Low-Cost UAV-Based Wildfire Smoke Detection

Roshan Krishnan*

Abstract

Early wildfire detection is important because fires become more difficult and dangerous to control as they grow. This paper examines whether a lightweight computer-vision model could support low-cost wildfire-smoke monitoring using small unmanned aerial vehicles. The project was motivated by a Crazyflie drone equipped with an AI Deck, although the model was trained and tested using the publicly available Boreal Forest Fire dataset rather than images collected directly by the drone. The dataset contained 4,954 labeled UAV images from four forest locations in Finland. Four YOLO11n models were trained using a leave-one-location-out approach. In each experiment, images from three locations were used for training and images from the fourth, previously unseen location were used for testing. At a resolution of 640×360, the models achieved overall smoke-detection scores between 76.0% and 93.0%. This shows that the models could detect smoke in new forest environments, although performance varied between locations. The paper also examined the effect of image resolution and analyzed false detections caused by features such as fog, clouds, haze, and bright backgrounds. The results provide a proof of concept for future wildfire smoke detection using low-cost, resource-constrained drones. The source code, trained models, annotations, evaluation results, and reproducibility materials are available in the project's GitHub repository.

Introduction

Wildfires cause extensive damage to forests, wildlife, and human property. Once a fire has grown to a large size, containing it becomes much more difficult and dangerous. For this reason, detecting a wildfire during its earliest stages is an important part of wildfire management. Existing detection methods include satellites, fixed camera networks, environmental sensors, and unmanned aerial vehicles (UAVs) [4]. Each method has advantages and limitations. Satellites can monitor large areas but may not observe the same location continuously, while fixed cameras are limited by their installation locations and line of sight. UAVs can instead fly over remote areas and collect images from different positions, making them a promising tool for local wildfire monitoring.

Smoke is an especially useful visual indicator because it may become visible before flames are clearly visible. However, automated smoke detection is challenging. Wildfire smoke does not have a fixed shape, color, or boundary, and its appearance changes with wind, distance, lighting, and atmospheric conditions. Thin smoke may be invisible, while dense smoke may resemble clouds, fog, dust, or reflections from water and produce false detections. Previous studies have therefore used deep learning object detection models to distinguish wildfire smoke from complex natural backgrounds [5].

In this project, a Crazyflie drone equipped with an AI Deck motivated the development of a low-cost wildfire-smoke detection system. The AI Deck contains a low-power GAP8 processor and a small monochrome camera which allows for a resource-constrained flying platform [2]. However, collecting images of real wildfires would be dangerous, expensive, and difficult to control. Therefore, the current study uses the publicly available Boreal Forest Fire dataset rather than collecting new wildfire data with the Crazyflie. The project should consequently be viewed as a proof of concept for future onboard smoke detection, rather than a complete field deployment on the Crazyflie.

*American High School

The Boreal Forest Fire dataset contains UAV images collected during controlled forest burns at four locations in Finland: Evo, Heinola, Karkkila, and Ruokolahti [7, 8]. The dataset includes images showing smoke under different viewing angles, distances, landscapes, and weather conditions. Smoke regions are labeled with bounding boxes, allowing the images to be used for supervised object detection. In total, the image subset used in this project contains 4,954 images. The presence of four geographically separate collection sites makes it possible to test whether a model has learned general visual features of smoke or has simply learned background patterns associated with a particular location.

Image resolution is another important consideration for small UAV systems. High-resolution images contain more visual detail, but they also require greater storage, communication bandwidth, and computational power. In contrast, low-resolution images can be processed more efficiently but may make thin or distant smoke difficult to recognize. The Crazyflie AI Deck camera itself operates at a relatively low resolution, making this trade-off especially relevant. This paper therefore evaluates the detector using images corresponding to 1920×1080, 1280×720, 640×360, and 320×180 pixels. The goal is to determine the lowest image resolution at which smoke can still be detected reliably.

The main research questions of this project are: (1) how well can a lightweight object-detection model identify wildfire smoke in UAV images, (2) how well does the model generalize to a forest location that was not included during training, (3) which location is the most difficult for the model, (4) how does reducing image resolution affect detection performance, and (5) which natural objects or atmospheric conditions produce false smoke detections? By answering these questions, the project evaluates both the accuracy and practical limitations of lightweight wildfire-smoke detection and explores its potential use on low-cost, resource-constrained UAV platforms.

Methods

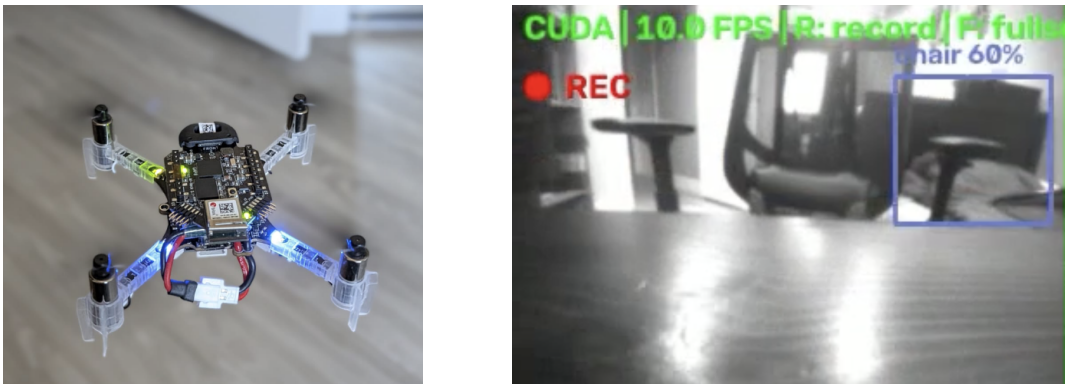

Figure 1: Crazyflie UAV hovering during testing (left) and the live camera feed from the AI Deck (right).

Hovering the drone

My preliminary goal was to make the drone hover at some height z_0 above the ground. To accomplish that, I implemented a PD controller. The thrust equation for the PD controller of the drone is given by:

$$T = mg - k_1(z - z_0) - k_2 v$$

Here, T is thrust, m is the mass of the drone, g is the acceleration due to gravity, k_1 and k_2 are positive constants, z is the present height, and v is the velocity of the drone.

The thrust is related to the propeller angular speed (ω) by the equation $T = C\omega^2$, where C is a constant. The PD controller fine-tunes the ω based on the required thrust. To estimate the current height of the drone, I used a Kalman Filter, for which the equation is

$$\hat{z} = z_p + K(z_s - z_p). \tag{1}$$

In the case of a drone, we have two variables: one is the sensor height z_s, which is measured by the flow deck sensor in the drone [3], and the other is the height z_p, which is calculated by simulating the dynamics physics. They each come with their own uncertainty; $sigma_s$ is the sensor uncertainty and σ_P is the physics uncertainty. The K in the Kalman Filter formula is the Kalman gain given by,

$$K = \sigma_p/(\sigma_s + \sigma_p). \tag{2}$$

To achieve stable drone hovering, I initially established a radio connection between my computer and the drone using OpenAI Codex [6] to generate Python code, which helped the drone hover and fly using Anthropic's Claude [1].

Getting video feed from the AI deck

To establish a stable video streaming pipeline from the Bitcraze AI-Deck, specific hardware modifications and firmware updates were required. Initially, a hardware conflict between the AI-Deck and the Flow Deck v2 prevented camera initialization. To resolve this, the Flow Deck v2 was completely detached from the drone chassis, leaving only the AI-Deck securely mounted to the expansion headers.

With the hardware isolated, the AI-Deck firmware was reflashed. Because the stock firmware on the deck dated back to 2024, updating to the latest stable release was necessary to ensure compatibility with modern streaming protocols and software libraries. This process was completed by referencing the official Bitcraze "Getting Started with AI-Deck 1.1" documentation alongside AI-guided troubleshooting.

Once the firmware was successfully compiled and flashed to the ESP32, the host computer could directly discover and connect to the AI-Deck's localized Wi-Fi network. It was then that I could view the video feed on the AI Deck (see Figure 1).

About the object detection model

The RoboticsBootcamp repository by achintzeus1994 serves as an excellent foundation for deploying real-time object detection models tailored for drone applications. Designed as a practical framework for robotics enthusiasts and developers, this codebase typically bridges the gap between raw computer vision principles and hardware implementation. By leveraging modern lightweight deep learning architectures—such as variations of MobileNet-SSD implemented via PyTorch or OpenCV—the repository focuses on the efficiency required for aerial edge computing. Integrating this framework into your drone's software stack enables it to process high-frame-rate spatial data, enabling critical autonomous behaviors such as real-time obstacle avoidance, precision landing, target tracking, and dynamic environment mapping directly from the onboard camera feed.

By prioritizing computational efficiency, the model achieves low single-digit inference times per frame. This makes it ideal for running locally on low-power embedded hardware, giving the drone the split-second situational awareness required for real-time path planning and reliable collision avoidance.

Results

The dataset used in this paper was the Boreal Forest Fire UAV dataset [8]. After preparing the data for YOLO-based smoke detection, the dataset contained 4,954 labeled UAV images from four

locations in Finland: Evo, Heinola, Karkkila, and Ruokolahti. Of these, 4,693 images contained at least one annotated smoke region, while 261 had empty label files. In total, the dataset contained 4,862 annotated smoke bounding boxes.

Performance on Unseen Locations

The first experiment tested whether a model trained on three locations could detect smoke at a fourth location that it had never seen during training. Table 1 and Figure 2 show the results at 640×360.

Table 1: Performance when each location was held out from training and used only for testing at 640×360.

Location	Precision	Recall	mAP@0.50	mAP@0.50–0.95
Evo	94.2%	91.2%	91.0%	44.3%
Heinola	78.9%	81.4%	76.0%	32.9%
Karkkila	79.7%	71.6%	84.7%	49.6%
Ruokolahti	99.7%	92.8%	93.0%	73.8%

Performance was generally strong but varied between locations. Ruokolahti produced the best results, with 99.7% precision, 92.8% recall, and 93.0% mAP@0.50. Evo also performed well, reaching 91.0% mAP@0.50. Heinola was the most difficult location overall, with the lowest mAP@0.50 of 76.0% and mAP@0.50–0.95 of 32.9%. Karkkila had the lowest recall at 71.6%. These differences show that the model could detect smoke in unseen forest environments, but its performance depended on the visual conditions at each location.

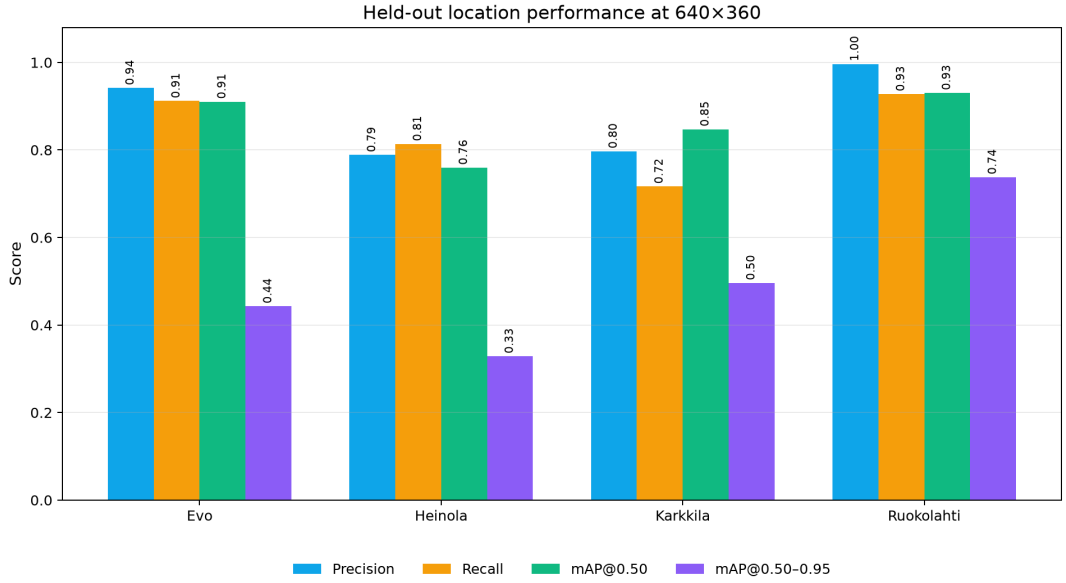

Figure 2: Smoke-detection performance for each held-out location at 640×360. Each location was excluded from training and used only for testing.

Effect of Image Resolution

The same four held-out models were next evaluated at four image resolutions. The results are summarized in Table 2 and Figure 3. Macro averages give each location equal weight.

Table 2: Average performance across the four held-out locations at different image resolutions.

Resolution	Precision	Recall	mAP@0.50	mAP@0.50–0.95	Worst Recall
1920 × 1080	17.1%	17.5%	8.3%	1.9%	0.6%
1280 × 720	55.4%	56.7%	50.6%	17.7%	28.2%
640 × 360	88.1%	84.2%	86.2%	50.1%	71.6%
320 × 180	88.3%	86.2%	87.1%	51.5%	77.1%

The two lower tested resolutions gave the strongest results. At 640 × 360, the macro recall was 84.2% and mAP@0.50 was 86.2%. Performance remained strong at 320 × 180, where recall increased to 86.2% and mAP@0.50 to 87.1%. The worst-location recall was also higher at 320×180, increasing from 71.6% to 77.1%.

The paper defined a reliable setting as having macro recall and mAP@0.50 of at least 80%, while maintaining at least 70% recall at every held-out location. As shown in Figure 3, both 640 × 360 and 320 × 180 satisfied these conditions. Therefore, 320 × 180 was the lowest tested resolution that maintained reliable smoke detection. This is important for small UAVs because lower resolution reduces the amount of image data that must be processed.

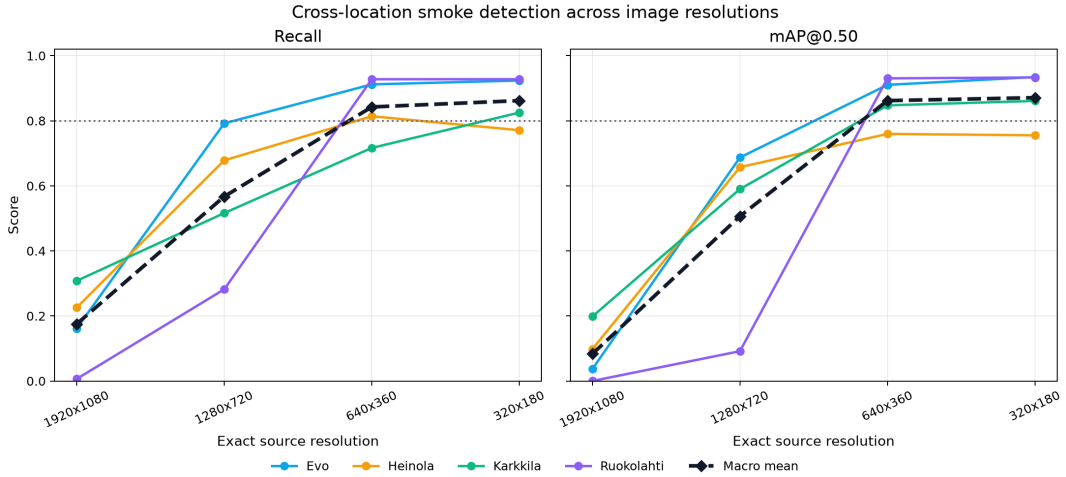

Figure 3: Recall and mAP@0.50 across the four tested image resolutions. The dashed line shows the macro average and the dotted line marks the 0.80 performance threshold.

Performance decreased sharply at 1280 × 720 and 1920 × 1080. However, this does not mean that higher-resolution cameras are worse for smoke detection. The model was trained using a 640-pixel input scale, while these tests also increased the model's inference canvas. This changed the apparent size of smoke regions relative to what the model had seen during training. A UAV could therefore still capture high-resolution images while resizing them before sending them to the detector.

Error Analysis

A detailed error analysis was performed at 320×180, the lowest resolution that passed the reliability criteria. Predictions with confidence of at least 0.25 were included, and a predicted box was counted as correct when it overlapped an unmatched ground-truth box with IoU ≥ 0.50.

Across the four locations, 592 images contained at least one unmatched prediction, corresponding to 639 unmatched predicted boxes. There were also 632 ground-truth boxes that were missed. However, these numbers do not all represent smoke being incorrectly detected in smoke-free scenes. Many unmatched predictions occurred in images that actually contained smoke and were caused

by duplicate or oversized bounding boxes.

The annotation audit revealed another important source of apparent error. Seven images with empty annotation files produced smoke detections. Visual inspection showed that six of these seven images actually contained visible smoke. An example is shown on the left side of Figure 4. The model's detection was therefore reasonable, but it was automatically counted as a false positive because no ground-truth box was available.

The right side of Figure 4 shows a clearer example of a genuine environmental false alarm. In this Heinola image, the model detected a bright, slightly hazy region of sky above the trees even though no smoke was visible.

Figure 4: Examples from the error analysis. Left: visible smoke was detected but was missing from the ground-truth annotation. Right: bright, hazy sky was incorrectly detected as smoke.

Figure 5 shows additional examples from the visual audit. The Heinola example is a genuine false alarm, while the Evo, Karkkila, and Ruokolahti examples contain visible smoke despite having empty label files. No clear cases were found in which fog alone or water reflections were mistaken for smoke at the selected confidence threshold. These results show that qualitative inspection is important because dataset annotation errors can make the measured false-positive rate appear worse than the model's actual smoke-detection behavior.

Conclusion

This paper showed that a lightweight YOLO11n model can detect wildfire smoke in UAV images from forest locations that were not seen during training. At 640×360, the models achieved smoke-detection scores between 76.0% and 93.0%, showing that the model could generalize to new environments, although performance varied between locations. The resolution experiments also showed that 320×180 maintained strong performance and was the lowest tested resolution that satisfied the paper's reliability criteria. This is encouraging for small drones, where computing power, memory, and battery capacity are limited.

The error analysis showed that not every measured false positive was a true model mistake. Several images contained visible smoke even though their annotation files were empty, causing correct detections to be counted as errors. However, the model also produced some genuine false alarms, such as detecting bright or hazy sky as smoke. The results provide a proof of concept that lightweight computer-vision models could support low-cost UAV-based wildfire monitoring.

Future work should test the system using images collected directly from a Crazyflie or other small UAV. The model could also be tested in a wider range of environments, including different forests, seasons, weather conditions, fog, clouds, and different types of smoke. Additional training data could help reduce false alarms and improve performance in difficult locations such as Heinola. Future work could also deploy the model directly on the Crazyflie AI Deck to measure real-time inference speed, power use, detection range, and flight performance. Finally, smoke detection could be combined with the drone's GPS location so that a detected smoke plume can be reported together with its approximate geographic location.

Figure 5: Selected error examples at 320 × 180. Predictions are shown in red and ground-truth boxes, when present, in yellow. The Heinola example is a genuine false alarm, while the other examples contain visible smoke that was not annotated.

References

[1] Anthropic. *Claude Code*. `https://anthropic.com`. Agentic command-line developer tool and coding assistant utilized for flight control logic optimization and script deployment. 2025.

[2] Bitcraze AB. *AI Deck 1.1*. `https://www.bitcraze.io/products/ai-deck/`. Accessed: 2026-07-29. 2026.

[3] Bitcraze AB. *Flow deck v2 Datasheet*. Rev 1. Bitcraze AB. 2018. URL: `https://www.bitcraze.io/documentation/hardware/flow_deck_2/flow_deck_2-datasheet.pdf`.

[4] Ankita Mohapatra and Truyen Trinh. "Early Wildfire Detection Technologies in Practice—A Review". In: *Sustainability* 14.19 (2022), p. 12270. DOI: `10.3390/su141912270`. URL: `https://doi.org/10.3390/su141912270`.

[5] Mukhriddin Mukhiddinov, Akmalbek Bobomirzaevich Abdusalomov, and Jinsoo Cho. "A Wildfire Smoke Detection System Using Unmanned Aerial Vehicle Images Based on the Optimized YOLOv5". In: *Sensors* 22.23 (2022), p. 9384. DOI: `10.3390/s22239384`. URL: `https://doi.org/10.3390/s22239384`.

[6] OpenAI. *OpenAI Codex*. `https://openai.com`. Autonomous AI coding agent and model family utilized for script automation and environment connection protocols. 2026.

[7] Julius Pesonen et al. *Boreal Forest Fire: UAV-Collected Wildfire Detection and Smoke Segmentation Dataset*. National Land Survey of Finland, Finnish Geospatial Research Institute, 2025. DOI: `10.23729/fd-72c6cf74-b8eb-3687-860d-bf93a1ab94c9`. URL: `https://doi.org/10.23729/fd-72c6cf74-b8eb-3687-860d-bf93a1ab94c9`.

[8] Julius Pesonen et al. "Boreal Forest Fire: UAV-Collected Wildfire Detection and Smoke Segmentation Dataset". In: *Scientific Data* 12 (2025), p. 1419. DOI: `10.1038/s41597-025-05634-0`. URL: `https://doi.org/10.1038/s41597-025-05634-0`.