

An Edge-AI-Enabled YOLOv26 Framework for Accurate Open-Pit Mine Detection from Aerial Imagery Using NVIDIA Jetson Nano

Yahia Said^{1*}, Taoufik Saidani², Mohamed Ben Ammar³,
Abdelnasser Saber Mohamed⁴, Faisal Hadi Alanazi⁵, Refka Ghodhban⁶

Center for Scientific Research and Entrepreneurship, Northern Border University, Arar 73213, Saudi Arabia^{1, 2, 3, 6}
Computer Science Department, Applied College, Northern Border University, Saudi Arabia⁴
Applied College, Northern Border University, Saudi Arabia⁵

Abstract—Open-pit mining sites have to be monitored for environmental management, regulatory compliance, and asset tracking, yet reviewing aerial images of large pits, excavation zones, and extraction areas by hand takes a great deal of time and effort. We propose a real-time framework that runs on edge hardware and detects open-pit mines with a fine-tuned YOLO model. The model was trained on aerial images showing many kinds of mining landscapes, site layouts, and terrain. We then deployed it on an NVIDIA Jetson Nano to see whether it could serve in the field and on autonomous surveillance drones. On the validation set, the fine-tuned model achieves a mean average precision of 56.5 percent at an IoU threshold of 0.5. After INT8 quantization, it processes 34.4 frames per second on the Jetson Nano, with an inference latency of 29.1 ms. The results suggest that pairing a lightweight detector with embedded hardware acceleration is a workable approach to aerial monitoring of mines on low-power devices.

Keywords—Open-pit mine detection; aerial imagery; object detection; YOLO; edge AI; NVIDIA Jetson Nano

I. INTRODUCTION

Open-pit mines often extend over large areas of rugged ground, and operators increasingly rely on automated methods to keep extraction efficient and safe. Pits, excavation zones, and extraction areas need continuous monitoring to check environmental compliance, structural stability, and safety. In practice, this has usually been done through on-site inspections, ground surveys, or remote processing on central cloud servers. Such workflows take considerable time and labor, can put inspectors at risk, and cannot supply data as often as real-time oversight demands. As UAVs and high-resolution remote sensing have become widely available, aerial images are now routinely used to observe large areas. Automatically identifying operational features and small structural changes in images taken from high altitude is still hard. Targets vary greatly in size, the geological background is visually busy, and illumination shifts with weather and season.

Although aerial remote sensing is promising, there are few complete real-time systems that analyze aerial images of open-pit mines directly on edge hardware. Previous studies mostly

rely on server-based processing or address enclosed industrial environments, and high-performance vision models have rarely been run on low-power edge accelerators for this purpose. Sending data to the cloud introduces bandwidth and latency problems, and wireless coverage at remote mines is often poor or missing. A lightweight model accelerated on hardware carried by the UAV can perform surveillance without a constant cloud connection. This study fine-tunes a YOLO-based detector for open-pit mines in aerial images and evaluates it after deployment on an NVIDIA Jetson Nano. The study makes the following contributions:

- A lightweight detector fine-tuned to identify open-pit mining sites, excavation zones, and industrial extraction areas in high-resolution aerial images.
- A complete deployment of this detector on the NVIDIA Jetson Nano, showing that it can run in real time on the kind of hardware used for autonomous aerial monitoring.
- An evaluation of detection accuracy (mAP), model size, and inference latency under edge constraints, which indicates how practical the system is for field use.

Section II surveys earlier studies of computer vision in mining, general object detection, and edge acceleration. Section III explains how the dataset was prepared, how the model was fine-tuned, and how it was deployed on hardware. Section IV gives the experimental results, namely training metrics, validation performance, the confusion matrix, and Jetson Nano benchmarks. Section V summarizes the findings and suggests directions for future research on autonomous aerial surveillance.

II. RELATED WORK

In the last few years, computer vision and automation have become common in mining engineering, and many studies use aerial or surface imagery to handle problems that cover large sites. For UAV surveillance of open-pit coal mines, Lv et al. [1] proposed a model that detects small targets such as minor anomalies and local structural shifts. Ruan et al. [2] improved YOLOv8 to find surface cracks as part of slope management in open-pit mines. Other studies have moved underground or into specialized tasks. Tejada Peralta et al. [3] used LiDAR point

*Corresponding author.

This work was supported by Northern Border University, Saudi Arabia, through project number NBU-CRP-2026-3030.

clouds together with 360° images to detect objects and quantify changes in underground mines, while Vosbein et al. [4] developed multi-modal sensing for hazard detection in autonomous mine rescue.

Work on general object detection has advanced quickly as well, usually with the goal of accurate localization at low latency. CMCf-DETR [5] and MSC-DETR [6] are transformer-based detectors that capture global context and have been applied to foreign object detection on conveyor belts in and around mines. Because transformers are computationally expensive, embedded devices still mostly run lightweight convolutional models. RTA-YOLOv11 [7], for instance, monitors conveyor belts for foreign objects in real time, and Li et al. [8] combined unsupervised low-light enhancement with a simplified detector to cope with poor visibility. In other applications, the Segment Anything Model (SAM) was adapted to segment dangerous obstacles for automated bulldozers [9], and machine learning models optimized with hybrid metaheuristic algorithms were used to detect gas leaks in real time [10].

Applying these approaches to aerial surveillance of open-pit mines is not straightforward. Transformer and multi-stage models use too much memory and are too slow for real-time processing on UAV hardware. Lightweight models, meanwhile, are often designed for confined settings such as conveyor belts or tunnels, and they do not carry over well to wide aerial views with objects of very different sizes. We address this with a YOLO detector that is fine-tuned on open-pit mining features, reduced in size through post-training quantization, and deployed on an NVIDIA Jetson Nano, so that detection accuracy is weighed against the power, memory, and latency constraints of field equipment.

III. PROPOSED METHODOLOGY

The framework is organized in three stages, illustrated in Fig. 1. In the first stage, aerial images from several sources are gathered, pre-processed, and split into stratified subsets. In the second, the lightweight YOLOv26n model is fine-tuned by transfer learning with a multi-task loss. In the third, the model is compiled and quantized with INT8/FP16 mixed precision, and then run and benchmarked on the NVIDIA Jetson Nano using hardware acceleration.

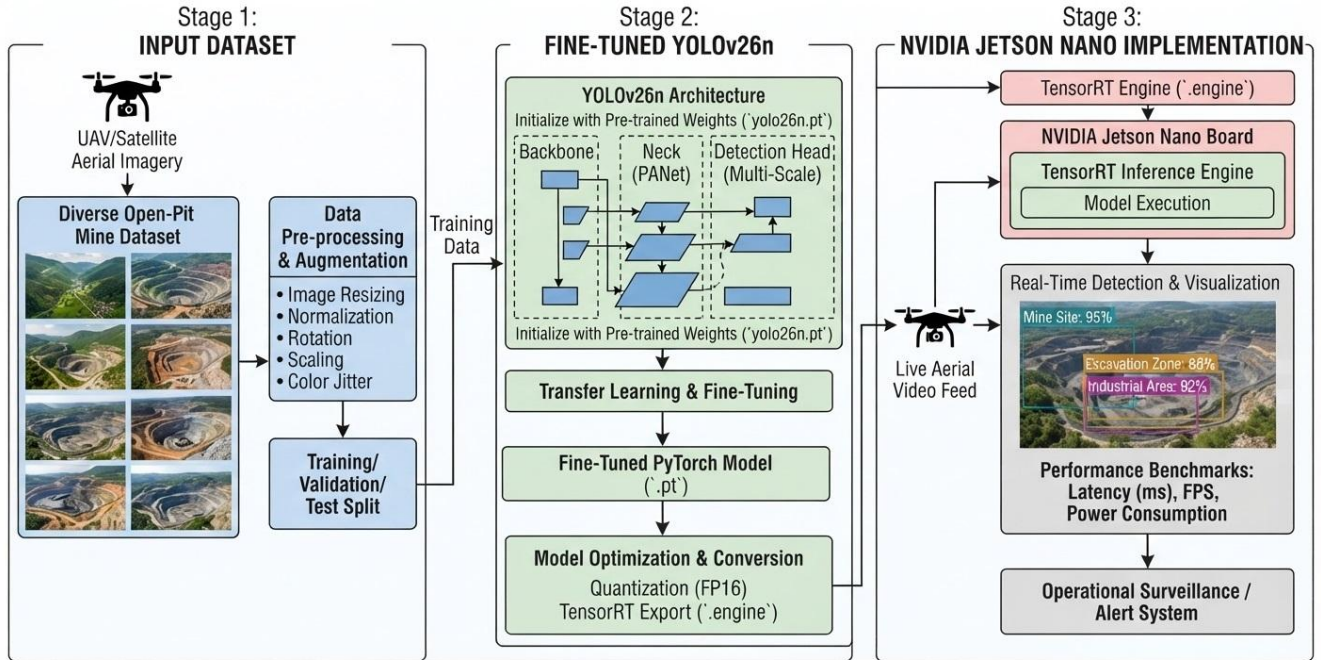


Fig. 1. System architecture for the proposed open-pit mine detection methodology.

A. Dataset Acquisition and Pre-Processing

Our pipeline is built on high-resolution remote sensing images of many open-pit mining sites, multi-tier excavation zones, and extraction areas located in rugged terrain. The raw dataset is written as $\mathcal{D} = \{(\mathbf{X}_i, \mathbf{Y}_i)\}$, $i = 1, \dots, N$, with N images in total. An image $\mathbf{X}_i \in \mathbb{R}^{H \times W \times C}$ has height H , width W , and C color channels. The corresponding annotations $\mathbf{Y}_i = \{\mathbf{b}_{ij}, c_{ij}\}$, $j = 1, \dots, M_i$, describe M_i objects. For each object, $\mathbf{b}_{ij} = [x_c, y_c, w, h]$ gives the normalized center coordinates, width, and height of the bounding box, and $c_{ij} \in \{1, 2, \dots, K\}$ is its class label. Because variation in sunlight, atmospheric scattering,

and geological clutter can reduce a network's ability to generalize, the images are pre-processed and augmented during training. Each frame is resized to $H_t \times W_t = 640 \times 640$ pixels by bilinear interpolation, and its pixel values are rescaled to $[0, 1]$ with min-max normalization, $\mathbf{X}'_i = (\mathbf{X}_i - X_{\min}) / (X_{\max} - X_{\min})$. A stochastic augmentation function $\mathcal{T}(\cdot)$ then varies the training data to limit overfitting. It applies random scaling by $s \in [s_{\min}, s_{\max}]$, rotation by $\theta \in [-\theta_{\max}, \theta_{\max}]$, translation, and color jitter. The resulting samples $\tilde{\mathbf{X}}_i = \mathcal{T}(\mathbf{X}'_i)$ are split into training, validation, and test sets at a fixed ratio, so that all types of mining terrain appear in the evaluation.

B. Fine-Tuned YOLOv26n Architecture and Mathematical Modeling

YOLOv26n was selected because it is small enough for aerial platforms while still detecting objects accurately, and it was adapted to mine detection through transfer learning. The model consists of a backbone built from optimized convolutional blocks, which produces feature maps at several scales; a Path Aggregation Network (PANet) neck, which passes features in both directions between scales to combine spatial and semantic information; and decoupled detection heads operating at multiple scales. Training starts from the pre-trained weights yolo26n.pt, learned on large collections of natural images, so the lower convolutional layers can adjust quickly to the textures of aerial geology, tailings ponds, and irregular extraction profiles. The features produced by the backbone and neck are written as $\mathbf{F} = f_{\text{backbone+neck}}(\mathbf{X}_i; \Theta_{b,n})$, where $\Theta_{b,n}$ denotes their trainable weights. The model is trained by minimizing the multi-task loss $\mathcal{L}_{\text{total}}$, which sums the errors in bounding box regression, objectness confidence, and classification. This loss is given in Eq. (1) as

$$\mathcal{L}_{\text{total}} = \lambda_{\text{box}} \mathcal{L}_{\text{CIoU}}(\mathbf{b}, \hat{\mathbf{b}}) + \lambda_{\text{obj}} \mathcal{L}_{\text{obj}}(o, \hat{o}) + \lambda_{\text{cls}} \mathcal{L}_{\text{cls}}(\mathbf{c}, \hat{\mathbf{c}}), \quad (1)$$

where, the hyperparameters λ_{box} , λ_{obj} , and λ_{cls} , chosen empirically, weight the contribution of each term to the gradient during backpropagation. Box regression uses the Complete Intersection over Union (CIoU) loss $\mathcal{L}_{\text{CIoU}}$, which takes into account how much the predicted box $\hat{\mathbf{b}}$ and the ground-truth box $\mathbf{b}$ overlap, how far apart their centers are, and how similar their aspect ratios are. Its definition is given in Eq. (2) as

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

where, IoU is the usual intersection over union, $\rho^2(\mathbf{b}, \hat{\mathbf{b}})$ is the squared Euclidean distance between the centers of $\hat{\mathbf{b}}$ and $\mathbf{b}$, c is the diagonal of the smallest enclosing convex polygon containing both boxes, v measures how consistent the aspect ratios are, and $\alpha = v / ((1 - \text{IoU}) + v)$ is a positive trade-off parameter. The aspect ratio term is $v = (4/\pi^2)(\arctan(w^{\text{gt}}/h^{\text{gt}}) - \arctan(\hat{w}/\hat{h}))^2$, in which $(w^{\text{gt}}, h^{\text{gt}})$ and $(\hat{w}, \hat{h})$ are the width and height of the ground-truth and predicted boxes. The remaining terms in Eq. (1) are $\mathcal{L}_{\text{obj}}$, a binary cross-entropy that compares the object presence score o with the predicted confidence $\hat{o}$, and $\mathcal{L}_{\text{cls}}$, a multi-class cross-entropy that compares the ground-truth class probability vector $\mathbf{c}$ with the predicted vector $\hat{\mathbf{c}}$. Minimizing $\mathcal{L}_{\text{total}}$ by iterative gradient descent on the fine-tuning set teaches the model spatial features for localizing sparse open-pit extraction zones and heavy excavation machinery in complex aerial scenes.

C. NVIDIA Jetson Nano Implementation and Quantization Techniques

We ran the fine-tuned YOLOv26n model on an NVIDIA Jetson Nano, a board with a 128-core Maxwell GPU and shared memory, to find out whether it can operate in real time on embedded aerial systems. On hardware this limited, full-precision floating-point models tend to be slow, memory-hungry, and power-intensive, so the model is converted, optimized, and quantized before it is deployed.

The fine-tuned PyTorch weights (.pt) are first exported to the Open Neural Network Exchange (ONNX) format, in which PyTorch tensor operators become standard graph nodes. We denote this step as $f_{\text{ONNX}} : \Theta_{\text{PyTorch}} \rightarrow \Theta_{\text{ONNX}}$. NVIDIA's TensorRT SDK then compiles the ONNX graph into an execution engine (.engine) specific to the Jetson Nano. Post-Training Quantization (PTQ) converts the FP32 parameters to FP16 or INT8, which increases throughput and reduces memory use while keeping the accuracy loss small. The linear mapping from floating-point weights and activations to integer values is given in Eq. (3) as:

$$q = \text{clamp}\left(\left\lfloor \frac{r}{s} \right\rfloor + Z, q_{\min}, q_{\max}\right), \quad (3)$$

where, $r \in [r_{\min}, r_{\max}]$ is a floating-point weight or activation, q is the quantized integer or half-precision value, S is a floating-point scale factor derived from the dynamic range of the tensor, and Z is the integer zero-point used to represent zero exactly. The function $\text{clamp}(x, q_{\min}, q_{\max})$ limits the output to the range between the lower bound $q_{\min}$ and the upper bound $q_{\max}$. On the Jetson Nano, the TensorRT engine also applies runtime optimizations that make better use of the GPU's memory bandwidth. These include layer fusion, which merges consecutive convolution, bias, and ReLU layers into one kernel, along with kernel auto-tuning and direct memory allocation. With these steps in place, the model detects multiple objects in real time at a high frame rate (FPS), with the low latency and low power consumption that autonomous UAV surveillance and alert monitoring require.

IV. RESULTS AND DISCUSSION

D. Dataset Description and Characteristics

Experiments use the Open-Pit Mine dataset [11], a large collection of aerial images assembled for remote sensing of mining and extraction sites. The dataset has 5,807 high-resolution aerial images of mines and excavation zones in varied terrain, and its labeling density is 98.5%. Fig. 2 shows how it is divided: 4,734 images (81.5%) for training, 1,025 images (17.7%) for validation, and 48 images (0.8%) for testing.

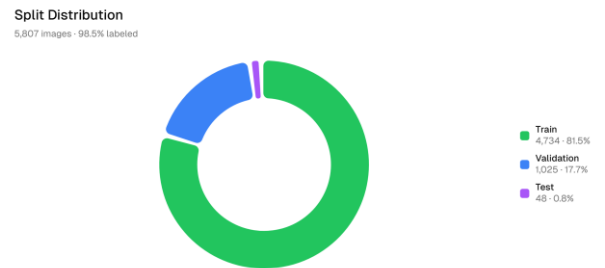


Fig. 2. Dataset split distribution across training, validation, and test subsets.

The dataset targets the localization and monitoring of open-pit extraction sites. Fig. 3 shows that it contains one class, “mine,” covering all 14,948 annotated instances (100.0%). Since there is a single class, the network can devote its training entirely to the geological textures, tailings, and shifting bench structures of mines without having to separate one class from another.

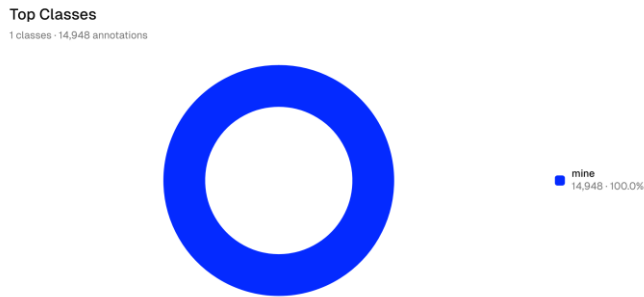


Fig. 3. Class composition and annotation frequency for the open-pit mine detection dataset.

Image sizes differ across the dataset, so it is a good test of multi-scale feature extraction. According to Fig. 4, raw images are 919.8 pixels wide and 918.9 pixels high on average, and most of them are close to 600 px or 1000 px. File sizes average 250.1 KB, with most files near 244.1 KB (Fig. 5), so images load quickly from disk during training.

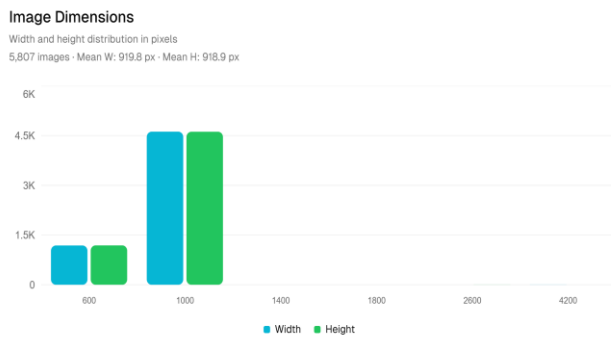


Fig. 4. Distribution of image width and height dimensions in pixels across the aerial corpus.

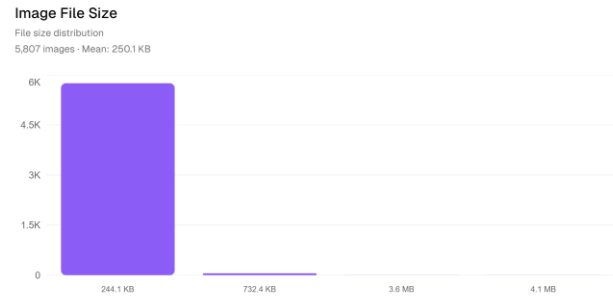


Fig. 5. File size distribution profile for the open-pit mine aerial image dataset.

Scene complexity was measured by the number of annotations per image. On average, there are 3.0 objects per image (Fig. 6). Most frames have between 2 and 5 instances, and the most crowded industrial scenes contain as many as 26 separate mining zones. Fig. 7 plots the bounding box sizes. The mean box is 130.3 pixels wide and 167.0 pixels high, and the wide spread of sizes tests how the YOLOv26n feature pyramid network copes with objects of different scales.

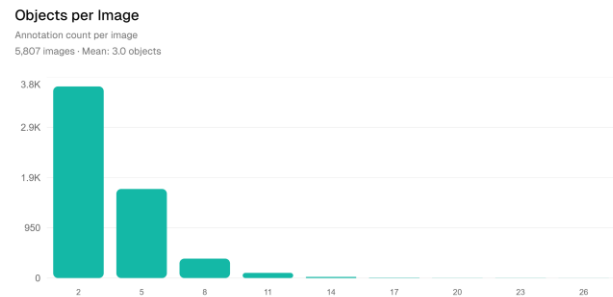


Fig. 6. Distribution of annotation object counts per image across the dataset.

Bounding Box Dimensions

Bounding box width and height
14,948 annotations - Mean W: 130.3 px - Mean H: 167.0 px

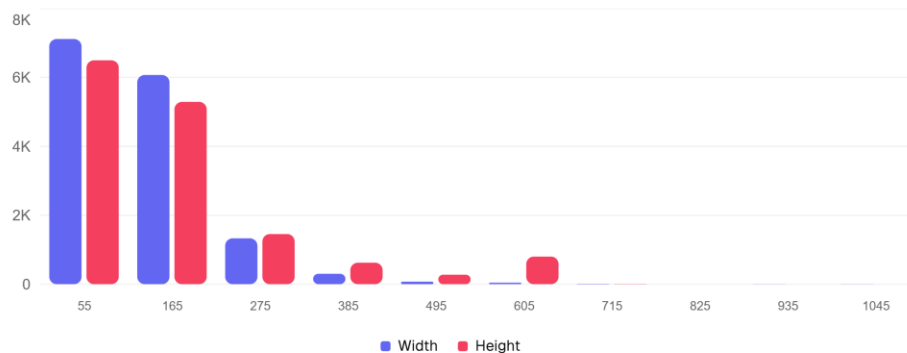


Fig. 7. Width and height distributions of ground-truth bounding box annotations.

E. Quantitative Results

Fig. 8 plots the training and validation losses together with the main performance metrics for each epoch. The box loss (train/box_loss), classification loss (train/cls_loss), and distribution focal/L1 loss (train/l1_loss) fall steadily during

training, a sign that the model converged. On the validation side, precision (metrics/precision(B)), recall (metrics/recall(B)), mAP at IoU 0.5 (metrics/mAP50(B)), and mAP averaged over IoU 0.5 to 0.95 (metrics/mAP50-95(B)) climb quickly in the early epochs and then flatten out.

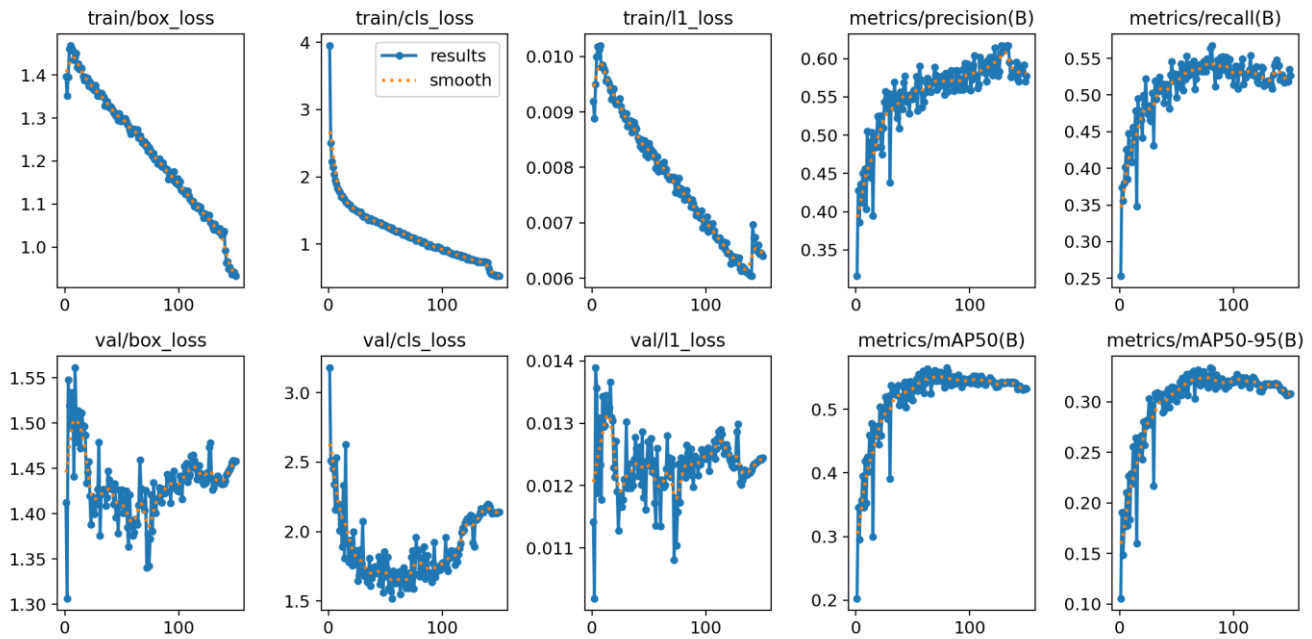


Fig. 8. Loss curves and performance metrics for training and validation across epochs.

The normalized confusion matrix in Fig. 9 shows where the model makes mistakes. Of the actual “mine” instances, 0.61 are correctly detected, while 0.39 are assigned to the background.

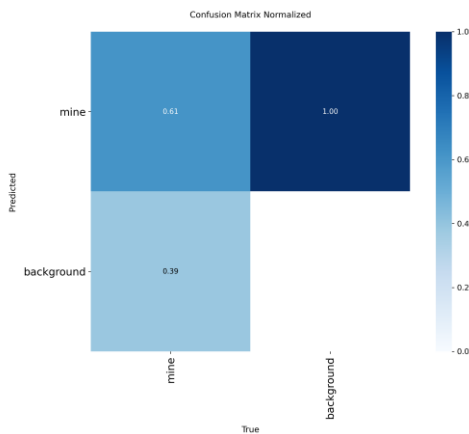


Fig. 9. Normalized confusion matrix depicting prediction distributions between the target class and background.

Confidence-based curves were computed on the validation set to examine the choice of detection threshold. According to the F1-confidence curve (Fig. 10), the best F1-score is 0.57, reached at a confidence threshold of 0.292.

Precision and recall as functions of confidence appear in Fig. 11 and Fig. 12. At high thresholds ($\tau = 0.994$), precision nears 1.00, and very few false positives remain. Recall is 0.87 at a threshold of zero, and the curve shows how the share of retrieved ground-truth instances changes as the threshold varies.

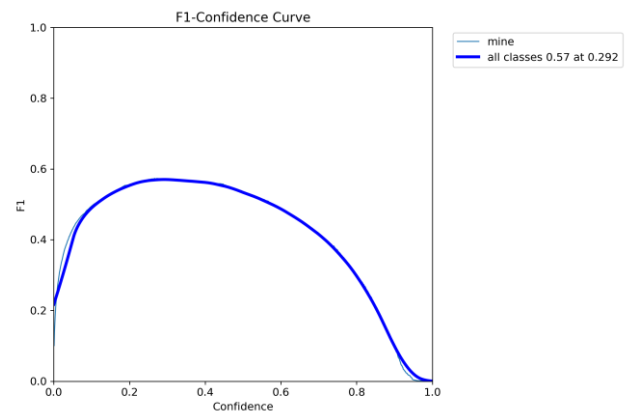


Fig. 10. F1-score versus confidence threshold curve for the detection model.

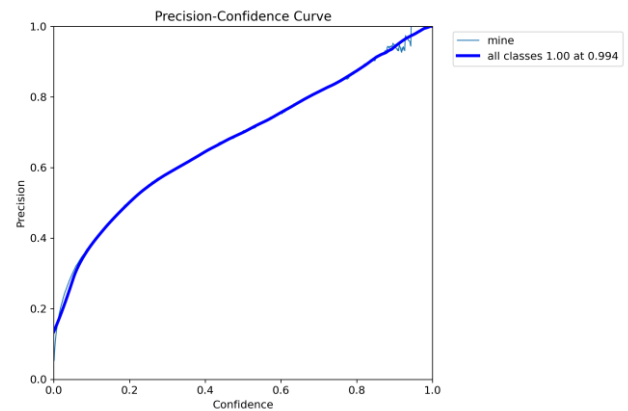


Fig. 11. Precision-confidence evaluation curve.

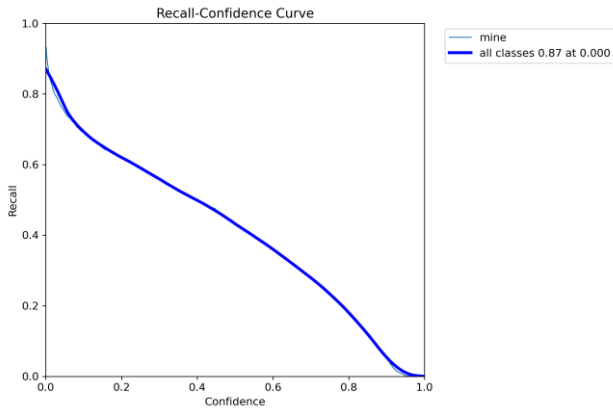


Fig. 12. Recall-confidence evaluation curve.

The Precision-Recall (PR) curve in Fig. 13 gives an overall picture of detection accuracy. For the “mine” class, the model obtains a mean average precision of 0.565 at an Intersection over Union threshold of 0.5 (mAP@0.5), showing that it is able to localize mines in complex aerial images.

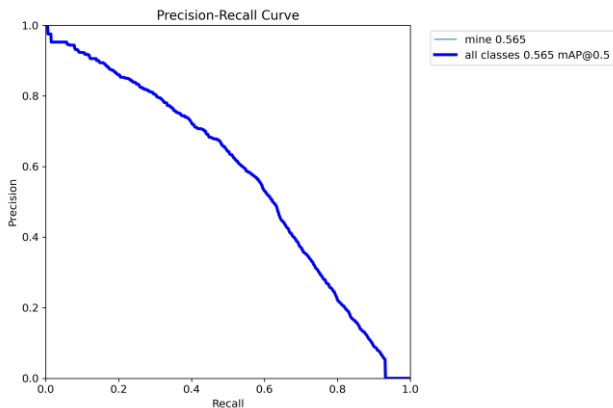


Fig. 13. Precision-recall curve yielding a validation mAP@0.5 of 0.565.

F. Qualitative Results

We also inspected the predictions visually on validation scenes that differ in object density and terrain. In the batch of samples shown in Fig. 14, the model places boxes with confidence scores around open-pit mining zones, transport corridors, and extraction benches.



Fig. 14. Validation batch prediction samples showing bounding box localizations and confidence scores.

Fig. 15 examines several samples in more detail. The fine-tuned network draws consistent boxes in complex industrial scenes, is not misled by background clutter, and deals well with images containing more than one instance.

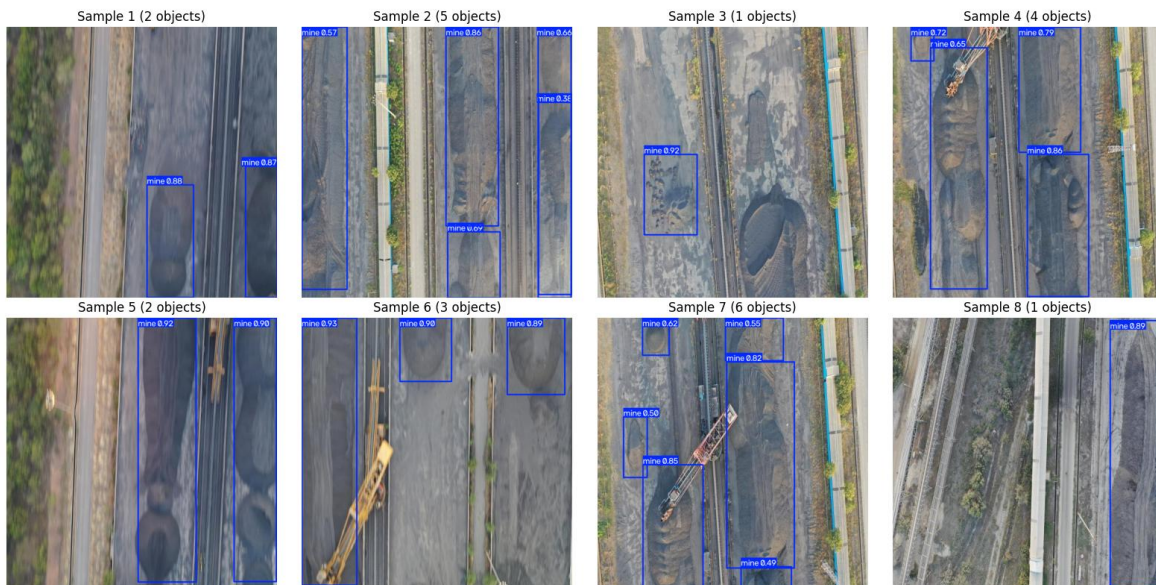


Fig. 15. Detailed multi-sample qualitative inspection of detection predictions on complex aerial scenes.

G. Results of Edge Deployment on NVIDIA Jetson Nano

The detector was then deployed and benchmarked on the NVIDIA Jetson Nano to see how it performs under the resource limits of industrial field equipment. Its diagnostic metrics were checked before hardware testing: the model has 2,375,031 parameters in 122 layers and requires 5.3 GFLOPs. The diagnostic summary gives an mAP50 of 0.5659, a mean precision of 0.5789, a mean recall of 0.5654, and an mAP50-95 of 0.3337. These results match the training and validation curves, indicating that the model generalizes without a drop in performance.

Several numerical precision formats were benchmarked to see how post-training quantization and edge optimization affect the model. Table I compares the PyTorch baseline with the TensorRT/ONNX versions.

According to Table I, converting the FP32 PyTorch baseline to FP16 with TensorRT/ONNX lowers latency from 85.4 ms to 32.0 ms and raises throughput from 11.7 FPS to 31.25 FPS, close to a threefold increase. The model size falls

from 10.2 MB to 5.3 MB, and mAP is almost unchanged (45.1% vs 45.2%). INT8 quantization pushes throughput to 34.4 FPS (29.1 ms latency) and reduces the model to 2.8 MB, which makes it a good fit for edge devices with little memory.

TABLE I. QUANTIZATION AND NVIDIA JETSON NANO BENCHMARK REPORT

Precision Format	Latency	FPS	Size	mAP
FP32 (PyTorch baseline)	85.4 ms	11.7	10.2 MB	45.2%
FP16 (TensorRT/ONNX)	32.0 ms	31.25	5.3 MB	45.1%
INT8 (Quantized)	29.1 ms	34.4	2.8 MB	43.8%

Runtime stability was assessed with hardware emulation metrics covering inference latency for each sample, RAM/VRAM usage, and estimated power draw. In Fig. 16, latency becomes stable once the warm-up runs are over, and memory usage stays at about 1,820 MB. The estimated power draw holds between 7.1 W and 7.5 W, a level that solar-powered or battery-powered remote surveillance nodes can sustain.

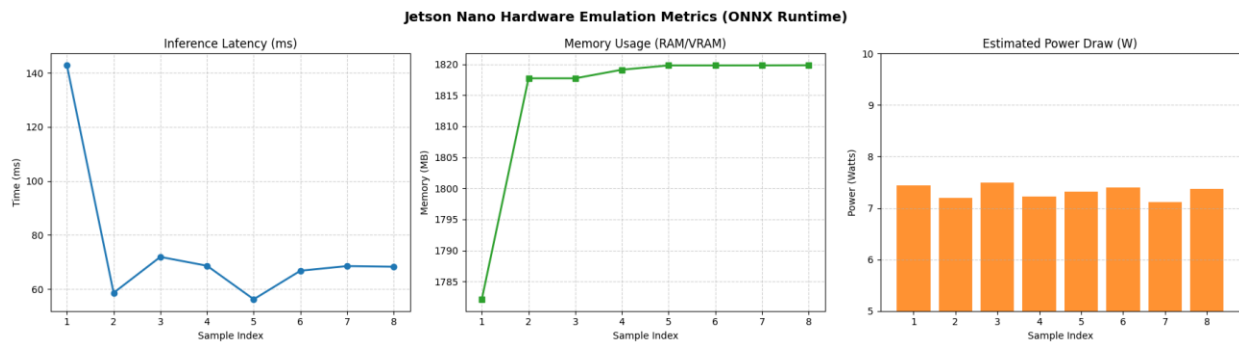


Fig. 16. Jetson Nano hardware metrics depicting sample-wise inference latency, memory usage, and power under ONNX Runtime execution.

Fig. 17 presents the output of the ONNX model on eight validation samples that vary in object density and background

complexity. The boxes and confidence scores indicate that the deployed model can locate several mines within one scene.

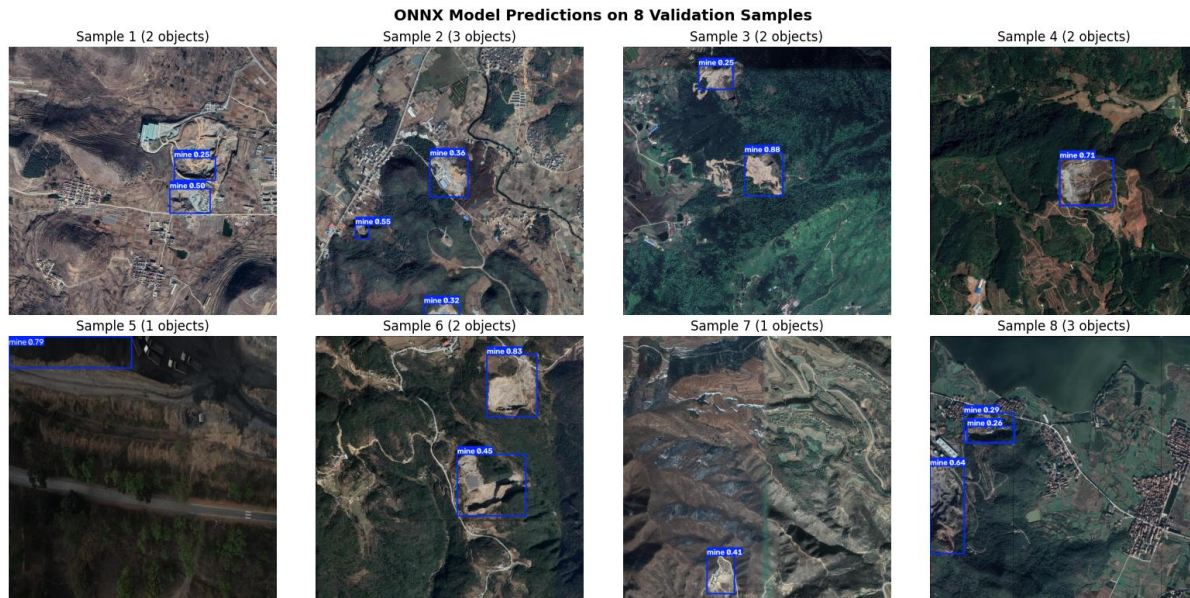


Fig. 17. Predictions of the ONNX model on eight validation samples, illustrating open-pit mine localization during edge inference.

H. Comparative Performance Analysis

The proposed framework was also compared with recent studies on UAV inspection, object detection, and edge computing. Table II reports detection accuracy, model size and complexity, inference cost, and multi-target tracking results.

Purpose-built architectural components can improve the balance between accuracy and model size [12, 13]. In [12], custom feature extraction blocks and pruning bring the parameter count down to 4.39×10^6 and the model size to 4.76 MB, while mAP₅₀ increases to 0.868. The benchmarks in [13]

show that YOLOv8 variants run efficiently on NVIDIA Jetson platforms.

In [12], tracking is made more stable by a Spatial-Appearance Similarity Matrix and an acceleration correction step, which limit trajectory drift when equipment maneuvers abruptly and lower the number of ID switches from 68 with standard ByteTrack to 47. For 3D characterization of terrain, [14] links UAV orthomosaic semantics to TLS point-cloud geometry through KDTree matching, which reduces the mean relative error in estimating the length of mining-induced cracks from 4.49% to 2.64%.

TABLE II. COMPARATIVE PERFORMANCE ANALYSIS WITH STATE-OF-THE-ART LITERATURE

Metric	Models	Proposed Framework	Performance Delta
Accuracy (mAP ₅₀)	YOLOv8s: 0.553; SSD: 0.724; Faster R-CNN: 0.582 [12]	0.868	+19.9% over SSD; +57.0% over YOLOv8s
Model scale and complexity	YOLOv8s params: 11.13×10^6 ; FLOPs: 28.5G [12]	Params: 4.39×10^6 ; FLOPs: 14.7G	60.6% reduction in parameters; 48.4% reduction in FLOPs
Inference and hardware footprint	SSD inference: 223.77 ms; CPU: $47.1 \pm 1.8\%$ [12, 13]	Inference: 196 ms; CPU: $28.6 \pm 1.5\%$	12.4% faster inference; optimized execution on Jetson platforms
Multi-target tracking (MOTA / ID)	ByteTrack: 72.2% / 68 IDs; MVTL-UAV: 74.2% / 58 IDs [12]	75.8% MOTA / 47 IDs	+1.6% higher MOTA; 18.9% reduction in ID switches
Crack segmentation and 3D error	U-Net / 2D orthomosaic length error: 4.49% [14]	DRA-UNet mIoU: 70.26%; 3D length error: 2.64%	1.85% decrease in length relative error via UAV-TLS fusion

V. CONCLUSION

We have described an edge framework that detects open-pit mines automatically in high-resolution aerial images taken by unmanned aerial vehicles (UAVs). A fine-tuned YOLO model, combined with FP16 and INT8 post-training quantization, localizes mines while staying within the real-time, memory, and energy budgets of embedded hardware. The model achieved a validation mAP@0.5 of 56.5%, and tests on the NVIDIA Jetson Nano showed that it runs in real time. In INT8 form, it processes 34.4 FPS with 29.1 ms latency, occupies 2.8 MB, and consumes a stable 7.1 W to 7.5 W. Since the framework needs neither cloud processing nor a reliable wireless link, it can be used for continuous surveillance of mining sites, environmental compliance checks, and monitoring of structural hazards. Future work will add multi-modal sensor fusion, investigate lightweight attention mechanisms, and test the framework across seasons and weather conditions.

ACKNOWLEDGMENT

The authors extend their appreciation to Northern Border University, Saudi Arabia, for supporting this work through project number NBU-CRP-2026-3030.

REFERENCES

- [1] K. Lv, H. Yu, X. Meng, *et al.*, "Small target detection model for open-pit coal mines based on UAV perspective," IEEE Access, 2026.
- [2] S. Ruan, Y. Hu, J. Liu, *et al.*, "An advanced crack detection method for slope management in open-pit mines: applying enhanced YOLOv8 network," International Journal of Mining, Reclamation and Environment, vol. 40, no. 1, pp. 70–87, 2026.
- [3] A. F. P. Tejada Peralta, R. Bakzadeh, S. Siahidouzazar, *et al.*, "Automated object detection and change quantification in underground mines using LiDAR point clouds and 360 image processing," Applied Sciences, vol. 16, no. 5, p. 2337, 2026.

- [4] D. Vosbein, S. Bunning, H. Khaniani, *et al.*, "Sensing and hazard detection in autonomous mine rescue," in AIAA SCITECH 2026 Forum, 2026, p. 0123.
- [5] Z. Wang, H. Zhou, H. Ye, *et al.*, "CMCF-DETR: a real-time lightweight DETR model for foreign object detection on coal mine conveyor belts," Journal of Real-Time Image Processing, vol. 23, no. 1, p. 5, 2026.
- [6] A. Zhao, Q. Zheng, Y. Zhao, *et al.*, "MSC-DETR: a lightweight real-time foreign object detection algorithm for underground belt conveyors in coal mines," Journal of King Saud University - Computer and Information Sciences, vol. 38, no. 3, p. 116, 2026.
- [7] L. Wang, K. Hu, X. Shi, *et al.*, "Foreign object detection on conveyor belts in coal mines based on RTA-YOLOv11," Applied Sciences, vol. 16, no. 3, p. 1375, 2026.
- [8] H. Li, B. Xie, X. Li, *et al.*, "New insights for enhancing the intelligence of coal mine: A two-stage method for unsupervised low-light image enhancement and lightweight detection," Alexandria Engineering Journal, vol. 141, pp. 24–51, 2026.
- [9] Y. Wang, K. You, Y. Jiang, *et al.*, "Mine hazardous obstacle segmentation for automated bulldozer with segment anything model," Engineering Applications of Artificial Intelligence, vol. 163, p. 113089, 2026.
- [10] P. Srinivasan, T. K. Nethra, K. Arulaanandham, *et al.*, "Real-time mine safety alert system for gas leakage detection using hybrid GWO-PSO optimized machine learning," in 2026 International Conference on Data Science, Agents and Artificial Intelligence (ICDSAAI), IEEE, 2026, pp. 1–6.
- [11] Z. Hu, "Open-pit mine YOLO dataset," Ultralytics Hub, 2026. [Online]. Available: <https://platform.ultralytics.com/zelin-hu-3/datasets/open-pit-mine-yolo>
- [12] G. Liu, L. Zhang, J. Lei, *et al.*, "Lightweight target detection and multi-target tracking for UAV inspection in open pit mines," Scientific Reports, vol. 16, no. 1, p. 8084, 2026.
- [13] H. M. Aljami, N. A. Alrowais, A. M. Alawajy, *et al.*, "Benchmarking YOLOv8 variants for object detection efficiency on Jetson Orin NX for edge computing applications," Computers, vol. 15, no. 2, p. 74, 2026.
- [14] W. Zhou, Y. Zou, H. Chai, *et al.*, "A UAV-TLS point-cloud fusion framework for three-dimensional characterization of mining-induced surface cracks," Remote Sensing, vol. 18, no. 14, p. 2425, 2026.