FPGA-Accelerated YOLOX with Enhanced Attention Mechanisms for Real-Time Wildfire Detection on UAVs

Anubhav Elhence, Anubhav Panda, Vinay Chamola *Senior Member, IEEE*, Biplab Sikdar *Fellow, IEEE*

Abstract—**Real-time wildfire detection is crucial for enabling prompt intervention and minimizing environmental and economic damages; however, deploying high-accuracy detection models on resource-constrained platforms like unmanned aerial vehicles (UAVs) presents significant challenges due to limitations in computational capacity and power availability. In this paper, we propose LCAM-YOLOX, an enhanced object detection framework that integrates a Layer-wise Channel Attention Module (LCAM) into the YOLOX architecture to improve detection accuracy while maintaining computational efficiency. The model is optimized for deployment on FPGA platforms through 8-bit integer quantization, facilitating efficient inference on devices with limited resources. We implement and evaluate the LCAM-YOLOX model on the Xilinx Kria KV260 FPGA platform, demonstrating that it achieves a quantized mean Average Precision (mAP) of 78.11%, outperforming other state-of-the-art models such as YOLOv3, YOLOv5, and YOLOX-m. Moreover, the LCAM-YOLOX model processes at 195 frames per second (FPS) using a single DPU core on the KV260, exceeding real-time processing requirements while consuming only 10.45 W of power, which translates to the highest performance per watt ratio among the tested platforms. These results highlight the suitability of the KV260 FPGA as an optimal choice for deploying high-performance, energy-efficient wildfire detection models on UAVs, enabling real-time monitoring in resource-constrained environments.**

Index Terms—**Quantized Neural Networks, Hardware-Software Co-Design, Aerial Robotics, Computer Vision for Other Robotic Applications, Energy and Environment-aware Automation, Field Robots, Intelligent Transportation Systems**

I. Introduction

Wildfires significantly threaten ecosystems, human lives, and property worldwide. Rapid detection and response are crucial to mitigate their devastating effects. Unmanned Aerial Vehicles (UAVs) have emerged as effective tools for real-time wildfire monitoring due to their flexibility and ability to access remote areas [1], [2]. However, implementing advanced fire detection algorithms on UAVs is challenging due to limitations in computational resources and energy constraints.

Deep learning models, particularly convolutional neural networks (CNNs), have been widely adopted for wildfire detection [3], [4]. Models like YOLOv5 and its variants have shown high accuracy in detecting fires from aerial images [5], [6]. Nevertheless, these models are often computationally intensive, making them unsuitable for real-time applications on resource-constrained UAVs. The constraints become even more critical when considering drone communication and battery limitations [7], [8], as limited battery capacity restricts flight time and continuous data transmission, directly impacting the operational efficiency of UAV-based monitoring systems. Efficient hardware and algorithmic optimizations are therefore imperative to ensure reliable and uninterrupted operation during critical wildfire monitoring tasks.

To address these challenges, lightweight models and optimization techniques have been explored [9], [10]. Quantization and pruning are common methods to reduce model size and computation without significantly sacrificing accuracy [5]. Hardware accelerators on Field-Programmable Gate Arrays (FPGAs) offer customizable architectures that can be optimized for specific tasks while consuming less power, making them ideal for UAV applications [10].

In this paper, we propose an FPGA-accelerated wildfire detection system using a modified YOLOX model enhanced with a Layer-wise Channel Attention Module (LCAM). LCAM is a lightweight attention mechanism designed to improve feature extraction by refining channel-wise feature maps at multiple network layers. By dynamically emphasizing important channels while suppressing less relevant ones, LCAM enhances the model's ability to detect objects, especially small and complex features such as wildfires.

The proposed LCAM-YOLOX model is quantized to an INT8 format for efficient deployment. Quantization refers to the process of reducing the precision of model weights and activations from 32-bit floating point (FP32) to 8-bit integer (INT8), which reduces computational complexity and memory usage while maintaining near-original accuracy. The quantized accuracy is measured using mean Average Precision (mAP), a standard metric that evaluates the precision of object detection models by computing the average precision across multiple object classes.

The LCAM-YOLOX model is deployed on the Xilinx Kria KV260 FPGA platform to achieve high detection accuracy and real-time performance suitable for UAV-based wildfire monitoring. The overall system architecture is illustrated in Fig. 1.

Our major contributions are:

1) We develop the LCAM-YOLOX model by integrating

Anubhav Elhence, Anubhav Panda and Vinay Chamola are with the Department of Electrical and Electronics Engineering & APPCAIR, BITS-Pilani, Pilani Campus, India 333031 (e-mail: elhenceanubhav@gmail.com, vinay.chamola@pilani.bits-pilani.ac.in).

Biplab Sikdar is with Departhment of Electronics and Communication Engineering, National University of Singapore (NUS) (email:bsikdar@nus.edu.sg)

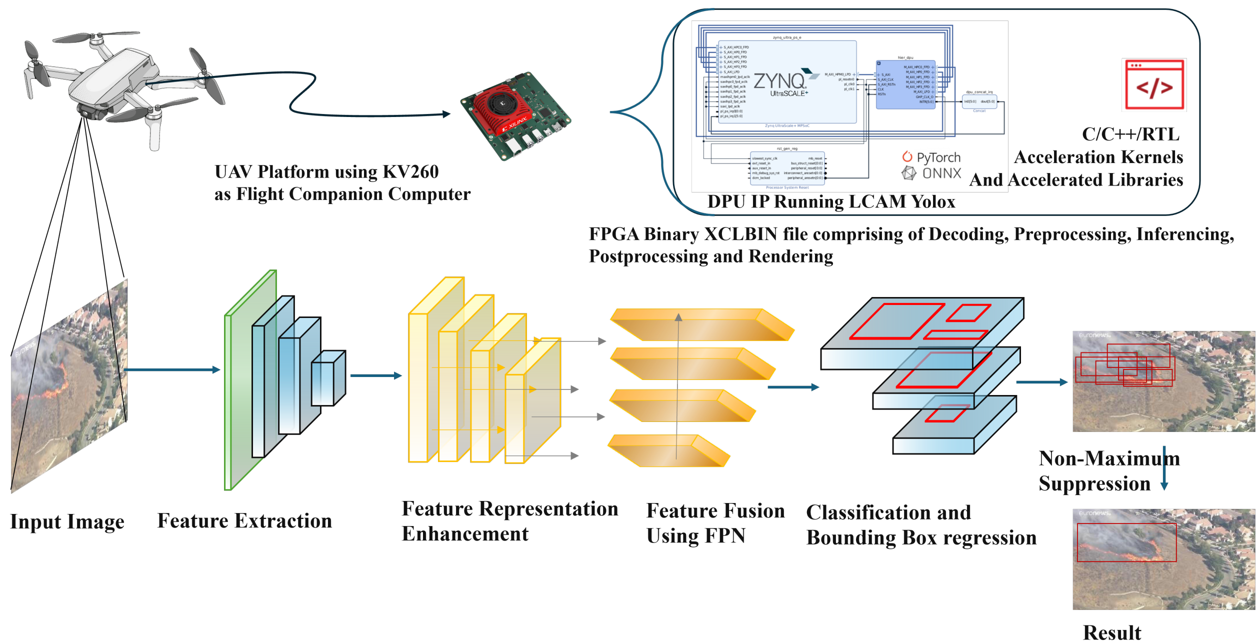


Fig. 1: Overall system architecture illustrating the end-to-end wildfire detection pipeline: from UAV image acquisition through FPGA-accelerated processing to fire detection.

a Layer-wise Channel Attention Module into YOLOX, enhancing detection accuracy in complex wildfire scenarios.
2) We apply quantization techniques to optimize the model for FPGA deployment, significantly reducing computational load and power consumption.
3) We achieve substantial improvements in frames per second (FPS), enabling real-time wildfire detection on UAV platforms.
4) We demonstrate the system's capability to function as a Flight Companion Computer (FCC) or replace traditional flight controllers on UAVs.

The rest of the paper is organized as follows: Section II reviews related works on wildfire detection using deep learning and hardware accelerators. Section III details our methodology, including the proposed LCAM-YOLOX model, quantization techniques, and FPGA deployment process. Section IV presents the experiments conducted to validate our approach. Section V discusses the results and analyzes the performance of our system compared to existing methods. Finally, Section VI concludes the paper and suggests future work directions.

II. Related Works

The application of Unmanned Aerial Vehicles (UAVs) for wildfire detection has garnered significant research interest due to their capability to provide real-time monitoring over extensive and inaccessible terrains. Recent studies have demonstrated the effectiveness of UAVs equipped with various sensing technologies and computer vision algorithms for early fire detection and response [11]–[14]. Concurrently, deploying machine learning (ML) models on resource-constrained UAV platforms has led to investigations into hardware acceleration using Field-Programmable Gate Arrays (FPGAs). This section reviews related works in these domains.

Several studies have focused on enhancing fire detection capabilities using UAVs equipped with advanced computer vision algorithms. The YOLO family of network models have been widely adopted, with various optimizations proposed for UAV applications. Zhou et al. [3] and Song et al. [6] developed lightweight variants of YOLOv5 using MobileNetV3 and RepVGG respectively, achieving improved computational efficiency suitable for real-time applications. Jiang et al. [4] proposed UAV-FDN, incorporating efficient attention modules and multi-scale fusion to enhance detection accuracy while reducing false positives and negatives.

Attention mechanisms and multi-modal approaches have been leveraged to improve detection performance in complex environments. Yang et al. [15] and Zhang et al. [16] enhanced detection systems with attention modules, specifically targeting early-stage and small-scale fire detection. Several researchers have explored multi-modal sensing approaches, with Xie et al. [10] utilizing dual-light vision and Choutri et al. [1] combining YOLO-based models with stereo vision for improved detection and geo-localization capabilities. The integration of hardware optimization techniques has been explored to meet the computational demands of real-time fire detection on UAVs. Recent work has focused on developing efficient system architectures and practical implementations for fire detection and response [2], [5], [9]. Researchers have also investigated ways to optimize computational resources, with Fouda et al. [17] and Hussain et al. [18] proposing adaptive frameworks that balance model complexity with detection accuracy.

Beyond fire detection, the deployment of ML models on

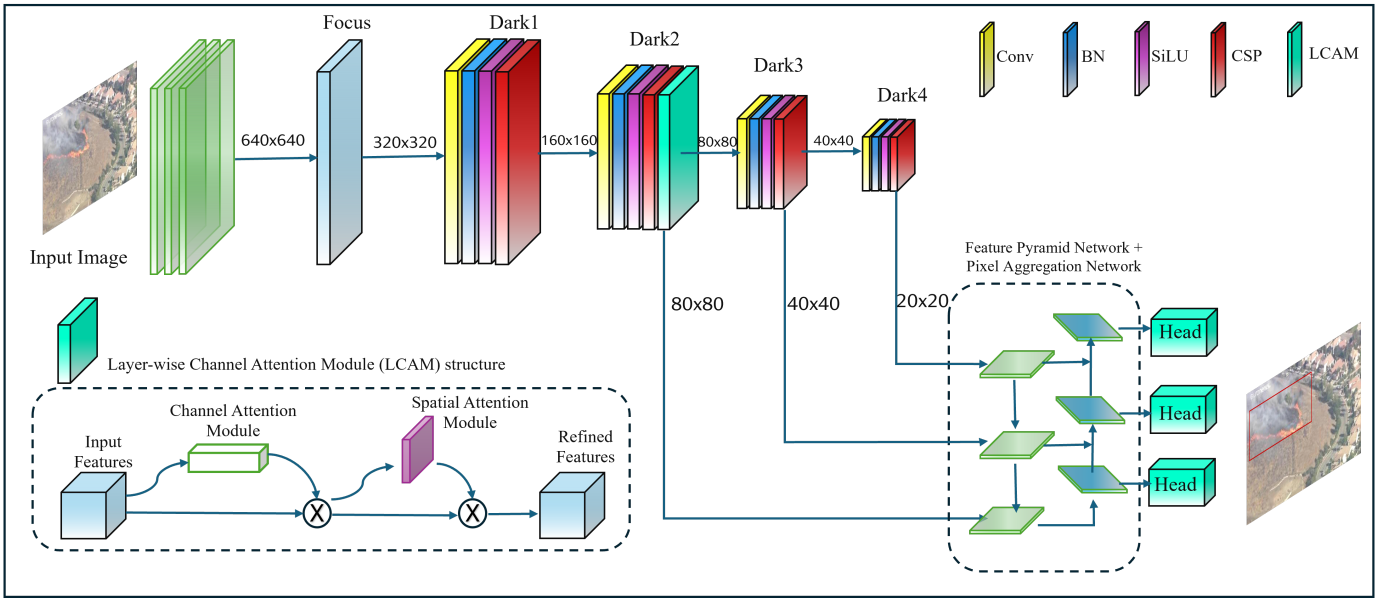


Fig. 2: Architecture of the LCAM-YOLOX model for wildfire detection. The model integrates Layer-wise Channel Attention Modules within the backbone to enhance feature extraction.

FPGAs for UAV applications has shown promising results. Malle et al. [19] and Kolpakov et al. [20] demonstrated the benefits of FPGA acceleration in various UAV perception tasks, highlighting improvements in latency and power consumption. Moreac et al. [21] introduced dynamic partial FPGA reconfiguration for UAV tasks, while Kaziha et al. [22] showed significant speedups using FPGA-accelerated neural networks optimized with genetic algorithms. These advances in FPGA acceleration, combined with techniques like map-reduce processing [23], have paved the way for the efficient implementation of complex vision algorithms on UAV platforms.

Recent advancements in UAV-based systems for real-time applications have demonstrated the potential of integrating advanced machine learning techniques to enhance performance. Ye et al. [24] propose RTD-Net, a real-time object detection network combining CNN and transformer models, leveraging a lightweight backbone and attention prediction head (APH) to improve detection speed and accuracy for small objects. Deng et al. [25] introduce a multi-camera-based indoor testbed for UAVs, enabling precise 3D tracking and control using smart cameras and an extended Kalman filter, highlighting its applications in scientific research and robotics. In dynamic environments with wind disturbances, Ma et al. [26] employ a reinforcement learning-based PRSW control method for UAV tracking, achieving robustness and superior learning effectiveness. Structural inspections have benefited from TinyML integration, as demonstrated by Zhang et al. [27], who utilize MobileNetV1 x0.25 for crack detection in concrete structures, achieving a high F1-score of 0.76 with minimal impact on UAV flight time. Similarly, Samanta et al. [28] propose the TinyAerialNet model for on-device aerial image classification on the ESP32 CAM board, achieving 88% accuracy on the AIDER dataset with low power consumption. For small and dense object detection, Ye et al. [29] present GLF-Net, which combines multiscale feature fusion and rotated regional proposal networks, achieving 86.52% mAP on the RO-UAV dataset. These works underline the progress in real-time detection, structural inspection, and energy-efficient models, aligning with our study's objectives in wildfire detection and hardware optimization.

These studies underscore the growing interest in combining advanced ML algorithms with hardware acceleration to enhance UAV capabilities for real-time fire detection and other applications. However, challenges remain in optimizing deep learning models for deployment on resource-constrained hardware without compromising detection accuracy.

Our work differentiates itself by developing an optimized LCAM-YOLOX model tailored for wildfire detection and deploying it on FPGA platforms, including the Xilinx Kria KV260, ZCU104, and ZCU102. By integrating a Layer-wise Channel Attention Module and employing quantization techniques, we address the computational and power constraints inherent in UAV platforms. This approach achieves high detection accuracy and real-time performance, advancing the state-of-the-art in UAV-based wildfire monitoring.

III. Methodology

In this section, we present our methodology for real-time wildfire detection using UAVs equipped with FPGA-accelerated deep learning models. We introduce the LCAM-YOLOX model, an enhanced version of YOLOX incorporating a Layer-wise Channel Attention Module (LCAM) to improve detection accuracy and robustness in complex wildfire scenarios. The model is optimized for deployment on FPGA platforms through quantization and efficient compilation processes.

A. LCAM-YOLOX Model

The proposed LCAM-YOLOX model integrates a hybrid domain attention mechanism within the YOLOX architecture to enhance its capacity for extracting multi-scale features crucial for wildfire detection. By incorporating LCAM into the backbone network, the model focuses on important features across different layers, improving detection capabilities and reducing false alarms.

An overview of the LCAM-YOLOX architecture is illustrated in Figure 2. The model processes high-resolution input images, and extracts features through a series of convolutional layers, attention modules, and feature pyramids to detect wildfires accurately.

1) Model Architecture: The LCAM-YOLOX model consists of the following key components:

1) **Input Image**: The model accepts a high-resolution input image of size $640 \times 640 \times 3$.
2) **Backbone Network**: The backbone includes several layers:
   - **Focus Layer**: Reduces the spatial dimensions by concatenating slices of the input, resulting in a tensor of size $320 \times 320 \times 12$.
   - **Dark Layers (Dark1 to Dark4)**: A series of convolutional layers and residual blocks that progressively downsample the feature maps while increasing the depth, extracting hierarchical features at different scales. The output sizes are $160 \times 160 \times 64$ (Dark1) down to $20 \times 20 \times 512$ (Dark4).
   - **Layer-wise Channel Attention Module (LCAM)**: Embedded within the backbone, LCAM enhances feature representation by applying attention weights to each channel in a layer-wise manner, allowing the network to focus on informative features relevant to wildfire detection.
3) **Neck Network**: Combines features from different scales using a Feature Pyramid Network (FPN) and a Path Aggregation Network (PAN) to improve the detection of objects of various sizes.
4) **Detection Heads**: Three detection heads correspond to different scales (80×80, 40×40, 20×20), enabling the detection of small, medium, and large wildfires by processing the aggregated features and predicting bounding boxes, objectness scores, and class probabilities.

2) Layer-wise Channel Attention Module (LCAM): The LCAM enhances the model's ability to focus on important features by applying attention mechanisms at each layer. This approach allows the network to adaptively recalibrate channel-wise feature responses, emphasizing informative features and suppressing less useful ones.

For an input feature map $\mathbf{F} \in \mathbb{R}^{C \times H \times W}$, where C, H, and W denote the number of channels, height, and width, respectively, the LCAM computes channel attention as follows:

$$\mathbf{M}_c = \sigma\left(\mathrm{MLP}\left(\mathrm{AvgPool}(\mathbf{F})\right) + \mathrm{MLP}\left(\mathrm{MaxPool}(\mathbf{F})\right)\right) \quad (1)$$

where:

- $\mathrm{AvgPool}(\mathbf{F})$ and $\mathrm{MaxPool}(\mathbf{F})$ are global average and max pooling operations applied along spatial dimensions, producing vectors of size $\mathbb{R}^{C \times 1 \times 1}$.
- MLP is a Multi-Layer Perceptron consisting of a bottleneck structure with one hidden layer. It reduces dimensionality to $\mathbb{R}^{C/r \times 1 \times 1}$ (where r is the reduction ratio) and then expands back to $\mathbb{R}^{C \times 1 \times 1}$.
- σ denotes the sigmoid activation function.

The attention weights $\mathbf{M}_c$ are applied to the input feature map $\mathbf{F}$ to produce the refined feature map $\mathbf{F}'$:

$$\mathbf{F}' = \mathbf{M}_c \otimes \mathbf{F} \quad (2)$$

where $\otimes$ denotes element-wise multiplication broadcasted across spatial dimensions.

This process enhances the representation of $\mathbf{F}$ by focusing on the most informative channels, thus improving the model's ability to detect wildfires in complex backgrounds.

3) Hybrid Domain Attention: The hybrid domain attention structure in LCAM-YOLOX combines channel and spatial attention mechanisms to further improve feature representation. While LCAM focuses on channel-wise attention, we also incorporate spatial attention to emphasize relevant regions within the feature maps.

The spatial attention module computes attention weights $\mathbf{M}_s \in \mathbb{R}^{1 \times H \times W}$ as:

$$\mathbf{M}_s = \sigma\left(f^{k \times k}\left([\mathrm{AvgPool}(\mathbf{F}'), \mathrm{MaxPool}(\mathbf{F}')]\right)\right) \quad (3)$$

where:

- $[\cdot, \cdot]$ denotes channel-wise concatenation.
- $f^{k \times k}$ is a convolution operation with a kernel size of $k \times k$ (typically $k = 7$).

The final refined feature map $\mathbf{F}''$ is obtained by applying spatial attention:

$$\mathbf{F}'' = \mathbf{M}_s \otimes \mathbf{F}' \quad (4)$$

This hybrid attention mechanism allows the model to focus on both the most informative channels and spatial regions, enhancing its capacity to detect wildfires across multiple scales and reducing false alarms.

The LCAM-YOLOX model employs a composite loss function to optimize detection performance. The total loss $\mathcal{L}_{\text{total}}$ is:

$$\mathcal{L}\text{total} = \lambda\text{loc}\mathcal{L}\text{CIoU} + \lambda\text{conf}\mathcal{L}\text{conf} + \lambda\text{cls}\mathcal{L}_{\text{cls}} \quad (5)$$

where λ_{loc}, λ_{conf}, and λ_{cls} are weighting factors for the localization, confidence, and classification losses, respectively.

a) Complete IoU Loss ($\mathcal{L}$CIoU):: Used for bounding box regression, CIoU loss considers the overlap area, center point distance, and aspect ratio between the predicted box $\mathbf{b}$ and the ground truth box $\mathbf{b}$gt:

$$\mathcal{L}\text{CIoU} = 1 - \text{IoU} + \frac{\rho^2(\mathbf{b}, \mathbf{b}\text{gt})}{c^2} + \alpha v \quad (6)$$

where:

- IoU is the Intersection over Union between $\mathbf{b}$ and $\mathbf{bgt}$.
- $\rho(\mathbf{b}, \mathbf{bgt})$ is the Euclidean distance between the centers of $\mathbf{b}$ and $\mathbf{bgt}$.
- c is the diagonal length of the smallest enclosing box covering both $\mathbf{b}$ and $\mathbf{bgt}$.
- v measures the similarity of aspect ratios.
- α is a positive trade-off parameter.

b) Confidence Loss ($\mathcal{L}_{conf}$):: Focal Loss is used to address class imbalance by down-weighting easy negatives:

$$\mathcal{L}_{\text{conf}} = -\alpha_t (1 - p_t)^{\gamma} \log(p_t) \tag{7}$$

where:

- p_t is the predicted confidence score.
- α_t balances the importance of positive and negative examples.
- γ is the focusing parameter that adjusts the rate at which easy examples are down-weighted.

c) Classification Loss ($\mathcal{L}_{cls}$):: Binary Cross-Entropy Loss is used for classifying whether an object is a wildfire:

$$\mathcal{L}_{\text{cls}} = -[y \log(p) + (1 - y) \log(1 - p)] \tag{8}$$

where y is the ground truth label and p is the predicted probability.

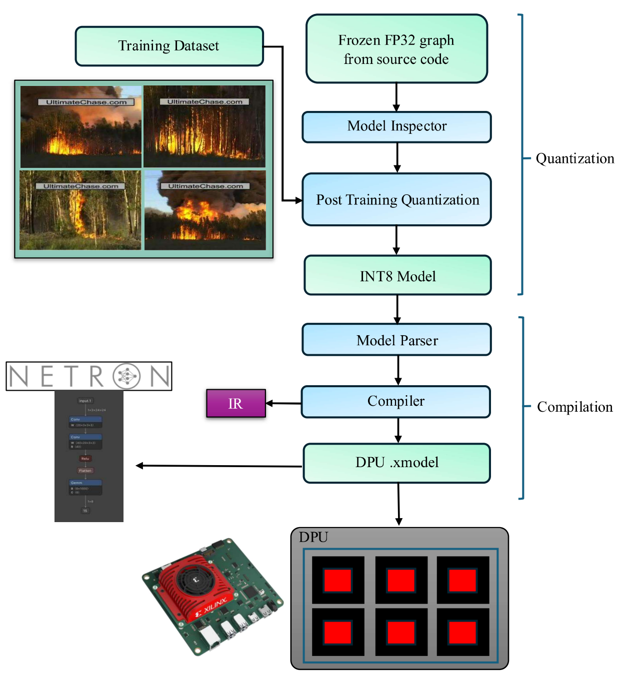


Fig. 3: Quantization flow chart illustrating the process of converting a trained model for deployment on FPGA hardware.

4) Soft Non-Maximum Suppression (Soft-NMS): To handle multiple overlapping detections and reduce false positives, we employ Soft Non-Maximum Suppression (Soft-NMS) within the LCAM-YOLOX model. Unlike traditional NMS, which discards overlapping boxes outright, Soft-NMS reduces the confidence scores of neighboring detections based on their overlap:

$$s_i = s_i \times \prod_{j=1}^{N} f(\text{IoU}_{ij}) \tag{9}$$

where:

- s_i is the confidence score of detection i.
- IoU_{ij} is the Intersection over Union between detections i and j.
- $f(\text{IoU}_{ij})$ is a decay function, typically $f(\text{IoU}_{ij}) = e^{-(\text{IoU}_{ij})^2/\sigma}$.

By retaining detections with lower confidence in overlapping regions, Soft-NMS is particularly effective in scenarios with closely located fire instances, such as dense wildfire regions. This approach enhances the model's reliability by improving recall while maintaining high precision.

5) Comparison with Other Models: We compared the LCAM-YOLOX model with YOLOv3, YOLOv5, and YOLOX-m to evaluate its performance. As shown in Table IV in Section V the LCAM-YOLOX model demonstrates improved detection accuracy and reduced false alarms, validating the effectiveness of the hybrid attention mechanism and architectural enhancements.

6) Notations and Definitions: The key notations used in this section are summarized in Table I.

TABLE I: Notations and Definitions

Notation	**Definition**
$\mathbf{F}$	Input feature map
C, H, W	Number of channels, height, and width of $\mathbf{F}$
$\mathbf{M}_c$	Channel attention weights from LCAM
$\mathbf{M}_s$	Spatial attention weights
$\otimes$	Element-wise multiplication
σ	Sigmoid activation function
MLP	Multi-Layer Perceptron
$\mathcal{L}_{\text{total}}$	Total loss function
$\mathcal{L}_{\text{CIoU}}$	Complete IoU loss for localization
$\mathcal{L}_{\text{conf}}$	Confidence loss (Focal Loss)
$\mathcal{L}_{\text{cls}}$	Classification loss (Binary Cross-Entropy)
IoU	Intersection over Union
$\rho(\mathbf{b}, \mathbf{b}_{\text{gt}})$	Center distance between predicted and ground truth boxes
c	Diagonal length of the smallest enclosing box
v, α	Aspect ratio term and trade-off parameter in CIoU loss
p_t, α_t, γ	Parameters in Focal Loss
y, p	Ground truth label and predicted probability
s_i, IoU_{ij}	Confidence score and IoU for Soft-NMS
$f(\cdot)$, σ	Decay function and decay rate in Soft-NMS

B. Quantization Process

To optimize the LCAM-YOLOX model for deployment on FPGA platforms, we employed an 8-bit integer (INT8) quantization process. Quantization reduces model size and computational complexity, enabling efficient inference on resource-constrained hardware while maintaining high accuracy.

The quantization flow, illustrated in Figure 3, involves the following steps:

1) **Quantization**: The trained floating-point model is converted to INT8 format using post-training quantization. A calibration dataset is used to determine optimal scaling factors for weights and activations.
2) **Compilation**: The quantized model is parsed and compiled into a hardware-compatible format (.xmodel file) using tools like Xilinx Intermediate Representation (XIR) and the Vitis AI compiler.
3) **Deployment**: The compiled model is loaded onto the FPGA's Deep Learning Processing Unit (DPU) for efficient inference on the target embedded device.

1) Quantization Details: The quantization process maps floating-point values to integer values using a linear transformation:

$$q = \text{round}\left(\frac{x - \beta}{s}\right) \tag{10}$$

where:

- x is the floating-point value.
- q is the quantized integer value.
- s is the scale factor.
- β is the zero-point offset to handle asymmetric quantization.

The dequantization reconstructs the approximate floating-point value:

$$\hat{x} = sq + \beta \tag{11}$$

By quantizing both weights and activations to INT8, we achieve significant reductions in model size and computational load, facilitating real-time inference on FPGA hardware without substantial loss in accuracy.

C. Compilation Process

The quantized model is transformed into an Intermediate Representation (IR) suitable for deployment on FPGA hardware. The IR tool flow comprises of four main libraries:

1) **Graph Library**: Acts as the central processing hub, integrating data from subgraph, operator, and tensor libraries to construct a comprehensive graph representation.
2) **Subgraph Library**: Defines manageable parts of the computational graph by partitioning it into subgraphs, enabling efficient optimization and execution.
3) **Operator Library**: Provides a collection of predefined operations (e.g., convolution, pooling) necessary for model execution.
4) **Tensor Library**: Manages data structures (tensors) that hold the model's inputs, outputs, and intermediate computations.

The compilation process involves:

1) **Parsing**: The quantized model is parsed into the IR format, eliminating framework-specific differences and providing a unified representation.
2) **Optimization**: The computational graph is optimized, including operator fusion (e.g., combining convolution and batch normalization).
3) **Subgraph Partitioning**: The graph is partitioned into subgraphs based on the capabilities of the target DPU, ensuring that only supported operations are mapped to hardware.
4) **Instruction Generation**: DPU instructions are generated for the subgraphs and attached to the IR graph.
5) **Serialization**: The optimized graph is serialized into a binary file (.xmodel) compatible with the FPGA hardware.

This process ensures that the model is efficiently converted into a format optimized for high-performance inference on the FPGA, enabling real-time wildfire detection on UAV platforms.

D. Deployment on FPGA Platforms

The final step involves deploying the compiled model onto FPGA platforms such as the Xilinx Kria KV260, ZCU104, and ZCU102, which are based on the Zynq UltraScale+ architecture. These platforms provide varying levels of FPGA fabric logic resources, allowing flexibility in accommodating different model sizes and computational requirements.

The deployment process includes:

1) **Model Loading**: The .xmodel file is loaded onto the FPGA, configuring the DPU for inference.
2) **Integration**: The model is integrated with the system's software stack, including pre-processing and post-processing pipelines.
3) **Inference**: Real-time inference is performed on input images captured by the UAV's camera, utilizing the FPGA's parallel processing capabilities for efficient computation.

Tools like Netron are used for model inspection, ensuring compatibility and correctness before deployment. By leveraging FPGA acceleration, we achieve significant improvements in inference speed and energy efficiency compared to traditional GPU-based approaches, making our system suitable for resource-constrained UAVs.

IV. Experimental Setup and Implementation Details

In this section, we describe the datasets used for training and testing, the implementation of our system, and the experimental setup employed to evaluate the performance of the proposed LCAM-YOLOX model on different FPGA platforms for real-time wildfire detection.

A. Datasets

We utilized the Foggia dataset [30], a widely recognized benchmark for fire and smoke detection algorithms. The dataset comprises 31 videos, with 14 containing fire instances and 17 featuring red objects and smoke to test false positive rates. The videos vary in resolution from 320×240 to 400×256 pixels and have frame rates between 9 and 29 FPS, simulating real-world variability. Each video contains between 80 and 6097 frames, providing a broad temporal scale for robust training and testing.

To improve the generalization ability of the model and enhance its robustness across diverse real-world wildfire scenarios, we supplemented the Foggia dataset with a custom-generated dataset. This custom dataset was constructed by incorporating images from three main sources:

- **Internet-Sourced Data:** Images were collected from publicly available online repositories [31]–[39] to capture varied fire conditions, including differences in fire size, intensity, and backgrounds (e.g., urban vs forest).
- **UAV-Based Aerial Imagery:** Aerial images were captured using UAVs equipped with multispectral and RGB cameras, providing unique perspectives, especially in large-scale fires and smoke-obscured areas. These images were collected across varying altitudes and angles.
- **Data Augmentation:** To further enhance dataset diversity, we applied augmentation techniques such as random rotations, cropping, flipping, varying lighting conditions, and adding synthetic smoke overlays. These augmentations ensured that the dataset adequately represented real-world scenarios involving varying illumination, smoke density, and fire severity.

The combined dataset spans diverse wildfire conditions, including low-light settings, high smoke density, varied fire intensities, and occlusions. These scenarios enhance the model's robustness and improve its ability to generalize to unseen environments, reducing false positives and improving detection accuracy in challenging conditions.

Our proposed model utilizes the custom wildfire dataset for training and object detection tasks, employing a class-based approach to categorize fire incidents. The classification ranges from Class 1 to Class 5, with higher classes representing more severe and widespread fires. The model also utilizes smoke as an auxiliary parameter to determine the fire's classification, with the intensity and volume of smoke being directly correlated to higher fire severity classes. This helps in segregating the fires based on their severity.

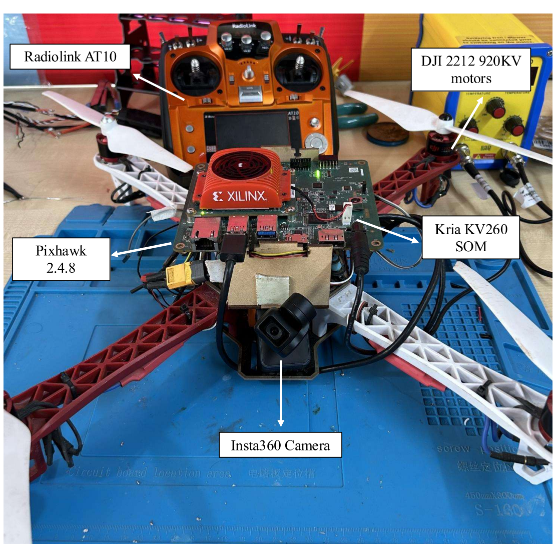


Fig. 4: FPGA-based UAV platform for real-time wildfire detection.

B. Implementation

Our implementation is based on three Xilinx FPGA platforms: the Kria KV260 System on Module (SOM), ZCU104, and ZCU102 development boards, all serving as onboard hardware accelerators for our UAV platform. These boards integrate key components that facilitate real-time video processing and model inference, each platform contributing to optimizing model performance and ensuring robustness in wildfire detection scenarios. The communication between UAVs and ground stations is secured using standard authentication protocols [40]. However, we decide to use the KV260 as the onboard hardware accelerator due to its performance-per-watt ratio. It is integrated into several key components of our UAV platform:

- **Insta360 USB Camera**: Provides real-time video input to the KV260 for wildfire detection.
- **Pixhawk 2.4.8 Microcontroller Module**: Connects to environmental sensors and controls the UAV's motion semi-autonomously through the KV260.

An image of the FPGA-based UAV platform is shown in Figure 4.

The KV260 functions as the onboard computer, handling real-time video processing and control tasks. We initially set up a test bench to validate the integration of the camera and the LCAM-YOLOX model for fire detection. Subsequently, we integrated the Pixhawk controller to provide full motion capabilities for the UAV.

C. Experimental Setup and Configuration

We evaluated the performance of the LCAM-YOLOX model on the KV260 FPGA platform by setting up a testbench which involved comparing it with state-of-the-art object detection models such as YOLOv3, YOLOv5, and YOLOX-m. The models were trained using the same configuration parameters as that for LCAM YOLOX listed in Table II.The codebase, along with detailed documentation on experimental setups, hardware configurations, and software dependencies, can be accessed at GitHub Repository https://github.com/halalboro/fpga-accelerators.git

The experimental setup involves processing input images instead of the live video input from the USB Camera for simplicity. The images are passed through the LCAM-YOLOX model deployed on the KV260's DPU running on the Programmable Logic (PL) with the configuration as shown in Table III. We designed a processing pipeline consisting of pre-processing, DPU inference, and post-processing stages in which the pre-processing and post-processing stages were running on the Processing System (PS) of the KV260 while the DPU inference ran in the PL. Figure 5 illustrates the pipeline for processing a batch of three images simultaneously while

TABLE II: Training Configuration Parameters for our proposed model, LCAM YOLOX

Parameter	Value
Model	LCAM YOLOX
Batch Size	8
Optimizer	SGD
Learning Rate	0.02
Momentum	0.937
Weight Decay	0.0005
Epochs	300

TABLE III: Deep Learning Processing Unit (DPU) Configuration Parameters

Parameter	Configuration
Core Architecture	
Component Name	DPUCZDX8G
Number of DPU Cores	3
Architecture	B4096
RAM Usage	Low
Channel Augmentation	Enabled
Computational Units	
Conv ReLU Type	ReLU + LeakyReLU + ReLU6
ALU Parallel	4
ALU ReLU Type	ReLU + ReLU6
ElementWise Multiply	Enabled
AveragePool	Enabled
Number of SFM cores	1
Implementation Details	
S-AXI Clock Mode	Independent
DPU 2x Clock Gating	Enabled
DSP48 Usage	High
DSP48 Max Cascade Length	4
Ultra-RAM Use per DPU	0
Timestamp auto-update	Enabled

varying the number of DPU cores allocated—single, dual, and triple core configurations (E2E_1, E2E_2, and E2E_3, respectively).

Processing Pipeline:

Pre-processing: This stage, handled sequentially on the PS of the FPGA, includes resizing and normalization. Resizing ensures uniform input dimensions of 640×640 pixels, necessary for model compatibility and optimized DPU processing, while retaining critical image features across diverse resolutions in the custom dataset. Normalization scales pixel values to [0, 1], reducing variability from lighting and intensities, improving feature extraction, and enhancing quantization robustness. These operations standardize the input format, ensuring consistent pre-processing time across configurations (Pre_1, Pre_2, Pre_3) and contributing to the model's accuracy and efficiency.

DPU Inference: The DPU cores on the Programmable Logic (PL) of the FPGA process the batch of images. In DPU_1, a single DPU core is allocated, while DPU_2 and DPU_3 have dual and triple DPU cores, respectively. Increasing the DPU core count does not change the inference time per image but improves throughput by processing multiple images concurrently. Thus, the batch processing time decreases with more DPU cores due to parallelism.

Post-processing: After inference, each image undergoes sequential post-processing on the PS. This stage includes tasks such as decoding, bounding box generation, and non-maximum suppression. The post-processing time per image remains consistent across configurations (Post_1, Post_2, Post_3), similar to pre-processing, as it is handled independently on the PS.

V. Results and Discussion

In this section, we present the experimental results of our proposed LCAM-YOLOX model for real-time wildfire detection on FPGA-accelerated UAV platforms. We compare the performance of LCAM-YOLOX with other state-of-the-art networks, analyze resource utilization and power consumption across different FPGA devices, and discuss the trade-offs between performance, cost, and energy efficiency. Our analysis emphasizes the suitability of the KV260 FPGA platform for UAV deployment due to its optimal balance of performance, power consumption, and cost.

A. Model Performance Comparison

We evaluated the performance of LCAM-YOLOX against a variety of object detection models, including YOLO series (YOLOv3, YOLOv5, YOLOX-m), lightweight detection models (MobileNetV2, EfficientDet), and classical detection algorithms (Faster R-CNN, RetinaNet). The comparison is presented in terms of Floating Point 32-bit (FP32) accuracy, Quantized (INT8) accuracy, root mean square error (RMSE), and frames per second (FPS) performance. Table IV summarizes the results.

Model Size and Complexity:

YOLOv3, with the largest parameter count (65.2 million) and smaller input size (416×416), achieves the lowest FP32 accuracy of 62.31% and quantized accuracy of 60.76%. This demonstrates that a higher parameter count does not necessarily result in better detection accuracy, especially when the architecture lacks optimization for efficient feature extraction.

YOLOv5, despite having significantly fewer parameters (7.2 million) and a larger input size (640×640), achieves an FP32 accuracy of 74.06% and quantized accuracy of 72.77%, showcasing the effectiveness of its lightweight design and optimized structure.

MobileNetV2 and EfficientDet, both lightweight detection models, have parameter counts of 4.3M and 3.9M, respectively. While they are efficient in terms of size, their FP32 accuracies (68.23% for MobileNetV2 and 73.42% for EfficientDet) fall behind YOLOX-m and LCAM-YOLOX. Additionally, they perform poorly in FPS (13 FPS and 98 FPS), making them unsuitable for real-time UAV-based applications.

Faster R-CNN and RetinaNet, as classical object detection models with larger input sizes (800×800), achieve relatively higher FP32 accuracies (79.12% and 78.21%, respectively).

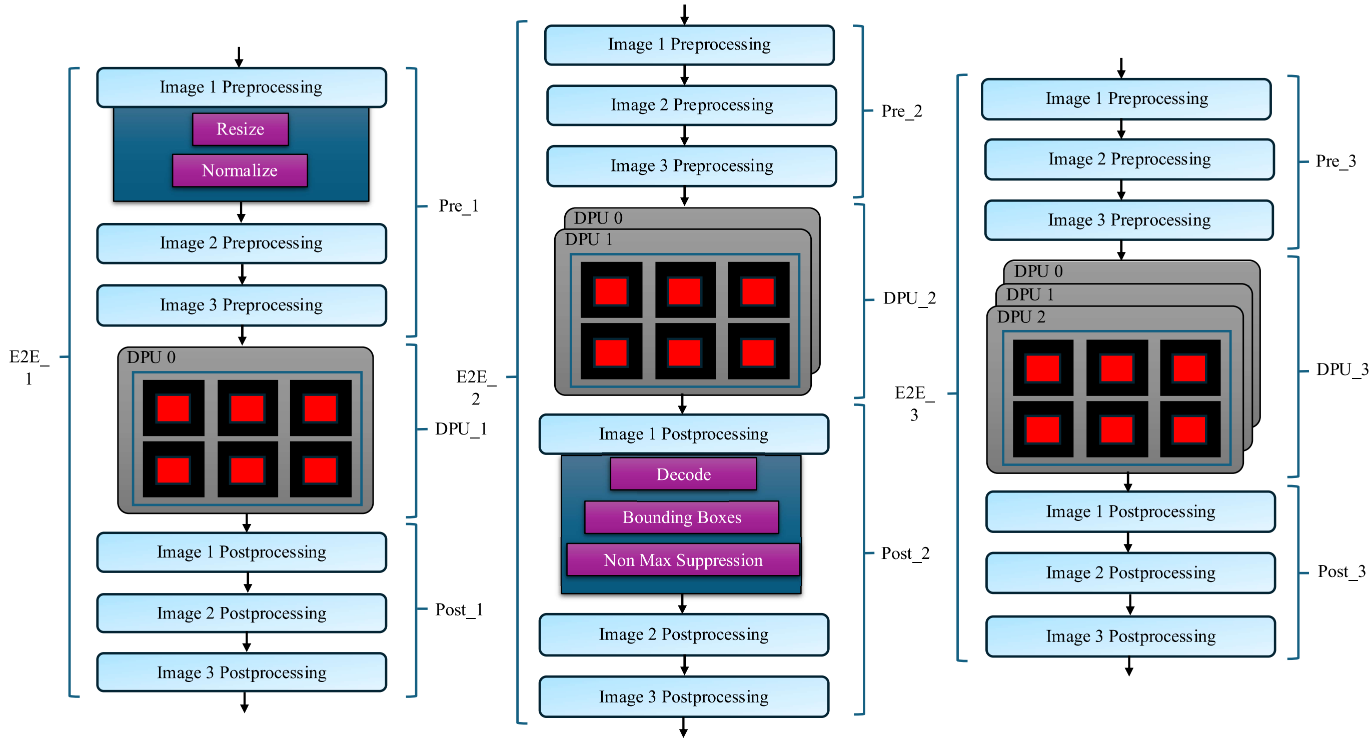


Fig. 5: Processing pipeline for a batch of three images with varying DPU core configurations.

TABLE IV: Comparison of Different Models

Network Model	Input Size	Model Parameters (M)	FP32 Accuracy (mAP)	Quantized Accuracy (mAP)	RMSE	FPS
YOLOv3	416	65.2	0.62306	0.6076	0.0842	145
YOLOv5	640	7.2	0.7406	0.7277	0.0651	182
YOLOX-m	640	25.3	0.7845	0.7694	0.0573	185
MobileNetV2	640	4.3	0.6823	0.6645	0.0912	13
EfficientDet	640	3.9	0.7342	0.7156	0.0784	98
Faster R-CNN	800	42.1	0.7912	0.7734	0.0635	12
RetinaNet	800	37.8	0.7821	0.7645	0.0678	16
LCAM YOLOX	**640**	**9.6**	**0.7989**	**0.7811**	**0.0489**	**195**

However, their FPS values are significantly lower (12 FPS for Faster R-CNN and 16 FPS for RetinaNet), which limits their feasibility for real-time deployment on resource-constrained platforms.

LCAM-YOLOX: Our proposed LCAM-YOLOX model achieves the highest performance with an FP32 accuracy of 79.89% and quantized accuracy of 78.11%. It outperforms YOLOX-m by 1.45% in FP32 accuracy while maintaining a moderate parameter count of 9.6M. The RMSE of LCAM-YOLOX is the lowest at 0.0489, further validating its precision in detection. With a real-time FPS of 195, LCAM-YOLOX achieves a strong balance between accuracy, efficiency, and computational cost, making it ideal for UAV-based wildfire detection systems.

Impact of Quantization: Quantization and pruning are critical for optimizing models on resource-constrained devices such as FPGA. As shown in Table VI, the LCAM-YOLOX model demonstrates strong resilience to these optimization techniques. Preprocessing steps, including resizing and normalization, improve the quantized model's robustness, increasing its mAP from 0.7989 to 0.8163. When quantized from FP32 to INT8 precision, the mAP slightly reduces to 0.7811, reflecting a retention of 95.7% accuracy compared to the FP32 baseline. This minimal drop highlights the robustness of LCAM-YOLOX, attributed to the Layer-wise Channel Attention Module (LCAM), which enhances feature representation and mitigates information loss during quantization. Furthermore, pruning the model by 30% reduces its weights (from 9.60M to 6.72M) and FLOPS (from 8.45G to 6.31G) while significantly boosting FPS from 76 to 195—a 2.56x improvement. These optimizations maintain accuracy while enhancing computational efficiency, making LCAM-YOLOX highly suitable for real-time UAV-based wildfire detection on FPGA platforms.

TABLE V: Ablation Study of LCAM YOLOX

Model Configuration	LCAM Module	Spatial Attention	Channel Attention	mAP	Parameters (M)
Baseline YOLOX				0.7645	9.0
YOLOX + Spatial Attention		✓		0.7723	9.2
YOLOX + Channel Attention			✓	0.7798	9.3
YOLOX + Both Attentions		✓	✓	0.7856	9.4
LCAM YOLOX (Full)	✓	✓	✓	**0.7989**	**9.6**

TABLE VI: Effect of Preprocessing, Quantization and Pruning on LCAM YOLOX

LCAM-YOLOX	Quantization(FP32 to INT8) and Pruning (30%)			Reduction
	Before	Preprocessing	After	
Weights [M]	9.60	9.60	6.72	X1.43
FLOPS [G]	8.45	8.45	6.31	X1.34
RMSE	0.0489	0.0472	0.0495	-
mAP	0.7989	0.8163	0.7811	-
FPS	76	76	195	X2.56

Overall Performance: Compared to all tested models, LCAM-YOLOX emerges as the best-performing model, balancing high detection accuracy, low RMSE, and real-time FPS performance. It significantly outperforms lightweight models (MobileNetV2, EfficientDet) in both accuracy and speed while surpassing classical models (Faster R-CNN, RetinaNet) in FPS. These results underscore the suitability of LCAM-YOLOX for UAV-based wildfire detection, where real-time performance, accuracy, and energy efficiency are critical.

B. Ablation Study of LCAM-YOLOX

To evaluate the contribution of the LCAM module to the overall performance of YOLOX, we conducted an ablation study, as presented in Table V.

Baseline YOLOX: Starting with the baseline YOLOX model, we observe an mAP of 0.7645 with 9.0M parameters.

Effect of Spatial and Channel Attention: Adding spatial attention improves the mAP to 0.7723, while adding channel attention alone achieves 0.7798. The combination of both attentions further enhances the performance to 0.7856, demonstrating their complementary effects.

LCAM (Full Integration): Finally, the full LCAM module integrates both spatial and channel attention mechanisms in a layer-wise fashion, resulting in the highest mAP of 0.7989 with a slight increase in parameters to 9.6M. This confirms the effectiveness of LCAM in enhancing the model's feature representation and detection accuracy.

C. FPGA Resource Utilization

We analyzed the resource utilization of the LCAM-YOLOX model across three FPGA devices—KV260, ZCU104, and ZCU102—with varying DPU core configurations. Table VII provides detailed utilization metrics for key resources.

Analysis of Resource Utilization:

On the KV260, LUT and FF utilization reach high levels even at 1-core configuration, indicating limited scalability beyond 2 cores. However, the BRAM and DSP utilization remain moderate, due to the availability of UltraRAM (URAM) on the device to manage the high memory bandwidth tasks.

ZCU104 shows a balanced resource utilization with moderate increases in LUT and FF usage across configurations. BRAM utilization is high in the 1-core configuration but decreases in the 2-core setup due to efficient memory allocation facilitated by enabling the URAM in the 2-core setup. It would otherwise exceed the BRAM size in a 2-core setup.

ZCU102, being the most expensive device due to its sea of resources, demonstrates lower resource utilization as expected in 1-core and 2-core configurations but reaches higher utilization at 3 cores, indicating better scalability for applications requiring more computational power.

Implications for Deployment:

The KV260's resource utilization suggests it is well-suited for applications requiring up to 2 DPU cores, making it an optimal choice for our UAV platform where size, weight, and power constraints are critical. Its efficient use of resources allows for real-time processing without exceeding the hardware capabilities.

D. Performance and Power Consumption Analysis

We compared the performance of the LCAM-YOLOX model across FPGA devices, Nvidia Jetson Nano, and STM32H7A3 using the metrics of frames per second (FPS) and peak performance in tera operations per second (TOPS). As shown in Table VIII, the KV260 achieves 195FPS with 1 core and 387FPS with 2 cores at a set DPU frequency of 300 MHz, outperforming Nvidia Jetson Nano (65 FPS, 0.56 TOPS) and STM32H7A3 (15 FPS, 0.28 TOPS) significantly. While Jetson Nano and STM32H7A3 offer cost-effective solutions, their FPS values are insufficient for high-throughput real-time wildfire detection tasks.

Performance Analysis:

TABLE VII: FPGA Resource Utilization Across Different Boards and Core Configurations

Boards	LUT (Used/Available)	FF (Used/Available)	BRAM (Used/Available)	URAM (Used/Available)	DSP (Used/Available)	LUTRAM (Used/Available)
ZCU102						
1-core	52,161/274,080 (**19.03%**)	98,249/548,160 (**17.92%**)	255/912 (**27.96%**)	–	710/2,520 (**28.17%**)	5,647/144,000 (**3.92%**)
2-core	107,237/274,080 (**39.12%**)	187,663/548,160 (**34.24%**)	512/912 (**56.14%**)	–	1,451/2,520 (**57.58%**)	12,344/144,000 (**8.57%**)
3-core	165,111/274,080 (**60.24%**)	302,294/548,160 (**55.15%**)	769/912 (**84.32%**)	–	2,138/2,520 (**84.84%**)	21,662/144,000 (**15.04%**)
ZCU104						
1-core	63,003/230,400 (**27.35%**)	107,833/460,800 (**23.40%**)	259/312 (**83.01%**)	–	718/1,728 (**41.55%**)	–
2-core	95,724/230,400 (**41.55%**)	188,110/460,800 (**40.82%**)	154/312 (**49.36%**)	96/96 (**100%**)	718/1,728 (**41.55%**)	–
KV260						
1-core	65,139/117,120 (**55.62%**)	108,532/234,240 (**46.33%**)	19/144 (**13.19%**)	42/64 (**65.62%**)	546/1,248 (**43.75%**)	–
2-core	84,511/117,120 (**72.15%**)	167,544/234,240 (**71.52%**)	57/144 (**39.58%**)	64/64 (**100%**)	918/1,248 (**73.56%**)	–

TABLE VIII: Performance and Cost Analysis of LCAM-YOLOX Model on Different Boards with Various DPU Configurations

Board	Device	DPU Cores	Performance (FPS)				Single Core Power (W)	Peak Performance (TOPS)	Cost ($)
			1C	2C	3C	4C			
KV260	XCK26 Ultrascale+	Up to 2	195	387	N/A	N/A	9.72	1.23	400
ZCU104	ZU7EV Ultrascale+	Up to 2	193	337	N/A	N/A	12.82	2.46	1,200
ZCU102	ZU9 Ultrascale+	Up to 3	185	319	481	N/A	20.53	3.45	3,000
NVIDIA Jetson Nano	Maxwell™ architecture	128 cores	65 FPS				8.61	0.56	200
STM32H7A3	Arm Cortex-M7 Processor	1 core	15 FPS				0.7	0.28	45

At a set DPU frequency of 300MHz, the KV260 achieves 195FPS with 1 core and 387FPS with 2 cores. ZCU104 yields similar performance with 193FPS (1-core) and 337FPS (2-core). ZCU102, supporting up to 3 cores, achieves a remarkable 481FPS at 3 cores. In comparison, the Nvidia Jetson Nano, with its 128-core Maxwell Architecture, achieves 65FPS and a peak performance of 0.56 TOPS, which is significantly lower than the FPGA platforms in terms of throughput and energy efficiency.

Power Consumption and Efficiency:

The KV260 consumes 9.72 W for single-core DPU workloads and 10.45 W at maximum DPU utilization (measured on *OWON P4305 Programmable Lab DC Power Supply*). ZCU104 consumes 13.20 W at maximum load, while the ZCU102 shows 22.13 W under similar conditions. Notably, the KV260 exhibits the highest performance per watt under both single-core and multi-core configurations, making it the most energy-efficient option among the tested devices.

The power consumption values reported in Table VIII were derived from actual on-board measurements using *OWON P4305*, while Figure 8 provides simulation-based power breakdowns using Vivado Power Analysis. Minor discrepancies between these datasets arise due to Vivado's conservative estimation approach, which does not account for real-world factors such as switching spikes and environmental influences.

Implications for UAV Deployment:

Considering the real-time video feed is typically limited to 60FPS, the KV260's single-core performance of 195FPS is sufficient for real-time processing requirements. Its low power consumption and high energy efficiency make it ideal for UAV applications where power resources are limited. For scenarios requiring multiple camera feeds or higher processing demands, boards like the ZCU102 may be considered despite their higher power consumption.

E. Scalability and Real-Time Processing

Figure 6 illustrates the performance (FPS) of the FPGA devices with varying DPU configurations.

The KV260, even in its single-core configuration, comfortably exceeds the real-time requirement of 60FPS,

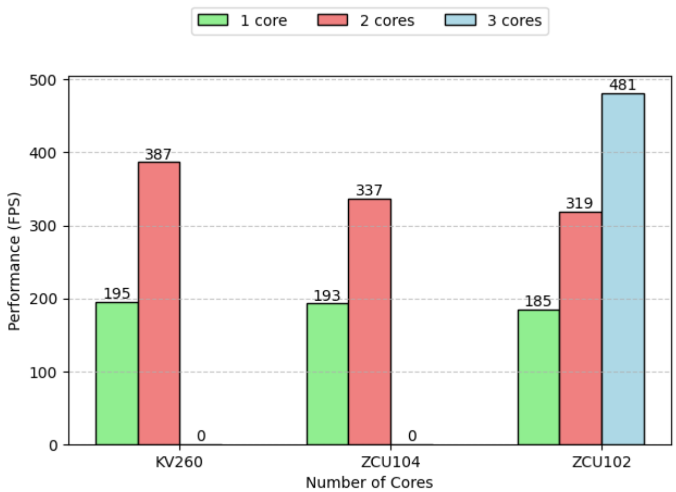


Fig. 6: Performance of FPGA devices with varying DPU configurations.

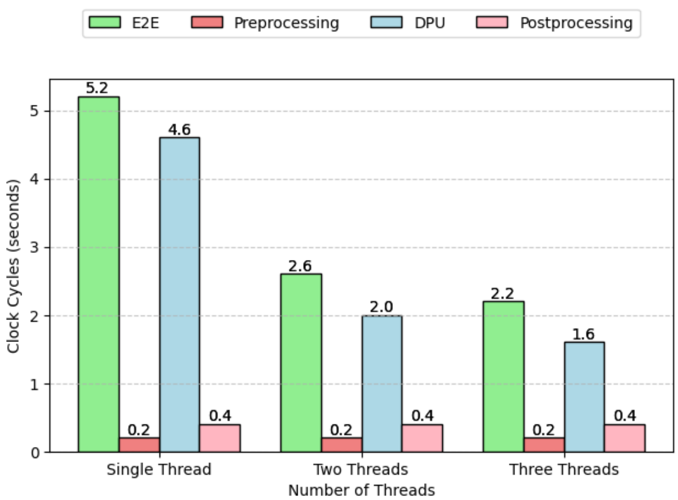


Fig. 7: Clock cycle analysis across different thread configurations.

handling real-time video feeds effectively. The ability to process at 195FPS provides headroom for additional computational tasks to occur simultaneously or even handle higher-resolution video inputs. When multiple camera feeds are necessary, higher-performance boards that support multiple DPU cores very efficiently, like the ZCU102 or the ZCU104, can be utilized, although with the added weight of increased power consumption and cost.

F. Impact of Multi-threading on Processing Time

We analyzed the effect of increasing the number of threads and DPU cores on the end-to-end (E2E) execution time. Figure 7 illustrates the processing pipeline for a batch of three images with varying threads.

The E2E processing time decreases significantly with the addition of threads and DPU cores due to parallel processing, particularly during DPU inferencing. However, pre-processing and post-processing times remain consistent, as they are handled sequentially on the Processing System (PS) of the FPGA device. This suggests that optimizing these stages could provide scope for further enhancement of overall performance.

G. Power Consumption Analysis

KV260: Consumes 10.45 W, with dynamic power constituting 93% of total consumption. Its high energy efficiency makes it ideal for UAV applications.

ZCU104: Consumes 13.20 W, balancing performance and power consumption effectively.

ZCU102: Has the highest power consumption at 22.13 W, making it less suitable for power-constrained UAV deployments.

Performance per Watt:

The KV260 outperforms the other boards in terms of performance per watt, achieving up to 37.15 FPS/W with 2 DPU cores under actual measurements. This highlights its suitability for energy-efficient real-time processing on UAV platforms.

H. Discussion

The experimental results demonstrate that the proposed LCAM-YOLOX model achieves superior detection accuracy while maintaining computational efficiency, making it highly suitable for deployment on FPGA-accelerated UAV platforms. Among the tested devices, the KV260 FPGA emerges as the optimal choice due to the following advantages:

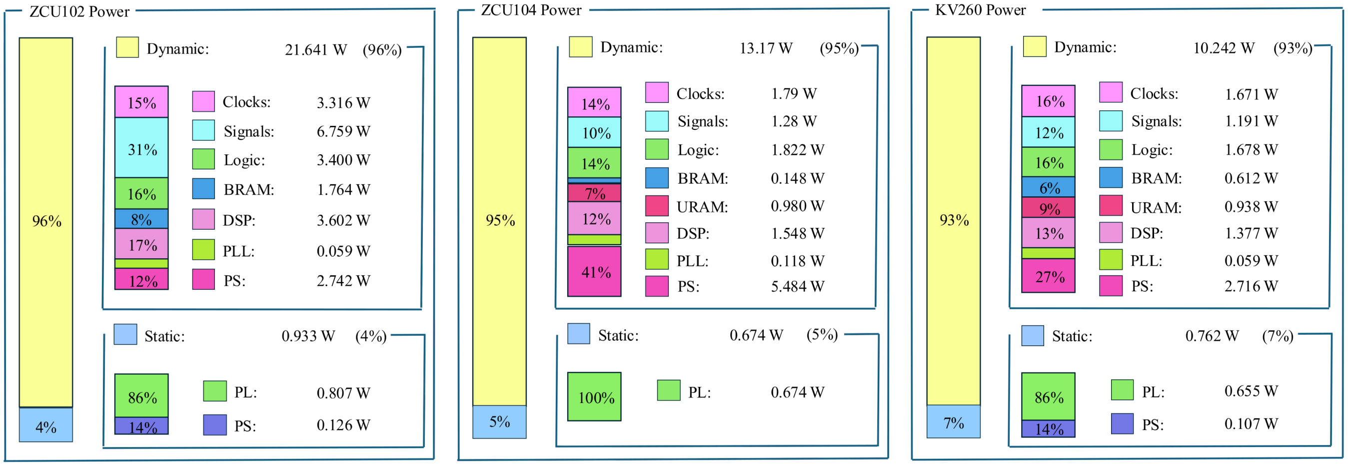
High Performance per Watt: The KV260 delivers exceptional energy efficiency, achieving up to 37.15FPS/W while consuming only 10.45W under real-world measurements. This minimal power consumption is ideal for energy-constrained UAV missions, where extended operational time is critical for real-time wildfire detection.

Power Analysis Discrepancies: Power consumption estimates from Vivado simulations (Fig. 8) are intentionally conservative, as they do not account for non-ideal conditions, switching spikes, or external factors. Actual consumption, as measured with *OWON P4305*, reflects these real-world conditions, providing more accurate data for deployment considerations.Table IX highlights these differences

TABLE IX: Comparison of Power Consumption: Vivado Simulation vs. Onboard Measurement

FPGA Board Configuration	Vivado Simulation (W)	Onboard Measurement (W)
ZCU102 (4 DPU Cores, 300 MHz)	21.64	22.13
ZCU104 (2 DPU Cores, 300 MHz)	13.17	13.20
KV260 (2 DPU Cores, 300 MHz)	10.24	10.45

Sufficient Processing Capability: At a single-core configuration, the KV260 comfortably exceeds the 60 FPS threshold required for real-time processing of a single-camera



(a) ZCU102 Power Consumption **(b) ZCU104 Power Consumption** **(c) KV260 Power Consumption**

Fig. 8: Power consumption from vivado power estimator reports at max DPU core configuration.

video feed. This ensures seamless operation with minimal latency, essential for live wildfire monitoring and detection.

Trade-offs with Higher-Performance Boards: While the ZCU102 and ZCU104 offer superior computational throughput—up to 481FPS with 3 cores on ZCU102—they come with significant trade-offs. The ZCU102, for example, consumes 22.13 W and costs $3,000, making it less suitable for power-limited UAV missions. However, these boards are better suited for specialized applications such as multi-camera real-time surveillance or tasks requiring extremely high computational throughput, where accuracy and latency take precedence over energy efficiency.

Impact of Preprocessing on Accuracy: The pre-processing pipeline, including resizing and normalization, plays a critical role in ensuring the model's robustness and performance. Resizing aligns the input images (from varied dimensions) to the model's required 640×640 input size, maintaining structural integrity and consistency. Normalization further improves the quantized model's robustness, enhancing mAP from 0.7989 to 0.8163, as shown in Table VI. This step mitigates quantization artifacts and ensures reliable detection accuracy across diverse environmental conditions.

Practical Applications: The KV260 platform is highly suitable for single UAV deployments tasked with real-time wildfire detection, where power efficiency and cost constraints are critical. For applications demanding higher computational throughput—such as multi-camera systems in disaster response coordination or urban surveillance—the ZCU102 and ZCU104 platforms offer significant advantages. Overall, the LCAM-YOLOX model, when paired with the KV260 FPGA, strikes an optimal balance between accuracy, energy efficiency, and affordability, making it a competitive solution for UAV-based wildfire detection in real-world deployment scenarios.

VI. Conclusion

In this paper, we presented the LCAM-YOLOX model, an enhanced object detection framework incorporating a Layer-wise Channel Attention Module (LCAM) to improve wildfire detection accuracy in UAV applications. Our extensive experiments demonstrated that LCAM-YOLOX outperforms other state-of-the-art models, achieving the highest quantized mAP of 78.11% while maintaining a moderate parameter count of 9.6 million. When deployed on the Xilinx Kria KV260 FPGA platform, the model not only exceeded real-time processing requirements with a single-core performance of 195 FPS but also offered the highest performance per watt ratio—up to 37.15 FPS/W under real-world measurements—compared to other tested boards. These findings validate the KV260's optimal balance of high detection accuracy, energy efficiency, and cost-effectiveness, making it an ideal choice for UAV-based real-time wildfire detection systems, particularly in resource-constrained environments where power and weight are critical considerations.

Limitations and Future Work:

While this study optimizes performance for single-camera input at 640×640 resolution, the model remains adaptable to higher-resolution inputs through preprocessing, ensuring real-time efficiency. Future work will explore native support for higher resolutions, analyzing the trade-offs between computational cost and accuracy. Additionally, we aim to integrate super-resolution techniques to enhance detection capabilities for high-resolution UAV-mounted cameras.

To improve scalability, we plan to evaluate multi-camera setups and extend dataset diversity by incorporating more complex wildfire scenarios, such as variations in lighting, smoke density, and fire intensity. Hybrid quantization strategies and dynamic frequency scaling will be explored to optimize power management. Furthermore, we will investigate comparisons against Transformer-based object detection models and specialized small-target detection algorithms. Expanding LCAM-YOLOX to alternative edge platforms and low-power microcontrollers remains a priority to ensure its versatility across UAV-based real-time detection applications.

References

[1] K. Choutri, M. Lagha, S. Meshoul, M. Batouche, F. Bouzidi, and W. Charef, "Fire detection and geo-localization using uav's aerial images and yolo-based models," *Applied Sciences*, 2023.

[2] R. Arora, M. Khosravi, S. Enayati, and H. Pishro-Nik, "Deployment of a uav-based fire detection system," *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, pp. 1–6, 2023.

[3] M. Zhou, L. Wu, S. Liu, and J. Li, "Uav forest fire detection based on lightweight yolov5 model," *Multimedia Tools and Applications*, pp. 1–12, 2023.

[4] M. Jiang, Y. Wang, F. Yu, T. Peng, and X. Hu, "Uav-fdn: Forest-fire detection network for unmanned aerial vehicle perspective," *J. Intell. Fuzzy Syst.*, vol. 45, pp. 5821–5836, 2023.

[5] M. Omar, R. Ibrahim, K. Bingi, W. M. H. W. M. Nadzri, and M. Faqih, "Real-time uav system integration for fire detection and classification," in *2023 IEEE International Conference on Computing (ICOCO)*, 2023, pp. 237–241.

[6] K. Song, Y. Zhang, B. Lu, W. Chi, and L. Sun, "Uav forest fire detection based on repvgg-yolov5," *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1277–1282, 2022.

[7] V. Hassija, V. Saxena, and V. Chamola, "Scheduling drone charging for multi-drone network based on consensus time-stamp and game theory," *Computer Communications*, vol. 149, pp. 51–61, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S014036641930948X

[8] V. Hassija, V. Chamola, A. Agrawal, A. Goyal, N. C. Luong, D. Niyato, F. R. Yu, and M. Guizani, "Fast, reliable, and secure drone communication: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 23, no. 4, pp. 2802–2832, 2021.

[9] S. R. Jambula, S. Mathi, S. N. Polu, and S. Yadav, "Advanced image-driven uav fire detection and suppression system with pneumatic deployment: Aerial fire response," in *2023 7th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2023, pp. 1723–1729.

[10] X. Xie, J. Wang, Y. Zhang, J. Xin, L. Mu, and X. Xue, "A uav forest fire detection method based on dual-light vision images," *2023 CAA Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS)*, pp. 1–6, 2023.

[11] K. Choutri, S. Fadloun, M. Lagha, F. Bouzidi, and W. Charef, "Forest fire detection using iot enabled uav and computer vision," in *2022 International Conference on Artificial Intelligence of Things (ICAIoT)*, 2022, pp. 1–6.

[12] B. Dilli and M. Suguna, "Early thermal forest fire detection using uav and saliency map," *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*, pp. 1523–1528, 2022.

[13] R. Haluska, S. Marchevský, M. Pleva, and S. Pillár, "Example of fire detection using uav and cellular networks," in *International Conference on Emerging eLearning Technologies and Applications*, 2023, pp. 159–164.

[14] H. U. Unlu, D. Chaikalis, A. Tsoukalas, and A. Tzes, "Uav indoor exploration for fire-target detection and extinguishing," *Journal of Intelligent Robotic Systems*, vol. 108, pp. 1–14, 2023.

[15] huan yang, J. Wang, and J. Wang, "Efficient detection of forest fire smoke in uav aerial imagery based on an improved yolov5 model and transfer learning," *Remote. Sens.*, vol. 15, p. 5527, 2023.

[16] Y. Zhang, S. Chen, W. Wang, W. Zhang, and L. Zhang, "Pyramid attention based early forest fire detection using uav imagery," in *Journal of Physics: Conference Series*, vol. 2363, 2022.

[17] M. Fouda, S. Sakib, Z. Fadlullah, N. Nasser, and M. Guizani, "A lightweight hierarchical ai model for uav-enabled edge computing with forest-fire detection use-case," *IEEE Network*, vol. 36, pp. 38–45, 2022.

[18] T. Hussain, H. Dai, W. Gueaieb, M. Sicklinger, and G. D. Masi, "Uav-based multi-scale features fusion attention for fire detection in smart city ecosystems," *2022 IEEE International Smart Cities Conference (ISC2)*, pp. 1–4, 2022.

[19] N. Malle, F. F. Nyboe, and E. Ebeid, "Onboard powerline perception system for uavs using mmwave radar and fpga-accelerated vision," *IEEE Access*, vol. 10, pp. 113 543–113 559, 2022.

[20] M. Kolpakov, M. Omarov, and M. Khasanov, "Development of an uav system architecture using machine vision algorithms with a single-board computer or fpga," in *2024 International Russian Smart Industry Conference (SmartIndustryCon)*, 2024, pp. 872–877.

[21] E. Moreac, E. Abdali, F. Berry, D. Heller, and J. Diguet, "Hardware-in-the-loop simulation with dynamic partial fpga reconfiguration applied to computer vision in ros-based uav," *2020 International Workshop on Rapid System Prototyping (RSP)*, pp. 1–7, 2020.

[22] O. Kaziha, T. Bonny, and A. Jarndal, "Genetic algorithm augmented inception-net based image classifier accelerated on fpga," *Multimedia Tools and Applications*, pp. 1–29, 2023.

[23] Y.-M. Choi and H. K.-H. So, "Map-reduce processing of k-means algorithm with fpga-accelerated computer cluster," *2014 IEEE 25th International Conference on Application-Specific Systems, Architectures and Processors*, pp. 9–16, 2014.

[24] T. Ye, W. Qin, Z. Zhao, X. Gao, X. Deng, and Y. Ouyang, "Real-time object detection network in uav-vision based on cnn and transformer," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–13, 2023.

[25] H. Deng, Q. Fu, Q. Quan, K. Yang, and K.-Y. Cai, "Indoor multi-camera-based testbed for 3-d tracking and control of uavs," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 6, pp. 3139–3156, 2020.

[26] B. Ma, Z. Liu, Q. Dang, W. Zhao, J. Wang, Y. Cheng, and Z. Yuan, "Deep reinforcement learning of uav tracking control under wind disturbances environments," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–13, 2023.

[27] Y. Zhang, L. S. Martinez-Rau, B. Oelmann, and S. Bader, "Enabling autonomous structural inspections with tiny machine learning on uavs," in *2024 IEEE Sensors Applications Symposium (SAS)*, 2024, pp. 1–6.

[28] R. Samanta, B. Saha, and S. K. Ghosh, "Tinyml-on-the-fly: Real-time low-power and low-cost mcu-embedded on-device computer vision for aerial image classification," in *2024 IEEE Space, Aerospace and Defence Conference (SPACE)*, 2024, pp. 194–198.

[29] T. Ye, W. Qin, Y. Li, S. Wang, J. Zhang, and Z. Zhao, "Dense and small object detection in uav-vision based on a global-local feature enhanced network," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2022.

[30] P. Foggia, A. Saggese, and M. Vento, "Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 9, pp. 1545–1556, 2015.

[31] M. Hu, Y. Ren, and H. Chai, "Forest fire detection based on improved yolov5," in *Proceedings of the 2021 4th International Conference on Artificial Intelligence and Pattern Recognition*, ser. AIPR '21. New York, NY, USA: Association for Computing Machinery, 2022, p. 172–178. [Online]. Available: https://doi.org/10.1145/3488933.3488985

[32] Q. xing Zhang, G. hua Lin, Y. ming Zhang, G. Xu, and J. jun Wang, "Wildland forest fire smoke detection based on faster r-cnn using synthetic smoke images," *Procedia Engineering*, vol. 211, pp. 441–446, 2018, 2017 8th International Conference on Fire Science and Fire Protection Engineering (ICFSFPE 2017). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877705817362574

[33] V. Gupta, S. Roy, V. Jaiswal, K. Bhardwaj, and P. S. Rana, "Drone assisted deep learning based wildfire detection system," in *2022 Seventh International Conference on Parallel, Distributed and Grid Computing (PDGC)*, 2022, pp. 162–166.

[34] J. Shi, W. Wang, Y. Gao, and N. Yu, "Optimal placement and intelligent smoke detection algorithm for wildfire-monitoring cameras," *IEEE Access*, vol. 8, pp. 72 326–72 339, 2020.

[35] P. R. A, G. P. Praneash, T. Rashmika, and A. Natarajan, "Forest fire detection using computer vision," in *2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 2022, pp. 1094–1103.

[36] Z. Zhanying and C. Xinyuan, "Research on forest fire detection algorithm based on yolov5," in *2021 International Conference on Intelligent Computing, Automation and Systems (ICICAS)*, 2021, pp. 354–357.

[37] B. Dilli and M. Suguna, "Early thermal forest fire detection using uav and saliency map," in *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*, 2022, pp. 1523–1528.

[38] M. Park, D. Q. Tran, J. Bak, and S. Park, "Advanced wildfire detection using generative adversarial network-based augmented datasets and weakly supervised object localization," *International Journal of Applied Earth Observation and Geoinformation*, vol. 114, p. 103052, 2022.

[39] D. Kinaneva, G. Hristov, J. Raychev, and P. Zahariev, "Application of artificial intelligence in uav platforms for early forest fire detection," in *2019 27th National Conference with International Participation (TELECOM)*, 2019, pp. 50–53.

[40] T. Alladi, Naren, G. Bansal, V. Chamola, and M. Guizani, "Secauthuav: A novel authentication scheme for uav-ground station and uav-uav communication," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15 068–15 077, 2020.