Patrol Agent: An Autonomous UAV Framework for Urban Patrol Using on board Vision Language Model and on cloud Large Language Model

Zihao Yuan
School of Aeronautics and Astronautics
Zhejiang University
Hangzhou, China
22324063@zju.edu.cn

Fangfang Xie
School of Aeronautics and Astronautics
Zhejiang University
Hangzhou, China
fangfang_xie@zju.edu.cn

Tingwei Ji*
School of Aeronautics and Astronautics
Zhejiang University
Hangzhou, China
zjjtw@zju.edu.cn
*Corresponding author

2024 6th International Conference on Robotics and Computer Vision (ICRCV) | 979-8-3315-2742-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICRCV62709.2024.10758606

***Abstract*—Unmanned Aerial Vehicles (UAVs) used for urban patrols typically require human control or supervision. To enhance the automation of UAVs in this context, we propose the Patrol Agent, which is able to patrol, identify and track a target in a fixed area autonomously without any human intervention. The Patrol Agent employs Vision Language Model (VLM) for accurate visual information, object detection model for rough detection about the target, and Large Language Model (LLM) deployed on cloud for analysis and action-deciding. During patrols, the agent uses a lightweight VLM to generate captions of the scenes it observes. These captions are then sent to the LLM on cloud for further analysis which provides responses regarding the danger level of the scene, appropriate actions to take, and the detailed reasons behind these actions. When the agent identifies and tracks a target, it activates the VLM only when the object detection model detects an object corresponding to the target. This approach conserves computing resources and enhances onboard operational speed. The proposed agent can identify and track targets without requiring fine-tuning data or human intervention. It outperforms Visual Question Answering (VQA) models in patrol and uses fewer computing resources compared to agents that solely rely on VLM for tracking.**

***Keywords*—UAVs patrols, Large-Language-Model, Vision-Language-Model, detection**

I. Introduction

For the past fifty years, Unmanned Aerial Vehicles (UAVs) have been extensively utilized across various fields, including surveillance, monitoring, search and rescue, healthcare, agriculture and more [1]–[6]. These UAVs assist humans in gathering environmental information, analyzing situations, and making informed decisions, especially in scenarios where it is uncertain or unsafe for humans to approach directly.

In an attractive yet challenging endeavor to enhance UAVs capabilities in motion planning, decision making, and precise perception, many researchers have combined UAVs with AI techniques such as Deep Reinforcement Learning (DRL), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM), among others [7]–[10]. However, these work often exhibit a certain degree of failure when encountering new environments or difficulties in training the models. A new trend involving Large Language Models (LLM) [11] and Vision-Language Models (VLM) [12] has emerged, offering a novel approach to building agents [13], which helps agents adapt to new environments. The General Robot Intelligence Development (GRID) platform facilitates robots in learning, composing, and adapting skills according to their physical capabilities, environmental constraints, and goals [14]. GRID can orchestrate a set of foundation models and other tools via APIs to address complex robotics tasks using both simulation and real-world data. With different instructions, the agent can perform various tasks. Some researchers have developed an AeroAgent [15], designed to think and act more like a human, with components corresponding to human organs such as the Observer, Mover, Cerebrum, and Cerebellum. Furthermore, this AeroAgent features an Automatic Plan Generator module, enabling it to plan autonomously without specific instructions. This allows the agent to search for and discover tasks independently during its operation.

However, existing works have the following problems:

(1) The use of large models can be resource-intensive. For example, GRID utilizes approximately ten models, and AeroAgent employs GPT-4V, which necessitates substantial computing resources and associated costs [16].

(2) Urban road scenes often receive less attention compared to other environments. Many studies primarily focus on larger scenes such as forests, maritime platforms, and similar settings.

(3) GPT-4, a closed-source large language model (LLM) and vision language model (VLM), can only be accessed through the OpenAI [17].

Our research proposes Patrol Agent, a system specifically designed for urban road scenes without requiring fine-tuning for the LLM and VLM. We utilize Florence2 [18], a lightweight VLM for caption generation and accurate object tracking, and You Only Look Once (YOLO)-v8 [19] for object detection. Both Florence2 and YOLO are deployed directly on the edge. We use the open-source Qwen2-72B [20] LLM on the cloud to analyze more complex captions and provide instructions to the agent for subsequent actions. The agent is running in a simulation environment for various tasks without human intervention; once initiated, it autonomously searches for targets or patrols the road.

The remainder of the paper is structured as follows: Section 2 details the chosen simulation, model, and proposed method. Section 3 describes the actions of Patrol Agent in the simulated road environment for patrolling, indetifying and tracking target, and analyzing how the method works. Section 4 concludes the paper.

II. Materials and methods

A. Use UE4 and AirSim to build environment

AirSim is an open-source platform designed to bridge the gap between simulation and reality, thereby aiding the development of autonomous vehicles [21], [22]. Running primarily on UE4, it can realistically simulate the physical world. Additionally, we utilize the model from [23] to enrich the scene. We built the map in AirSim, as shown in Fig 1.

For Patrol Mode, we created three different scenarios for the agent to respond to. For Track Mode, we placed a single tracking target on the map, allowing it to walk along the road to test whether the agent can detect and follow it.

With AirSim, we can primarily focus on the image returned from the agent to test our method. However, it also supports Hardware-in-the-Loop (HIL) simulations for more realistic UAV testing. For this research, we only used position and velocity control policies to simplify the model.

B. Choose for reasoning and vision part

With the extremely fast progress in Transformer-based language models [24], there are numerous options to choose from. Considering both open-source availability and performance, we selected the Large Language Model (LLM) Qwen2-72B for our agent, which we temporarily acquired via APIs. We used Florence-2, a lightweight yet powerful Vision-Language Model (VLM) , to assist with captioning and accurately detecting the target. Additionally, we employed the object detection model You Only Look Once (YOLO)-v8 to help conserve computing resources on the edge.

C. Proposed Method

When the agent operates, its functionality can be divided into three main components: the Agent Body, the Model, and the Output.

Agent Body: This component primarily consists of the hardware, flight control systems, and other essential elements that ensure the agent can execute actions based on the output and transmit perceived information to the model for further processing.

Model: This component includes the models used in building the agent and their deployment configurations. Florence2 and YOLO-v8 are deployed on board, while Qwen2-72B is deployed on the cloud. This setup allows the agent to use minimal computing resources on board, ensuring high operating speed while maintaining excellent performance.

Output: This component consists of the outputs generated by the model. For example, given an image and its ground truth (e.g., ”a group of people walking”), the agent processes the image using the model and generates a caption such as ”... a group of people walking on the ground.” With specific instructions (e.g., ”man in white clothes”), the agent can identify and box the target in the image. Additionally, the agent can analyze the situation depicted in the image to some extent, determine whether it constitutes an emergency, and instruct the agent body on the appropriate actions to take.

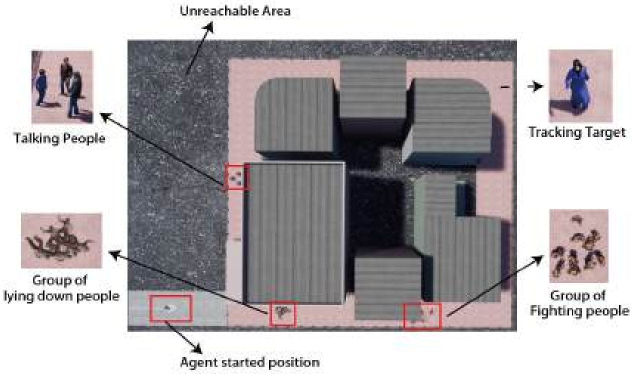


Fig. 1. The map built in UE4. For Patrol Mode, there are talking people, Group of Fighting people and Group of lying down people staying on the map. For Track Mode, there are a tracking target walk along the road.

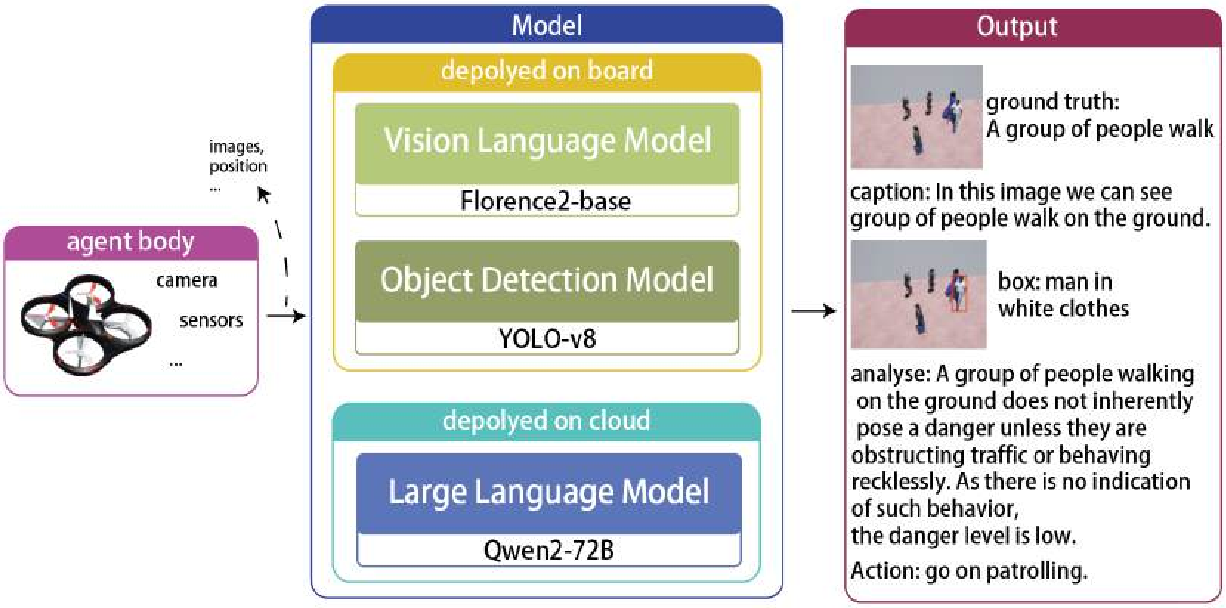
III. Patrol Agent simulates in AirSim

A. Agent patrols

When the agent is patrolling, it continuously processes images using a VLM. During patrols, the VLM provides different descriptions for two images. When it sees a scene with only a white arrow, it finds it difficult to grasp the context and requests the LLM to perform a deeper analysis. Due to the influence of light, if the scene shows the number 88 without other descriptions, it implies that the image only contains the shadow of a UAV. To save computing resources and improve operating speed, the agent will not ask the LLM to perform further analysis on this description. This process is as depicted in the Fig 3. In addition to obtaining desired responses from the Large Language Model (LLM), we use a prompt before making a request. The prompt involves describing the agent’s role, skills, restrictions, and examples of scenarios that will not occur in our simulation.

There are three main scenes in the simulation. Scene one contains a group of people standing and talking, scene two contains a group of people engaged in fierce fighting, and scene three contains a group of people lying on the ground.

In scene one, three people are standing and talking. This is a scene with no risk, so the agent should continue patrolling when it encounters this scene. Most of the time, the agent can distinguish that the view is safe. However, the captions generated by the VLM are not always identical; there are two main captions it generates (see Generation 1 and 2 in Fig 4). We found that although the VLM sometimes recognizes



ground truth:
A group of people walk

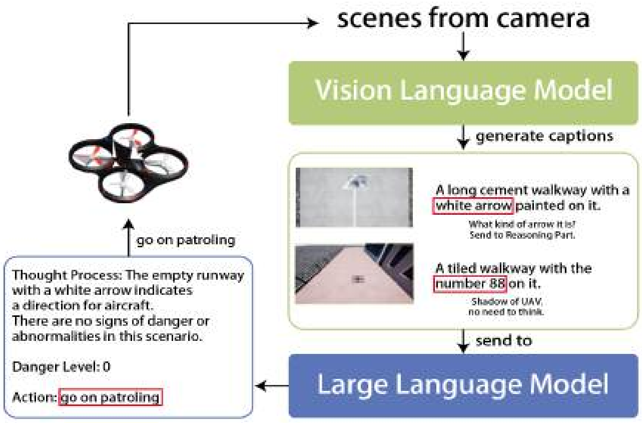
caption: In this image we can see group of people walk on the ground.

box: man in white clothes

analyse: A group of people walking on the ground does not inherently pose a danger unless they are obstructing traffic or behaving recklessly. As there is no indication of such behavior, the danger level is low.

Action: go on patrolling.

Fig. 2. Architecture of proposed method.



A long cement walkway with a white arrow painted on it.

What kind of arrow it is?
Send to Reasoning Part.

A tiled walkway with the number 88 on it.

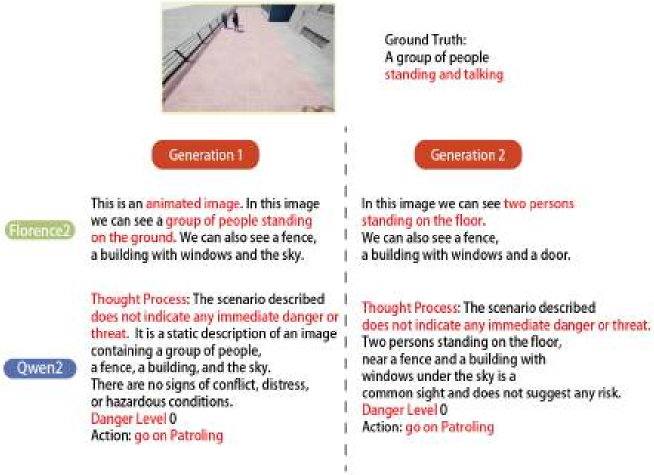
Shadow of UAV,
no need to think.

Thought Process: The empty runway with a white arrow indicates a direction for aircraft.
There are no signs of danger or abnormalities in this scenario.

Danger Level: 0

Action: go on patroling

Fig. 3. The process of agent patrolling.



Ground Truth:
A group of people
standing and talking

This is an animated image. In this image we can see a group of people standing on the ground. We can also see a fence, a building with windows and the sky.

In this image we can see two persons standing on the floor.
We can also see a fence,
a building with windows and a door.

Thought Process: The scenario described does not indicate any immediate danger or threat. It is a static description of an image containing a group of people,
a fence, a building, and the sky.
There are no signs of conflict, distress,
or hazardous conditions.
Danger Level 0
Action: go on Patroling

Thought Process: The scenario described does not indicate any immediate danger or threat.
Two persons standing on the floor,
near a fence and a building with
windows under the sky is a
common sight and does not suggest any risk.
Danger Level 0
Action: go on Patroling

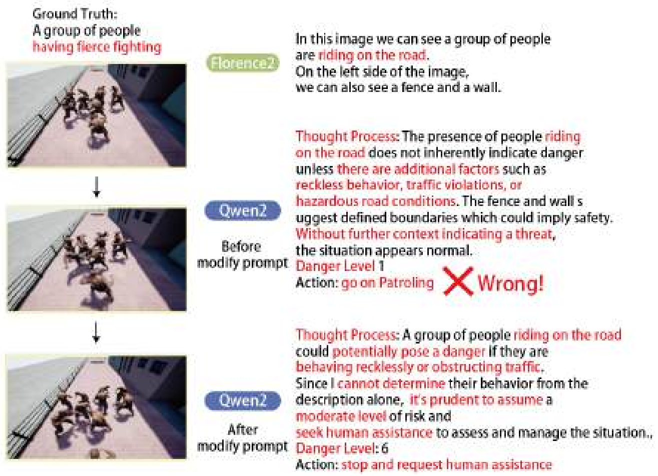
Fig. 4. Camera view of Agent for Scene 1. Three people are talking.

the view as a simulation, the Large Language Model (LLM) still understands what should be done, and the agent will not simply stop.

In scene two, a group of people is engaged in a fierce battle. This scene is so dangerous that it requires human intervention. VLM perceives it as a group of people riding on the road, which does not seem problematic. LLM senses the red flags behind the scene but still assigns a low danger level Before modifying the prompt (see Fig 5). Once we modify the prompt to include descriptions of reckless behavior and traffic violations, LLM re-generates the answer (see Fig 5). After the modification, the agent can now distinguish this scene and return desired action, and this process is easy to implement.

In scene three, a group of people is lying down on the road. This scene is so urgent that it requires human assistance. Although VLM does not provide detailed information about the scene, LLM compensates well by considering this situation as an accident or illness. The agent should stop and monitor them, waiting for human intervention (see Fig 6).

We also tried patrolling with the Visual Question Answering (VQA) model Qwen-VL-Max [25], using the same prompt that LLM used in our Patrol Agent. In Table I, we compare the two methods. The correct rate for captions represents the ratio of correct captions among all returned captions during patrol; the correct rate for actions represents the ratio of correct actions among all returned actions during patrol. By using the VQA, we found that it sometimes provides answers like ”Cannot Determine,” which never appear in the prompt. Moreover, it pays too much attention to the examples in the prompt



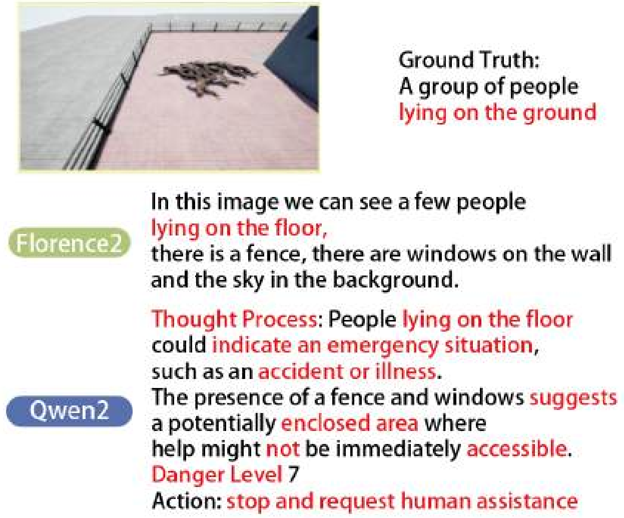
Ground Truth:
A group of people
having fierce fighting

In this image we can see a group of people
are riding on the road.
On the left side of the image,
we can also see a fence and a wall.

Thought Process: The presence of people riding
on the road does not inherently indicate danger
unless there are additional factors such as
reckless behavior, traffic violations, or
hazardous road conditions. The fence and wall s
uggest defined boundaries which could imply safety.
Without further context indicating a threat,
the situation appears normal.
Danger Level 1
Action: go on Patroling

Thought Process: A group of people riding on the road
could potentially pose a danger if they are
behaving recklessly or obstructing traffic.
Since I cannot determine their behavior from the
description alone, it's prudent to assume a
moderate level of risk and
seek human assistance to assess and manage the situation.,
Danger Level: 6
Action: stop and request human assistance

Fig. 5. Camera view of Agent for Scene 2. A group of people are having fierce fight.



Ground Truth:
A group of people
lying on the ground

In this image we can see a few people
lying on the floor,
there is a fence, there are windows on the wall
and the sky in the background.

Thought Process: People lying on the floor
could indicate an emergency situation,
such as an accident or illness.
The presence of a fence and windows suggests
a potentially enclosed area where
help might not be immediately accessible.
Danger Level 7
Action: stop and request human assistance

Fig. 6. Camera view of Agent for Scene 3. A group of people are lying down on the road.

and repeatedly gives the same answer shown in the prompt. This severe LLM hallucination causes the agent to repeatedly believe it is in danger [26], [27], constantly switching between stopping and moving forward, greatly slowing down its speed.

B. Agent identifies and tracks target

When the agent is identifying and tracking a target, it can also use LLM for scene reasoning. However, to demonstrate the effectiveness of a lightweight VLM deployed on board, we use only the VLM and an object detection model for this task (see Fig 7). The agent needs to identify and track a woman in a blue dress on the map; this is the only instructions the agent receives. We cannot use object detection model for tracking in this task because object detection model does not inherently understand the concept of ”woman in a blue dress” without fine-tuning on specific data for that category. Additionally, when a group of people appears, object detection model can only determine the number of people present but cannot accurately box the target (see Fig 8), and it may even produce incorrect detection results. However, the VLM infers much slower than the object detection model, so we do not want it to run all the time. The object detection model is very helpful in this regard. Before the agent encounters the target, there are many objects along the road, such as walls, windows, bricks, and so on. Although the object detection model cannot identify the woman in a blue dress, it is proficient at finding the class ”People”, and it is faster. Even if it makes mistakes occasionally, it can correct itself in the next moment (see Fig 9).

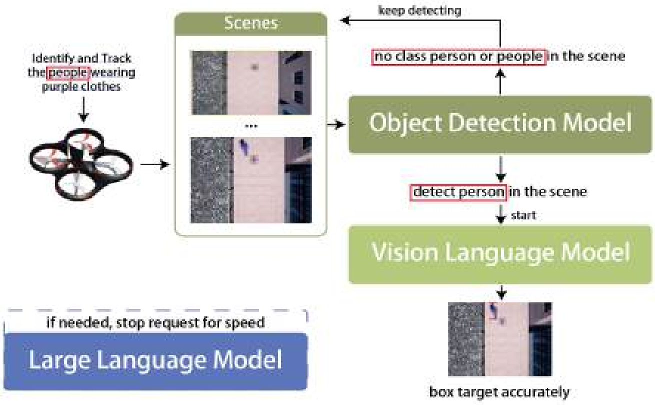


Fig. 7. The process of agent indentifying and tracking target.

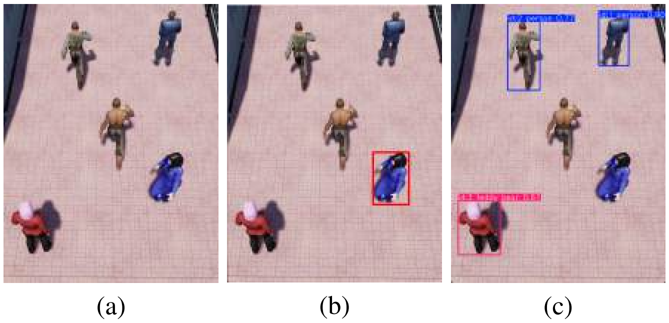


Fig. 8. (a) Original camera view of a group of people. (b) Detect result of Florence2. (c) Detect result of YOLO.

Once the VLM starts running, it can focus on the ”woman in a blue dress.” Regardless of whether the target is in corners, under shadows, or on roads with strong light, the agent can follow the target closely (see Fig 10). When we need to find and track someone in a fixed area, we no longer need to gather a large number of pictures to fine-tune the vision model. We also don’t need to worry about poor network connectivity in some areas, as the lightweight VLM can be deployed directly on the edge device and handle everything independently. To demonstrate that using the object detection model allows the Patrol Agent to perform faster tracking while using fewer computing resources, we shut down the object detection model and kept the VLM running until it found the target. In Table II,

TABLE I
COMPARATIVE ANALYSIS OF PATROL AGENT WITH VQA

Method Name	*Time to patrol a lap*	*Correct rate for captions*	*Correct rate for actions*
VQA (Qwen-VL-Max)	208 seconds	24.13%	36.67%
Patrol Agent	150 seconds	66.10%	89.47%

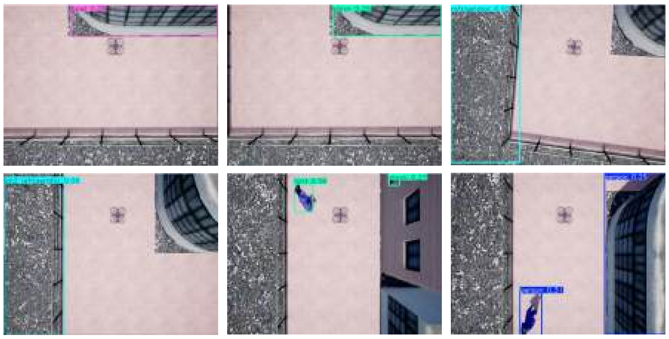


Fig. 9. The detect result before YOLO finds the people.

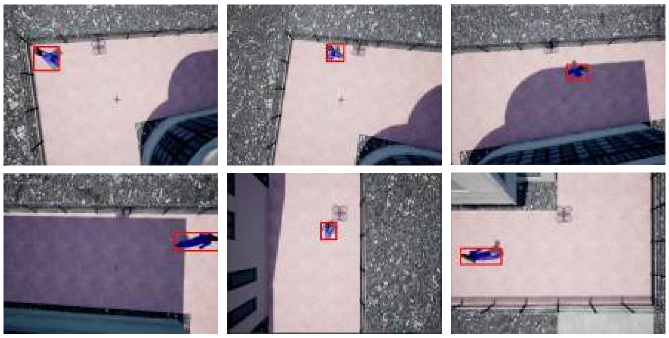


Fig. 10. The detect result when Florence2 is running.

although the VLM model used is already a lightweight one, it is still relatively large. Therefore, if we keep the VLM running before it detects a person, it would waste significant computing resources and place a heavy burden on the agent's heat dissipation system.

TABLE II
COMPARATIVE ANALYSIS OF KEEPING FLORENCE2 ALWAYS RUNNING OR NOT

Florence2 run time	*Time to find target*	*Graphics Memory*
Always	64 seconds	about 4.5 GB
Until YOLO find person	45 seconds	about 0.3 GB

IV. CONCLUSION

Our research demonstrates the functionality and impressive performance of the Urban Road Patrol Agent in patrolling, identifying and tracking. Remarkably, it can discern events within its view and suggest feasible actions without requiring fine-tuning on data. The proposed method exhibits distinct advantages in the following three facets:

(1) Model Efficiency: We utilize lightweight models such as Florence2-base (0.23B), YOLO-v8n (3.2M). LLM Qwen2-72B (72B), which are significantly smaller than models like GPT-4 or GPT-4V.

(2) Urban Road Patrol Focus: Our research concentrates on urban road patrol scenarios, including tracking, street fights, and individuals lying down. The Agent is specifically designed to adapt to these situations.

(3) Open-Source Models: Our Agent operates using open-source models, allowing others interested in this work to conduct further research on model behavior during the Agent's operation. Additionally, both models are trained on real-world data but can be applied directly in simulated environments. This demonstrates our method's potential to reduce the gap between real-world and simulated-world decision-making to a certain degree.

However, the Agent has areas that require improvement due to limitations in the robotic system it runs on:

(1) Response Speed: In AirSim running on Windows, using the simGetImages API to obtain images from the UAV's camera view takes about 100ms. After processing with Florence-2 and YOLO and waiting for Qwen2 responses, the Agent perceives the world at approximately 3-4Hz. This frequency is insufficient for rapid movements, such as following a person quickly. By integrating ROS, faster publishing and subscribing of sensor data could be facilitated, raising the frame rate to the theoretical upper limit of 30Hz while ensuring the parallel operation among all modules of the system. Additionally, applying Model Quantization to reduce the size of Florence-2 can further improve the perception rate. Running Qwen2 on the edge rather than on the cloud can also contribute to faster response times.

(2) Scene Variety: The real world is far more complex than our simulations, affecting image quality and UAV flight control in various ways. For instance, some scenarios may involve indirect information, requiring analysis of the connections among different pieces of information. To address these challenges, it is essential to design a container for the agent to extract and store information, giving it a form of ”memory” to analyze new scenes in the context of past experiences. Additionally, to enhance decision-making accuracy, a new structure must be designed to endow the agent with a certain level of ”judgment.” This will allow the agent to guide itself in generating reasonable and executable answers.

(3) Real-World Application: Although the Agent performs well in simulations, we aim to design a more robust physical body for it. This includes considerations for obstacle avoidance, tool carrying and utilization, and improved motion control. For example, we can equip the agent with more

powerful computational devices like jetson, employ more advanced inertial odometry, and utilize sophisticated path planning algorithms. Enhancing these aspects will enable the Agent to effectively reduce human stress in practical applications.

References

[1] X. Li and A. V. Savkin, "Networked unmanned aerial vehicles for surveillance and monitoring: A survey," *Future Internet*, vol. 13, no. 7, 2021.

[2] H. Ren, Y. Zhao, W. Xiao, and Z. Hu, "A review of uav monitoring in mining areas: current status and future perspectives," *International Journal of Coal Science & Technology*, vol. 6, pp. 320 – 333, 2019.

[3] R. A. Khalil, N. Saeed, and M. Almutiry, "Uavs-assisted passive source localization using robust tdoa ranging for search and rescue," *ICT Express*, vol. 9, no. 4, pp. 677–682, 2023.

[4] S. Ullah, K.-I. Kim, K. H. Kim, M. Imran, P. Khan, E. Tovar, and F. Ali, "Uav-enabled healthcare architecture: Issues and challenges," *Future Gener. Comput. Syst.*, vol. 97, p. 425–432, aug 2019.

[5] B. H. Y. Alsalam, K. Morton, D. Campbell, and F. Gonzalez, "Autonomous uav with vision based on-board decision making for remote sensing and precision agriculture," in *2017 IEEE Aerospace Conference*, pp. 1–12, 2017.

[6] S. Javaid, N. Saeed, and B. He, "Large language models for uavs: Current state and pathways to the future," 2024.

[7] T. Hickling, N. Aouf, and P. Spencer, "Robust adversarial attacks detection based on explainable deep reinforcement learning for uav guidance and planning," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 10, pp. 4381–4394, 2023.

[8] C. Wang, J. Wang, J. Wang, and X. Zhang, "Deep-reinforcement-learning-based autonomous uav navigation with sparse rewards," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6180–6190, 2020.

[9] Y. Lu, Z. Xue, G.-S. Xia, and L. Zhang, "A survey on vision-based uav navigation," *Geo-spatial information science*, vol. 21, no. 1, pp. 21–32, 2018.

[10] X. Chen and Q. Meng, "Self-learning vehicle detection and tracking from uavs," *International Journal of Mechanical Engineering and Robotics Research*, vol. 5, no. 2, pp. 149–155, 2016.

[11] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large language models: A survey," 2024.

[12] J. Zhang, J. Huang, S. Jin, and S. Lu, "Vision-language models for vision tasks: A survey," 2024.

[13] T. Guo, X. Chen, Y. Wang, R. Chang, S. Pei, N. V. Chawla, O. Wiest, and X. Zhang, "Large language model based multi-agents: A survey of progress and challenges."

[14] S. Vemprala, S. Chen, A. Shukla, D. Narayanan, and A. Kapoor, "Grid: A platform for general robot intelligence development," 2023.

[15] H. Zhao, F. Pan, H. Ping, and Y. Zhou, "Agent as cerebrum, controller as cerebellum: Implementing an embodied lmm-based agent on drones," 2023.

[16] "Gpt-4v(ision) system card," 2023.

[17] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, *et al.*, "Gpt-4 technical report," 2024.

[18] B. Xiao, H. Wu, W. Xu, X. Dai, H. Hu, Y. Lu, M. Zeng, C. Liu, and L. Yuan, "Florence-2: Advancing a unified representation for a variety of vision tasks," 2023.

[19] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO," Jan. 2023.

[20] J. Bai, S. Bai, Y. Chu, *et al.*, "Qwen technical report," *arXiv preprint arXiv:2309.16609*, 2023.

[21] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," 2017.

[22] R. Madaan, N. Gyde, S. Vemprala, M. Brown, K. Nagami, T. Taubner, E. Cristofalo, D. Scaramuzza, M. Schwager, and A. Kapoor, "Airsim drone racing lab," *arXiv preprint arXiv:2003.05654*, 2020.

[23] S. Casao, A. Otero, Álvaro Serra-Gómez, A. C. Murillo, J. Alonso-Mora, and E. Montijano, "A framework for fast prototyping of photo-realistic environments with multiple pedestrians," 2023.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need."

[25] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou, "Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond," *arXiv preprint arXiv:2308.12966*, 2023.

[26] P. Sui, E. Duede, S. Wu, and R. J. So, "Confabulation: The surprising value of large language model hallucinations," 2024.

[27] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," 2023.