

Sharif: Edge Computing and Energy Consumption Optimization for UAVs at Various Altitudes

Mohamed Benaly

Faculty of Sciences, Laboratory of Electronic Systems, Information Processing, Mechanics and Energetics, Ibn Tofail University, Morocco
mohamed.benaly@uit.ac.ma (corresponding author)

Azzedine El Mrabet

Laboratory of Advanced Systems Engineering, Ibn Tofail University, Kenitra, Morocco
azzedine.elmrabet@uit.ac.ma

Hajar El Karch

Laboratory of Advanced Systems Engineering, Ibn Tofail University, Kenitra, Morocco
hajar.elkarch@uit.ac.ma

Rachid El Gouri

Laboratory of Advanced Systems Engineering, Ibn Tofail University, Kenitra, Morocco
rachid.elgouri@uit.ac.ma

Abdelkader Mezouari

Faculty of Sciences, Laboratory of Electronic Systems, Information Processing, Mechanics and Energetics, Ibn Tofail University, Morocco
abdelkader.mezouari1@uit.ac.ma

Laamari Hlou

Faculty of Sciences, Laboratory of Electronic Systems, Information Processing, Mechanics and Energetics, Ibn Tofail University, Morocco
laamari.hlou@uit.ac.ma

Received: 11 April 2025 | Revised: 16 May 2025 and 24 May 2025 | Accepted: 27 May 2025

Licensed under a CC-BY 4.0 license | Copyright (c) by the authors | DOI: <https://doi.org/10.48084/etasr.11413>

ABSTRACT

Unmanned Aerial Vehicles (UAVs) are essential for enhancing situational awareness and mitigating risks in mission areas prone to threats. However, integrating deep learning architectures into UAVs is challenging due to payload and energy limitations, which can cause rapid battery depletion and reduced operational duration, limiting their effectiveness in extended missions. This paper presents a novel electronic architecture for UAVs to optimize energy consumption during UAV missions while maintaining high performance. The architecture dynamically switches between two accelerators, a Visual Processing Unit (VPU) and a Graphics Processing Unit (GPU), based on real-time mission requirements, enabling seamless transitions and efficient handling of both simple and complex tasks. Experimental evaluations show that this approach reduces energy consumption by up to 26% compared to traditional systems, thereby extending UAV operational endurance and ensuring high service quality across multiple altitudes.

Keywords-Unmanned Aerial Vehicle (UAV); mission area; electronic architecture; hardware accelerator; deep learning architecture; power consumption

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have become prominent due to their agility and real-time monitoring capabilities, with advancements in Artificial Intelligence (AI) and hardware significantly enhancing their performance and safety. Despite this progress, integrating real-time aerial object detection and embedded processing remains challenging due to object scale, orientation, density variations, and power constraints. Deep learning, particularly through Convolutional Neural Networks (CNNs) such as Region-based CNN (R-CNN), Faster R-CNN, Mask R-CNN, and You Only Look Once (YOLO), has advanced detection accuracy and robustness. Recent frameworks, such as Object Based Image Analysis CNN (OBIA)-CNN for avalanche detection [1], transformer-CNN hybrids for UAV imagery [2], and Diminutive Multi-Dimensional Locality Coding based CNN (DMLC-CNN) for human activity recognition [3], exemplify these improvements. Mask R-CNN and Detection Transformer (DETR) have also been used for tree crown delineation using multispectral and UAV lidar data [4], whereas the YOLO family has been deployed in applications such as seat belt detection, fault diagnosis, crack identification, and weed mapping [5-8].

The development of energy-efficient, high-performance hardware has enabled UAVs to run complex machine learning and deep learning algorithms in real-time. Hardware accelerators such as Neural Processing Units (NPUs), Tensor Processing Units (TPUs), Graphics Processing Units (GPUs), and Field-Programmable Gate Arrays (FPGAs) from Intel, Nvidia, and Google [9-15], along with edge AI platforms like Intel's OpenVINO [16], Nvidia's TensorRT [17], and Google's Coral [18], have been instrumental in supporting these capabilities. Although FPGAs are energy-efficient and suitable for real-time processing, their complexity limits widespread use [19]. UAVs still face constraints due to limited payload and energy capacity, especially when executing demanding deep learning models. These challenges underscore the need for innovative approaches that not only optimize hardware and neural networks but also adapt to dynamic operational requirements across varying altitudes [20-24].

UAVs are increasingly adopted for urban surveillance to counter threats such as illegal Micro Aerial Vehicle (MAV) flights, yet many existing systems struggle with detection or are cost-prohibitive. To tackle this, authors in [20] developed a UAV system with a vision-based deep learning interceptor, validated through successful aerial tests. Authors in [21] introduced the PAL system, using deep learning for real-time action recognition and location mapping via a Pixel2GPS converter. Authors in [22] proposed a UAV-based railway intrusion detector featuring Fused Convolutional Long Short-Term Memory (Fused-ConvLSTM), attention modules, and lightweight strategies. Authors in [23] presented a Deep Reinforcement Learning (DRL) framework for energy-efficient UAV navigation, validated using a Digital Twin and Parrot ANAFI drone. Authors in [24] tested modular ML systems on constrained UAV hardware for tasks such as autonomous landing and people detection. Authors in [25] used dual-camera systems and multi-frame deep learning for efficient drone

detection. Authors in [26] enhanced CenterNet with depth features, a Gaussian kernel, and DIoU loss to boost small target detection. Authors in [27] introduced an AF vision system leveraging autoencoders for UAV camera focus control, achieving high accuracy. Authors in [28] highlighted RetinaNet's superior performance on the Stanford Drone Dataset (SDD) for UAV object detection. Authors in [29] built an edge-enabled drone network for bird vocalization-based classification using low-resource machine learning. Authors in [30] proposed a UAV-based avalanche victim detector using CNNs and Support Vector Machines (SVMs) with enhanced pre/post-processing.

Despite these innovations, a lack of focus on efficient onboard processing systems continues to hinder UAV surveillance performance, reducing flight time and increasing energy use, thus limiting their operational effectiveness in prolonged and remote missions.

II. PROPOSED TECHNIQUE

This paper proposes "Sharif," a novel dynamic switching framework designed to optimize energy consumption while maintaining precise target tracking across various UAV altitudes. Sharif leverages advanced deep learning and edge computing strategies to adaptively adjust computational resources based on mission requirements. By intelligently balancing energy usage and processing power, Sharif enables UAVs to operate efficiently in diverse scenarios, addressing the challenges of limited battery life and high computational demands.

Our system design methodology is centered around an edge system, chosen based on the UAV's operating altitude, which serves as the primary hardware for data processing. The proposed system automatically detects any moving objects within the monitored area, providing real-time positional data via the UAV. This approach enhances UAV efficiency and reduces energy consumption. The conceptual diagram of the complete proposed system is depicted in Figure 1.

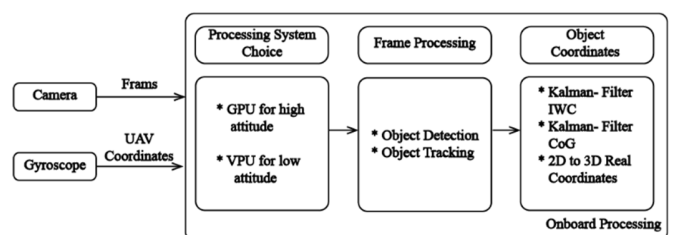


Fig. 1. Conceptual diagram of the proposed system.

The main stages of the methodology are summarized as follows:

- Processing system selection: The decision to use either a Visual Processing Unit (VPU) or GPU in the processing system is determined by the UAV's altitude.
- Frame processing: Employing a neural network architecture in conjunction with a Kalman filter to process and track objects. This process is carried out on the selected processing system.

- Pixel to real-world coordinates conversion: Translating digital image pixel coordinates to their corresponding physical locations in the real world through camera calibration. This involves geometric camera calibration or camera resectioning to estimate the parameters of the camera lens and image sensor.

A. Processing System Choice

The swap between two hardware processing systems, based on the UAV's altitude (Figure 2), involves exchanging processing units responsible for handling real-time frames from the camera, which reflect the UAV's orientation and motion. This swap enables a seamless transition between the two processing systems, ensuring the UAV's altitude control remains uninterrupted. The swap is triggered when a predetermined altitude condition is reached. The new processing system swiftly takes over the responsibilities of the previous system, utilizing its hardware capabilities to ensure optimal performance. This seamless swap ultimately guarantees smooth and reliable UAV operation, enhancing overall stability, control, and energy consumption.

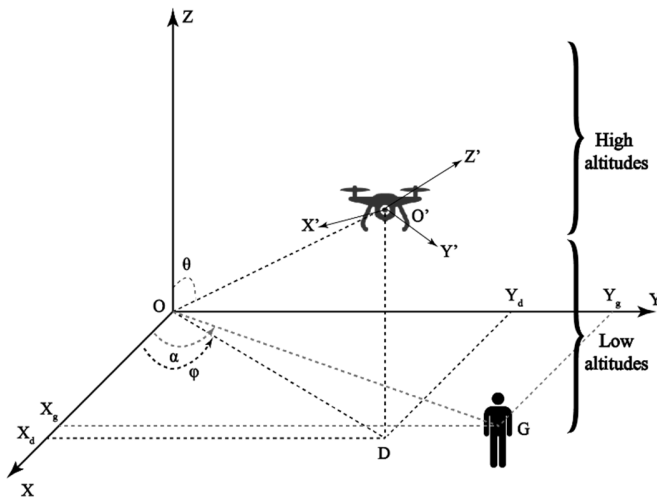


Fig. 2. UAV altitudes within the monitored area.

B. Deep Learning for Edge Computing

After selecting a model for deployment, it is important to determine which edge AI platforms can support its execution in the proposed system. This study specifically examines OpenVINO and TensorRT, developed by Intel and Nvidia, respectively. These toolkits allow AI developers to deploy pre-trained deep learning models using high-level C++ Inference Engine APIs integrated with application logic. Before utilizing the OpenVINO inference engine to deploy a deep learning model, the model must be converted to a format compatible with edge AI hardware using the OpenVINO toolkit. Following this conversion, the model is structured into an Intermediate Representation (IR) network, which the Inference Engine can effectively utilize. The IR comprises two binary files, .xml and .bin, essential for performing detection model inference within the OpenVINO virtual environment. Figure 3 summarizes the typical workflow for deploying the model on Neural Compute Stick 2 (NCS2).

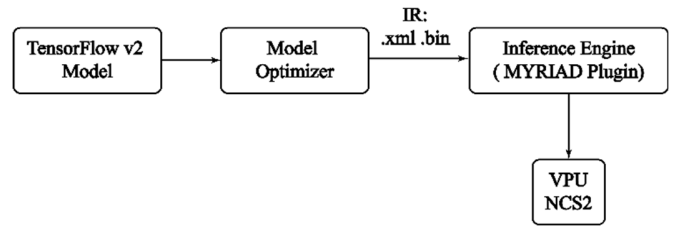


Fig. 3. Workflow to convert and deploy a TensorFlow v2 model to IR using Intel OpenVINO.

With the TensorRT Toolkit, the model format contains all the information required to share or deploy a trained model. In native TensorFlow, the workflow typically involves loading the saved model and running inference using TensorFlow runtime. In TensorFlow–TensorRT (TF-TRT), there are additional steps involved, including applying TensorRT optimizations to the TensorRT-supported subgraphs of the model, and optionally pre-building the TensorRT engines to convert the model. After conversion, the model is restructured into a Dynamic Shape Profile Strategy network, which consists of .engine files that allow the application to execute inference in the TensorRT virtual environment [22]. Figure 4 illustrates the workflow for deploying a TensorFlow v2 model using TF-TRT on a GPU.

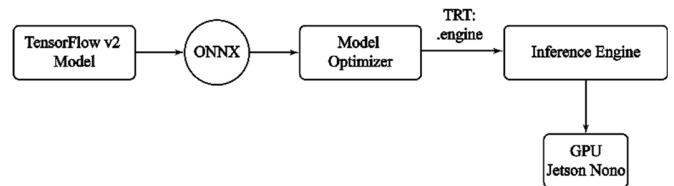


Fig. 4. Workflow to convert and deploy a TensorFlow v2 model to TRT using Nvidia TensorRT.

This study focuses on TensorFlow as the framework for implementing AI edge solutions on Intel VPU and Nvidia GPU platforms due to its optimization capabilities. TensorFlow integrates seamlessly with Intel's libraries, enhancing performance on Intel-based devices. It also supports Nvidia GPUs through CUDA, enabling accelerated training and inference via parallel computing. Furthermore, TensorFlow supports the Intel Movidius NCS2, designed for efficient edge AI applications by offloading computation and freeing system resources.

III. ELECTRONIC SYSTEM AND SOFTWARE ARCHITECTURE

The hardware architecture of the system is illustrated in Figure 5. The electronic architecture employed in our work combines two hardware accelerators: the Nvidia GPU (128-core Nvidia Maxwell architecture GPU, 921 MHz) and the Intel VPU (Intel Movidius Myriad X Vision Processing Unit, 700 MHz). The Nvidia Jetson Nano CPU (Quad-core ARM Cortex-A57 MPCore processor, 1.43 GHz) provides excellent multitasking capabilities simultaneously between input (frames and UAV position) and image processing units (Nvidia GPU and Intel VPU). A camera is integrated as an input device to the CPU, capturing frames from the UAV's perspective for subsequent processing. Additionally, a gyroscope serves as a

position sensor, providing real-time orientation and movement data. This integration enables accurate UAV position tracking, ensuring precise control and maneuverability.

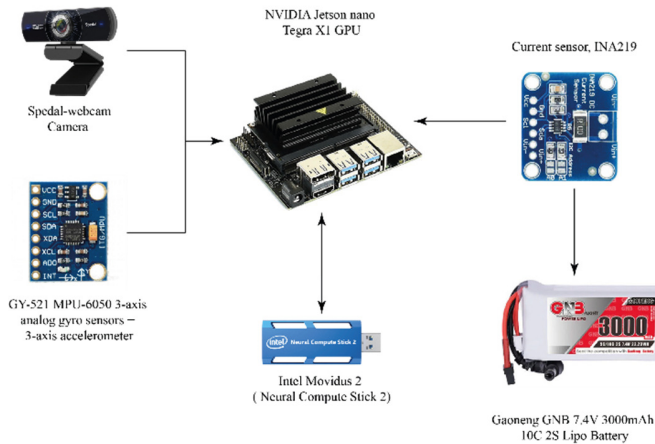


Fig. 5. Hardware architecture of the proposed electronic system.

The Intel Movidius Myriad X VPU (NCS2) accelerates AI model execution for real-time analysis of low-altitude UAV images, whereas the Nvidia Maxwell architecture GPU excels in processing detailed high-altitude images. The Spedal Webcam Camera, with full HD or 4K resolution and a wide-angle lens, enhances image quality and preprocessing on the Jetson Nano. For accurate altitude measurement, the GY-521 MPU-6050 gyroscope module integrates seamlessly with the Jetson Nano. Additionally, the INA219 sensor measures power consumption, voltage, and current, transmitting data via the I2C bus protocol for subsequent power calculations in the electronic architecture. The OpenVINO and TensorRT toolkits allow us to optimize and deploy neural network architecture on the target hardware. The Model Optimizer and Inference Engine components are employed in the application. Figure 6 depicts the overall software architecture. The multitasking CPU functions as a sensor gateway to support the camera and gyroscope. Based on the application, we ensured that all communication among the VPU-CPU-GPU components communicate via the PCIe/UART bus. For high UAV altitudes, the CPU uses the Nvidia GPU to transfer sensor data to the discrete GPU for processing. For low UAV altitudes, the CPU uses the Intel VPU to transfer data to the host VPU for processing.

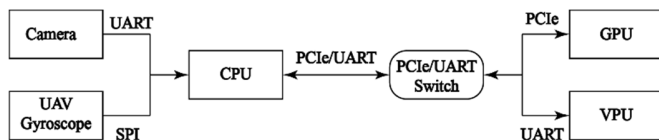


Fig. 6. Software architecture of the proposed system.

IV. FIELD TEST OF THE UAV SYSTEM

A. UAV System Prototype

The UAV onboard application is deployed on the Nvidia Jetson Nano (8 GB RAM) with Intel NCS2, which serves as

the onboard computer for our UAV (Figure 7). The Jetson Nano, which is mounted on the Auvideo J120 carrier board, connects to the UAV using a USB-to-TTL/UART connection. The UAV is powered by a Gaoneng GNB 7.4 V 3000 mAh 10C 2S Lipo battery, which also supplies power to the onboard computer via a custom-made voltage regulator. The Spedal Webcam Camera and GY-521 MPU-6050 are attached to the UAV, with the camera fixed at a pre-defined angle. The total payload of our onboard setup used to run the PAL system (including onboard computer, webcam, power adapter, and cables) is 0.98 kg.

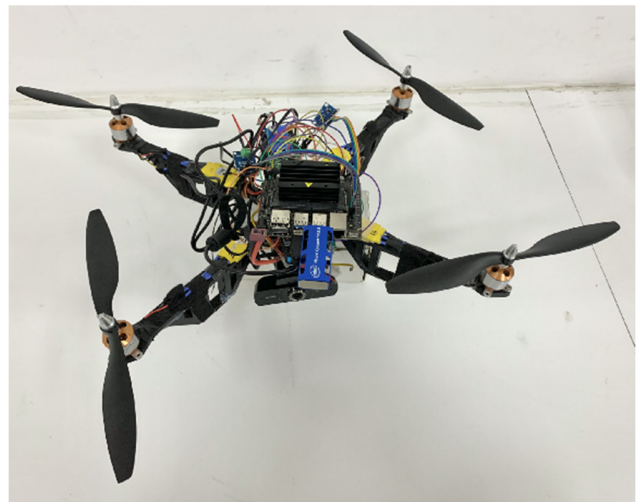


Fig. 7. UAV system prototype.

B. Evaluation

We evaluate our system by randomly placing three mobile objects (representing three people) on a 2D plane. These objects follow three predetermined paths. The UAV velocity and camera orientation remain fixed for each run, but we vary the UAV altitude. The camera's Angle of View (AOV) is set at 64°, and the camera orientation is fixed at a tilt angle of 55° throughout the experiments. The altitude, AOV, and tilt angle all impact the final real-world coordinates.

V. RESULTS AND DISCUSSION

During the UAV's initial flight, the primary objective is to determine the minimum altitude L_{min} for seamless system transition. The VPU detects objects within the surveyed area while ascending until detection capabilities are impaired, observed at 12 m. Thus, the optimal transition altitude L_{min} is identified as $L_{min} = 12$ m. The second UAV flight aims to leverage both systems' capabilities, utilizing the VPU for low-altitude and the GPU for high-altitude monitoring. This entails continuous detection of objects and estimation of their positions in each loop, allowing for comprehensive coverage of the area. The flight aims to quantify the total power consumption of the integrated system throughout the mission. Figure 8 illustrates the UAV's trajectory alongside the paths of three tracked objects, as computed by our system. This visualization highlights the UAV's movement and the system's ability to accurately track multiple objects in real-time.

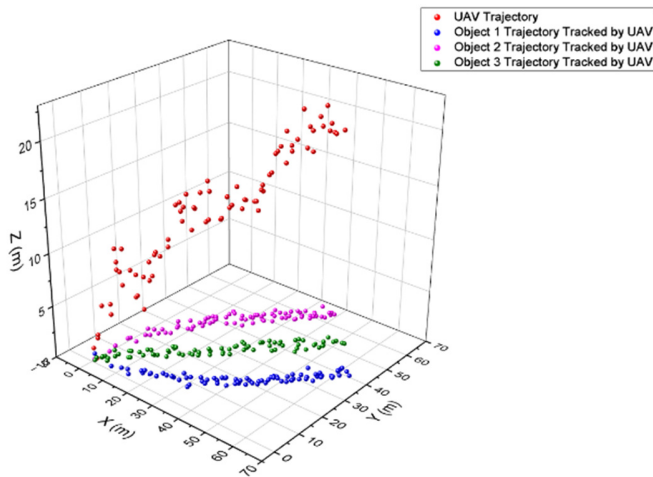


Fig. 8. UAV trajectory and the trajectories of three tracked objects.

Figure 9 illustrates the system's energy consumption at various altitudes. At the minimum altitude $L_{min} = 12$ m, the Intel VPU demonstrates low energy consumption, emphasizing its efficiency in performing low-altitude processing tasks. Conversely, at altitudes above L_{min} , the GPU exhibits higher energy consumption, reflecting its handling of more complex, energy-intensive processes. This contrast underscores the differing efficiency and resource utilization of the two systems.

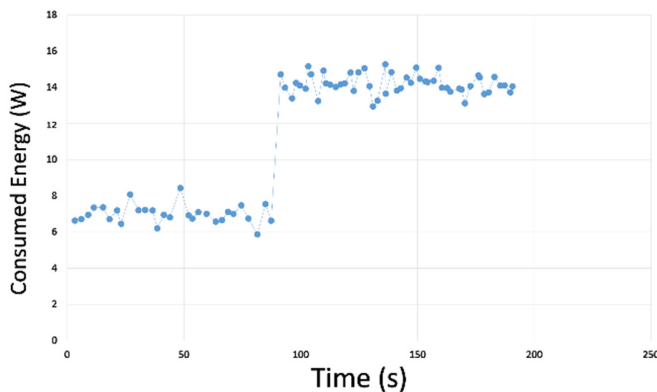


Fig. 9. Energy consumption graph.

Figure 10 illustrates the average energy consumption of the three systems: the VPU, the GPU, and our proposed combined system. This figure provides a comparative analysis, showcasing the efficiency of our combined system in terms of energy usage. By juxtaposing the energy consumption of each processing unit with our integrated approach, it becomes evident that the combined system optimizes energy efficiency.

Most UAV studies focus on neural networks and specialized applications but lack dynamic systems for energy efficiency. While some achieve real-time detection and low energy use, they fail to optimize for varying altitudes. Sharif's approach uniquely integrates dynamic system swaps, reducing energy consumption by 26%, enhancing efficiency, extending flight time, and setting a benchmark. Table I provides a comparative analysis of UAV systems from prior studies and

the proposed Sharif system. This project, while innovative, has several limitations that warrant consideration. The architecture is designed for individual UAVs, and its scalability to swarm operations has not been validated. Additionally, the system requires periodic updates during missions, a process that becomes more complex when operating with a swarm of UAVs.

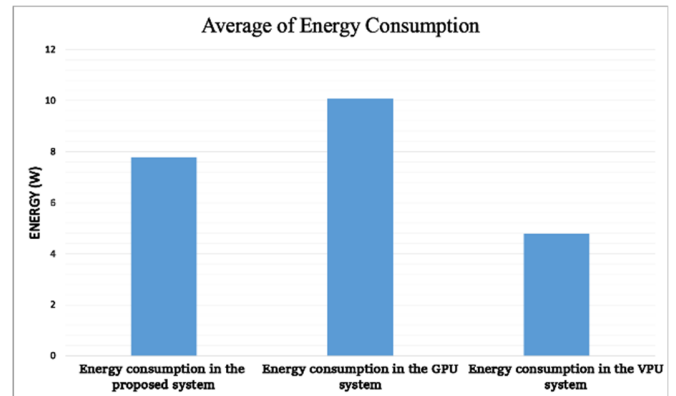


Fig. 10. Average energy consumption per system.

TABLE I. COMPARISON OF UAV SYSTEMS

Study/ approach	Performance	Energy efficiency	Dynamic system swap	Unique contribution
[20]	Detection speed of 9.4 FPS with YOLOv4-tiny	-	No	Effective for interception but lacks energy focus
[21]	PAL with 0.667 FPS	-	No	Enhances awareness but lacks energy optimization
[22]	Fused-ConvLSTM with 29.3 ms/img	-	No	Utilizes AI for specific application scenarios
[23]	DRL policy with 0.41 min/mission	0.39 kJ/mission	No	Focused on navigation in dynamic environments
[24]	LeNet with 45 ms/frame	0.246 J/frame	No	Energy-focused, but lacks dynamic altitude adaptation
Sharif (proposed system)	Detection speed of 25 FPS for GPU and 5 FPS for VPU with SSD MobileNet v2 320x320	26% reduction in energy consumption	Yes	First to implement system swap for altitude-based energy efficiency

VI. CONCLUSION

In this study, we presented an innovative Unmanned Aerial Vehicle (UAV) system designed to enhance monitoring capabilities across diverse scenarios. By integrating a combined hardware accelerator topology, the proposed system achieves significant reductions in energy consumption while maintaining minimal object loss during operations. Our

comparative analysis identified the neural network topology most suited to the specific requirements of the processing systems, ensuring optimal performance and efficiency. Extensive experiments validated the system's ability to dynamically transition between hardware accelerators, achieving seamless adaptability to varying operational conditions. These findings highlight the system's superiority in minimizing energy consumption and improving throughput compared to state-of-the-art UAV mission mechanisms.

Beyond the immediate improvements in energy efficiency and detection accuracy, the findings have broader implications for the UAV research community. They demonstrate the potential of dynamic hardware integration in optimizing UAV operations, paving the way for more sustainable and versatile aerial systems. Future work will focus on optimizing power consumption for a swarm of UAVs operating in complex environments. This will involve the development of collaborative algorithms and efficient energy-sharing mechanisms to enhance the overall performance, longevity, and orchestration of UAV fleets integrated with advanced hardware.

REFERENCES

- [1] S. K. Dewali, K. Jain, D. Varshney, S. Dhamija, and E. Pundir, "Combining OBIA, CNN, and UAV photogrammetry for automated avalanche deposit detection and characterization," *Advances in Space Research*, vol. 72, no. 8, pp. 3109–3132, Oct. 2023, <https://doi.org/10.1016/j.asr.2023.06.033>.
- [2] W. F. Hendria, Q. T. Phan, F. Adzaka, and C. Jeong, "Combining transformer and CNN for object detection in UAV imagery," *ICT Express*, vol. 9, no. 2, pp. 258–263, Apr. 2023, <https://doi.org/10.1016/j.icte.2021.12.006>.
- [3] K. P. Sinha and P. Kumar, "Human activity recognition from UAV videos using a novel DMLC-CNN model," *Image and Vision Computing*, vol. 134, June 2023, Art. no. 104674, <https://doi.org/10.1016/j.imavis.2023.104674>.
- [4] S. Dersch, A. Schöttl, P. Krzystek, and M. Heurich, "Towards complete tree crown delineation by instance segmentation with Mask R-CNN and DETR using UAV-based multispectral imagery and lidar data," *ISPRS Open Journal of Photogrammetry and Remote Sensing*, vol. 8, Apr. 2023, Art. no. 100037, <https://doi.org/10.1016/j.ojphoto.2023.100037>.
- [5] Y. Liu *et al.*, "Detection method of the seat belt for workers at height based on UAV image and YOLO algorithm," *Array*, vol. 22, July 2024, Art. no. 100340, <https://doi.org/10.1016/j.array.2024.100340>.
- [6] B. J. Souza, S. F. Stefenon, G. Singh, and R. Z. Freire, "Hybrid-YOLO for classification of insulators defects in transmission lines based on UAV," *International Journal of Electrical Power & Energy Systems*, vol. 148, June 2023, Art. no. 108982, <https://doi.org/10.1016/j.ijepes.2023.108982>.
- [7] Q. Qiu and D. Lau, "Real-time detection of cracks in tiled sidewalks using YOLO-based method applied to unmanned aerial vehicle (UAV) images," *Automation in Construction*, vol. 147, Mar. 2023, Art. no. 104745, <https://doi.org/10.1016/j.autcon.2023.104745>.
- [8] O. G. Ajayi, J. Ashi, and B. Guda, "Performance evaluation of YOLO v5 model for automatic crop and weed classification on UAV images," *Smart Agricultural Technology*, vol. 5, Oct. 2023, Art. no. 100231, <https://doi.org/10.1016/j.atech.2023.100231>.
- [9] S. J. Wei, D. F. Al Riza, and H. Nugroho, "Comparative study on the performance of deep learning implementation in the edge computing: Case study on the plant leaf disease identification," *Journal of Agriculture and Food Research*, vol. 10, Dec. 2022, Art. no. 100389, <https://doi.org/10.1016/j.jafr.2022.100389>.
- [10] S. H. Lee, H. Goëau, P. Bonnet, and A. Joly, "New perspectives on plant disease characterization based on deep learning," *Computers and Electronics in Agriculture*, vol. 170, Mar. 2020, Art. no. 105220, <https://doi.org/10.1016/j.compag.2020.105220>.
- [11] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," *Frontiers in Plant Science*, vol. 7, Sept. 2016, Art. no. 1419, <https://doi.org/10.3389/fpls.2016.01419>.
- [12] J. Mas, T. Panadero, G. Botella, A. A. Del Barrio, and C. García, "CNN Inference acceleration using low-power devices for human monitoring and security scenarios," *Computers & Electrical Engineering*, vol. 88, Dec. 2020, Art. no. 106859, <https://doi.org/10.1016/j.compeleceng.2020.106859>.
- [13] D. Rivas, F. Guim, J. Polo, P. M. Silva, J. Ll. Berral, and D. Carrera, "Towards automatic model specialization for edge video analytics," *Future Generation Computer Systems*, vol. 134, pp. 399–413, Sept. 2022, <https://doi.org/10.1016/j.future.2022.03.039>.
- [14] L. Yang, G. Liu, J. Wang, H. Bai, J. Zhai, and Y. Dai, "Fast3DS: A real-time full-convolutional malicious domain name detection system," *Journal of Information Security and Applications*, vol. 61, Sept. 2021, Art. no. 102933, <https://doi.org/10.1016/j.jisa.2021.102933>.
- [15] E. Badidi, K. Moumane, and F. E. Ghazi, "Opportunities, Applications, and Challenges of Edge-AI Enabled Video Analytics in Smart Cities: A Systematic Review," *IEEE Access*, vol. 11, pp. 80543–80572, 2023, <https://doi.org/10.1109/ACCESS.2023.3300658>.
- [16] "Quick Start Guide — NVIDIA TensorRT Documentation." Nvidia. <https://docs.nvidia.com/deeplearning/tensorrt/latest/getting-started/quick-start-guide.html>.
- [17] "OpenVINO 2025.3 — OpenVINO™ documentation — Version(2025)." OpenVINO. <https://docs.openvino.ai/2025/index.html>.
- [18] "Coral." Google for Developers. <https://developers.google.com/coral>.
- [19] B. Moyer and Y. Watanabe, "Chapter 13 - Hardware Accelerators," in *Real World Multicore Embedded Systems*, B. Moyer, Ed. Oxford, UK: Newnes, 2013, pp. 447–480, <https://doi.org/10.1016/B978-0-12-416018-7.00013-4>.
- [20] Z. W. Lee, W. H. Chin, and H. W. Ho, "Air-to-air Micro Air Vehicle interceptor with an embedded mechanism and deep learning," *Aerospace Science and Technology*, vol. 135, Apr. 2023, Art. no. 108192, <https://doi.org/10.1016/j.ast.2023.108192>.
- [21] R. Galdes *et al.*, "UAV-Based Situational Awareness System Using Deep Learning," *IEEE Access*, vol. 7, pp. 122583–122594, 2019, <https://doi.org/10.1109/ACCESS.2019.2938249>.
- [22] H. Huang *et al.*, "Railway intrusion detection based on refined spatial and temporal features for UAV surveillance scene," *Measurement*, vol. 211, Apr. 2023, Art. no. 112602, <https://doi.org/10.1016/j.measurement.2023.112602>.
- [23] A. Soliman *et al.*, "AI-based UAV navigation framework with digital twin technology for mobile target visitation," *Engineering Applications of Artificial Intelligence*, vol. 123, no. B, Aug. 2023, Art. no. 106318, <https://doi.org/10.1016/j.engappai.2023.106318>.
- [24] A. Albanese, M. Nardello, and D. Brunelli, "Low-power deep learning edge computing platform for resource constrained lightweight compact UAVs," *Sustainable Computing: Informatics and Systems*, vol. 34, Apr. 2022, Art. no. 100725, <https://doi.org/10.1016/j.suscom.2022.100725>.
- [25] E. Unlu, E. Zenou, N. Riviere, and P.-E. Dupouy, "Deep learning-based strategies for the detection and tracking of drones using several cameras," *IPSN Transactions on Computer Vision and Applications*, vol. 11, no. 1, July 2019, Art. no. 7, <https://doi.org/10.1186/s41074-019-0059-x>.
- [26] X. Zhizhong, W. Jingen, H. Zhenghao, and S. Yuhui, "Research on multi UAV target detection algorithm in the air based on improved CenterNet," in *2020 International Conference on Big Data & Artificial Intelligence & Software Engineering*, Bangkok, Thailand, 2020, pp. 367–372, <https://doi.org/10.1109/ICBASE51474.2020.000084>.
- [27] A. Ahmed and R. N. Farhan, "Autofocus Vision System Enhancement for UAVs via Autoencoder Generative Algorithm," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 18867–18872, Dec. 2024, <https://doi.org/10.48084/etasr.8519>.
- [28] X. Wang, P. Cheng, X. Liu, and B. Uzochukwu, "Fast and Accurate, Convolutional Neural Network Based Approach for Object Detection from UAV," in *IECON 2018 - 44th Annual Conference of the IEEE*

Industrial Electronics Society, Washington, DC, USA, 2018, pp. 3171–3175, <https://doi.org/10.1109/IECON.2018.8592805>.

- [29] N. Das, N. Padhy, N. Dey, A. Mukherjee, and A. Maiti, "Building of an edge enabled drone network ecosystem for bird species identification," *Ecological Informatics*, vol. 68, May 2022, Art. no. 101540, <https://doi.org/10.1016/j.ecoinf.2021.101540>.
- [30] M. B. Bejiga, A. Zeggada, A. Nouffidj, and F. Melgani, "A Convolutional Neural Network Approach for Assisting Avalanche Search and Rescue Operations with UAV Imagery," *Remote Sensing*, vol. 9, no. 2, Feb. 2017, Art. no. 100, <https://doi.org/10.3390/rs9020100>.