Vehicle Density-Aware Adaptive Offloading for UAV-Based Road Traffic Monitoring

Mohammad Dwipa Furqan
Department of Computer Science
University of Tsukuba
Tsukuba, Japan
dwipa.mohammad@sd.cs.tsukuba.ac.jp

Fumio Machida
Department of Computer Science
University of Tsukuba
Tsukuba, Japan
machida@cs.tsukuba.ac.jp

Ermeson Andrade
Department of Computing
Federal Rural University of Pernambuco
Recife, Brazil
ermeson.andrade@ufrpe.br

***Abstract*—Unmanned Aerial Vehicles (UAVs) have emerged as a transformative technology for real-time road traffic monitoring, offering enhanced efficiency and responsiveness to modern traffic management systems. However, the resource limitations of UAVs and the dynamic nature of traffic densities present significant challenges for continuous operation. To address these constraints, this study proposes a vehicle-density-aware adaptive offloading mechanism that dynamically alternates between local processing and task offloading to fog nodes, based on real-time traffic conditions. The mechanism operates in three distinct modes: Low-CPU Mode for low vehicle density, Full Offloading Mode for moderate density, and Local Processing Mode for high-density scenarios. Preliminary results reveal that the proposed VD-aware adaptive offloading mechanism effectively balances performance, resource efficiency, and communication costs. It maintains competitive accuracy, optimizes throughput, and dynamically manages CPU utilization and communication overhead. These findings highlight the adaptability and efficiency of the proposed mechanism, making it an ideal solution for UAV-based road traffic monitoring in dynamic and resource-constrained environments.**

***Index Terms*—Fog computing, Road traffic monitoring, Task offloading, UAV, Vehicle density**

I. Introduction

Road traffic monitoring has become a critical concern due to increasing challenges in managing congestion and ensuring safety[1][2][3][4]. Modern traffic monitoring systems provide real-time insights into traffic patterns and incidents, enabling authorities to optimize infrastructure, enhance safety, and reduce environmental impacts[5][6][7][8]. Traffic congestion significantly affects economic productivity, environmental quality, and road safety by increasing fuel consumption, costs, and pollution. For instance, in 2014, road congestion in the United States led to $160 billion in additional costs, with similar economic impacts observed globally [9][10]. This issue is exacerbated by the growing number of vehicles and aging transportation infrastructure, emphasizing the need for intelligent and efficient traffic monitoring solutions.

Traffic congestion often results from incidents such as traffic accidents, which impede flow and create risks to public safety. Advanced monitoring systems, including UAV-based and smartphone-based solutions, have been developed to address these challenges [11][12][13]. While smartphone-based systems are accessible and convenient, they rely on user reports, which can be delayed or inaccurate [14][15][16]. In contrast, UAV-based systems provide real-time aerial monitoring, enabling comprehensive road coverage, timely accident detection, and access to hard-to-reach locations. This makes them highly effective for improving traffic management and public safety. Traditional monitoring methods, such as fixed infrastructure-based systems, lack the adaptability and coverage needed to address dynamic traffic patterns in modern cities. UAVs, with their mobility and scalability, have emerged as a promising alternative, offering high-resolution real-time data. However, UAVs are constrained by limited computational power, memory, and energy, which present challenges in real-time applications. Additionally, the dynamic nature of traffic density—ranging from low to high—requires varying levels of computational effort. Existing processing methods, such as full offloading to fog nodes or exclusive local processing, often fail to balance performance and resource efficiency. Full offloading can create network bottlenecks, while local processing can overwhelm the UAV's limited resources.

To address these challenges, we propose a vehicle-density-aware (VD-aware) adaptive offloading mechanism designed to dynamically adjust task allocation based on real-time traffic density. The VD-aware adaptive offloading mechanism operates in three modes: Low-CPU Mode for resource-efficient processing in no/low-density scenarios, Full Offloading Mode for maximizing throughput in moderate-density conditions, and Local Processing Mode for ensuring accuracy in high-density situations. To effectively evaluate the adaptability of our proposed approach, we designed a dynamic evaluation scenario that simulates real-world traffic conditions with varying vehicle densities. The implementation uses an NVIDIA Jetson Nano as the on-board processing UAV and a PC as the fog node to process offloaded tasks. These devices are connected via a Wi-Fi access point, simulating a real-world UAV-fog computing environment. Our evaluation considers key performance metrics, including accuracy, throughput, CPU usage, and communication cost to comprehensively compare the VD-aware adaptive offloading mechanism with existing approaches. The evaluation results demonstrate that the VD-aware adaptive offloading method achieves a balanced trade-off across accuracy, throughput, resource usage, and communication cost compared to existing methods. By integrating our adaptive mechanism, we aim to improve the overall performance and resource efficiency of UAV-based road traffic monitoring systems.

The rest of the paper is organized as follows. Section 2 reviews related research on UAV-based road traffic monitoring and computational offloading strategies, highlighting the gaps addressed in this research. Section 3 presents the proposed VD-aware adaptive offloading mechanism, detailing the operational mode and decision-making process. Section 4 describes the experimental setup, including the evaluation scenario, hardware configuration, and performance metrics. Section 5 discusses the experimental results, comparing the VD-aware adaptive offloading mechanism with existing approaches in terms of accuracy, throughput, CPU usage, and communication cost. Finally, Section 6 concludes the paper and outlines potential directions for future research.

II. Related Work

Fog and edge computing have emerged as essential paradigms for real-time data processing, enabling efficient and low-latency solutions for applications like road traffic monitoring. By decentralizing computational resources, fog computing reduces dependence on centralized cloud systems, enhancing both latency and bandwidth utilization [17]. UAV-based systems, in particular, significantly benefit from this paradigm, as they require quick and resource-efficient decision-making to effectively support real-time tasks such as road traffic monitoring.

A. UAV-Based Road Traffic Monitoring

UAVs are increasingly employed in road traffic monitoring due to their mobility, flexibility, and ability to capture aerial views of traffic conditions. Applications include vehicle counting, congestion detection, and accident reporting. These systems rely heavily on image processing techniques such as real-time object detection, classification, and tracking [18][19]. Recent research has demonstrated the effectiveness of fog computing in traffic management systems. For example, [20] proposed an Intelligent Traffic Congestion Mitigation System (ITCMS) that utilizes fog computing to address traffic congestion in densely populated smart cities. The system uses edge processing to reduce communication bandwidth and enable real-time decision-making, resulting in improved traffic efficiency and reduced latency. Furthermore, edge computing platforms have been developed for real-time traffic monitoring using computer vision techniques [21]. These systems can perform tasks such as congestion detection and speed monitoring without relying on cloud infrastructure, demonstrating the potential for low-latency, on-site processing in traffic management applications.

However, the dynamic nature of road traffic presents challenges in processing high-resolution data in real-time, particularly under resource-constrained environments. Existing studies often focus on either improving detection accuracy or reducing processing latency but rarely address the balance between these factors in dynamic traffic scenarios.

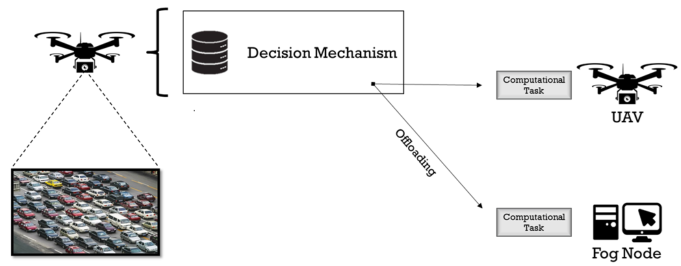

Fig. 1 System Design

B. Computational Offloading Strategies

Several computational offloading strategies have been proposed to optimize resource utilization in UAV-enabled fog computing systems. For example, adaptive offloading techniques, such as PA-offload, focus on balancing performance and service availability by dynamically deciding between local processing and fog offloading [22]. These methods often use probabilistic models to account for resource constraints and environmental uncertainties. Fuzzy-based offloading approaches [23] utilize object detection confidence levels to determine whether tasks should be offloaded to fog nodes or processed locally, achieving a balance between accuracy and computational load. Resource-aware frameworks like Resource-aware video streaming (RAViS) [24] extend these strategies by continuously monitoring CPU usage and adjusting processing rates dynamically. Such frameworks ensure resource-efficient operations, especially in UAV systems with limited onboard computational capabilities. These approaches emphasize adaptability in managing workloads, which is important requirement for UAV-based road traffic monitoring.

C. Contribution of This Work

Unlike existing approaches that primarily focus on static metrics such as resource availability or detection accuracy, our proposed VD-aware adaptive offloading mechanism introduces vehicle density as a key decision criterion. By dynamically switching between Low-CPU, Full Offloading, and Local Processing modes based on real-time vehicle density, the proposed method is expected to balance performance and resource efficiency. A simple illustration of our system can be seen in Figure 1. This contribution bridges the gap between dynamic workload adaptation and resource-constrained UAV systems in the context of road traffic monitoring.

III. Proposed Approach

To address the challenges of dynamic traffic conditions and resource limitations in UAV-based road traffic monitoring, we propose a VD-aware adaptive offloading mechanism as shown in Figure 2. This approach dynamically allocates computational tasks based on real-time vehicle density, switching between three operational modes: Low-CPU Mode, Full Offloading Mode, and Local Processing Mode. Vehicle density (ρ_v), defined as the number of vehicles per square meter ($vehicles/m^2$), is chosen as the key decision criterion due

TABLE I SUMMARY OF DIFFERENT COMPUTING MODES

Mode	Processing Location	Advantages	Disadvantages
Local Processing	UAV	High Accuracy & Network Communication-independent	High UAV Load & Low Throughput
Low-CPU	UAV	Low UAV Load & Network Communication-independent	Low throughput & Low Accuracy
Full Offloading	Fog Node	Low UAV Load & High Throughput	Low Accuracy & Network Communication-dependent

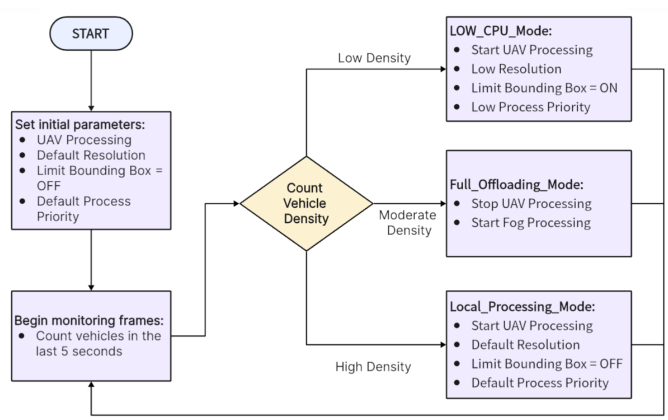

Fig. 2 VD-Aware Adaptive Offloading Mechanism

to its direct correlation with computational demands. This correlation has been observed in our preliminary experiments. As ρ_v increases, the number of detected objects in each frame rises, leading to higher processing complexity, increased inference time, and greater resource consumption. In low-density scenarios, fewer objects are detected, requiring minimal computational resources and enabling resource-efficient operation. In moderate-density conditions, faster vehicle movements and varying object detection requirements make fog-based offloading crucial to ensure faster detection (high throughput). High-density traffic scenarios, on the other hand, necessitate local processing to maintain detection accuracy of detection by large number of vehicles. Table 1 summarizes the three operational modes of the proposed VD-aware adaptive offloading mechanism, highlighting the corresponding processing locations, advantages, and disadvantages of each mode.

In Low_CPU_Mode, which is activated when the vehicle density is low, the UAV processes frames with reduced resolution (e.g., 300x200 pixels) and limits the number of detected vehicle to the top 5 based on confidence scores. Additionally, the process priority is lowered to conserve energy and minimize resource usage. This mode ensures that the system remains efficient while maintaining readiness for any workload changes.

The Full_Offloading_Mode is triggered under moderate traffic density. In such conditions, gaps in the traffic allow vehicles to move faster, requiring quicker frame analysis. The UAV offloads frames to a fog node, where the server performs object detection. By leveraging the computational resources of the fog node, this mode maximizes throughput while reducing the UAV's computational burden.

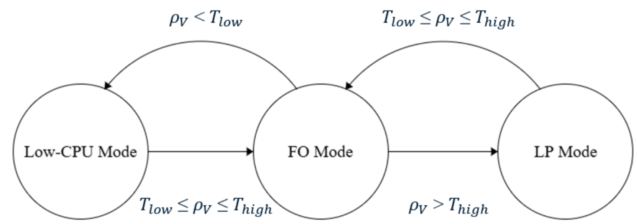

Fig. 3 Decision-making process state diagram

When the traffic density is high, the Local_Processing_Mode is activated to ensure high detection accuracy. Under these conditions, vehicles often move slowly or are stationary, making local processing a more efficient and reliable option compared to offloading. This mode eliminates potential network delays caused by offloading, which could reduce detection accuracy. To further illustrate the dynamic decision-making process, Figure 3 provides a state diagram depicting the transitions between the three operational modes. The transitions are determined by the real-time vehicle density (ρ_v) monitored every 5 seconds.

Threshold variables T_{low} and T_{high} are introduced to determine transitions between modes, with their values adjustable depending on the scenario. Starting from the Low-CPU Mode, if the ρ_v increases and exceeds T_{low}, the system transitions to the Full Offloading Mode, enabling the UAV to offload computational tasks to the fog node for efficient throughput in moderate-density traffic. Similarly, as ρ_v surpasses T_{high}, the system shifts to Local Processing Mode to ensure accurate detection by handling the computational workload directly on the UAV. Conversely, when the ρ_v decreases below T_{high}, the system transitions from Local Processing Mode back to Full Offloading Mode to balance throughput and resource usage during moderate traffic. Finally, as ρ_v below T_{low}, the system returns to Low-CPU Mode, conserving resource by reducing resolution and limiting processing demands. These transitions, monitored every 5 seconds, ensure adaptability to dynamic traffic conditions while maintaining optimal performance and resource efficiency.

IV. Experimental Setup

To evaluate the performance of the proposed vehicle-density-aware adaptive offloading mechanism, an experimental setup was designed with the following components:

Fig. 4 Sample of Image Datasets

A. Hardware and Software Configuration

To evaluate the performance of the proposed VD-aware adaptive offloading mechanism, an experimental setup was designed using specific hardware and software. The hardware included an NVIDIA Jetson Nano 4GB as the UAV processing unit and a PC with 16GB RAM and an Intel Core i3-13100H serving as the fog node. The Jetson Nano simulated the UAV's onboard processing, while the PC acted as the fog node to handle offloaded tasks. Both devices were connected to the same Wi-Fi access point, which served as the communication link, simulating a base station's role in a real-world UAV-fog computing scenario. The experiments were conducted using Python 3.8 with libraries such as OpenCV, PyTorch, and psutil for performance monitoring. The YOLOv8 model was used for vehicle detection, and the input resolution of the road traffic scene captured by the UAV was set to 600x400.

B. Data Collection

The dataset used for training the YOLOv8 model consisted of 1041 annotated vehicle images from real-world bird's-eye view perspectives as shown in Figure 4 [25]. The experiments themselves were conducted using GTA V as a simulator to generate diverse traffic scenarios [26][27]. These scenarios represented varying traffic densities: low density, moderate density, and high density in four minute. This combination of real-world data for training and simulated environments for testing ensured a comprehensive evaluation of the proposed mechanism.

C. Evaluation Metrics

The evaluation focused on four key metrics: accuracy, throughput, CPU usage, and communication cost. Accuracy was measured as the proportion of correctly detected vehicles compared to the ground truth, while throughput was assessed in frames per second (FPS) to evaluate processing speed. CPU usage was monitored on both UAV nodes to analyze computational load. Communication cost was measured as the time required for data transmission between the UAV and the fog node. The detailed explanation of each evaluation metric is as follows:

- **Accuracy**:
  We manually evaluated the accuracy of the object detection on a frame-by-frame basis using the following formula:

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{1}$$

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{2}$$

$$F1 \text{ Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{3}$$

  where TP is True Positives (correctly detected vehicles), FP is False Positives (incorrectly detected vehicles), and FN is False Negatives (missed vehicles that should have been detected). Finally, we obtained an average of F1 score. We used F1 Score metric to measure accuracy. The F1 score is particularly suitable in situations where there is an imbalance between positive and negative detections, as it considers both precision (the proportion of correct detections out of all detected) and recall (the proportion of actual vehicles correctly detected).
- **Throughput**:
  Throughput is measured in Frames Per Second (FPS) and indicates how many frames the system can process in one second. To obtain the FPS value, we need to determine the detection time. Detection time represents the total time for running the object detection model until obtain the final detection output. In the full offloading scenario, the total detection time is the sum of the UAV processing time (UPT), the communication delay between the UAV and the fog node (CD), and the fog node processing time (FPT).

$$\text{Detection Time (offloading)} = \text{UPT} + \text{CD} + \text{FPT}, \tag{4}$$

  FPS provides insight into the system's ability to handle real-time traffic monitoring. Since there are 1000 milliseconds in one second, the FPS can be calculated using the formula as:

$$FPS = \frac{1000}{\text{Detection Time (ms)}}. \tag{5}$$

- **CPU Usage**:
  We monitored CPU usage during the experiments. This metric helps assess how much computational resources were consumed by the UAV node, indicating the efficiency of different mechanism. The decision to prioritize CPU usage over GPU usage stemmed from practical considerations. CPU performance often represents a bottleneck in UAV systems. Many real-world UAV deployments face constraints in onboard CPU resources, making it critical to optimize CPU usage for energy efficiency and system performance. This focus provided valuable insights into the system's behavior under conditions where GPU acceleration may not always be feasible.

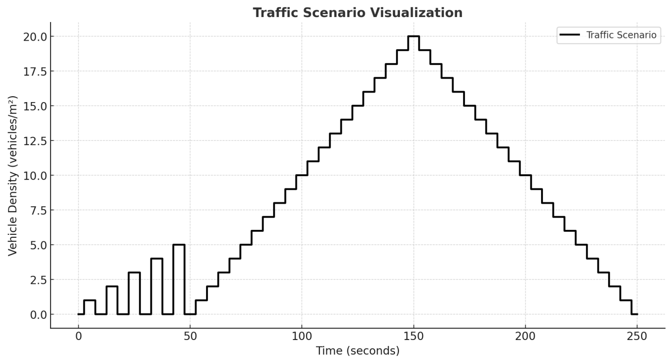

Fig. 5 Varying vehicle density of the evaluation scenario

- **Communication Cost**:
We evaluated the communication cost associated with data transfer between the UAV and the fog node. This metric captures the total time of transmitting video frames from the UAV to the fog node and receiving the processed results. Comparing the communication cost across all approaches provides insights into the trade-offs between offloading strategies. By analyzing this metric, we aim to highlight the conditions under which our VD-aware adaptive offloading mechanism outperforms other methods, balancing processing efficiency and communication overhead.

D. Evaluation Scenario

The evaluation scenario was designed to simulate dynamic traffic conditions with varying vehicle densities over a 4-minute video as shown in Figure 5. In the initial phase, the road alternated between being empty and having an incremental increase in vehicle density. For example, during the first 5 seconds, there were 0 $vehicles/m^2$. In the next 5 seconds, 1 $vehicle/m^2$ appeared, followed by 0 $vehicles/m^2$ again. This alternating pattern continued with an increasing vehicle density until reaching a maximum of 5 $vehicles/m^2$. In the middle phase, the scenario transitioned to a cumulative traffic pattern where vehicle density increased one by one every 5 seconds without disappearing from the frame, reaching up to 20 $vehicles/m^2$. After this peak density, vehicles began decreasing one by one every 5 seconds, simulating a gradual reduction in traffic until the road was completely empty. This multi-phase scenario was designed to comprehensively evaluate the system's adaptability to varying traffic densities.

E. Comparison Target

In this evaluation, we compare the performance of the proposed approach with four different baseline methodologies as presented below.

- **VD-aware Adaptive Offload**:
We implemented the VD-aware adaptive offloading approach presented in Section III. This approach dynamically adjusts task allocation among Low-CPU, Full Offloading, and Local Processing modes based on real-time vehicle density ρ_v. In this evaluation, we set T_{low} and T_{high} to 3 and 15 ($vehicles/m^2$), respectively. These values are determined by preliminary analysis and observations to switch the computation modes effectively depending on vehicle densities.
- **Fuzzy-based Decision**:
Tasks are offloaded to a fog node when object detection confidence scores are lower and processed locally when confidence scores exceed a threshold, using a fuzzy neural network [23]. To replicate the decision-making mechanism from the referenced study, we implemented a fuzzy control system to determine task offloading decisions. The system employs three key parameters: the average confidence of detected objects P_{obj}, the number of detected objects N_{obj}, and the confidence level of the decision C_{obj}. The input variables P_{obj} and N_{obj} were mapped to seven linguistic variables Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (ZO), Positive Small (PS), Positive Medium (PM), Positive Big (PB). A fuzzy rule base, derived from the referenced study, was constructed to guide the control mechanism. This rule base includes 49 rules, where combinations of P_{obj} and N_{obj} determine the output C_{obj} using Sugeno-style fuzzy inference. Based on the computed C_{obj}, the system decides whether to offload computations to the Fog Node or process locally at the Edge Node. A threshold R_{obj} of 0.5 for C_{obj} was empirically chosen to balance processing accuracy.
- **RAViS framework**:
RAViS framework aims to optimize resource availability while maintaining accuracy [24]. In this case, we use CPU usage thresholds to decide whether to continue process or stop the tasks. This method is designed to optimize resource usage by dynamically monitoring the system's CPU utilization and temporarily suspending object detection when the CPU usage exceeds a predefined threshold. The system enters a "paused" state for 10 seconds whenever the CPU usage surpasses 85%. While in this state, the object detection process is stopped, and video frames are annotated with a "Detection Paused" message, ensuring continuity in video output without detection overhead. This is expected to reduce CPU overload and ensure stable system performance.
- **Full Offloading**:
Full Offloading is a strategy where all computational tasks, such as object detection and tracking, are entirely offloaded from the UAV to a fog node. This approach leverages the fog node's higher processing power to achieve fast and reliable results, making it well-suited for scenarios with high computational demands or limited UAV resources. The main advantage is its ability to handle large workloads without overloading the UAV. However, it is highly dependent on network reliability and communication latency.
- **Local Processing**:
Local Processing refers to handling all computational

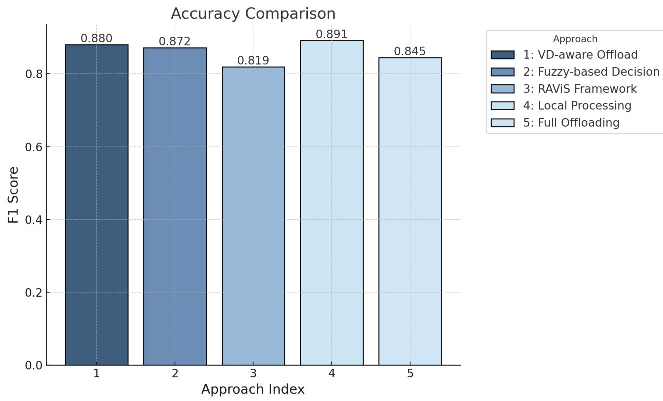

Fig. 6 Accuracy comparison of different approaches

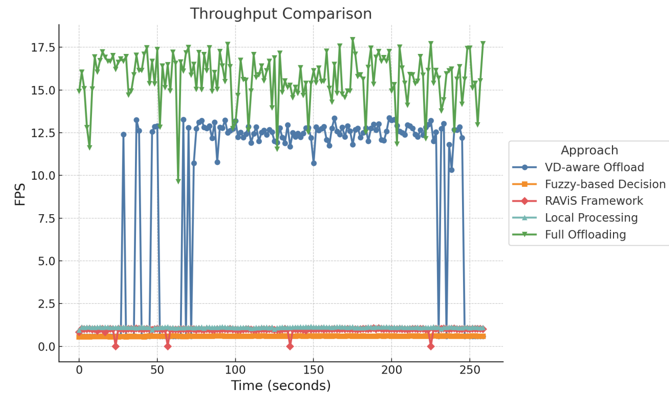

Fig. 7 Throughput comparison of different approaches

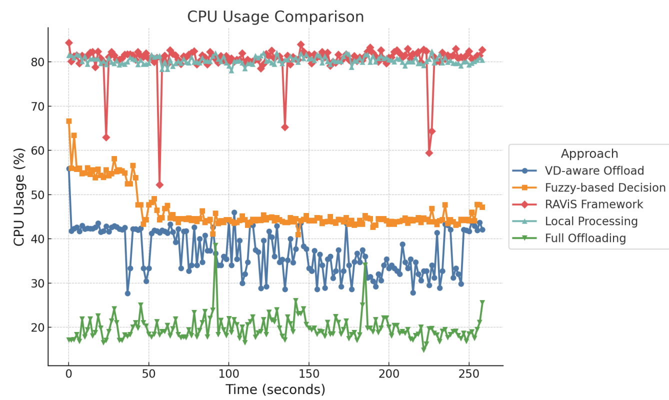

Fig. 8 CPU usage comparison of different approaches

tasks on the UAV itself without offloading to external nodes. This approach avoids network dependencies, ensuring consistent performance even in environments with poor connectivity. Local Processing is advantageous in scenarios requiring high accuracy and low latency, as it eliminates potential delays caused by network communication. However, the UAV's onboard computational resources are often limited, which can lead to higher CPU usage and lower throughput.

V. Results

We evaluated our proposed approach in comparison to baseline methods, focusing on accuracy, throughput, CPU usage, and communication cost. The analysis highlights the adaptability, resource efficiency, and trade-offs of each method under varying traffic conditions, showcasing the advantages and limitations of the proposed approach.

A. Accuracy

VD-aware adaptive offloading mechanism achieves a competitive F1 Score of 0.88, demonstrating its ability to balance accurate detection with adaptability to varying traffic densities as shown in Figure 6. While slightly outperformed by the Local Processing (LP) method, which achieves the highest F1 Score of 0.89. The Fuzzy-Based Decision approach follows closely with an F1 Score of 0.87. The RAViS Framework, driven by CPU usage thresholds, achieves the lowest F1 Score of 0.81, indicating that its focus on resource management comes at the cost of detection accuracy. On the other hand, the Full Offloading (FO) approach achieves an F1 Score of 0.84, slightly lower than the VD-aware adaptive offloading mechanism, due to communication overhead.

B. Throughput

As shown in Figure 7, the throughput of VD-aware adaptive offloading mechanism fluctuates significantly during moderate-density conditions, where the system effectively offloads tasks to the fog node. However, the throughput drops during low-density conditions, where resource usage is minimized with Low-CPU mode, and during high-density conditions, where local processing mode dominates. This fluctuation reflects the adaptability of the VD-aware adaptive offloading mechanism to dynamically changing traffic workloads. In contrast, the Fuzzy-Based Decision, RAViS Framework, and Local Processing (LP) methods maintain steady but low throughput, averaging around 1-2 FPS. The low throughput of the Fuzzy-Based Decision method occurs because, even when tasks are offloaded to the fog node, local processing still performs calculations, which increases detection time and limits throughput. The RAViS Framework and LP methods are constrained by the limited computational resources of the UAV as it relies on localized processing. The Full Offloading (FO) method achieves the highest throughput, maintaining an average of approximately 14-15 FPS throughout the experiment due to its reliance on the powerful fog node for all computational tasks. However, this comes with a significant dependency on network stability.

C. CPU Usage

The CPU usage of VD-aware adaptive offloading mechanism remains stable and moderate, averaging around 40% throughout the experiment as shown in Figure 8. This reflects the system's ability to dynamically transition between modes based on traffic density, avoiding excessive strain on the UAV while maintaining efficient operation. In contrast, the Fuzzy-Based Decision approach shows slightly higher CPU usage,

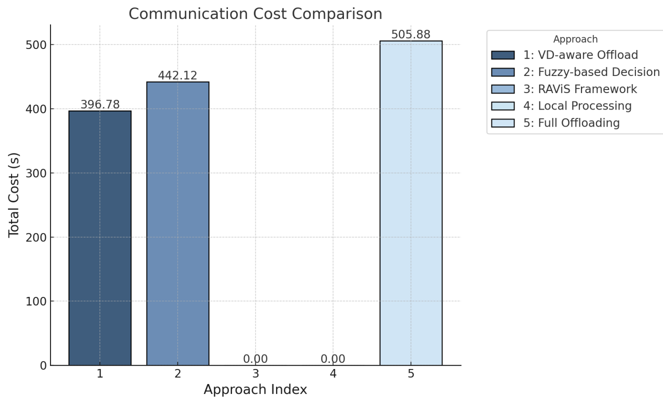

Fig. 9 Communication cost comparison of different approaches

averaging between 45%. While it doesn't employ an explicit Low-CPU Mode like the VD-aware adaptive offloading mechanism, its lower CPU usage compared to Local Processing (LP) and RAViS Framework is due to its selective offloading mechanism. The approach relies on fuzzy logic to decide whether to offload tasks or process them locally based on bounding box confidence and count. The RAViS Framework and LP methods exhibit the highest CPU usage, averaging close to 80%, as both rely entirely on onboard processing and lack the adaptability to reduce computational load during lighter traffic conditions. The RAViS Framework exhibits the highest and most erratic CPU usage, attributed to its reliance on continuous local processing and the absence of an adaptive mechanism to distribute tasks or reduce computational load dynamically. Additionally, periodic pauses in processing lead to noticeable CPU utilization drops. On the other hand, the Full Offloading (FO) approach achieves the lowest CPU usage, averaging between 15-20%, by offloading all tasks to the fog node.

D. Communication Cost

The VD-aware adaptive offloading mechanism demonstrates the lowest communication cost, effectively balancing offloading decisions by dynamically adapting to vehicle density as shown in Figure 9. During low-density conditions, the system operates in Low-CPU Mode, minimizing the need for communication. In high-density scenarios, the Local Processing Mode further reduces communication costs by eliminating offloading altogether, while offloading is strategically utilized in moderate-density conditions to optimize throughput. In contrast, the Fuzzy-Based Decision approach incurs slightly higher communication costs compared to the Proposed Method. This is because its offloading mechanism relies on confidence scores rather than vehicle density, leading to more frequent task transmissions to the fog node, regardless of traffic conditions. The Full Offloading (FO) method exhibits the highest communication cost, as it continuously sends all tasks to the fog node, making it heavily dependent on the network. On the other hand, the RAViS Framework and Local Processing (LP) approaches incur zero communication costs, as all processing is performed locally on the UAV. However, this comes at the expense of higher CPU usage and limited adaptability to varying traffic conditions. These findings underscore the efficiency of the VD-aware adaptive offloading mechanism in managing communication overhead, making it a more scalable and cost-effective solution compared to other approaches, particularly in scenarios where network reliability and resource efficiency are critical.

E. Multiple Experiment Statistics

To evaluate the effectiveness of the proposed VD-aware adaptive offloading mechanism, we conducted the same experiments multiple times, each run three times to ensure statistical reliability. The evaluation focused on key performance metrics, including accuracy, throughput, CPU usage, and communication cost. In UAV-based monitoring, key performance indicators (KPIs) must be defined to ensure the system meets real-time operational requirements. In this study, we establish the following primary KPIs: a target accuracy of ≥ 0.85 to ensure detection reliability, a throughput of ≥ 5 FPS to maintain real-time responsiveness, CPU usage in the UAV $\leq 80\%$ to ensure stable operation, and a communication cost of ≤ 60 ms to minimize unnecessary overhead. For each approach, the mean and variance of these metrics were calculated to assess both their central tendencies and variability under dynamic traffic conditions. Table II reveals that the VD-aware adaptive offloading mechanism consistently delivers competitive accuracy, comparable to other methods while maintaining high throughput. This highlights its adaptability in managing varying traffic densities without compromising detection performance.

In terms of resource utilization, the VD-aware adaptive offloading mechanism showcased efficient CPU usage, effectively balancing computational demands to avoid overburdening the UAV. Unlike approaches that rely heavily on onboard processing or simplistic offloading strategies, the VD-aware adaptive offloading mechanism dynamically adapts its resource usage, demonstrating better scalability and reliability under diverse conditions. Additionally, the VD-aware adaptive offloading mechanism achieved lower communication costs compared to other offloading approaches, reducing the dependency on network reliability. Overall, based on Table II, only VD-aware adaptive offloading mechanism successfully meets our defined KPIs.

VI. Conclusion

The experimental results show the effectiveness of our VD-aware adaptive offloading mechanism in achieving balanced performance, resource efficiency, and communication cost. By dynamically adapting to varying vehicle densities, the proposed mechanism demonstrated competitive accuracy, optimized throughput, while efficiently managing resource usage. Unlike other approaches, such as Fuzzy-Based Decision, RAViS, and Local Processing (LP), which either lacked adaptability or overburdened the UAV, our VD-aware adaptive

TABLE II MEAN AND VARIANCE (σ) OF METRICS FOR EACH APPROACH

Approach	Accuracy (F1 score) mean (sd)	Throughput (FPS) mean (sd)	CPU Usage (%) mean (sd)	Communication Cost (ms) mean (sd)
VD-Aware Offload	0.88 (0.004)	6.28 (3.902)	41.80 (4.370)	50.87 (49.601)
Fuzzy-Based Decision	0.87 (0.004)	0.57 (0.010)	48.17 (2.320)	56.68 (31.681)
RAViS Framework	0.82 (0.016)	0.92 (0.148)	79.83 (3.517)	00.00 (00.000)
Local Processing	0.89 (0.002)	1.05 (0.018)	80.64 (0.626)	00.00 (00.000)
Full Offloading	0.85 (0.007)	14.10 (1.251)	20.21 (2.421)	64.86 (35.712)

offloading mechanism maintained a stable balance between computational demands and resource efficiency.

Additionally, While Full Offloading (FO) achieved high throughput and low UAV resource usage, but it introduced significant communication costs due to network dependency. The VD-aware adaptive offloading mechanism minimized these costs by selectively offloading task. Our VD-aware adaptive offloading mechanism presents a comprehensive solution that is well-suited for dynamic and resource-constrained UAV-enabled fog computing systems for road traffic monitoring.

Future work will focus on dynamic threshold optimization using machine learning to adapt to varying traffic and environmental conditions. Additionally, Incorporating energy-aware considerations to improve the system's sustainability, particularly in long-duration deployments is also a challenge addressed in future work.

Acknowledgment

This work was supported in part by a grant from the Telecommunications Advancement Foundation.

References

[1] E. Udin, "South korea will use drones and AI to monitor traffic conditions next year," 2023. [Online]. Available: https://www.gizchina.com/2023/12/27/south-korea-drones-ai-traffic-monitoring/

[2] A. J. Tobias, "Ohio gets approval for drone traffic surveillance program," 2024. [Online]. Available: https://www.govtech.com/transportation/ohio-gets-approval-for-drone-traffic-surveillance-program

[3] C. U. N. Staff, "Hamburg is pioneering urban drone traffic management," 2024. [Online]. Available: https://www.commercialuavnews.com/europe/hamburg-is-pioneering-urban-drone-traffic-management

[4] B. E. Putra, "6 drones 5 kilometers ready to monitor movements on traffic jams," 2023. [Online]. Available: https://voi.id/en/news/341544

[5] H. Today, "Police now using drones to detect traffic offenders," 2024. [Online]. Available: https://hungarytoday.hu/police-now-use-drones-to-detect-traffic-offenders/

[6] TradeTrucks, "Drones to be deployed in NSW to monitor traffic," 2024. [Online]. Available: https://www.tradetrucks.com.au/drones-to-be-deployed-in-nsw-to-monitor-traffic/

[7] G. Singh, "WATCH: Bengaluru traffic police deploys 10 drone cameras to monitor traffic, aide commuters," 2024. [Online]. Available: https://www.news9live.com/city/bengaluru/watch-bengaluru-traffic-police-deploys-10-drone-cameras-to-monitor-traffic-aide-commuters-2425761

[8] I. Singh, "Police drones to monitor spring break crowds at miami beach," 2024. [Online]. Available: https://dronedj.com/2024/02/29/spring-break-2024-miami-drone/

[9] A. C. Sutandi, "Its impact on traffic congestion and environmental quality in large cities in developing countries," *Proceedings of the Eastern Asia Society for Transportation Studies*, pp. 180–180, 2009.

[10] T. Afrin and N. Yodo, "A survey of road traffic congestion measures towards a sustainable and resilient transportation system," *Sustainability*, vol. 12, no. 11, 2020.

[11] C. Zhu, J. Zhu, T. Bu, and X. Gao, "Monitoring and identification of road construction safety factors via uav," *Sensors*, vol. 22, no. 22, 2022.

[12] J. T. Lee, M. S. Ryoo, M. Riley, and J. K. Aggarwal, "Real-time illegal parking detection in outdoor environments using 1-d transformation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 1014–1024, 2009.

[13] D. T. Mane, S. Sangve, S. Kandhare, S. Mohole, S. Sonar, and S. Tupare, "Real-time vehicle accident recognition from traffic video surveillance using YOLOV8 and OpenCV," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 11, no. 5s, pp. 250–258, 2023.

[14] J. White, C. Thompson, H. Turner, B. Dougherty, and D. C. Schmidt, "WreckWatch: Automatic traffic accident detection and notification with smartphones," *Mob. Netw. Appl.*, vol. 16, no. 3, pp. 285–303, 2011.

[15] M. S. Amin, M. A. S. Bhuiyan, M. Bin Ibne Reaz, and S. S. Nasir, "Gps and map matching based vehicle accident detection system," in *2013 IEEE Student Conference on Research and Developement*, 2013, pp. 520–523.

[16] T. Laor and Y. Galily, "In waze we trust? gps-based navigation application users' behavior and patterns of dependency," *PLOS ONE*, vol. 17, no. 11, pp. 1–17, 11 2022.

[17] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, ser. MCC '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 13–16.

[18] E. V. Butilă and R. G. Boboc, "Urban traffic monitoring and analysis using unmanned aerial vehicles (uavs): A systematic literature review," *Remote Sensing*, vol. 14, no. 3, 2022.

[19] I. Bisio, C. Garibotto, H. Haleem, F. Lavagetto, and A. Sciarrone, "A systematic review of drone based road traffic monitoring system," *IEEE Access*, vol. 10, pp. 101 537–101 555, 2022.

[20] A. Elsayed, K. Mohamed, and H. Harb, "Enhanced traffic congestion management with fog computing - a simulation-based investigation using IFOG-simulator," *Int. J. Comput. Netw. Commun.*, vol. 16, no. 3, pp. 29–48, May 2024.

[21] G. Liu, H. Shi, A. Kiani, A. Khreishah, J. Lee, N. Ansari, C. Liu, and M. M. Yousef, "Smart traffic monitoring system using computer vision and edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12 027–12 038, 2022.

[22] F. Machida, Q. Zhang, and E. Andrade, "Performability analysis of adaptive drone computation offloading with fog computing," *Future Generation Computer Systems*, vol. 145, pp. 121–135, 2023.

[23] Y. Yuan, S. Gao, Z. Zhang, W. Wang, Z. Xu, and Z. Liu, "Edge-cloud collaborative uav object detection: Edge-embedded lightweight algorithm design and task offloading using fuzzy neural network," *IEEE Transactions on Cloud Computing*, vol. 12, no. 1, pp. 306–318, 2024.

[24] A. M. Shiddiqi, E. D. Yogatama, and D. A. Navastara, "Resource-aware video streaming (ravis) framework for object detection system using deep learning algorithm," *MethodsX*, vol. 11, p. 102285, 2023.

[25] DFN9, "Dfn9vehicle dataset," Roboflow, 2025. [Online]. Available: https://universe.roboflow.com/dfn9/dfn9vehicle

[26] R. Games, "Grand theft auto V," 2013. [Online]. Available: https://www.rockstargames.com/gta-v

[27] ItsJustCurtis, "Menyoo 2.0," GTA5-Mods, 2024. [Online]. Available: https://www.gta5-mods.com/scripts/menyoo-2-0