020/papers/Li_Overcoming_Classifier_Imbalance_for_LongTail_Object_Detection_With_Balanced_Group_CVPR_2020_paper.pdf).
- [54] M. Tomaszewski and J. Osuchowski, "Effectiveness of Data Resampling in Mitigating Class Imbalance for Object Detection," *CEUR Workshop Proceedings*, vol. 3628, 2023. Available: <https://ceur-ws.org/Vol-3628/paper14.pdf>.
- [55] Roboflow, "YOLOv8 vs. YOLOv4 Tiny: Compared and Contrasted," Roboflow, 2023. Available: <https://roboflow.com/compare/yolov8-vs-yolov4-tiny>.
- [56] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2017, pp. 2980–2988.
- [57] Ultralytics, "Pretrain YOLOv8 with Semi-supervised Learning," GitHub Issue #4373, 2023. Available: <https://github.com/ultralytics/ultralytics/issues/4373>.