OPEN ACCESS

*International*
*Journal*
*Of Modern Engineering Research (IJMER)*

# The Study of a Following and Obstacle -Avoidance Vehicle

Wen-Kung Tseng [1], Pin-Cheng Lin [2]

[1]*Professor, Graduate Institute of Vehicle Engineering, National Changhua University of Education, Taiwan R.O.C.*

[2]*Postgraduate Student, Graduate Institute of Vehicle Engineering, National Changhua University of Education, Taiwan R.O.C.*

*Corresponding Author: andy007@cc.ncue.edu.tw*

***ABSTRACT:*** *With the continuous advancement of technology, research and development in autonomous vehicles are becoming widespread and increasingly progressive. This study aims to utilize radar for path planning in smart autonomous vehicles, enabling autonomous obstacle avoidance patrols. Additionally, network cameras for continuous monitoring were used. The captured images were synchronously analyzed using YOLO image recognition to differentiate between intruders and personnel. Upon confirming an intruder, the system would activate an electric gun for repulsion purposes. By utilizing ROS (Robot Operating System) and YOLO (You Only Look Once) as the main core components for designing the autonomous vehicle, this concept could be fully realized. It successfully achieved the establishment of a machine security system with repulsion capabilities, replacing human security personnel. This innovation helps prevent injuries to human guards, reduce personnel expenses, and actualize the vision of automated security.*

***KEY WARDS:*** *Autonomous vehicles, Radar, Path planning, YOLO.*

---

Date of Submission: 05-02-2025 Date of acceptance: 16-02-2025

---

## I. INTRODUCTION

Home burglary has been a significant issue affecting people's livelihoods since ancient times. Burglary incidents frequently occur in private residences, roadside shops, and factories, posing severe threats to personal property and corporate assets. These incidents not only lead to material losses but can also cause psychological trauma to victims. According to data from the Ministry of the Interior, there were 4,411 home burglary incidents in 2023, with 14.01% of these cases involving stolen cables from factories or warehouses [1]. The high market value of these stolen cables makes them an attractive target for criminals. Currently, most factories and warehouses primarily rely on security systems or human security patrols to prevent theft. However, these security measures are not always entirely effective in deterring criminal activity. On one hand, fixed security systems, such as surveillance cameras and intrusion sensors, may not cover all areas and blind spots. On the other hand, while human security patrols offer mobility, they have inherent limitations, particularly at night or when personnel are insufficient. Additionally, if intruders are armed, the safety of security personnel could be compromised.

In the field of neural networks, there have been significant advancements in object detection systems. Joseph Redmon [2] and his colleagues introduced the YOLO9000 (You Only Look Once) model, a real-time object detection system capable of recognizing over 9,000 object categories. This innovation has greatly contributed to the field of object detection by expanding the range of objects that could be accurately identified. Moreover, Juan Du's review of modern object [3] detection algorithms shed light on the comparison between the CNN series and YOLO. YOLO has shown superior practical applications, achieving faster frame rates of up to 155FPS during experiments. These improved frame rates signified a substantial progression in real-time object detection, enhancing performance and efficiency in image processing and intelligent detection tasks. On a practical front, Huang Yucheng's development of a smart factory self-propelled vehicle [4] showcases the application of these advancements. By utilizing the robust ROS platform, machine control, sensor data recognition, and sophisticated navigation capabilities, the autonomous vehicle designed by Huang Yucheng offered automated navigation and handling functionalities. This self-driving vehicle, tailored for industrial or indoor settings, exemplified the integration of advanced technologies for real-world applications.

To enhance theft prevention efficiency, reduce security costs, and mitigate the dangers associated with human patrols, this study proposed a novel security system utilizing YOLO image recognition technology integrated with autonomous patrol vehicles. YOLO is a rapid and accurate object detection algorithm suitable for real-time applications, capable of significantly improving the accuracy and response speed in detecting intruders. The autonomous patrol vehicles were designed to navigate and patrol independently, with the primary recognition target being humans. Workers wearing reflective vests were excluded from being targeted, eliminating the risk of accidental harm. When the target was identified as a human intruder, the vehicle-mounted electric gun could fire bullets upon detecting the intruder, acting as a deterrent to expel the intruder. This dynamic patrol approach overcomes the limitations of traditional fixed-point detection security systems, providing comprehensive security coverage without blind spots.

## II. PROCEDURES AND METHODS

YOLO is a widely used object detection model that has been extensively applied in recent years [5]. It can locate objects in an image with a single prediction. The YOLO algorithm is built on a CNN (Convolutional Neural Network) model. When an image is input, it is divided into an S×S grid, where each grid cell predicts bounding boxes. Each grid cell predicts only two bounding boxes, along with five confidence scores and corresponding class or label probabilities. The predicted bounding box quantities and confidence scores of the two grid cells are then processed using the Non-Maximum Suppression (NMS) algorithm to obtain the final detection results. In this study, the YOLOv8 model was used, which was developed by the Ultralytics team and released in 2023[5]. YOLOv8 is an improved design based on YOLOv5. The installation process for YOLOv8 is simplified, making it easier for users to enter this field. The basic architecture of the model was largely the same as YOLOv5, as shown in Figure 1. There are several features that have been improved from YOLOv5, with references to the Neck design of YOLOv7. Specifically, YOLOv8 integrates a Feature Pyramid Network (FPN) architecture, which allows it to handle feature maps of different sizes. Key features of YOLOv8 include:

1. ***Anchor-Free Detection***: This method directly predicts the center of objects in the image, without relying on preset bounding boxes, enhancing the model's adaptability.
2. ***Mosaic Data Augmentation***: YOLOv8 adopts the Mosaic data augmentation technique from YOLOv5, which stitches four images together during training to provide more contextual information. This method is deactivated during the last 10 epochs of training to optimize training accuracy and improve training speed.
3. ***YOLOv8 Backbone***: Inspired by the FPN architecture of YOLOv7, it incorporates the core structure of YOLOv5 but replaces the C3 modules with C2F modules. This modification, through residual connections, enhances network performance and alleviates the problem of gradient vanishing

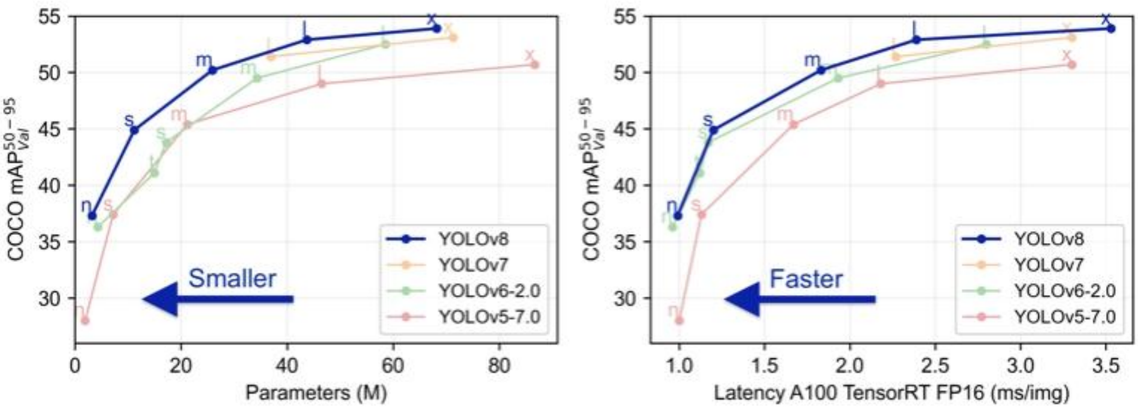


**Figure 1: YOLOV8 configuration**

With advancements in computer and semiconductor technologies, the application of deep learning in object recognition has been increasing. The emergence of Convolutional Neural Networks (CNNs) represents a milestone in object recognition technology, significantly enhancing recognition accuracy and precision. CNNs in deep learning are primarily used to process and analyze data with a grid structure, such as images or videos[6]. They possess the ability to automatically learn and extract features from images, replacing the manual feature extraction process in traditional object recognition techniques. Through neural network computations, CNN models achieve high efficiency and precision, making them the core technology in modern object recognition. The structure of CNNs includes multiple convolutional layers, pooling layers, and fully connected layers. The key operations within CNNs involve convolution and pooling, performed by the convolutional and pooling layers, respectively. These operations extract and compress features necessary for object recognition. Subsequently, the fully connected layers perform the final feature extraction and classification. The architecture of Convolutional Neural Networks is shown in Figure 2.

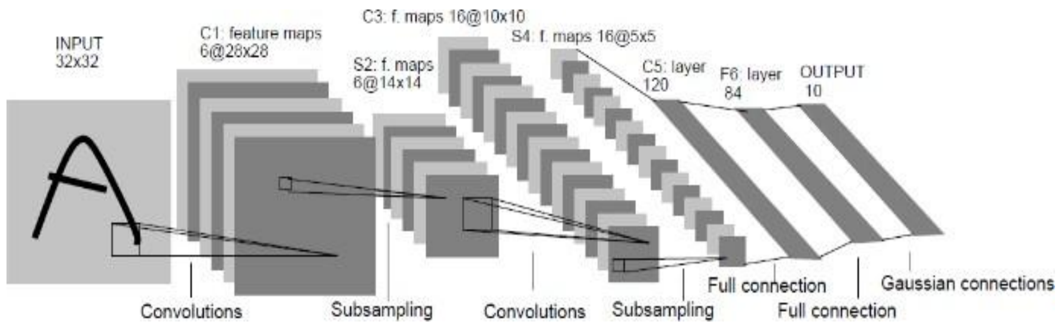


**Figure 2: CNN architecture**

Convolutional Layer: The primary function of the convolutional layer is to extract features from images. Before performing convolution operations, it is necessary to define the size of the convolution kernel (Kernel), such as 3x3, 5x5, etc. The convolution kernel moves from left to right and top to bottom, with the distance it moves referred to as the stride. The larger the stride, the fewer times the convolution kernel moves, which can reduce the computational load and memory usage of the model. However, it also results in smaller feature maps and may cause some features to be ignored. Therefore, when designing convolutional layers, it is important to consider the model's requirements for localization. Through convolution operations, the convolutional layer can identify various shapes, curves, colors, and other features in an image. Consequently, the size of the convolution kernel also affects the resulting feature map. The figure below illustrates the convolution kernel operation process.

Pooling Layer: The primary purpose of the pooling layer is to filter the feature map, removing unnecessary features while retaining key information during the compression process to reduce image data and simplify the model as shown in Figure 3. This enhances the feature extraction process, with the pooling layer configuration progressively reducing the data space, thereby indirectly controlling overfitting. Pooling operations typically use two methods: MaxPooling and Average Pooling [7].

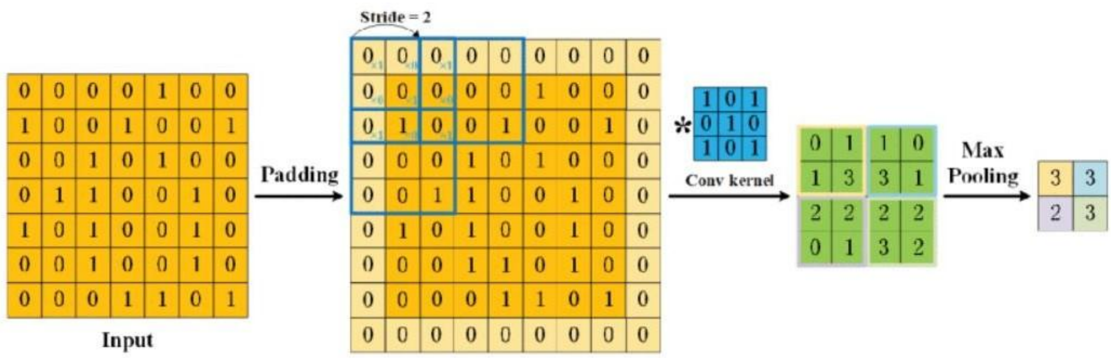


**Figure 3: Pooling Layer Structure**

Max Pooling: Max pooling operates similarly to convolutional operations, both utilizing a sliding window approach. Prior to computation, the size of the window must be specified. For instance, when the stride is set to 2, a 2x2 pooling operation is performed. Max pooling selects the maximum value within the window as the output and repeats this process to compress the feature map. This method helps retain the most prominent features within each region.

Average Pooling: In contrast to max pooling, average pooling takes the average of all values within the window as the output of the pooling operation. This process is illustrated in figure (b). This method considers all the information within the window, providing more features compared to max pooling. Average pooling can make better use of all the features from the input, offering a more comprehensive utilization of the information.

Fully Connected Layer: The architecture of the fully connected layer is similar to that of a multilayer perceptron (MLP). The primary function of the fully connected layer is to perform classification and prediction. It flattens the features extracted by convolutional and pooling layers, converting the two-dimensional matrix into a one-dimensional array. This array is then mapped to the final output to complete the classification process.

In this study, self-propelled vehicles refering to vehicles capable of autonomous patrol was developed, which can navigate without manual operation by personnel, also known as unmanned vehicles or intelligent vehicles. The operation of self-propelled vehicles relies on sensors such as radar, cameras, LiDAR, etc., to detect the surrounding environment, including roads, vehicles, pedestrians, obstacles, etc. Based on ROS (Robot Operating System) [8], the vehicle makes decisions and moves to achieve safe, efficient, and accurate driving. The development of self-propelled vehicles has garnered global attention and is gradually becoming a trend for the future, making it one of the key focal points of global technological advancements. This technology can enhance road safety, reduce traffic accidents, conserve energy, and decrease environmental pollution. ROS is an operating system based on the Linux operating system Ubuntu, designed for developing robot applications. Unlike traditional operating systems such as Android, Apple macOS, and Windows, ROS provides functionality similar to an operating system, including low-level driver management and inter-process communication as shown in Figure 4 and Figure 5.

Its primary function is to serve as a framework for communication between different parts of a robot. When moving a robotic arm to the correct position, for example, control of motors and other sensors are necessary to avoid obstacles. ROS provides many similar functions, enabling communication between motor control programs and sensor control programs, optimizing training precision and improving training speed. Its primary function is to serve as a platform for ROS supporting distributed computing and open collaboration among different components of robots. It provides a comprehensive toolset covering areas such as modeling, simulation, visualization, control, perception, and navigation. Through ROS, different parts of robot systems can operate on separate computers, enhancing system flexibility and scalability. Developers can program and communicate between components using familiar programming languages. ROS is currently extensively utilized in robotics, autonomous driving, and artificial intelligence fields. It boasts a large support community and contributions from developers, making it a feature-rich framework.

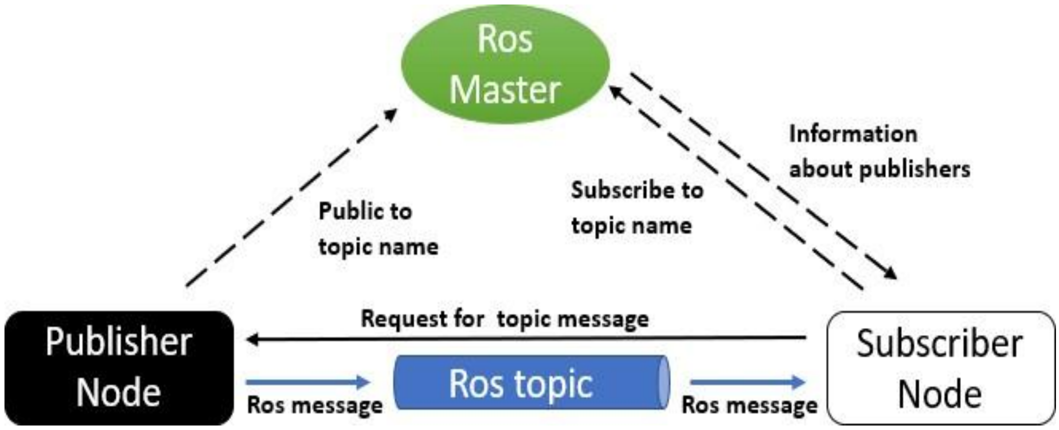


**Figure 4: ROS Framework [8]**

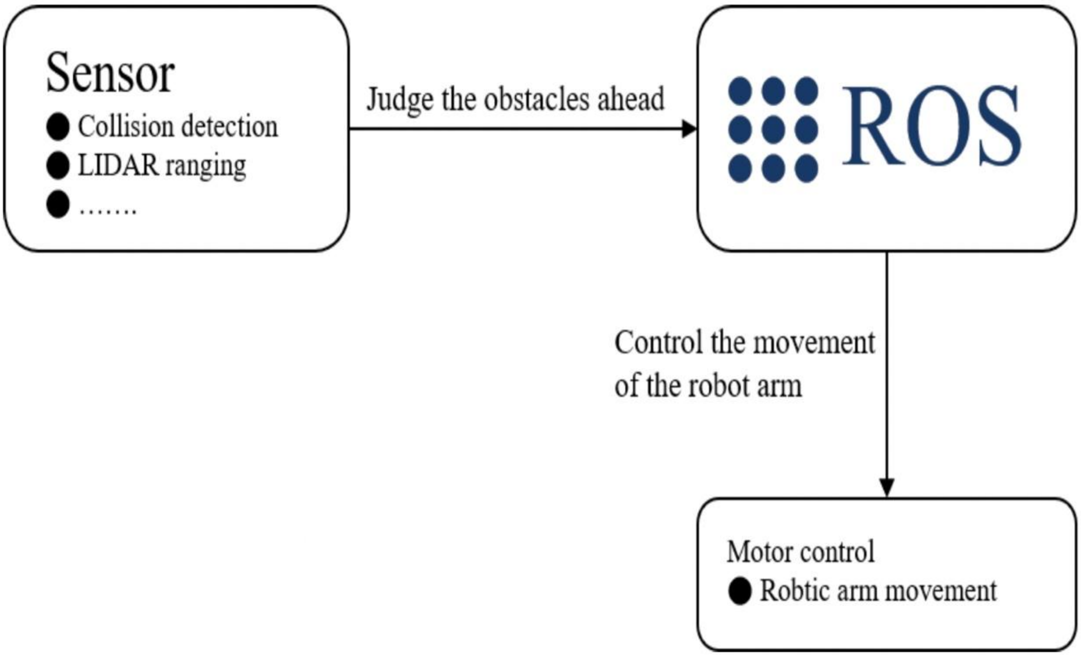


**Figure 5: ROS Operation**

## III. RESULTS AND DISCUSSIONS

The objective of this research was to create a security autonomous robot that combines components or systems such as ROS, YOLO, LiDAR, Arduino, etc. The robot would be capable of autonomously patrolling, avoiding obstacles, and distinguishing between individuals as intruders or authorized personnel. It will have the functionality to repel intruders effectively. The experimental framework of this research consisted of two main components: the ROS system and YOLO image recognition. In the ROS segment, SLAM (Simultaneous Localization and Mapping) [8] scanning was employed to generate maps and plan paths. Through the NVIDIA system and motor control board, the robot base was controlled to achieve obstacle avoidance functionality. In the YOLO segment, a model was trained on Ubuntu 16.04 to recognize individuals or reflective vests. Subsequently, Arduino was used to control the electric gun for activation or deactivation, aiming to achieve the repulsion effect.

At the outset of the experiment, it was necessary to gather an adequate number of target images. In the context of this research, images of individuals and reflective vests need to be collected and a database established. Once the data preparation was completed, the training of the YOLOv8 model commences. Upon completion of the training, it was integrated with the autonomous robot. In the robotics segment, using JETSON NANO, the current environment map was constructed. With the assistance of the SLAM map, the robot could autonomously patrol and navigate obstacles. Simultaneously, real-time detection was conducted using a network camera. During the patrol, if the recognized target was a person, the electric gun would be activated for shooting. However, if the individual was wearing a reflective vest, the electric gun would not be activated. The self-propelled vehicle is shown in Figure 6.

**Figure 6: Self-propelled Vehicle**

The graph below illustrates the training results of the YOLOv8 model as shown in Figure 7, Figure 8 and Figure 9. After training the YOLOv8 model, the results were visualized in the graph, which presents performance metrics. Following 150 training epochs, the best results were achieved, with a precision of 0.922 and a recall of 0.85. It is noteworthy that the precision reached 1 at a confidence level of 0.906, indicating a high probability of correct identification. A higher precision signifies a higher probability of correct identification, while a higher recall indicates a higher probability of correct classification results and a lower probability of misidentification errors.

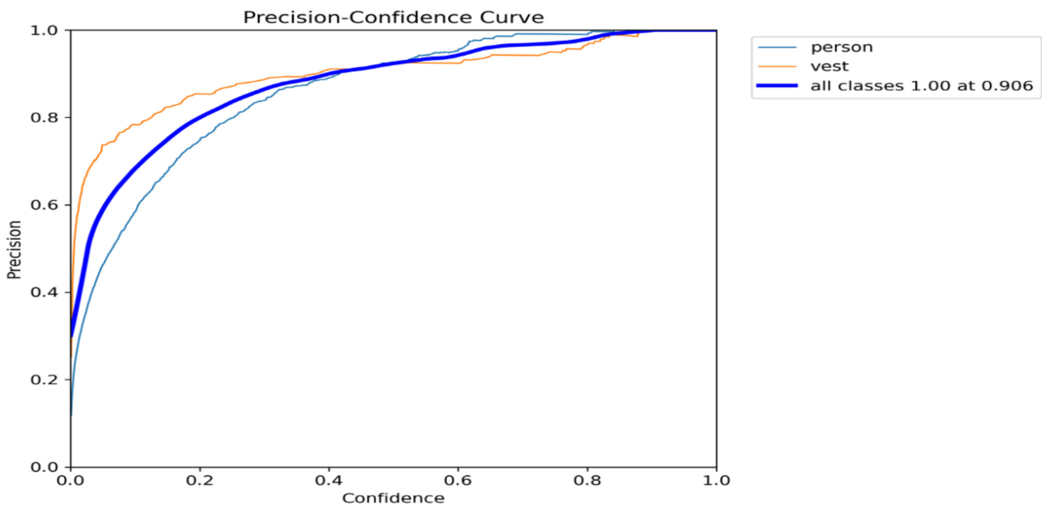


**Figure 7: Precision-Confidence Curve**

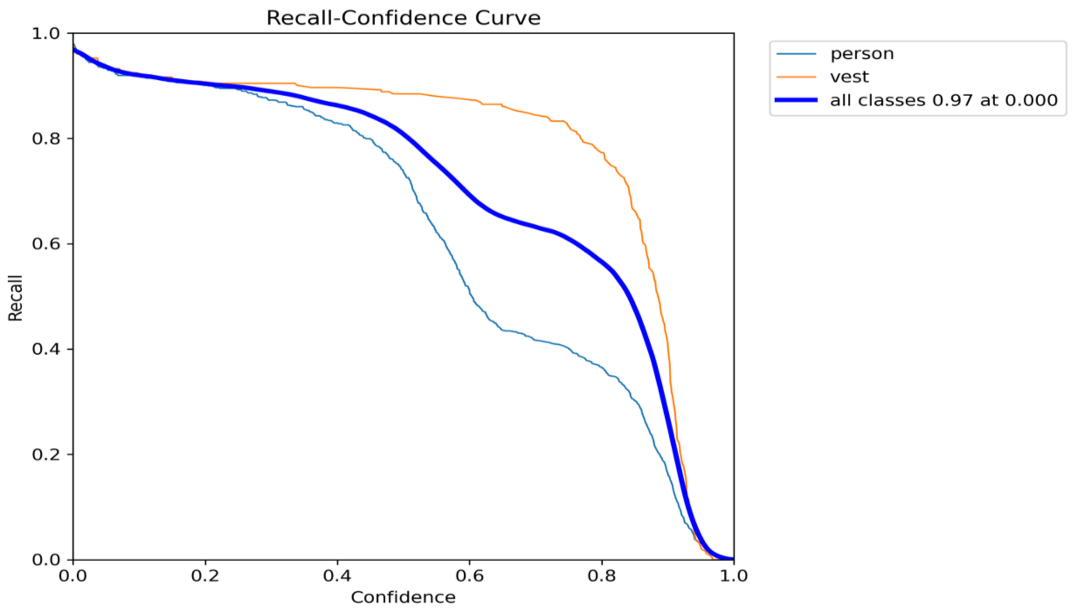


**Figure 8: Recall-Confidence Curve**

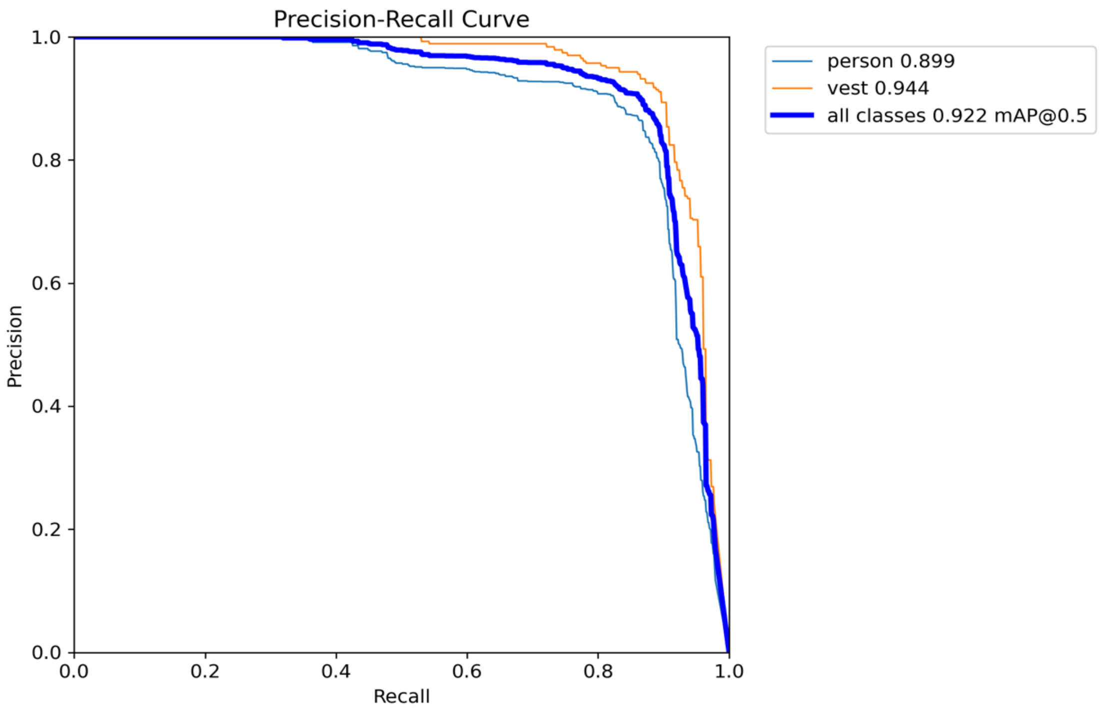


**Figure 9: Precision-Recall Curve**

Figure 10 shows the real-time detection using a network camera. It can be observed that after training, real-time detection is able to correctly identify the classes of "person" and "vest" on the screen. The recognition speed reaches 0.011, demonstrating accurate identification even on computer screens. Due to space limitations, the tests were conducted successfully within a range of 10 meters.


**Figure 10: Real-Time Detection Reflective Vest**

Figure 11 depicts the status of the autonomous vehicle testing obstacle avoidance functionality. When the autonomous vehicle detected obstacles in the path it would circumvent them.

(a) (b)

(c)

**Figure 11: Self-propelled Vehicle Obstacle Avoidance**

## IV. CONCLUSIONS

The experiment encompassed two primary components: autonomous vehicle patrolling and obstacle avoidance, along with object image recognition. Traditional unmanned patrol vehicles often rely on infrared or temperature sensors for detection, which can limit their ability to discern intruders from regular personnel effectively. In contrast, the autonomous vehicle in this study adopted LiDAR scanning to create spatial maps, enabling efficient autonomous patrolling and obstacle avoidance. For detection, a camera was used, and a trained YOLOv8 model was employed to distinguish targets, triggering the Arduino-controlled electric gun alarm if an intruder is identified. Additionally, the YOLO model included training for recognizing personnel wearing reflective vests to avoid inadvertently triggering the alarm on authorized individuals. The YOLO model demonstrated rapid real-time detection at 0.012 seconds, indicating a swift and effective identification process. Achieving a precision rate of 0.944 and mAP of 0.922, the model showcased high performance metrics. By integrating YOLO image recognition with the autonomous vehicle's navigation and obstacle avoidance capabilities, the experiment proved to be a viable solution. The successful addition of an electric gun deterrent feature, coupled with the model's ability to distinguish friend from foe, substantiates the overall success of the experiment.

## REFERENCES


[1]. Ministry of the Interior, National Police Agency, (2021). 2021 Investigation report.

[2]. Joseph Redmon and Ali Farhadi, (2017). "YOLO9000: Better, Faster, Stronger." Computer Vision and Pattern Recognition (CVPR). pp. 6517-6525.

[3]. Juan Du , (2018). "Understanding of Object Detection Based on CNN Family and YOLO" , Journal of Physics,Vol.1004.

[4]. Huang, Yu, (2020). "Autonomous Vehicle Based on ROS", Master Thesis, National Taipei University of Technology.

[5]. Ultralytics, ''ALGORITHM PRINCIPLES AND IMPLEMENTATION WITH YOLOV8'', Available: https://mmyolo.readthedocs.io/en/latest/recommended_topics/algorithm_descriptions/yolov8description.html..

[6]. L. Yann, B. Yoshua and H. Patrick, (1998). ''Gradient-based Learning Applied to Document Recognition'', Proceeding of IEEE, pp. 2278-2324, November.

[7]. Chen, Yi-Ja, (2015). *TensorFlow and Keras: Python deep learning*. Chi-Biao Book Company.

[8]. H. Durrant-Whyte, T. Bailey, (2006). "Simultaneous Localization and Mapping: PartI," IEEE robotics & automation magazine, vol. 13, pp 99-110, June.