Real-Time Fire and Smoke Detection for Trajectory Planning and Navigation of a Mobile Robot

Pham Van Bach Ngoc
Sefas Department, Space Technology Institute, Vietnam Academy of Science and Technology, Vietnam
pbngoc@imech.vast.vn (corresponding author)

Le Huy Hoang
Sefas Department, Space Technology Institute, Vietnam Academy of Science and Technology, Vietnam
hoanghuylea218218@gmail.com

Le Minh Hieu
Sefas Department, Space Technology Institute, Vietnam Academy of Science and Technology, Vietnam
leminhhieu21042001@gmail.com

Ngoc Hai Nguyen
Sefas Department, Space Technology Institute, Vietnam Academy of Science and Technology, Vietnam
nguyenngochai1412001@gmail.com

Nguyen Luong Thien
Sefas Department, Space Technology Institute, Vietnam Academy of Science and Technology, Vietnam
nlthien@sti.vast.vn

Van Tuan Doan
Hung Yen University of Technology and Education, Vietnam
doanvantuan@utehy.edu.vn

Received: 14 August 2023 | Revised: 25 August 2023 and 31 August 2023 | Accepted: 2 September 2023

 DOI: https://doi.org/10.48084/etasr.6252

ABSTRACT

Mobile robots have many industrial applications, including security, food service, and fire safety. Detecting smoke and fire quickly for early warning and monitoring is crucial in every industrial safety system. In this paper, a method for early smoke and fire detection using mobile robots equipped with cameras is presented. The method employs artificial intelligence for trajectory planning and navigation, and focus is given to detection and localization techniques for mobile robot navigation. A model of a mobile robot with Omni wheels and a modified YOLOv5 algorithm for fire and smoke detection is also introduced, which is integrated into the control system. This research addresses the issue of distinct objects of the same class by assigning each object a unique identification. The implementation not only detects fire and smoke but also identifies the position of objects in three-dimensional space, allowing the robot to map its environment incrementally for mobile navigation. The experimental results demonstrate the high accuracy achieved by the proposed method in identifying smoke and fire.

Keywords-mobile robots; robot mapping; smoke and fire real-time detection; dynamic environment; localization; presumptive environment; robot navigation; object identification

I. INTRODUCTION

The accurate navigation of autonomous vehicles depends on precise information about the robot's state and surroundings, including position, orientation, speed, and acceleration. While GPS is widely used for position estimation, it can be limited by environmental conditions and sensor cost. To address these limitations, a camera-based method for position estimation that

eliminates the reliance on GPS is proposed. This method utilizes visual information to improve accuracy and reduce cost in autonomous vehicle navigation. Cameras with depth sensors, such as monocular, stereoscopic, or omnidirectional cameras, can be used in different localization strategies. These strategies include map-building systems, map-based systems, and maple systems [1]. While SLAM and deep learning-based approaches have been proposed for path planning, they may not adequately handle dynamic environments. To address this, we present a visual-based localization approach implemented through a deep learning architecture in this article. Furthermore, we propose a method that utilizes a camera equipped with YOLOv5 + object tracking for detecting and tracking objects like fire and smoke, enabling the construction of 3D maps of these objects. Through this method, the robot can navigate towards the identified location of the fire.

II. RELATED WORK

There is a growing interest in the field of smoke and fire detection, primarily due to its significant applications in various domains. Detecting smoke and fire in its early stages plays a vital role in preventing potential disasters and minimizing associated damages. As a result, researchers are extensively investigating and developing innovative techniques and algorithms to enhance the accuracy and efficiency of smoke and fire detection systems. A large body of related work exists in this domain, spanning across multiple disciplines such as computer vision, image processing, machine learning, and sensor technology. Early approaches primarily focused on using traditional image processing methods, relying on color-based features and intensity transformations. In addition, smoke, temperature, or gas sensors can also be used for the purpose of early fire detection. For instance, authors in [2] used an MQ-5 sensor to continuously monitor the surrounding environment and prevent gas leakage, thus reducing the risk of fire and damage [2]. On the other hand, as mentioned above, a feasible method can be used to detect smoke using several techniques such as Haar features, Bhattacharya distance method, SIFT descriptors, Gabor wavelets approach, and SVM classifier to identify the smoke through video processing [3].

Regarding object tracking algorithms, this method is commonly used to track and detect objects, such as the implementation in [4] which combines the Scale Invariant Feature Transform (SIFT) and Singular Spectrum Analysis (SSA) methods. Based on these results, we propose utilizing a combination of the YOLOv5 deep learning model with object tracking algorithms to detect and provide early alerts of fires in real-time through a camera.

III. PROPOSED APPROACH

A. *Object Detection with the Modified YOLOv5 Model*

The YOLOv5 network consists of three main components, namely the backbone, neck, and head. The backbone is responsible for aggregating and generating image features from various fine-grained images through a Convolutional Neural Network (CNN). The neck, on the other hand, comprises a series of network layers that mix and combine these image features and transfer them to the prediction layer. Finally, the head utilizes the image features to produce bounding boxes and predict the categories of objects. To create different versions of the YOLOv5 network model, the width and depth of the backbone network are adjusted using the parameters depth_multiple and width_multiple. These adjustments result in 4 versions termed as YOLOv5x, YOLOv5l, YOLOv5m, and YOLOv5s. Among these versions, YOLOv5s is the simplest, with a depth_multiple of 0.33 and width_multiple of 0.5 [5]. It has the least model parameters and offers the fastest detection speed.

The purpose of the SPP (Spatial Pyramid Pooling) structure [6] in YOLOv5 network is to generate a fixed size feature vector as the output of a fully connected layer, regardless of the input image size. The SPP structure utilizes three convolution kernels with sizes of 3, 5, and 9, in order to extract features through the use of maximum pooling. This enhances the expressive capability of the feature graph and expands the network's receptive field. Figure 1(a) illustrates the SPP structure, which first applies 1×1, 3×3, 5×5, and 9×9 maximum pooling operations to the data transferred from the convolutional normalization activation function (Convolution + Banch Normalization + SiLU (Sigmoid Linear Unit), CBS (Conv-BN-SiLU)) in parallel. These results are then joined to the CBS structure using concatenation splicing to accomplish feature fusion and complete the feature extraction process. However, the original SPP structure increases the computational workload due to the parallel pooling operations with differently-sized convolutional kernels, which affects performance. As a result, the Spatial Pyramid Pooling Fast (SPPF) structure is introduced to improve the pooling performance while reducing computational requirements. The SPPF structure substitutes the parallel maximum pooling operation from three convolutional kernels with different sizes in the original SPP, with a sequential operation using three convolutional kernels of the same size. As depicted in Figure 1, the SPPF structure bears resemblance to the SPP structure. Initially, the SPPF layer conducts a successive 5×5 maximum pooling operation on the data obtained from the CBS structure [7]. Subsequently, the data are merged into the CBS structure through concatenation splicing, enabling acceleration in processing speed while simultaneously obtaining a more comprehensive feature extraction.

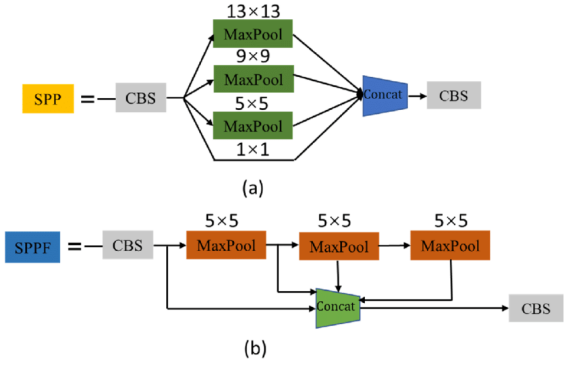

Fig. 1. (a) SPP structure diagram and (b) SPPF structure diagram.

Authors in [5] made some modifications to the SPP module with quite promising results. However, with the proposed multi-scale architecture of SPPD, there may be some difficulties such as more complex computations for the computer. With the embedded Jetson Nano computer, we need a lightweight model that still maintains faster recognition speed. Therefore, it is necessary to modify the YOLOv5 model by reducing the number of max pooling layers and replacing the activation function with Leaky ReLU (Leaky Rectified Linear Unit) in the SPPF module. Modifying the YOLOv5 model may have some impacts on the model. Some possible impacts are the reduction of the number of max pooling layers. Max pooling is a transformation that not only reduces the size of feature maps but also decreases the accuracy of features by selecting the maximum value within the max pooling region. Reducing the number of max pooling layers can help retain more detailed information about the object's positions in the image.The advantages of replacing SiLU with Leaky ReLU can be described as:

- No gradient vanishing problem: The biggest advantage of Leaky ReLU over SiLU is that it avoids the issue of the gradient approaching zero. In SiLU, the derivative of the activation function is nearly zero as the input value approaches infinity, making backpropagation difficult and leading to gradient loss. On the other hand, with Leaky ReLU, the gradient does not vanish and can be easily propagated, helping improve training efficiency.
- Ability to learn complex models: Leaky ReLU allows the model to learn more complex non-linear properties by introducing negative outputs. This helps the model capture non-linear transformations in the data, while increasing flexibility and modeling ability.
- Positivity and fast computation: The Leaky ReLU activation function simply multiplies negative inputs by a small non-zero coefficient, while positive inputs remain unchanged. This makes the positivity of Leaky ReLU higher than SiLU. Moreover, the computation of Leaky ReLU is simpler than SiLU, which helps increase computation speed in the model.

Therefore, reducing the number of max pooling layers and replacing the activation function with leaky ReLU in the SPPF module of YOLOv5 can improve the model's object recognition capability and help the model converge better during training. Figure 2 shows the modified SPPF module of YOLOv5. Figure 3 demonstrates the training results of the modified YOLOv5 model.

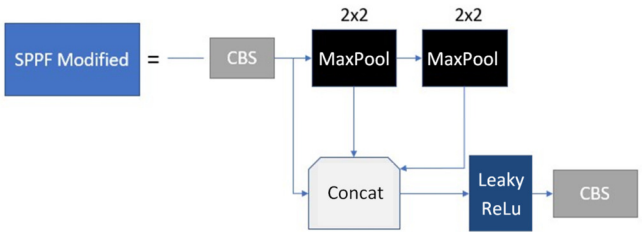

Fig. 2. Modified SPPF structure.

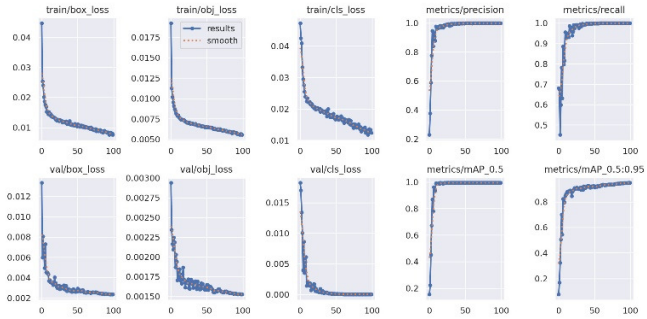

Fig. 3. Model training graph.

Through training the model, we observed the accuracy and strength of the early fire detection model using the modified YOLOv5 algorithm. This enables early fire alerts to be efficiently and quickly triggered, thereby helping to prevent or minimize damage. The basic system meets the necessary accuracy requirements and avoids data overfitting, while the output probability meets the evaluation and early fire alert goals. After experimenting with the Jetson Nano Developer Kit, it was observed that the system can detect at a speed of about 62 frames per second, meeting the set target and utilizing the available devices we have.

B. Data Preparation and Processing

Detecting smoke accurately in practice is a challenging task due to complex possible scenarios and limited image information. Creating a training dataset with paired smoke images and bounding box positions is a major challenge for comprehensive learning. However, well-constructed datasets can effectively train discriminative models as smoke detectors, ensuring accurate detection of smoldering fires. The dataset Smoke 100k [9] was used with options such as high smoke, low smoke, and medium smoke. Fire data, including videos, were collected from the internet. Cleaning the data is crucial but time-consuming, involving removing irrelevant data, outliers, filling missing values, standardizing patterns, and ensuring data confidentiality. Once cleaned, validation is essential to test for errors in the preparation process. The validation step helps identifying any remaining issues that need to be addressed before moving forward. All collected data were divided into 3 categories corresponding for smoke and fire: low, medium, and high. However, to keep it simple in programming, we only used 2 labels: fire and smoke. After collecting the data, we labeled them using the online platform makesense.ai [8]. Subsequently, we converted the annotated image into a text format based on the YOLO format.

C. Object Tracking Algorithm

In this algorithm, we use object tracking in combination with YOLOv5. Object tracking is the task of automatically identifying objects in a video and calculating the trajectories of each object. Object tracking is often used in the surveillance fields such as automatic number plate recognition. So that, we can see the difference with usual object detection algorithms is YOLO does not distinguish objects of the same class and YOLOv5 + object tracking gives a unique id in each object.

Assumption: The detector produces a detection per frame for every object to be tracked. Detections of an object in consecutive frames have a high overlap rate.

$$IOU\left(S_1, S_2\right) = \frac{S_1 \cap S_2}{S_1 \cup S_2} \quad (1)$$

Inputs:

$$\left\{D_0, D_1, ..., D_{F-1}\right\} = \begin{Bmatrix} \left\{d_0, d_1, ..., d_{N-1}\right\}, \left\{d_0, d_1, ..., d_{N-1}\right\}, \\ ... \left\{d_0, d_1, ..., d_{N-1}\right\} \end{Bmatrix} \quad (2)$$

where D_f represents the detections at frame f and d_f represents the j^{th} detection at that frame.

Figure 4 shows the flowchart of the proposed algorithm. To fully supervise the workspace, the robot's motion trajectory is determined based on the workspace. Figure 5 shows the robot's trajectory as it moves in the working environment, with coordinates determined by the encoder signal during its movement.

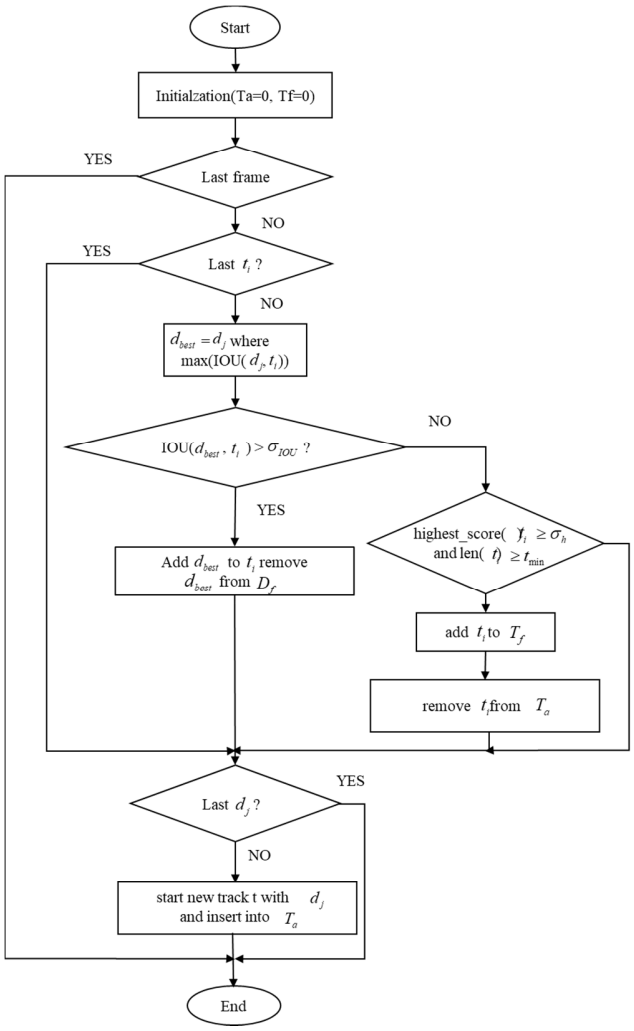

Fig. 4. Algorithm of YOLOv5 + object tracking.

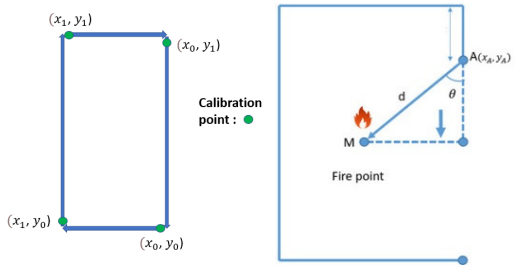

Fig. 5. The trajectory of the robot in the workspace and when fire is detected.

When a fire point is detected, the robot's trajectory is adjusted based on the distance and angle calculated using the stereo camera. To approach and alert the fire point, the trajectory changes to a vector with the robot's position at the detection point as the starting point and the location of the fire as the endpoint.

D. Navigation

The robot has omni wheels to enhance its flexibility when operating within the workspace. Moreover, the omni wheels, in combination with the three-wheeled robot structure, are highly suitable for adjusting the camera's vision flexibly. During movement, based on the omnidirectional nature of the omni wheels, the robot can follow the desired trajectories or rotate at desired angles in a stationary position. This not only enables the robot to fast and flexibly reach fire-risk locations but also facilitates camera vision adjustments. In this way, the camera can be directed more directly toward the fire location, thus enhancing the accuracy of the fire hazard recognition model.

The robot navigates along the predetermined trajectory, enabling it to perform monitoring tasks and alerting upon fire detection. The starting position serves as the initial point, and if no fire is detected, the robot continues to navigate throughout the supervised area. The completion of a navigation cycle is indicated when the robot returns to its initial position. This navigation process is repeated in the absence of fire detections. When a fire is detected, the robot must accurately navigate along a new trajectory to make a θ degree turn and move a distance of d .

The multi-directional robot using omni wheels is described in [10]. The robot is positioned within the $\left(O_w \ \hat{X}_w \ \hat{Z}_w\right)$ coordinate system. This mobile robot consists of the robot body and three omnidirectional wheels labeled 1, 2, and 3. The wheels are arranged equidistantly from the center of the robot chassis with a spacing of 120 degrees. The parameters for an omnidirectional mobile robot are:

M_t : the total mass of the mobile robot, including the robot's body mass and the components' mass.

L_C : the radius of the mobile robot.

f_1, f_2, f_3 : the reaction force applied by the ground to the omni-directional wheel, where the direction is vertical to wheel axes 1, 2, and 3, respectively

R_w: the radius of the omni- directional wheels.

δ_C : the angle between wheel 2 and the $\hat{Z}_M$ -axis of the mobile coordinate system.

φ: the rotation angle of the omnidirectional mobile robot.

I_y: the moment of inertia of the omni-directional mobile robot about the $\hat{Y}_M$ -axis.

$\theta_1, \theta_2, \theta_3$: the rotation angle of omni-directional wheel 1, 2, and 3, respectively.

ω_1, ω_2, ω_3: the angular velocity of omni-directional wheels 1, 2, and 3, respectively.

$\begin{bmatrix} Z_W & X_W \end{bmatrix}^T$: the position for the center of mass of the robot relative to the coordinate system.

The dynamics and kinematics of the mobile robot can be described by:

$$\begin{bmatrix} \dot{z}_W \\ \dot{x}_W \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} M_t & 0 & 0 \\ 0 & M_t & 0 \\ 0 & 0 & I_y \end{bmatrix} \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -0.5 & -0.5 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \\ L_c & L_c & L_c \end{bmatrix} \quad (3)$$

where $\dot{z}_W$, $\dot{x}_W$, $\dot{\phi}$ are the recent coordinates of the robot, determined by input parameters like M_t, I_y, $\sin\phi$, $\cos\phi$, and L_c. Equation (14) describes the relationship between these parameters.

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \frac{1}{R_w} \begin{bmatrix} 1 & 0 & L_C \\ -0.5 & \sqrt{3}/2 & L_C \\ -0.5 & -\sqrt{3}/2 & L_C \end{bmatrix} \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{z}_W \\ \dot{x}_W \\ \dot{\phi} \end{bmatrix} \quad (4)$$

Based on the recent coordinates of the robot, the velocity in each wheel is calculated by (14). We can see that the velocity in each wheel depends on the radius of the Omni directional wheel. In the first matrix, the 3×3 parameters of the matrix are $\frac{1}{2}; \frac{-1}{2}; \frac{\sqrt{3}}{2}; -\frac{\sqrt{3}}{2}; 0$ created by the initial angle (30 degrees) and the radius of the robot. The next 3×3 matrix is created by rotating by angle θ and the final 1×3 matrix is based on the recent robot coordinates. The omnidirectional wheels in this study were powered by DC servo motors. The model for a DC motor can be simplified to:

$$\tau_m = \frac{K_t}{R_a} u - \frac{K_t^2}{R_a} \omega_m \quad (5)$$

where τ_m is the motor torque, calculated by u which is the control voltage, K_t is the motor torque constant, ω_m is the angular velocity of the motor, and R_α is the armature resistance. The traction force f of the wheel is given by the calculated motor torque [11]:

$$f = \frac{n}{R_\omega} \tau_m \quad (6)$$

where n is the gear ratio. The three motors used in this mobile robot are identical. Therefore, combining (5) and (6), the relationship between f_1 , f_2 , u, and ω is given by:

$$\begin{bmatrix} f_1 & f_2 & f_3 \end{bmatrix}^T = \frac{nK_t}{R_\omega R_a} \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^T - \frac{n^2 K_t^2}{R_\omega R_a} \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^T \quad (7)$$

The dynamics of the robot can be presented as:

$$\ddot{P}_\omega = A_\omega \dot{P}_\omega + B_\omega(\phi) U_C \quad (8)$$

where:

$$P_\omega = \begin{bmatrix} z_W & x_W & \phi \end{bmatrix}^T \quad (9)$$

$$U_C = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^T \quad (10)$$

$$a_1 = \frac{-3n^2 K_t^2}{2R_\omega^2 M_t R_a} \quad (11)$$

$$a_2 = \frac{-3n^2 K_t^2 L_C^2}{R_\omega^2 I_y R_a} \quad (12)$$

$$b_1 = \frac{nK_t}{2R_\omega M_t R_a} \quad (13)$$

$$b_2 = \frac{nK_t L_C}{R_\omega I_y R_a} \quad (14)$$

$$A_\omega = \begin{bmatrix} a_1 & 0 & 0 \\ 0 & a_1 & 0 \\ 0 & 0 & a_2 \end{bmatrix} \quad (15)$$

$$B_\omega(\phi) = \begin{bmatrix} 2b_1\cos\phi & -b_1\cos\phi - \sqrt{3}b_1\sin\phi & -b_1\cos\phi + \sqrt{3}b_1\sin\phi \\ 2b_1\sin\phi & -b_1\sin\phi + \sqrt{3}b_1\cos\phi & -b_1\sin\phi - \sqrt{3}b_1\cos\phi \\ b_2 & b_2 & b_2 \end{bmatrix} \quad (16)$$

We assume that the predicted fire-point of the ball in the $\left(O_w \quad \hat{X}_w \quad \hat{Z}_w\right)$ coordinate system is $\begin{bmatrix} z_{b\omega}(t) & x_{b\omega}(t) \end{bmatrix}^T$. The position reference to the mobile robot is set to be:

$$P_{b\omega}(t) = \begin{bmatrix} z_{b\omega}(t) & x_{b\omega}(t) & \phi \end{bmatrix}^T \quad (17)$$

where the rotation angle φ is assumed to be 0. The tracking error is defined as:

$$e(t) = P_{b\omega}(t) - P_\omega(t) \quad (18)$$

The tracking error is definitely important because of its role in the PID controller, which is described by:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d e(t) \frac{d}{dt} \quad (19)$$

From $e(t)$, $\ddot{e}$ can be given by:

$$\ddot{e} = \ddot{P}_{b\omega} - \left[A_\omega \dot{P}_\omega + B_\omega(\phi) U_C \right] \quad (20)$$

The definition of a new control input U is:

$$U \triangleq \ddot{P}_{b\omega} - \left[A_{\omega}\dot{P}_{\omega} + B_{\omega}(\phi)U_C \right] \quad (21)$$

From the above, we obtain:

$$\frac{d}{dt}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} U \quad (22)$$

The feedback control U_C can be written as:

$$U_C = B_w(\phi)^{-1}\left[\ddot{P}_{bw} - A_w\dot{P}_w - U \right] \quad (23)$$

In (23), the system is decoupled into a linear form, and the PID control algorithm is employed to achieve tracking control and calibrate the speed as expected. Consequently, the subsequent PID control strategy applies in this scenario:

$$\dot{\varepsilon} = e \text{ and } U = -K_p e - K_d \dot{e} - K_i \varepsilon \quad (24)$$

where K_d, K_p, and K_i are 3×3 diagonal PID gain matrixes with equal diag$\{k_{d_i}\}$, diag$\{k_{p_i}\}$, and diag$\{k_{i_i}\}$, respectively, and i = 1, 2, 3. The closed-loop tracking error system using PID controller is given by:

$$\frac{d}{dt}\begin{bmatrix} \varepsilon \\ e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ -K_i & -K_p & -K_d \end{bmatrix}\begin{bmatrix} \varepsilon \\ e \\ \dot{e} \end{bmatrix} \quad (25)$$

According to the Routh–Hurwitz criterion, PID gain values k_p, k_i, and k_d must satisfy: $k_{i_i} < k_{d_i} k_{p_i}, i = 1,2,3.$

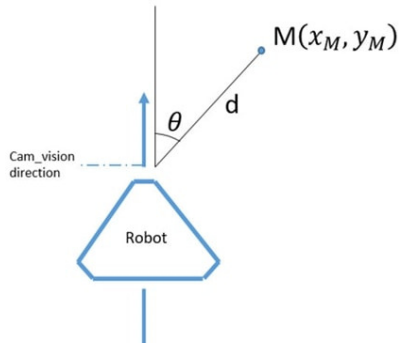

Fig. 6. Rotation and movement of the robot.

The PID control gain values are determined using the proposed control method in order to achieve closed-loop stability. The phase margin and gain margin have been set to 45°. The maneuverability and controllability of a 3-omni wheel robot are inversely related. The robot exhibits exceptional maneuverability due to the flexibility of its omni wheels. However, precise control is necessary to maintain trajectory accuracy and prevent unwanted movements. To achieve this, encoders are installed on each driving motor of the robot to measure wheel speed and distance traveled. In the event of fire detection, the robot's trajectory is modified, and navigation is required to guide the robot along the new path towards the fire. The Jetson Nano and stereo camera determine the distance d and angle θ and the robot rotates by θ degrees and moves straight for the distance d based on this information (Figure 6).

IV. RESULTS AND DISCUSSION

The omni-directional 3-wheel robot successfully detected smoke and fire, navigated to their location utilizing a camera and YOLOv5 algorithm. Object tracking algorithms enabled the tracking of these objects across multiple frames with the help of a camera and Jetson Nano developer kit for detection and processing. Once an object is detected, YOLOv5 identifies its type and provides positional information. The object tracking algorithm tracks the object across frames and calculates the optimal path. With advanced algorithms and information processing, the robot can navigate accurately and safely toward the source of smoke or fire. This feature is valuable in emergency and firefighting scenarios. The results can be demonstrated using an embedded computer connected to a screen, enabling the robot to adjust its trajectory based on fire recognition signals captured by the camera (Figures 7-8). Figure 7 was captured after completing the trajectory change and reaching the fire detection point. In Figure 8, the robot follows a pre-programmed path, but alters its trajectory towards the detected fire location once it receives the camera signal.

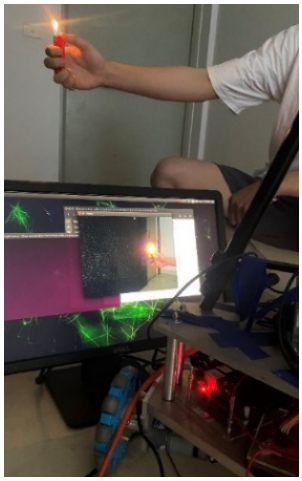

Fig. 7. The result after the robot detects fire and moves to the fire point.

Fig. 8. Initial trajectory, change in trajectory, and movement to the fire location.

V. CONCLUSION

The current research has successfully developed an autonomous omni-directional robot equipped with a camera for navigation and early fire detection. The proposed method enables the robot to detect smoke and fire, calculate the

trajectory, and accurately navigate to the smoke/fire location. The control of the robot model moving along the trajectory is performed precisely according to the kinematics and dynamics problems. By applying the versatility of the omni wheels, the robot moves flexibly and accurately. In addition, the use of modified YOLOv5 in combination with object tracking provides a new solution for object recognition. The model's verification in fire and smoke detection has proven that YOLOv5 performs considerably better than the original model, with an accuracy of up to 95% on our fire and smoke dataset. Moreover, the enhanced YOLOv5 can detect 62 frames per second, meeting real-time demands. The omni robot has demonstrated its ability to identify and locate smoke and fire using the camera, YOLOv5, and object tracking. This advancement holds promise for various applications in robotics technology. However, the trajectory planning method used in this research may not have a high success rate in handling obstacles near the target location. Future work could focus on addressing this limitation to enhance the overall performance of the system.

ACKNOWLEDGMENT

This research is funded by the Vietnam Academy of Science and Technology under grant number VAST01.01/21-22.

REFERENCES

[1] M. S. Güzel, "Autonomous Vehicle Navigation Using Vision and Mapless Strategies: A Survey," *Advances in Mechanical Engineering*, vol. 5, Jan. 2013, Art. no. 234747, https://doi.org/10.1155/2013/234747.

[2] B. F. Alshammari and M. T. Chughtai, "IoT Gas Leakage Detector and Warning Generator," *Engineering, Technology & Applied Science Research*, vol. 10, no. 4, pp. 6142–6146, Aug. 2020, https://doi.org/10.48084/etasr.3712.

[3] P. Matlani and M. Shrivastava, "An Efficient Algorithm Proposed For Smoke Detection in Video Using Hybrid Feature Selection Techniques," *Engineering, Technology & Applied Science Research*, vol. 9, no. 2, pp. 3939–3944, Apr. 2019, https://doi.org/10.48084/etasr.2571.

[4] A. Ozturk and I. Cayiroglu, "A Real-Time Application of Singular Spectrum Analysis to Object Tracking with SIFT," *Engineering, Technology & Applied Science Research*, vol. 12, no. 4, pp. 8872–8877, Aug. 2022, https://doi.org/10.48084/etasr.5022.

[5] Z. Wu, R. Xue, and H. Li, "Real-Time Video Fire Detection via Modified YOLOv5 Network Model," *Fire Technology*, vol. 58, no. 4, pp. 2377–2403, Jul. 2022, https://doi.org/10.1007/s10694-022-01260-z.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," in *Computer Vision – ECCV 2014*, Cham, 2014, pp. 346–361, https://doi.org/10.1007/978-3-319-10578-9_23.

[7] M. Qiu, L. Huang, and B.-H. Tang, "ASFF-YOLOv5: Multielement Detection Method for Road Traffic in UAV Images Based on Multiscale Feature Fusion," *Remote Sensing*, vol. 14, no. 14, Jan. 2022, Art. no. 3498, https://doi.org/10.3390/rs14143498.

[8] "Make Sense." https://www.makesense.ai/.

[9] H.-Y. Cheng, J.-L. Yin, B.-H. Chen, and Z.-M. Yu, "Smoke 100k: A Database for Smoke Detection," in *2019 IEEE 8th Global Conference on Consumer Electronics (GCCE)*, Osaka, Japan, Jul. 2019, pp. 596–597, https://doi.org/10.1109/GCCE46687.2019.9015309.

[10] S.-T. Kao and M.-T. Ho, "Ball-Catching System Using Image Processing and an Omni-Directional Wheeled Mobile Robot," *Sensors*, vol. 21, no. 9, Jan. 2021, Art. no. 3208, https://doi.org/10.3390/s21093208.

[11] M.-T. Ho and H.-S. Wang, "Pid Controller Design With Guaranteed Gain And Phase Margins," *Asian Journal of Control*, vol. 5, no. 3, pp. 374–381, 2003, https://doi.org/10.1111/j.1934-6093.2003.tb00129.x.