International Journal on Semantic Web and Information Systems
Volume 20 • Issue 1

Unmanned Aerial Vehicle Fire Detection Platform Based on Semantic Yolov5 and Autonomous Recognition

Yihang Chen, Shenyang Aerospace University, China
Xiaoying Song, Shenyang Aerospace University, China*
https://orcid.org/0009-0001-4929-3567

ABSTRACT

In recent years, as fire image detection has become a research hotspot. One class of methods is color-based methods, which are very sensitive to brightness and shadows. As a result, the number of false alarms generated by these methods is high. Aiming at the task requirements of airborne binocular vision obstacle avoidance and target tracking, this paper establishes the verification platform architecture of UAV (Unmanned Aerial Vehicle) binocular vision obstacle avoidance and target tracking. For the update and maintenance of boundary regions, we can also continuously extract richer information from the boundary, make more elaborate plans, and develop an incremental method to detect locally updated maps within the boundary. The fire point can be independently and quickly identified through deep learning to extinguish the fire accurately. Assuming that the system incorrectly identifies 2 out of 80 non-fire sources as fire sources, so the results indicate a precision of about 88%, a recall of 90%. However, the traditional fire detection is around 80%.

KEYWORDS

Autonomous Detection, Binocular UAV, Path Planning, Target Tracking

BACKGROUND

Fire threatens human life and property safety, and in severe cases, it can cause huge economic losses and casualties. Nowadays, with the rapid development of the economy and the increasing scale of urban buildings, the fire situation has become more complex, and the difficulty of firefighting also increases. Currently, the main method of firefighting is still manual firefighting by firefighters, which is often accompanied by injuries or even sacrifices of firefighters. Therefore, developing more advanced firefighting methods and using robots to replace manual firefighting has become a research trend. Fire robots mainly use cameras to explore the situation of the fire scene and accurately recognize flame targets based on video images (Chen, 2023; Liang et al., 2024). This is the key to efficient robotic firefighting, and fire robots have significant real-time and accuracy requirements for flame target detection. With the increasing maturity of unmanned aerial vehicle (UAV) technology and the

DOI: 10.4018/IJSWIS.344026
*Corresponding Author

further expansion of aerial photography technology, UAV is increasingly being used in large-scale rescue equipment and intelligent fire detection (Chen, 2022; Wang, 2017).

Purpose of the Study

This article combines UAV platforms, airborne binocular vision, airborne processing computers, development workstations, and visual navigation integrated development environments. Deep learning systems (Kim & Muminov, 2023) can identify fire points quickly, and fire hazard inspections, on-site rescue command, fire detection, and prevention and control can be carried out on complex terrain and structural buildings in the air (Li, 2023). These methods solve the problems of traditional fire detection methods and improve the efficiency and accuracy of fire detection.

RELATED WORK

Research Status of UAV Fire Detection

In the field of UAV fire detection, researchers usually use image processing and machine learning techniques to achieve fire recognition. Among them, the methods based on image processing include extracting color features, texture features, and shape features, and motion detection based on optical flow and inter-frame difference techniques. Machine learning methods include traditional support vector machines (SVM), random forests, and deep learning methods such as convolutional neural networks (CNNs). In recent years, deep learning-based methods have achieved remarkable results in fire recognition. Among them, the you only look once v5 (YOLOv5) algorithm is an efficient object detection algorithm suitable for processing large-scale data sets and achieving real-time object detection. The YOLOv5 algorithm combines the characteristics of fast, accurate, simple, and lightweight and has high computational efficiency and recognition accuracy.

In UAV fire detection, researchers can use the YOLOv5 algorithm for fire recognition. By training the model to use the fire image data set taken by the UAV, the rapid detection and location of the fire area can be realized, and disaster relief measures can be taken in time. In addition, the YOLOv5 algorithm can combine other sensor data, such as infrared images and smoke sensor data, to improve the accuracy and reliability of fire detection. In general, fire recognition methods based on image processing and machine learning have achieved certain results in the field of UAV fire detection, and YOLOv5 algorithm, as an efficient object detection algorithm, is expected to provide more possibilities for the further development of fire detection technology.

YOLOv5 Algorithm and Its Application in Object Detection

YOLOv5 is a lightweight object detection algorithm that uses a single-stage detection method and combines deep learning and object detection technology. The basic principle is to segment the whole image into different grid cells, and each grid cell performs object detection by predicting the center coordinates, width and height of the bounding box, and the object category score. Compared with YOLOv4, YOLOv5 simplifies and optimizes the network structure and improves the speed and accuracy of object detection.

The advantages of the YOLOv5 algorithm are mainly reflected in its simplicity and efficiency. YOLOv5 simplifies and optimizes the network structure, reduces the number of parameters and calculations, and improves the speed and efficiency of object detection. By introducing a new data augmentation strategy and optimizing the training method, YOLOv5 achieves higher accuracy in object detection tasks. Cross-platform deployment, YOLOv5 supports the deployment of multiple frameworks and hardware, which is convenient for application and debugging in different environments.

In practical applications, YOLOv5 is widely used in image and video object detection, including face recognition, vehicle detection, pedestrian detection, and other scenes. Compared with traditional object detection algorithms, YOLOv5 has faster inference speed and higher accuracy, especially on large-scale data sets. At the same time, due to its simple and lightweight structure, YOLOv5 can also achieve efficient object detection on mobile terminals and embedded devices, which has a wide range of application prospects.

The Role of Semantic Segmentation Techniques in Fire Recognition

The role of semantic segmentation technology in fire recognition is mainly reflected in the classification and segmentation of fire images at the pixel level, and the fire area is extracted and assisted in fire detection and recognition. Through semantic segmentation technology, the accurate segmentation and location of fire images can be realized, and then the accuracy and efficiency of fire recognition can be improved. In fire recognition, semantic segmentation technology can help identify fire features such as flame and smoke and segment fire areas effectively. By analyzing the pixel information of the fire area, the real-time monitoring and warning of the fire can be realized, and the corresponding rescue measures can be taken in time to improve the efficiency and accuracy of the fire response.

Combined with the target detection algorithm, the performance of fire recognition can be further improved. The target detection algorithm can detect fire alarm facilities, fire equipment, and other fire-related targets. Combined with the segmentation results of semantic segmentation technology, the specific target recognition and location in the fire image can be realized, a more comprehensive understanding of the fire scene, and the speed of fire recognition and response can be accelerated. Semantic segmentation technology has important application potential in fire recognition. By combining with object detection algorithms, it can realize the extraction and analysis of multi-dimensional information of fire recognition, improve the timeliness and accuracy of fire response, and provide strong support for fire prevention and rescue work.

DESIGN OF FIRE DETECTION PLATFORM

Overall Platform Architecture

The fire detection robot includes a UAV platform and other elements such as airborne binocular vision, airborne processing computer, development workstation, and visual navigation integrated development environment. Stereo-matching technology calculates the depth of information in the two-dimensional (2D) image. At the same time, visual-inertial fusion technology estimates the motion attitude of the camera itself (Galiautdinov, 2021). The depth information of multiple frames is fused to reconstruct the voxel map of the spatial environment. The obstacle distance information is obtained from the map to generate the executable path.

The visual development platform consists of a visual development training environment (Xie et al., 2023), an airborne visual processing development system, a visual positioning platform (Zhang et al., 2024), and an environment reconstruction system. The information interaction and task control platform include information interaction programs, obstacle avoidance and path planning algorithms, and UAV position navigation control programs. The fire detection robot can quickly identify the fire point, and the fire hazard inspection, on-site rescue command, fire detection and prevention, and control of complex terrain and complex structure buildings through the air become a new choice for the fire force.

Our UAV experimental platform is shown in Figure 1, which includes a UAV platform, onboard binocular vision, onboard processing computer, development workstation, and visual navigation integrated development environment. The software framework is shown in Table 1, and the hardware is shown in Figure 2. Each hardware directly communicates through a serial port.

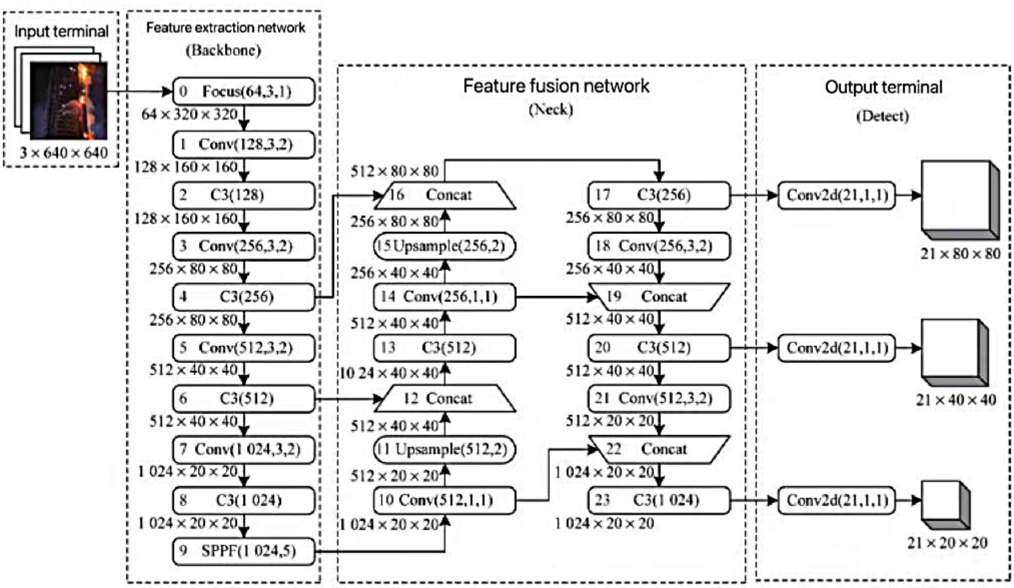
The flight control of the UAV experimental platform adopts Pixhawk 4. Pixhawk 4 is one of the core hardware components of the PX4 (UAV autonomous driving platform) flight control system. It is an open-source flight control software for drones and unmanned aerial vehicles. This paper provides

Figure 1. The UAV Experimental Platform

Table 1. The UAV Software Framework

Type	Content
Flight controller software	PX4 1.8.2
Airborne operating system	Ubuntu 18.04
ROS edition	ROS Melodic
Communication framework	Mavros
Positioning framework	vSLAM

Figure 2. The UAV Hardware Connection Relationship



a flexible toolset for UAV developers to share technology and thus create tailored solutions for UAV applications. The PX4 provides standards for drone hardware support and a software stack, thus allowing the ecosystem to build and maintain the hardware and software in a scalable manner. PX4 is one of the most popular open-source flying control boards. The PX4 software system is firmware, and its core operating system (OS) is the embedded RTOS (NuttX) real-time Acorn RISC Machine (ARM) system. The firmware comes with a series of toolsets, a system driver module, and a peripheral software interface layer, all this software (including user-defined flight control software), together with the OS kernel, is compiled into the firmware and then uploaded to the flight control board to realize the software configuration of the flight control board.

Some of the main functions of the PX4 are that it can control many different equipment racks and types, including aircraft (multirotor, fixed-wing, and vertical take-off and landing), ground vehicles, and underwater vehicles. It is suitable for hardware selection for equipment controllers, sensors, and other peripherals. It also has a flexible and powerful flight mode and safety features. The PX4 ground control station, QGroundControl, is an integral part of the PX4 self-drive system and can run on multiple platforms, such as Windows, macOS, and Linux. With QGroundControl, you can burn the PX4 firmware to the hardware, set the machine, change different parameters, get real-time flight information, and create and perform fully autonomous tasks.

Regarding control protocol and logic, the PX4 provides good native support for the micro air vehicle link (MAVLink) protocol (one of the most common UAV flight control protocols). This protocol can be used both for ground station (GCS) control of UAV and for UAV information feedback of GCS. The flight control scenario is usually like this:

1. Manual flight control: GCS → (MAVLink) → UAV
2. Information collection: GCS ← (MAVLink) ← UAV
3. Autonomous flight control: User App → (MAVLink) → UAV

If you want to realize the ground station control flight, the ground station uses the MAVLink protocol to send control commands to the UAV through the radio frequency channel (or Wi-Fi). To achieve autonomous drone flight, you can write your own application (running on the drone system) to send local control commands to the drone using the MAVLink protocol. However, in order to achieve the flexibility of the flight control architecture and avoid dependence on the underlying implementation details, in PX4, developers are not encouraged to use MAVLink directly in custom flight control programs but are encouraged to use a micro-object request broker (uORB), which requests the Broker messaging mechanism. In fact, uORB is conceptually equivalent to the named pipe in the portable operating system interface (POSIX), which is essentially an inter-process communication mechanism. Since PX4 uses the NuttX real-time ARM system, uORB is equivalent to multiple processes (driver-level modules) opening the same device file, and multiple processes (driver-level modules) interact and share data through this file node.

In the uORB mechanism, messages exchanged are called topics, and a topic contains only one message type (that is, a data structure). Each process (or driver module) can *subscribe* to or *publish* multiple topics; there can be multiple publishers for a topic, and a subscriber can subscribe to multiple topics. Because of the existence of the uORB mechanism, the above flight control scenario becomes:

1. Manual flight control: GCS → (MAVLink) → (uORB topic) → UAV
2. Information collection: GCS ← (MAVLink) ← (uORB topic) ← UAV
3. Autonomous flight control: User App → (uORB topic) → (MAVLink) → UAV

With the above background, you can write your own flight control logic, just add a custom module in the PX4 source code, and then use uORB to subscribe to relevant information (such as sensor messages) and publish relevant control information (such as flight mode control messages). The specific uORB API and uORB message definitions refer to the PX4 documentation and source code, and all control commands are in the msg of the firmware code and are not described.

Finally, it is worth mentioning that in the PX4 system, there is also a dedicated module called MAVLink, the source code is in the firmware folder *src/modules/mavlink*, MAVLink is quite similar to the Linux console command tool set. It can be used as a command under the NTT controller or as a system module to load and run in the background. It implements the following functions: (1) uORB message parsing, the actual translation of uORB messages into specific MAVLink underlying instructions, or vice versa, and (2) obtaining or sending MAVLink messages through serial RF communication interface takes into account the development mode of user-written programs and also applies to the development mode of script toolchain similar to Linux, which is very flexible to use.

Airborne Board

The Nvidia Jetson TX2 delivers exceptional speed and energy efficiency to users' embedded AI computing devices. This supercomputer module uses Nvidia Pascal GPU, up to 8 GB of memory, 59.7 GB/s of memory bandwidth, and provides a wealth of standard hardware interfaces, which perfectly adapt to various products and form factors to realize a true AI computing terminal. The 50 mm × 87 mm module size allows deep learning technology to be truly applied to small products such as drones. It has good performance—more than doubled—and twice the energy efficiency compared to Jetson TX1, thanks to Jetson TX2 256core Nvidia Pascal architecture and 8 GB of RAM for faster calculations and more reasoning power (Wang, 2016).

Optimize user energy efficiency, and with Jetson TX2, you can instantly run large deep neural networks on edge devices, achieving greater accuracy. With a power consumption of only 7.5 watts, it is more than 25 times more energy-efficient than today's most advanced desktop-class central processing units (CPUs). This makes it very suitable for real-time processing processes in applications with high bandwidth and latency requirements. These include factory robots, commercial drones, enterprise collaboration devices, and smart cameras for smart cities, among others.

Jetson TX2 features Nvidia Pascal graphic processing units (GPUs) with 256 compute unified device architecture (CUDA) capable cores. The CPU complex consists of two clusters of ARM v8 64-bit CPUs connected by a high-performance coherent interconnect structure. A Denver-2 (dual-core) CPU cluster is optimized to improve single-thread performance. The second CPU cluster is an ARM Cortex-A57 QuadCore, which is more suitable for multi-threaded applications.

The memory subsystem contains a 128-bit controller, which provides high-bandwidth low-power double data rate 4 (LPDDR4) synchronous dynamic random access memory (SDRAM) support. The module also integrates 8 GB LPDDR4 main memory and 32 GB embedded multimedia card (eMMC) flash memory. The design change from 64 bits to 128 bits for the TX1 is a major performance boost.

The module also supports hardware video encoders and decoders that support 4K ultra-high definition (UHD) video at 60fps in different formats. This is slightly different from the hybrid Jetson TX1 module, which uses dedicated hardware and software running on the Tegra SoC to accomplish these tasks. Also included is an audio processing engine with full hardware support for multi-channel audio. Jetson TX2 supports Wi-Fi and Bluetooth wireless connectivity. Wi-Fi is a big improvement over the previous Jetson TX1. It also includes a Gigabit Ethernet .Includes Gigabit Ethernet built on top of the basic Ethernet standard..

Importantly, the TX2 has two CPUs and a GPU with 256 CUDA cores. One CPU is a dual-core Denver 2, and the second is an ARM CortexA57, with 8 GB of running memory and 32 GB of flash memory. This powerful developer kit enables motherboard hardware functions and interfaces to take full advantage of the Linux development environment (Fatemidokht et al., 2021). It also supports the Nvidia Jetpack software development kit (SDK), which includes a board support package (BSP), deep learning libraries, computer vision, GPU computing, multimedia processing, and many more features.

The JetsonDevelopment Pack (JetPack) is an on-demand, all-in-one package that bundles and installs all development software tools for the Nvidia Jetson embedded platform. The Nvidia JetPack SDK is the most comprehensive solution for building AI applications. It bundles all of the Jetson platform software, including TensorRT, CUDA deep neural network (cuDNN) library, CUDA toolkit, VisionWorks, Streamer, and OpenCV, which are L4T based on the LTS Linux kernel. JetPack includes:

- Deep learning: TensorRT, cuDNN, and Nvidia deep learning GPU training system (DIGITS) workflows
- Computer vision: Nvidia VisionWorks, OpenCV
- GPU computing: Nvidia CUDA, CUDA library
- Multimedia: internet service provider (ISP) support, camera images, video codec
- Robot operating system (ROS) compatibility (Liu et al., 2021; Fang et al., 2024; Yang, 2019), OpenGL, advanced developer tools, and more.

While considering the stability of the UAV system, we also need to consider the hardware platform, software architecture, and the overall flexibility of the system design. Nvidia Jetson TX2 As a high-performance embedded computing platform, Jetson TX2 provides enough computing power to support complex deep learning models and real-time processing. The modular hardware design improves the scalability of the system. For example, hardware components such as sensors and communication modules can be easily upgraded or replaced by using standardized interfaces and components.

The PX4 flight control software provides a highly configurable flight stack that supports multiple UAV types and configurations. By adjusting the parameters and modules of the PX4, it can adapt to the flight characteristics and mission requirements of different UAVs. The ROS provides a flexible software framework to support the development of various robotic applications. ROS's nodal architecture and rich packages make it easy to add new functions and services to the system to adapt to different tasks and environments.

It is necessary to perform performance matching for different UAV scales, capabilities, or different indoor environments. For small or low-power UAVs, it may be necessary to optimize algorithms and software to reduce computational requirements. For example, a lightweight object detection model can be adopted, or model quantization and pruning techniques can be used to reduce the model size and computational complexity. Select the right combination of sensors. For example, for UAVs that require high-precision obstacle avoidance, lidar or depth cameras can be integrated, while for simple surveillance tasks, using a common red, green, and blue wavelengths (RGB) camera may be sufficient. For UAVs flying for long periods of time, energy management is crucial. Energy consumption can be reduced by optimizing algorithms and flight strategies, or high energy density batteries and energy recovery techniques can be employed.

Camera

Xiaomi binocular camera series products all adopt the binocular camera scheme of "vision + structured light + inertial navigation" fusion (Yang, 2024). As a 3D sensor based on visual recognition technology, Xiaomi binocular camera is suitable for both indoor and outdoor environments. Without fear of an outdoor bright light environment, the recognition distance can reach 15m+, and it can also work in a completely dark indoor environment. In addition, the standard IR active light helps the Xiaomi binocular camera perfectly solve the problem of recognizing indoor white walls and texture-less objects. The "binocular + inertial measurement unit (IMU)" navigation scheme can provide accurate cloud complementary data (Dwivedi, 2022) for visual simultaneous localization and mapping (vSLAM) (Li, 2023) applications and has higher accuracy and robustness than other single schemes.

MYNTAI-D can provide a direct-output depth scheme based on the depth calculation chip without the host computer and can provide depth data output of up to 720p/60fps. In addition, it also provides rich SDK interfaces and vSLAM open-source project support, which can help customers quickly integrate solutions, accelerate the process of product development, and realize the rapid productization and landing of solutions.

Apart from the above important reasons that the Xiaomi binocular camera series products have been favored by the industry since their launch, in addition, each Xiaomi binocular camera has passed more than a dozen reliability tests such as high temperature and humidity continuous work, low-temperature dynamic aging, and sinusoidal vibration of the whole machine in Foxconn laboratory, which has extremely strong industrial control performance. At the same time, self-developed global shutter technology, indoor and outdoor light sensitivity adaptive technology, automatic white balance technology, hardware level frame synchronization, and IMU. Synchronization technology is adopted to ensure that the Xiaomi binocular camera can provide real-time and stable image sources to cope with complex changes in indoor and outdoor scenes.

In addition to the separate demonstration of the Xiaomi binocular structured light depth inertial navigation camera, this time, Xiaomi Intelligent also provides a new live interactive demonstration of the robot visual navigation obstacle avoidance scheme. The Xiaomi intelligent robot is based on a self-developed binocular vision sensor, fuses multiple high-quality solutions, recognizes the display area environment on the spot, reconstructs the 3D map, plans the walking path autonomously, and can easily avoid obstacles and navigate to the specified place autonomously.

Control Station

The simultaneous localization and mapping (SLAM) control station is a desktop host installed with the Ubuntu 18.04 operating system. Its main function is to realize the SLAM UAV remote start, flight trajectory display, path planning route display, and other functions. The station is equipped with a QGC ground station, which can monitor the aircraft parameter state online. The SLAM control station acts as the master node in ROS distributed control to realize multi-computer control.

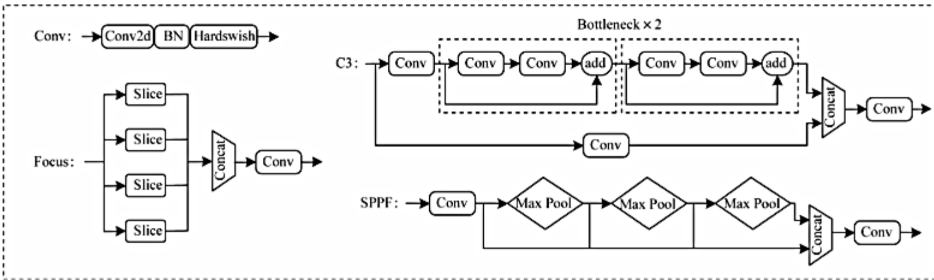
METHODOLOGY: AUTONOMOUS PATH PLANNING ALGORITHM

Autonomous path planning (Lin et al., 2023) environment mapping algorithm plays a crucial role in UAV fire detection systems. These algorithms can help the UAV accurately understand its surroundings and, based on this, plan an optimal path to complete the mission. For the UAV fire detection system, accurate and efficient path planning is of key significance for timely detection and treatment of fires. The importance of the environment mapping algorithm in this system is as follows.

- **Fire scene perception:** The environment mapping algorithm can help the UAV perceive and understand the environmental situation of the fire scene, including buildings, roads, and obstacles. By mapping, the UAV can determine the best path to reach the target area in order to detect and monitor the fire situation.
- **Determining the best path:** Environment mapping and path planning algorithms can help drones determine the best path to reach or fly over the fire as quickly as possible. This can improve the response speed, reduce the risk of fire spread, and provide better support for fire rescue.
- **Obstacle avoidance and risk avoidance:** There may be various obstacles at the fire scene, such as smoke, flames, and buildings. The environment mapping algorithm can help the UAV identify these obstacles and avoid them. In this way, the possibility of UAV encountering risks in performing tasks can be reduced and its safety can be guaranteed.
- **Update the map in real time:** Due to the complexity of the fire scene, changes in the environment can be very fast and drastic. The environment mapping algorithm can update the map in real time and provide accurate environmental information, so that the UAV can make appropriate decisions and actions.

In summary, the importance of the autonomous path planning environment mapping algorithm in a UAV fire detection system is self-evident. Through these algorithms, UAVs can perform tasks in complex and dangerous environments, improve the efficiency of fire detection and treatment, and ensure the safety of people’s lives and property.

Figure 3. Fast-Planner Pseudocode



Application Method

Fast Planner is an efficient global path planning algorithm mainly used to generate long-term planning paths for exploring environments, boundaries, encountering obstacles, and other situations during disaster detection in order to minimize the time and distance of the entire path. This algorithm generates an optimal path between the starting and ending points based on known map information and the current state of the drone. It is very fast in running time and can usually complete path planning within a few seconds, as shown in Figure 3.

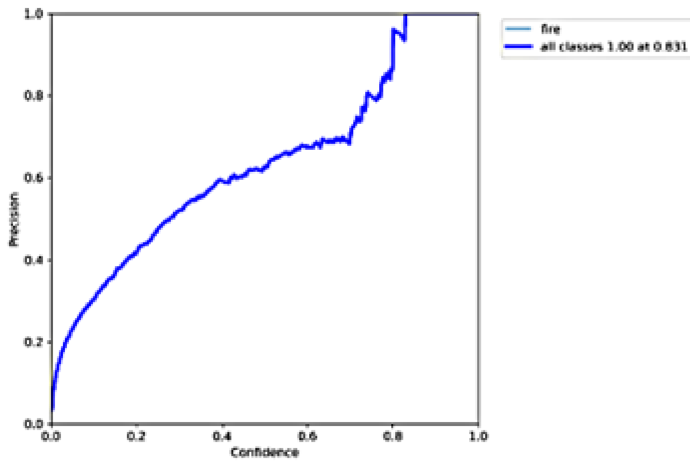
````
Input: Start position (s), Goal position (g), Environmental map (E)
Output: Planned path (P)
1: Initialize P as empty
2: Identify frontiers in E using an exploration strategy (e.g.,
   Voronoi diagram)
3: For each frontier f in frontiers do
4: Calculate the heuristic cost h(f) from f to g
5: If h(f) is lower than the current best option then
6: Add f to P as a potential intermediate waypoint
7: End if
8: End for
9: If P is not empty, then
10: Sort waypoints in P by increasing cost h(f)
11: Connect s to the first waypoint in P, then to each subsequent
    waypoint, and finally to g using a local path planning algorithm
12: Return P
13: Else
14: Return Failure (No feasible path found)
15: end if
```
````

Global Path Planning: The Hybrid A * Algorithm

Search Method

At first, the A* algorithm (Wang et al., 2024; Sun et al., 2024; Hu & Xu, 2023) is applied to search in a 2D grid, essentially simplifying the aircraft into particles with fixed or four directions of movement and a fixed distance of movement. However, this does not match the actual aircraft model. So, the hybrid A * algorithm and the heading angle are introduced to transform the search into three dimensions of
````

Figure 4. Comparison Between the A * Algorithm and the Hybrid A * Algorithm



space, which is consistent with the aircraft kinematic model. Both algorithms are based on the grid world, in which A * assigns a corresponding cost to the center points of each grid and the algorithm only accesses these center points, while hybrid A * first selects points from these grids that meet the 3D continuous state of the aircraft and assigns the cost to these points.

The hybrid A * algorithm cannot guarantee finding the minimum cost solution because it combines continuous coordinate states of the same unit. However, the obtained path is guaranteed to be drivable. Through comparison, as shown in Figure 4, we found that the solution of the hybrid A * algorithm is usually located in the neighborhood of the global optimal solution, allowing for local improvement of the path through gradient descent, thereby achieving the global optimal solution.

Introduction of the Reeds-Shepp Curve

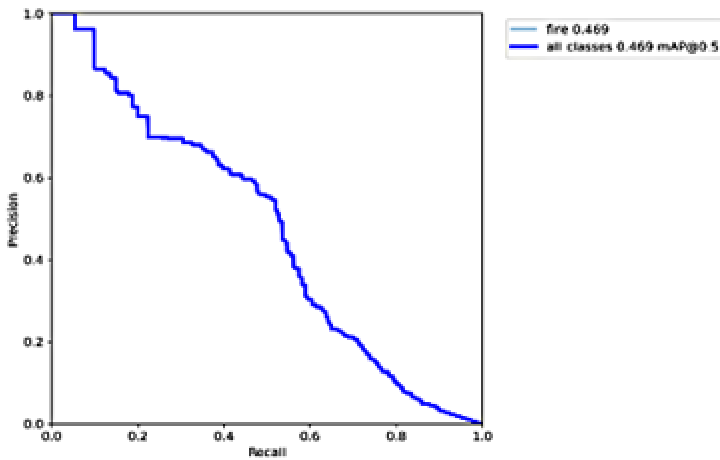
The Reeds–Shepp curve proves that if the shortest path of any starting and ending pose contains a circle and a straight line, it must be obtained at the minimum turning radius (Li, 2023). When the front wheel steering angle is such a value, the path is the shortest and the time consumption of the aircraft is also shorter. This means that each state can lead to six other states, similar to the classical A* moving from one state to eight other states in eight directions. However, classical A* is state sampling, that is, jumping from one state to another, ignoring the kinematic process therein. However, hybrid A* is controlled sampling, which aligns more with the kinematic reality. Of course, real motion may not always continue to move according to the minimum turning radius, but this is done here for the sake of simplicity.

The Reeds–Shepp curve does not consider obstacle avoidance factors at all. Still, due to its simplicity and ease of calculation, the construction speed is very fast, making it possible to construct it first and then check for collisions. Therefore, unlike traditional A *, which does not consider intermediate processes, it is necessary to sample the entire time period for collision detection uniformly. If there is no collision in this trajectory segment, it will be added to the search tree as a candidate trajectory, as shown in Figure 5. When using the Reeds–Shepp curve search, if Reeds–Shepp curves appear, they will collide, and the classic A * search needs to be performed again. This hybrid search improves the search speed.

Definition of the Heuristic Function

In theory, the heuristic function should be the shortest path length from the current node pose to the endpoint node pose while satisfying obstacle avoidance and aircraft kinematic constraints. This

Figure 5. Candidate Trajectories in the Search Tree



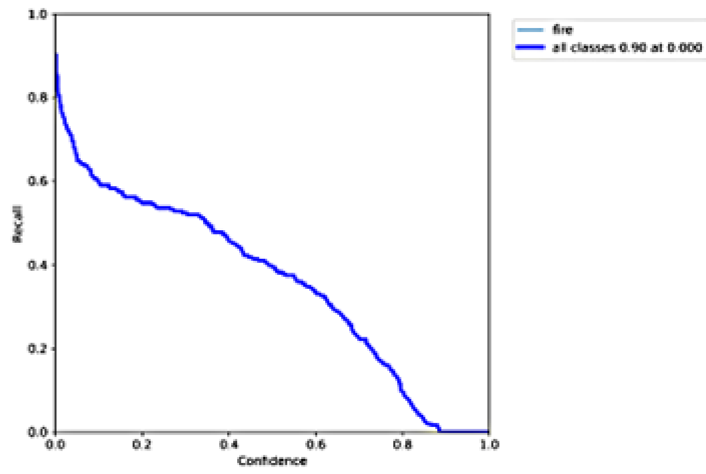
is the true path, but it is difficult to know it before sampling and searching the remaining location environment. So, hybrid A * designs two sub-functions of the heuristic function, one representing the shortest path that meets the aircraft kinematic constraints but ignores collision factors and the other representing the shortest path that meets the obstacle avoidance constraints but ignores the aircraft kinematic constraints. The heuristic function is defined as the maximum value of both.

Constrained heuristics. When the current node is closer to the endpoint, more attention should be paid to the aircraft kinematic constraints. Ignoring obstacles means that the path does not depend on any information from the online scene. All Reeds–Shepp curves can be sampled and enumerated offline, and the path length can be recorded in advance. When calling this function online, only indexing and interpolation are needed to return the function value. At the same time, the main purpose of this heuristic function is to prune the branches of the search tree that are incorrect when approaching the target value (Kim & Muminov, 2023), ensuring that the final termination pose can be accurately connected. Figure 6 shows the path generated using a heuristic function, which can be seen as continuous. The advantage is that compared to directly extending the number of nodes using Euclidean distance.

Unconstrained heuristics. When the current node is far from the endpoint, more attention should be paid to obstacle avoidance driving to prevent getting stuck in a dead end. The traditional A * heuristic function should be used for calculation. Figure 6 shows the discrete path generated under the heuristic function (similar to the results obtained by the traditional A * algorithm). The advantage is that introducing this heuristic function can discover all U-shaped obstacles and dead ends in 2D space. The design of heuristic functions in this way is beneficial for improving search speed, and the loss function used in the algorithm is the maximum value of the two heuristic functions. Figure 6 shows the mixed A * pseudocode.

```
function RounDSTATE(x)
x.Posx = max{m∈Z | m ≤ x.Posx}
x.Posy = max{m∈Z | m ≤ x.Posy}
x.Ange = max{m∈Z | m ≤ x.Ange}
return x
end function
```

Figure 6. Mixed A * Pseudocode: Algorithm 5 Hybrid A* Search




```
function EXISTS(°Csucc,C)
if {x ∈ C|roundState(x) = roundState(csuce)} ≠ ϕ then
return true
else
return false
end if
end function
0 = ϕ
C = ϕ
Pred(xs) ← null
O.push(xs)
while O ≠ ϕ, do
x ← O.popMin()
C.push(x)
if roundState(x) = roundState(x_g), then
return x
else
for u∈U(x), do
Tsuce ← f(x,u)
if-exists(xsucc,C), then
g ← g(x) + l(x,u)
if-exists(xsucc,O) or g < 9(xsucc), then
Pred(xsucc) ← x
g(3succ) → g
h(xsuce)-Heuristic(succ,g)
if-exists(xsucc,O), then
O.push(xsucc):88else
else
O.decreaseKey(xsuce)
end if
end if
```

```
end if
end for
end for
end while
```

Update and Maintenance of Boundary Areas

The boundary is defined as a known free voxel adjacent to an unknown voxel, which is grouped into clusters to guide navigation. The information extracted by traditional algorithms is too rough to make detailed decisions. In addition, boundaries are retrieved by processing the entire map, which is not scalable for large scenes and high planning frequencies. The author extracts richer information from boundaries, makes more refined planning, and develops an incremental method to detect locally updated maps within boundaries (Wei, 2021; Shui, 2019; Ping, 2018).

Information Structure of Frontiers

Fast UAV Exploration using Incremental (FUEL) organizes boundary regions into clusters one by one. Each boundary area information structure (FIS) contains five types of information:

1. The positions of all cells in the cluster.
2. The average position of all cells.
3. The axis-aligned bounding box (AABB) of the entire cluster, which is the smallest rectangular bounding box along the coordinate direction that can contain the entire cluster.
4. All viewpoints possible to observe the cluster.
5. Cost to other clusters.

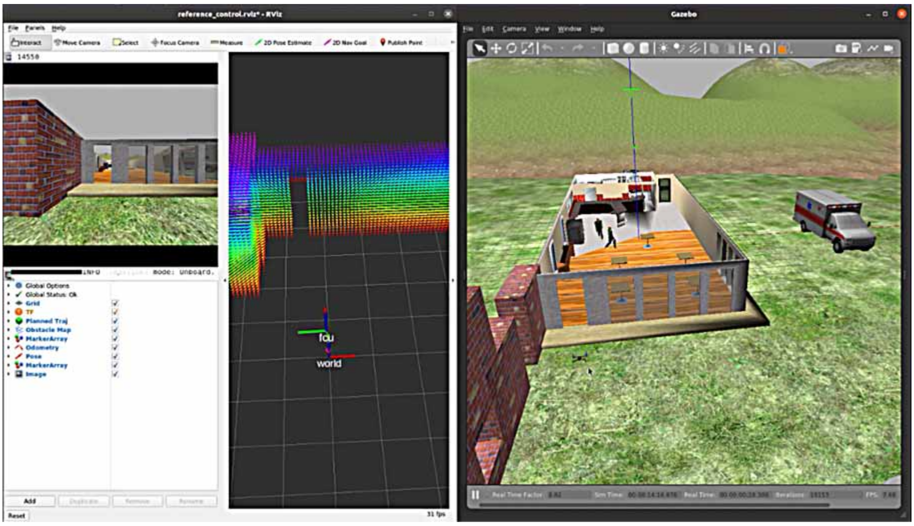
Frontiers Update Strategy

The boundary region update strategy in FUEL can be summarized as: generating the latest, removing duplicates, ignoring those that are too small, and segmenting those that are too large. The specific introduction is as follows:

1. The map updates with sensor information input.
2. Calculate the AABB (Bm) of the updated area and the AABB (Bi) of the previous cluster and record the clusters where Bi and Bm overlap (as shown in the upper left corner of Figure 7).

Figure 7. Update Strategy of Frontiers

Figure 8. TSP for Global Path Planning



3. Conduct a detailed inspection of clusters with overlapping elements and remove the overlapping cells (as shown in the upper right corner of Figure 7).
4. Search for frontiers again and re-cluster to generate new clusters.
5. After deleting smaller clusters, perform principal component analysis (PCA) on each cluster (as shown in Figure 7, bottom left).
6. If the maximum eigenvalue exceeds the threshold, it will be split in half along the main axis direction (as shown in the bottom right corner of Figure 7).

This completes an update to the frontier.

Layered Motion Planning

In the motion planning section, this project combines the needs of exploring search and rescue missions, uses traveling salesman problem (TSP) for global path planning, and selects a more optimal perspective combination within a limited range. As shown in Figure 8, starting from the green asterisk, FUEL conducts path planning in three levels (Zhao et al., 2023): global path planning (green line) → local viewpoint optimization (blue line) → running trajectory optimization (red line).

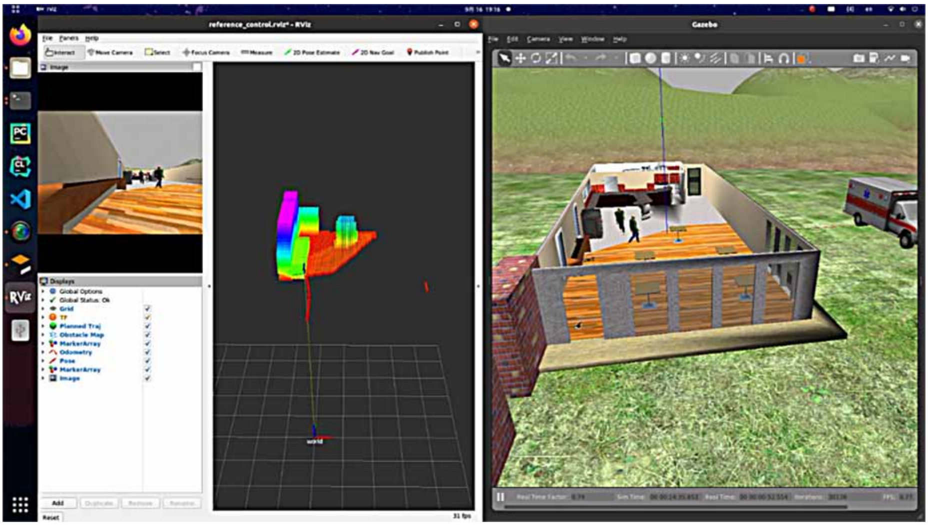
Local Viewpoint Optimization

In this process, FUEL is optimized within the current position distance R_ {rf}, as shown in Figure 9. The specific optimization method can be summarized in a sentence: select the viewpoints within this range and use the Dijkstra algorithm to get the lowest sum of path t_ {lb} of each distance. Information gain can be easily integrated into the optimization process, with both info-gain and cost. However, in the specific application, info-gain does not actually participate in this optimization. Instead, Dijkstra directly gives the connection line of each optimized point, and then each point takes the maximum angle of coverage.

Minimum Time B-Spline Trajectory

Given discrete viewpoints, continuous trajectories are necessary for smooth navigation. Our quadcopter trajectory planning (Fan et al., 2022) is based on a method of generating smooth, safe, and dynamically feasible B-spline trajectories (Yu et al., 2024). Next is to optimize all parameters of the B-spline

Figure 9. FUEL Local Viewpoint Optimization



to minimize the total trajectory time and enable the quadcopter to utilize its dynamic performance fully. The minimum-time B-spline trajectory is an optimization method used to generate a smooth trajectory for a robot or moving object to pass through a specified path in the shortest possible time during motion. This method combines B-spline curves and dynamic constraints for more efficient and natural motion planning.

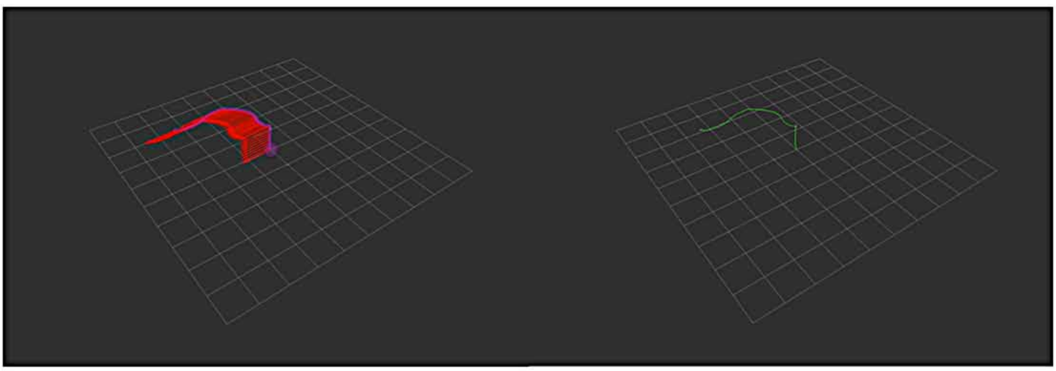
The key to generating the minimum-time B-spline trajectory is to determine the node vectors and control points of the curve and further optimize the motion's time allocation. The minimum-time B-spline trajectory generation method has great flexibility and adaptability for trajectory planning problems. It is widely used in fields such as robot navigation, autonomous driving, and aerospace. It can meet different motion requirements by adjusting parameters and constraints, such as shortest time, minimum energy consumption, and smoothness. Meanwhile, this method can also combine obstacle avoidance algorithms and path planning algorithms to achieve safe path planning and motion control in complex environments. In summary, the minimum time B-spline trajectory generation method comprehensively considers B-spline curves and dynamic constraints to achieve smooth trajectory planning for robots or moving bodies to complete motion in the shortest possible time on a given path. It is a flexible and efficient motion planning method, which is of great significance for application scenarios that require precise control of motion time. Explore the original scene and the completed scene as shown in Figures 10 and 11.

Implementation of Semantic YOLOv5 Algorithm

Before the you only look once (YOLO) algorithm was proposed, two-stage object detection algorithms represented by the region-based convolutional neural network (R-CNN) series (Zeng et al., 2024; Li et al., 2019; Huang et al., 2024) were widely used due to their advantages of high detection accuracy. However, these algorithms have slow detection speeds, and make it difficult to meet real-time requirements, limiting their application in embedded devices. YOLOv5 (Jain et al., 2022) is currently the latest model in the YOLO series of algorithms (Tian et al., 2022; Zhou et al., 2023; Chen, 2023). It is mainly composed of input, feature extraction network, feature fusion network, output, and other parts. Figures 12 and 13 show the overall network structure and submodule structure of YOLOv5 (Yu et al., 2018).

Figure 10. Original Scene to Be Detected

Figure 11. Explores The Completion Scenario



In the decision-making process of the UAV, the priority of the information is transformed according to different purposes; if the main task is fire detection, the output of the YOLOv5 fire detection algorithm will have a high priority. If the environment suddenly becomes complex or dangerous, the information provided by the binocular vision system may temporarily gain higher priority to ensure the safety of the UAV. At the same time, our UAV also performs dynamic adjustment, and the decision-making system of the UAV dynamically adjusts the priority of information processing according to real-time feedback. For example, if the fire source is confirmed and the situation is urgent, the system may temporarily ignore other non-critical information and concentrate its resources on fire location and response.

In practice, the integration process involves the UAV's CPU or flight controller receiving a real-time data stream from the binocular camera and the YOLOv5 algorithm. This data is fed into an integrated data processing module, which is responsible for fusing visual data and fire detection results and making decisions based on preset rules and real-time environmental information.

The data processing module will fuse the depth information provided by the binocular vision system and the fire detection results of the YOLOv5 algorithm to generate a comprehensive environment model and fire source location information. Based on this comprehensive information, the path planning algorithm will calculate an optimal flight path that can avoid obstacles and quickly reach the fire source location. According to the path planning results, the flight control algorithm adjusts the UAV's flight attitude and speed to ensure that it can reach the target position safely and accurately. Through this integration and priority determination mechanism, UAVs are able to detect and respond effectively to fire in complex indoor environments while ensuring their own safety and the successful completion of their missions.

In order to effectively solve the complexity and computational requirements of the algorithm in the UAV indoor fire detection system and ensure the real-time performance and feasibility of the system in the actual environment. In the image preprocessing stage, images with different resolutions are used for multi-scale analysis to filter out most of the non-fire source areas quickly. Then, high-resolution images are used for detailed analysis of the suspected fire source areas. This way, the dependence on high computational resources can be reduced while maintaining the detection accuracy. The parallel processing ability of GPU is used to accelerate computation-intensive tasks such as image processing and path planning.

On the on-board computing platform of UAVs, small but efficient GPUs can be integrated, or CPUs with parallel processing capabilities can be used. The resource allocation strategy is dynamically adjusted according to the urgency and importance of the task. For example, in the fire source detection task, CPU and memory resources can be preferentially allocated to the fire source detection algorithm. By monitoring system resource usage in real time, it ensures that critical tasks can obtain sufficient computing resources.

Compared with you only look once v3 (YOLOv3) and you only look once v4 (YOLOv4), YOLOv5 has made the following improvements: at the network input end, anchor boxes are pre-set in advance, which can adaptively calculate anchor boxes in different training sets during training. By continuously iterating and updating, the best anchor box value is found, effectively improving detection accuracy. In the feature extraction section (Nhi & Le,2022; Chu et al., 2022; Zheng et al., 2022), YOLOv5 uses the Focus operation, which can effectively reduce the floating-point operations (FLOPs) of the model to improve detection speed. The C3 module and the spatial pyramid pooling fast (SPPF) module (Wang et al., 2020) are respectively improvements to the corresponding feature extraction module in YOLOv4, which can improve training speed while reducing duplicate gradient information, making the YOLOv5 network have better learning ability; In the feature fusion section, YOLOv5 continues the multi-scale feature fusion method of feature pyramid network (FPN) + path

Figure 12. YOLOv5 Original Network Structure

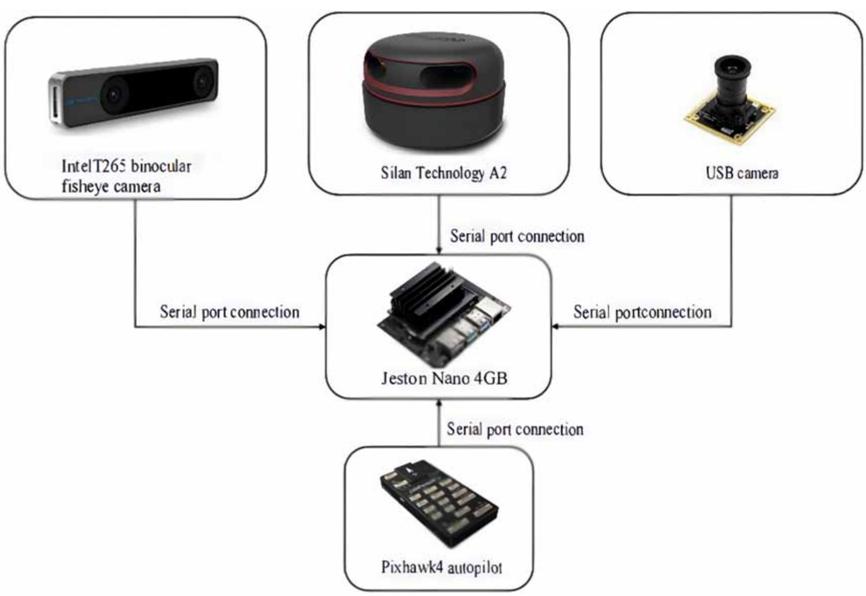
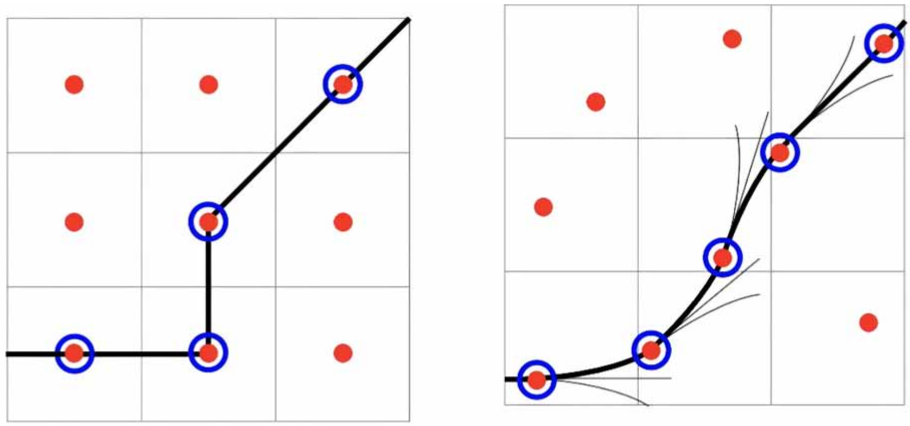


Figure 13. YOLOv5 Submodule Structure



aggregation network (PAN) in YOLOv4, with three detection layers set at different scales. Compared with YOLOv4, although the detection accuracy of YOLOv5 has slightly decreased, the training time and detection speed have been greatly improved, and the flexibility is stronger than YOLOv4. Moreover, the weight file of the model is smaller, which has a strong advantage in the fast deployment of the model and is more suitable for application in embedded devices. Therefore, this article chooses YOLOv5 as the basic architecture for implementing flame detection (Feng, 2022).

Introducing the Python Torch (PyTorch) Framework

The YOLO-TensorFlow (YOLO-TF) algorithm used in this paper is developed based on the python torch (PyTorch) framework, a machine learning-based neural network tensor optimization library specifically (Qian et al., 2022) for image and video domain research. It is an open-source Python-based machine-learning framework developed by Facebook. It provides a flexible way to build deep learning models using dynamic computation graphs for model construction and training and static computation graphs for model deployment. The core data structures in PyTorch are tensors, which are data classes similar to multidimensional arrays.

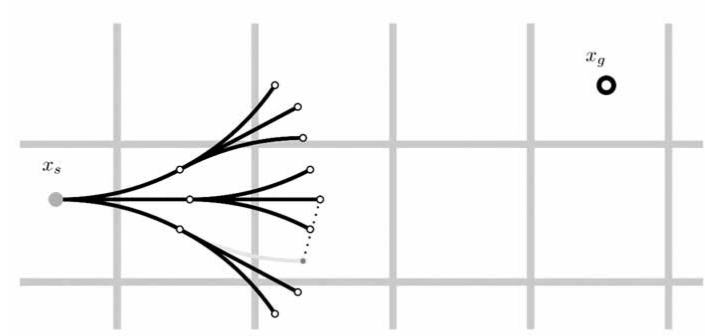
Type, which can be accelerated on GPU and CPU. PyTorch provides many common tensor operations, such as convolution (Sun et al., 2021), pooling, and linear layers, which can be combined to form more complex neural networks. PyTorch also provides automatic differentiation, which automatically calculates derivatives and is used in backpropagation. This makes PyTorch ideal for building gradient descent optimization algorithms and for training deep learning models. In summary, PyTorch is a powerful deep learning framework. It provides flexible model construction, efficient computing power, and convenient automatic differentiation functions, which can help researchers and engineers quickly build and train deep learning models. Therefore, this paper uses PyTorch framework to build YOLO-TF algorithm to realize the construction and analysis of forest fire detection model.

EXPERIMENTS AND RESULTS

Experimental Setup

The YOLOv5S(YOLOv5small) is generated after 10 training iterations with an input size of 640×640 on the fireworks data set. To ensure the effectiveness of the model in real-world indoor fire scenarios, the selection of the dataset is crucial. As much as possible, the dataset should cover a wide range of possible indoor fire situations, including different types of fire sources, various indoor objects, and

Figure 14. Training Results



environmental factors that may affect fire detection, such as smoke and light changes. In addition, the dataset should also contain some extreme cases, such as nighttime fires or large-scale fires, to test the model's performance in these cases.

Analysis of Experimental Results

Assuming that the system incorrectly identifies two out of 80 non-fire sources as fire sources, the results indicate a precision of about 88%, a recall of 90%, and a false positive rate of 2.5%. The average time to process each image and decide is 200 ms. The system's performance also changes under different lighting conditions. Under good lighting conditions, the detection accuracy is 95%. Under moderate light conditions, it drops to 85%. In low light conditions, it drops to 75%. This indicates that the system's robustness under different lighting conditions is decreased, but it remains at an acceptable level. In order to compare the performance of the system with existing methods, we can choose several recognized fire detection algorithms for comparison, such as traditional color or texture analysis methods or other object detection algorithms based on deep learning (such as Single Shot MultiBox Detector (SSD)and faster R-CNN. The system has advantages in detection accuracy. However, the response time and resource consumption of the system are also relatively high.Figure 14 shows the result graph after training ten times.

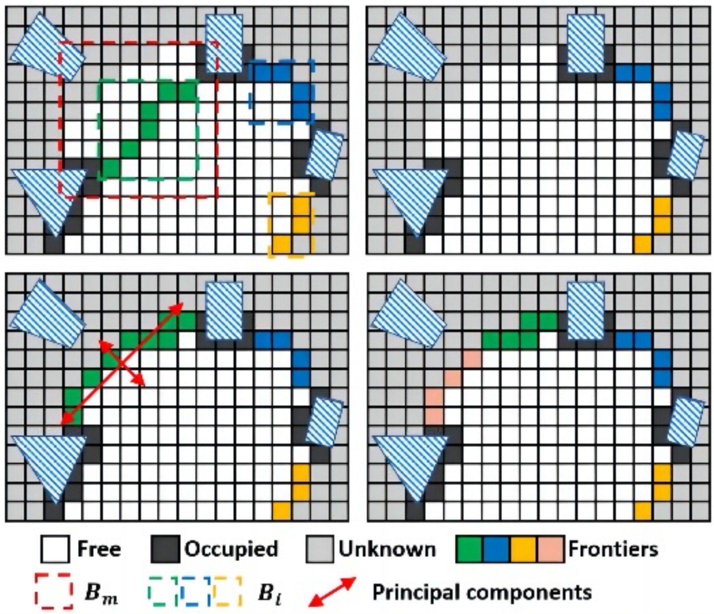
Although the model only underwent a few training epochs, the fire detection results were still quite good. However, it is observed that trained models tend to predict the red emergency lights on the top of police cars as fires. This may be due to the training dataset containing (Behera et al., 2023) only a few hundred negative samples. We can solve this problem by adding images of unmarked flame objects as negative samples and further improve the model's performance. The author who created YOLOv5 suggests using a background image of approximately 0–10% to help reduce false positives (Zhu et al., 2022). The results of fire detection are shown in Figure 15.

USE CASES

Simulation Scenario

Configure and launch the Gazebo simulation environment and robot operating system visualization (RViz) tool on the PX4 UAV platform. Then, the ROS package is used to call the UAV sensor data (such as light detection and ranging (LiDAR) and camera) for environmental perception. Next, the algorithm presented in this article will be used for real-time map construction, and the created map data will be published to RViz for visualization. Meanwhile, LiDAR data was used for obstacle detection and distance estimation. Using navigation algorithms and local path planners, identify and utilize the constructed map and perception information to plan safe paths while avoiding collisions.

Figure 15. Fire Detection Results



LiDAR is a commonly used sensor for detecting objects and obstacles in the surrounding environment. It determines the distance and position of an object by firing a laser beam and measuring the time it takes to reflect back. This makes lidar a very effective method to detect and avoid obstacles.

LiDAR can provide reliable data to help robots or autonomous vehicles avoid obstacles in a variety of challenging environments. For example, in complex urban environments, buildings, vehicles, and pedestrians may suddenly appear on the road, and lidar can detect these obstacles in time and take corresponding actions. In severe weather conditions, such as rain, snow, and fog, LiDAR can also maintain good performance because it is not affected by light. However, in some specific environments, LiDAR may encounter some challenges. For example, insufficient illumination may cause the LiDAR to identify the shape and contour of the object inaccurately. In multi-target detection, if the objects are too close to each other, the lidar may be confused, resulting in inaccurate detection. In addition, LiDAR may also produce errors on certain types of obstacles, such as glass and reflective objects.

To overcome these challenges, we also combine other sensors (such as cameras and ultrasonic sensors) to enhance the detection performance of LiDAR. At the same time, the applicability and accuracy of lidar in various environments can be improved through optimization of algorithms and software. Overall, LiDAR is a powerful tool capable of reliably detecting obstacles in a variety of challenging environments. The detection performance of lidar can be further improved by integrating multiple sensors and optimization algorithms, leading to a safer and more efficient automated system.

Field Testing

Finally, the generated waypoints are sent to the UAV for autonomous navigation through control algorithms. After the entire process is completed, the real-time map constructed by the UAV and the effect of obstacle avoidance navigation can be observed in RViz, as shown in Figures 16 and 17. The actual operation results of the UAV (He et al., 2024) are shown in Figures 18 and 19. By comparing

Figure 16. Simulation Results

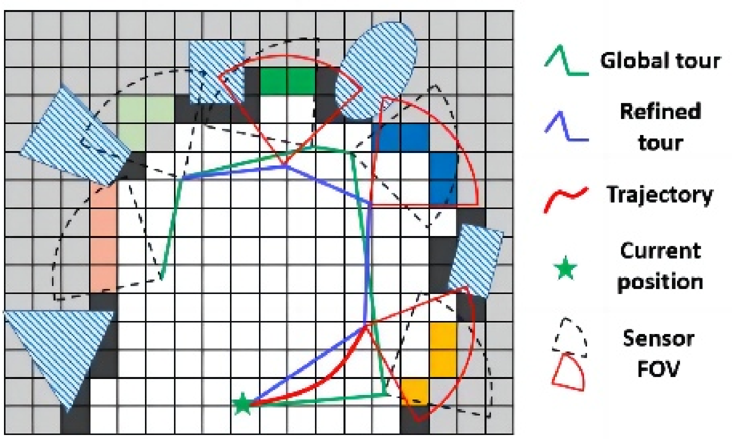
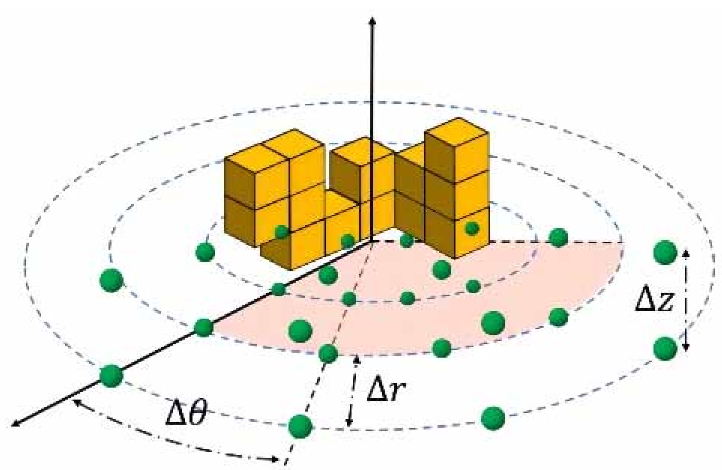


Figure 17. Simulation Results



the simulated environment with the actual flight effect, it can be seen that our UAV has sufficient accuracy, attention, reliability, and spatial awareness.

CONCLUSION

Summary of Research

Fuel-Planner introduced the *high-speed jump*, which can skip unwanted nodes during the search process, thus improving the search speed. We chose to use the hybrid A * algorithm for the actual aircraft kinematics. We introduced the hybrid A * algorithm, introduced the heading angle, and conducted the search in three dimensions consistent with the aircraft kinematic model. The YOLOv5 recognition algorithm is used in vision for target identification and detection. With the autonomous identification of the UAV as the background, the YOLOv5 uses the Focus operation.

Figure 18. Autonomous Flight Attitude and Trajectory of UAV

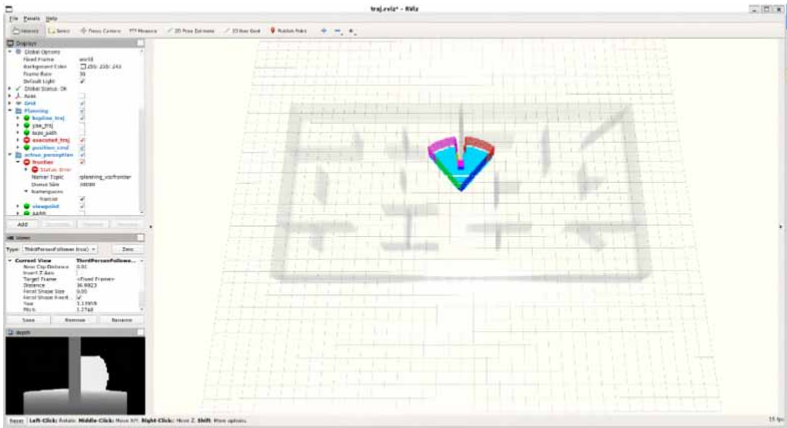
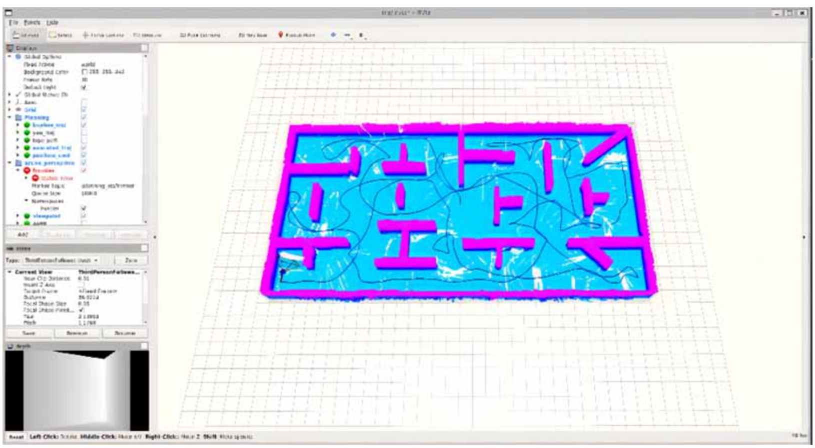


Figure 19. Autonomous Flight Attitude and Trajectory of UAV



Practicability and Prospect

In the future, we can realize multi-UAV collaborative operation, and through the joint work of multiple UAVs, timely monitoring and response to large-scale fires can be realized, and the coverage and operation efficiency of the system can be improved. The UAV and ground rescue equipment were integrated to realize that the UAV could quickly guide the rescue team to the disaster scene after fire detection, improving rescue efficiency and disaster disposal speed. The system's adaptability to harsh environmental conditions, such as high temperatures and smoke, is strengthened to ensure the system's stable operation and accurate monitoring in various environments. We need to continue improving these directions, such as improving our UAV's performance and solving the current limitations.

Future Directions

Future fire detection technology will be more intelligent and automated, and more accurate and rapid-fire detection can be carried out by introducing artificial intelligence and machine learning technology. At the same time, the automatic fire alarm system can respond in time and take corresponding measures to reduce the loss of personnel and property from the fire. Future fire detection technologies will also be more networked and interconnected, enabling data sharing and collaboration with other devices and systems. More intelligent fire prevention and disposal can be achieved by integrating fire detection systems with building management and fire protection systems.

We can study fire detection algorithms in depth to improve the accuracy and sensitivity of detection. Techniques such as deep learning and neural networks can be explored to improve the intelligence level of fire detection systems. Fire detection technology can be applied to more fields, such as subways, high-speed rails, and automobiles, through different sensors and algorithms, timely monitoring, and early warning of fire in various scenarios can be realized.

CONFLICTS OF INTEREST

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

FUNDING STATEMENT

This research was supported by 2022 Liaoning Province General Higher Education Undergraduate Teaching Reform Research Project --- Approval General Project “Research and Practice on Customized Teaching Mode of College English Curriculum in Engineering Colleges Based on BOPPPS Teaching Model in the Context of New Liberal Arts”.

CORRESPONDING AUTHOR

Correspondence should be addressed to Xiaoying Song (songxy_sau_sy@163.com)

PROCESS DATES

Received: 3/5/2024, Revision: 3/26/2024, Accepted: 3/26/2024

REFERENCES

Behera, T. K., Bakshi, S., Sa, P. K., Nappi, M., Castiglione, A., Vijayakumar, P., & Gupta, B. B. (2023). The NITRDrone dataset to address the challenges for road extraction from aerial images. *Journal of Signal Processing Systems for Signal, Image, and Video Technology*, *95*(2–3), 197–209. doi:10.1007/s11265-022-01777-0

Chen, L. M., Cao, Y. Y., Huang, M., & Xie, X. G. (2023). Flame detection method based on the improved YOLOv5. *Computer Engineering*, *49*(8), 291–309.

Chen, X. (2022). Application of intelligent UAV in Fire monitoring and rescue. *[Engineering Technology Edition*]. *Journal of Qingdao University*, *37*(3), 83–86.

Chen, X. S., Wu, C. P., Dang, P., Liang, J., Liu, S. J., & Wu, T. (2023). Shrimp meat shell identification method based on the improved lightweight YOLOv4 model. *Journal of Agricultural Engineering*, *39*(9), 278–286.

Chu, J., Zhao, X., Song, D., Li, W., Zhang, S., Li, X., & Liu, A. A. (2022). Improved semantic representation learning by multiple clustering for image-based 3D model retrieval. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–20. doi:10.4018/IJSWIS.297033

Dwivedi, R. K. (2022). Density-based machine learning scheme for outlier detection in smart forest fire monitoring sensor cloud. *International Journal of Cloud Applications and Computing*, *12*(1), 1–16. doi:10.4018/IJCAC.305218

Fan, Q., Cai, Y. Q., & Jia, S. N. (2022). Track and planning and simulation of industrial robots with multiple methods. *[Natural Science Edition*]. *Journal of North China University of Science and Technology*, *44*(4), 60–67.

Fang, Y. X., Lin, S. X., Shi, L. F., & Ren, B. (2024). Indoor navigation robot research based on ROS and visual SLAM. *Journal of Dongguan Institute of Technology*, *31*(1), 59–64.

Fatemidokht, H., Rafsanjani, M. K., Gupta, B. B., & Hsu, C. (2021). Efficient and secure routing protocol based on artificial intelligence algorithms with UAV-assisted for vehicular ad hoc networks in intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, *22*(7), 4757–4769. doi:10.1109/TITS.2020.3041746

Feng, Y. Y. (2022). *Methods and research on dynamic path planning method of multi-target points of garbage removal robot[Master* dissertation]. University of Electronic Science and Technology.

Galiautdinov, R. (2021). Methods of information processing of relative motion in the flying groups of UAV. [IJSSCI]. *International Journal of Software Science and Computational Intelligence*, *13*(1), 90–108. doi:10.4018/IJSSCI.2021010106

He, C. L., Zhang, Q., & Wu, Y. H. (2024). Incremental dynamic inverse flight attitude control based on fixed time convergence. *Flight Mechanics*, 1–8.

Hu, Z., & Xu, B. (2023). Dynamic path planning combining the A~ * algorithm with the artificial potential field method. *Combined Machine Tool and Automatic Processing Technology*, *7*, 46–56.

Huang, J. H., Wu, S. L., & Xu, J. W. (2024). Research and application of cone-barrel recognition technology based on YOLOv5. *Journal of Wuhan Textile University*, *37*(1), 89–93.

Jain, A. K., Yadav, A., Kumar, M., García-Peñalvo, F. J., Chui, K. T., & Santaniello, D. (2022). A cloud-based model for driver drowsiness detection and prediction based on facial expressions and activities. *International Journal of Cloud Applications and Computing*, *12*(1), 1–17. doi:10.4018/IJCAC.312565

Kim, S., & Muminov, A. (2023). Forest fire smoke detection based on deep learning approaches and unmanned aerial vehicle images. *Sensors (Basel)*, *23*(12), 5702. doi:10.3390/s23125702 PMID:37420867

Li, D., Deng, L., Gupta, B. B., Wang, H., & Choi, C. (2019). A novel CNN based security guaranteed image watermarking generation scenario for smart city applications. *Information Sciences*, *479*, 432–447. doi:10.1016/j.ins.2018.02.060

Li, H. S. (2023). The application of UAV technology in modern fire rescue combat. *Proceedings of the 2023 Symposium on Fire Fighting and Emergency Rescue Technology-Fire Fighting Technology Equipment and Equipment Management*, 30–32.

Li, L. L., Zhang, Z., Yang, Y. F., Liang, L., Wang, X., & Han, Y. Q. (2023). Range method based on dot-line vision / inertial SLAM and object detection. *Chinese Journal of Inertial Technology*, *31*(3), 237–244.

Li, Y. F., Wang, S. T., Xiong, T. F., Xie, Y. L., & Li, H. (2023). Path planning of a mobile robot that takes into account start and stop characteristics and corner time consumption. *Journal of Xi'an Jiaotong University*, *57*(2), 192–202.

Liang, J. X., Zhao, J. F., Zhou, Y. L., & Shi, B. J. (2024). Fire image recognition method based on an improved YOLOv5s network model. *Sensors and Microsystems*, *43*(1), 157–161.

Lin, Z. J., Liu, K., & Lin, Q. X. (2023). Review of studies on pathway planning algorithms. *Modern Information Technology*, *7*(4), 75–80.

Liu, J. W., He, S. T., & Ma, S. Z. (2021). Design of an intelligent medical transport robot based on the ROS. *Electronic Design Engineering*, *32*(3), 161–165.

Nhi, N. T. U., & Le, T. M.Thanh The Van. (2022). A model of semantic-based image retrieval using C-tree and neighbor graph. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–23. doi:10.4018/IJSWIS.295551

Ping, H. D. (2018). *Study on continuous object boundary region detection in wireless sensor networks*[Master dissertation]. China University of Geosciences (Beijing).

Qian, W., Li, H., & Mu, H. (2022). Circular LBP prior-based enhanced GAN for image style transfer. *International Journal on Semantic Web and Information Systems*, *18*(2), 1–15. doi:10.4018/IJSWIS.315601

Shui, Q. (2019). *Research on the autonomous exploration and mapping technology of indoor mobile robot*[Master dissertation]. South China Agricultural University.

Sun, D. Y., Liu, H., Zhang, J. H., Sun, L. Y., Wang, Q., & Zhao, J. M. (2021). Classification of marine images based on deep convolutional neural network. *Ocean and Lake*, *52*(5), 1160–1169.

Sun, Y. T., Wang, R. J., & Jiang, D. S. (2024). Dynamic path planning of a surface watercraft integrating A * with the DWA algorithm. *Journal of Instrumentation : An IOP and SISSA Journal*, •••, 1–9. http://kns.cnki.net/kcms/detail/11.2179.th.20240204.1354.023.html

Tian, M. J., Li, X. L., Kong, S. H., Wu, L. C., & Yu, J. Z. (2022). Visual detection algorithm based on YOLOv4 (English). *Frontiers of Information Technology and Electronic Engineering*, *23*(8), 1217–1229. doi:10.1631/FITEE.2100473

Wang, H., Li, Z., Li, Y., Gupta, B. B., & Choi, C. (2020). Visual saliency guided complex image retrieval. *Pattern Recognition Letters*, *130*, 64–72. doi:10.1016/j.patrec.2018.08.010

Wang, H. F. (2017). Rationalized application of multi-rotor UAV in fire fighting. *Fire Protection, 62*(6).

Wang, J. N. (2016). *Design and implementation of the background data processing module in the mobile wireless network test and evaluation system*[Master dissertation]. Beijing University of Posts and Telecommunications.

Wang, Z. T., Luo, L. P., & Liao, Y. K. (2024). Improved A * algorithm fusion improved DWA algorithm. *Computer Engineering*, 1–19. 10.19678/j.issn.1000-3428.00.00EC0068483

Wei, Q. K. (2021). *Research on the construction and updating of the digital map of unstructured roads in closed areas*[Master dissertation]. Hunan University.

Xie, Y., Li, P., Nedjah, N., Gupta, B. B., Taniar, D., & Zhang, J. (2023). Privacy protection framework for face recognition in edge-based Internet of Things. *Cluster Computing*, *26*(5), 3017–3035. doi:10.1007/s10586-022-03808-8 PMID:36415683

Yang, C. F. (2019). *Research on the robotic multi-target path planning method based on ROS*[Master dissertation]. Tianjin University.

Yang, L. (2024). Surface reconstruction algorithm for 3 D scenes based on binocular vision. *Modern Electronic Technology*, *47*(4), 71–75.

Yu, C., Li, J., Li, X., Ren, X., & Gupta, B. B. (2018). Four-image encryption scheme based on quaternion Fresnel transform, chaos and computer generated hologram. *Multimedia Tools and Applications*, *77*(4), 4585–4608. doi:10.1007/s11042-017-4637-6

Yu, X. Y., Liu, Y. Z., & Ou, L. L. (2024). A continuous path planning method for mine-loading vehicles based on curvature and B spline. *High-tech Communications*, *34*(1), 72–82.

Zeng, J. W., Geng, L., Lu, C. Y., & Ren, L. H. (2024). A virtual track train path identification method based on binocular stereo vision. *Locomotive Electric Transmission*, *1*, 145–151.

Zhang, J. N., Liu, X. L., Zhi, Y. C., Zhang, X. P., & Zhang, J. H. (2024). Design of the artifact identification and positioning system based on machine vision. *Innovation in Science and Technology*, *4*, 64–67.

Zhao, J. T., Li, L., Xue, Z. J., & Zhang, Z. H. (2023). Hierarchical motion planning method for in-parking cruise based on hybrid A star. *Jixie Gongcheng Xuebao*, *59*(24), 290–298. doi:10.3901/JME.2023.24.290

Zheng, Z., Zhou, J., Gan, J., Luo, S., & Gao, W. (2022). Fine-grained image classification based on cross-attention network. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–12. doi:10.4018/IJSWIS.315747

Zhou, H. P., Jin, S. X., Zhou, L., Guo, Z. L., Sun, M. M., & Shi, M. H. (2023). Classification and identification of field camellia fruit based on transfer learning and YOLOv8n. *Journal of Agricultural Engineering*, *39*(20), 159–166.

Zhu, C., Jiang, W., Zhou, W., & Xiao, H. (2022). Similarity retrieval based on image background analysis. *International Journal of Software Science and Computational Intelligence*, *14*(1), 1–14. doi:10.4018/IJSSCI.309426

Yihang Chen, Student, undergraduate. Her research interests include Flight Vehicle Propulsion Engineering.

Xiaoying Song, associate professor, Master, Graduated from Liaoning University in 2001. Worked in Shenyang Aerospace University and as a visiting scholar in CUNY. Her research interests include Linguistics and Applied Linguistics.