Quantization-Aware Training for Efficient Object Detection on FPGAs: Case Studies

Xuanshu Luo[1], Gabor Fogarasi[1], Alan Syrgak[1], Paul Walther[1], Martin Werner[1]

[1] Technical University of Munich, Germany - (xuanshu.luo, gabor.fogarasi, alan.syrgak, paul.walther, martin.werner)@tum.de

Keywords: Model Quantization, Quantization-aware Training, Object Detection, FPGA, Computer Vision, Deep Learning.

Abstract

Deploying object detection models for resource-constrained remote sensing applications necessitates on-board model inference capabilities. While Field Programmable Gate Arrays (FPGAs) offer massive parallelism as energy-efficient hardware platforms, model quantization remains essential to further balance computational efficiency with detection accuracy. Compared to post-training quantization methods that involve multiple-stage development with consistent dependency on domain datasets, quantization-aware training (QAT) integrates quantization constraints into training, providing a simpler pipeline for model compression. However, QAT introduces quantization errors to which smaller objects are more vulnerable. To address this issue, we propose object-scale-aware (OSA) regularization that amplifies quantization error penalties for smaller targets. Our approach is validated through two case studies: bird detection at airports and aerial-view building detection. We perform 8-bit QAT on YOLOX series models using the MVA2023 dataset and the Bavarian Building Dataset for the respective studies. Our method achieves up to 50.2 times inference acceleration with minimal accuracy loss on Xilinx Kria KV260 FPGAs compared to full-precision models. The ablation study and detection examples further demonstrate the effectiveness of OSA regularization in small object detection.

1. Introduction

Object detection is a fundamental computer vision task that locates and classifies instances of interest in images or videos (Zou et al., 2023), playing a pivotal role in geoscience and remote sensing applications, such as monitoring, planning, disaster mapping, and resource management (Chen et al., 2021, Wang et al., 2022, Wu et al., 2022, Ijaz et al., 2023, Caricchio et al., 2025). Recent advancements in deep learning have yielded object detection models pre-trained on large-scale image datasets, e.g., COCO (Lin et al., 2014) and Objects365 (Shao et al., 2019), which possess robust and comprehensive understanding of visual patterns and serve as dependable foundations for subsequent task-specific adaptations (Girshick, 2015, Ge et al., 2021). With the increasing availability of annotated remote sensing and earth observation imagery (Werner et al., 2023, Li et al., 2025), many approaches tailor pre-trained models regarding data modality adaptation and feature transfer (Macias et al., 2022, Yan et al., 2022, Nie et al., 2024, Qi et al., 2024, Caricchio et al., 2025), achieving superior object detection capabilities in corresponding tasks without the computational costs of training models from scratch (Girshick, 2015, Li et al., 2018).

However, many of these applications in practice are deployed on hardware platforms operating in environments with limited network connectivity and power supply, such as satellites and aerial vehicles (Chen et al., 2021, Wu et al., 2022, Qi et al., 2024), impeding stable communication with remote computing infrastructure. Therefore, it is imperative to redistribute model inference computation from the cloud to local resource-constrained edge devices (Li et al., 2020). In this context, model efficiency represents a feasibility-critical requirement to ensure fast on-board inference and circumvent reliance on networks, which particularly benefits latency-sensitive remote sensing applications, such as disaster response (Wang et al., 2022, Ijaz et al., 2023, Li et al., 2025) and environmental monitoring (Luo et al., 2023, Nie et al., 2024, Caricchio et al., 2025). Hence, a holistic consideration across hardware platform selections and model optimization strategies is essential (Deng et al., 2020).

Field-Programmable Gate Arrays (FPGAs) emerge as an ideal deployment platform for superior on-board deep learning model efficiency. An FPGA contains a large matrix of configurable logic blocks interconnected via a programmable routing fabric, enabling customizable circuit pipelines for parallelism during model inference. This architecture aligns with the feed-forward nature of deep learning models (Sze et al., 2017) and thus achieves superior performance-per-watt compared to general-purpose processors (Nguyen et al., 2020). Furthermore, its reconfigurability offers flexibility for post-deployment refinements and updates in situ, which is essential for remote sensing applications, where diverse sensor modalities and varying computational workloads demand adaptive processing architectures. Given these synergistic advantages, FPGAs have become a competitive platform for model inference acceleration (Xu et al., 2022, Macias et al., 2022, Nechi et al., 2023). We therefore adopt FPGAs as the deployment targets in this study.

Beyond hardware selection, model optimization can further enhance on-board inference efficiency for deploying models in resource-constrained environments. Quantization is a universal model compression technique that reduces the precision of model weights and activations, e.g., from 32-bit floating-point numbers to 8-bit integers, thereby reducing power consumption and memory usage (Horowitz, 2014, Han et al., 2016). While post-training quantization (PTQ) is widely adopted, it requires domain datasets throughout the entire development process, typically including domain adaptation, calibration, and potential fine-tuning (Wei et al., 2021, Zhang et al., 2022), significantly complicating model deployment and updates. In contrast, quantization-aware training (QAT) incorporates quantization constraints directly into the training procedure, requiring domain datasets only once to produce the final deployable domain-adapted quantized model, thereby establishing a fairly more streamlined development routine (Jacob et al., 2018).

Motivated by the succinctness of QAT, this study investigates its application to object detection tasks. Through signal-to-noise ratio analysis, we demonstrate that QAT induces quantization

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-2-2026-697-2026

errors inversely proportional to object scale, leading to larger quantization noise for smaller targets. This poses a substantial challenge for remote sensing and earth observation tasks, where small objects dominate detection targets, potentially degrading detection performance despite computational efficiency gains. To address this challenge, this paper proposes a QAT-integrated development pipeline with object-scale-aware (OSA) regularization that penalizes quantization errors for smaller objects more heavily during training. This regularization term is incrementally added to the detection loss function during QAT to facilitate convergence and ensure detection accuracy.

The proposed method is validated through two case studies: (1) bird detection for bird strike prevention at airports; (2) aerial-view building detection. Both studies employ YOLOX series models (Redmon et al., 2016, Ge et al., 2021) pre-trained on the COCO dataset (Lin et al., 2014) as foundations to perform QAT with OSA regularization. Specifically, single-precision floating-point (`FP32`) model weights and activations are quantized to 8-bit integers (`INT8`) and fine-tuned using the Bavaria Building Dataset (BBD) (Werner et al., 2023) and the MVA2023 small-bird detection dataset (Kondo et al., 2023) for the two studies, respectively. Models are deployed on Xilinx Kria KV260 FPGAs for evaluation. Empirical results illustrate that our method achieves up to 50.2× acceleration for end-to-end inference with negligible performance degradation compared to full-precision counterparts. The ablation study further demonstrates the effectiveness of OSA regularization in mitigating quantization noise for small objects in QAT pipelines[1].

2. Related Work

2.1 Object Detection Models

Object detection simultaneously localizes and classifies multiple objects in input vision data, producing bounding boxes along with associated class labels and confidence scores (Zou et al., 2023). A fundamental challenge in object detection is handling objects at vastly different scales within the same image. Feature Pyramid Networks (FPN) address this challenge by extracting a pyramidal hierarchy of feature maps with lateral connections (Lin et al., 2017), enabling models to detect objects across diverse scales, from small targets captured in high-resolution feature layers to large objects represented in low-resolution layers. The notation P_k is generally utilized to denote a specific feature map level in the pyramid, where the corresponding spatial downsampling stride $s = 2^k$, resulting in feature maps of $1/s$ input size. The FPN architecture yields multiscale representations that have proven particularly effective for remote sensing applications, where objects exhibit substantial scale variations (Wu et al., 2022, Qi et al., 2024).

In addition to effective feature extraction by FPN, model efficiency constitutes another criterion for model selection in this study. Modern object detection models can be broadly categorized into one-stage and two-stage detectors (Zou et al., 2023). We prioritize one-stage detectors that perform detection in a single forward pass (Girshick et al., 2014), avoiding the computational overhead of the separate region proposal and classification stages required by two-stage methods, which is essential for remote sensing applications demanding on-board model inference efficiency to satisfy stringent latency requirements (Martone et al., 2019, Wei et al., 2021, Zhang et al., 2022).

[1] The code is available at `https://github.com/tum-bgd/QATDet`.

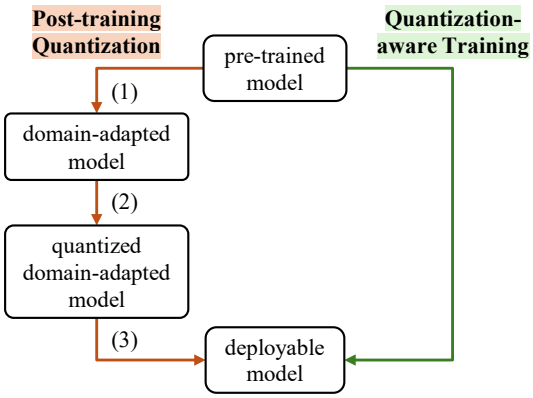

Figure 1. Comparison of development paradigms between post-training quantization and quantization-aware training.

The above analysis motivates us to select one-stage, FPN-based object detection architectures for development, with the iconic YOLOX series models (Ge et al., 2021) chosen as our implementation foundation. YOLOX models are developed based on conventional YOLO models in an anchor-free fashion that try to detect key/center points and estimate sizes rather than fitting into pre-defined anchor boxes (Redmon et al., 2016, Law and Deng, 2018). Such flexibility is particularly beneficial for remote sensing imagery with targets of arbitrary scale and aspect ratio (Tian et al., 2019). Given the on-board efficiency concerns addressed in this paper, we consider two light variants, YOLOX-Nano and YOLOX-Tiny, with 0.91 and 5.06 million parameters, respectively. Despite their lightweight design, experiments reveal that direct deployment of these compact models onto FPGAs for high-resolution remote sensing imagery, e.g., 4K resolution (2176×3840 pixels), results in inference latencies exceeding 20 seconds per frame, rendering them unsuitable for real-time applications. This observation underscores that model compression remains essential even for inherently small models for remote sensing applications in edge environments with latency requirements and high-resolution inputs, which reflects the motivation of this paper.

2.2 Model Quantization

Quantization reduces the numerical precision of model parameters and activations from higher-bit to lower-bit representations, thereby compressing model size and accelerating inference (Han et al., 2016, Jacob et al., 2018). Post-training quantization (PTQ) and quantization-aware training (QAT) are two predominant paradigms for model quantization. PTQ methods quantize models through statistical calibration on a small subset of training data (Krishnamoorthi, 2018, Wei et al., 2021). While seemingly computationally efficient, PTQ often suffers from accuracy degradation, as the quantization process lacks gradient feedback to compensate for discretization errors (Wang et al., 2019, Zhang et al., 2022). Meanwhile, selecting representative samples for calibration remains non-trivial and computationally expensive. Consequently, a fine-tuning step after calibration is typically essential to recover accuracy, yielding a conventional PTQ development routine comprising three steps as depicted in the left of Figure 1: (1) domain adaptation with potential data modality alignment, (2) quantization through calibration, (3) fine-tuning quantized models for accuracy restoration. Throughout the entire development process, the domain dataset is required by all three steps. This consistent domain data dependency significantly complicates model deployment.

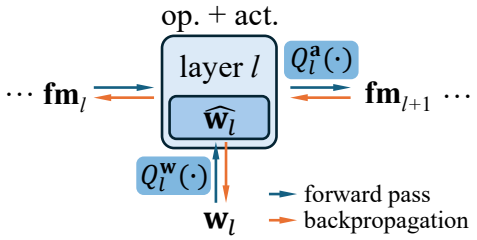

Figure 2. The QAT procedure for a given layer l, which contains layer operations (op.), e.g., convolution, and activation (act.).

In contrast, QAT simulates quantization effects during training by incorporating fake quantization operations in the forward pass while maintaining full-precision gradients in the backward pass to enable models to adapt their parameters to quantization constraints (Jacob et al., 2018, Krishnamoorthi, 2018). This gradient-guided adaptation enables QAT to achieve superior accuracy compared to PTQ, at the cost of increased training time. For domain adaptation scenarios where task-specific fine-tuning is required, QAT naturally integrates quantization into the adaptation process, producing deployment-ready quantized models in a single training phase, as illustrated in the right of Figure 1. Therefore, this paper employs QAT pipelines for development. However, as demonstrated in subsection 3.2, QAT introduces quantization errors in which smaller objects are more vulnerable, which motivates us to propose OSA regularization to address this issue as elaborated in subsection 3.3.

3. Methodology

This section elaborates on QAT principles and demonstrates that quantization error impedes small-scale object detection, which can be mitigated by the proposed OSA regularization.

3.1 Quantization-Aware Training

Integrating quantization into training requires full-precision model weights to be quantized for forward passes to simulate the resulting quantized model and measure the corresponding loss for backpropagation. This process impels models to learn parameters that are robust to the effects of discretization. Specifically, given a full-precision weight vector $\mathbf{w}$, its quantized version $\hat{\mathbf{w}}$ is calculated by the quantization function $Q(\cdot)$:

$$\hat{\mathbf{w}} = Q(\mathbf{w}, b) = \left\lfloor \frac{\text{clip}(\mathbf{w}, \alpha, \beta) - \alpha}{\Delta} + 0.5 \right\rfloor \cdot \Delta + \alpha, \quad (1)$$

where b is the target bit width, $\text{clip}(\cdot, \alpha, \beta)$ is a clipping function ranging from α to β, and $\Delta = (\beta - \alpha)/(2^b - 1)$ is the step size for the quantized value range $[\alpha, \beta]$. The clipping function ensures values are within range, while the $\lfloor \cdot + 0.5 \rfloor$ operation performs rounding to the nearest integer index before scaling back by Δ. Notably, α and β are learnable parameters, allowing the model to dynamically determine the optimal clipping range for each layer during training. A similar quantization procedure is also applied to activated feature maps $\mathbf{fm}$.

The QAT procedure is visualized in Figure 2. During the forward pass of layer l, full-precision weights $\mathbf{w}_l$ are first mapped to their quantized version $\hat{\mathbf{w}}_l$ through the weight quantization function $Q_l^{\mathbf{w}}(\cdot)$ to process the input $\mathbf{fm}_l$. The activated output is subsequently quantized by the activation quantization function $Q_l^{\mathbf{a}}(\cdot)$ to obtain the final output $\mathbf{fm}_{l+1}$, which serves as the input of the layer $l+1$. In this manner, the forward pass can simulate the deployment scenario at the target reduced precision. In contrast to the forward pass, backpropagation requires full-precision gradients to update $\mathbf{w}_l$ after loss calculation. However, the gradient of the loss $\mathcal{L}$ with respect to $\mathbf{w}_l$ is

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}_l} = \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{w}}_l} \cdot \frac{\partial \hat{\mathbf{w}}_l}{\partial \mathbf{w}_l} = \frac{\partial \mathcal{L}}{\partial Q_l^{\mathbf{w}}(\mathbf{w}_l, b)} \cdot \frac{\partial Q_l^{\mathbf{w}}(\mathbf{w}_l, b)}{\partial \mathbf{w}_l}, \quad (2)$$

where $\partial Q_l(\mathbf{w}_l, b)/\partial \mathbf{w}_l = 0$ almost everywhere, as $Q(\cdot)$ involves non-differentiable operations, e.g., clipping and rounding. This means $\mathbf{w}_l$ cannot be updated by conventional gradient descent, and consequently, neither can $\hat{\mathbf{w}}_l$.

To enable training, a common solution is to employ a straight-through estimator (STE) (Bengio et al., 2013), which assumes that $\partial Q_l(\mathbf{w}_l, b)/\partial \mathbf{w}_l = \mathbb{1}$, where $\mathbb{1}$ is an identity function. In practice, we refine this approach by dynamically calibrating gradients for layer-specific adaptation. Specifically, a learnable scaling factor $\theta_l^{\mathbf{w}}$ (rather than a simple $\mathbb{1}$) is assigned to the gradient of $Q_l^{\mathbf{w}}(\cdot)$, such that the effective gradient used for updating the full-precision weights is $\partial \mathcal{L}/\partial \mathbf{w}_l \approx \partial \mathcal{L}/\partial \hat{\mathbf{w}}_l \cdot \theta_l^{\mathbf{w}}$. Similarly, there are also $\theta_l^{\mathbf{a}}$ for $Q_l^{\mathbf{a}}(\cdot)$. This layer-wise scaling introduces more flexibility than $\mathbb{1}$ to facilitate convergence.

3.2 Quantization Error on Small Object Detection

Employing the above QAT pipeline allows us to obtain a deployable quantized model in a single fine-tuning step using domain datasets. However, we find that the discretization inherent in QAT introduces a scale-dependent quantization error. Concretely, consider an object *ob* with area A_{ob} represented by N_{ob} pixels at a pyramid level with stride s in FPN, we have $A_{ob} \propto s^2 N_{ob}$. For QAT using uniform quantization, the step size Δ is fixed, leading to the quantization error being statistically uniformly distributed in the range $[-\Delta/2, \Delta/2]$. Hence, the power (variance) of this error $\mathbf{p}_e$ is

$$\mathbf{p}_e = E[e^2] = \frac{\Delta^2}{12}. \quad (3)$$

For the quantization range $[\alpha, \beta]$, where $R = \beta - \alpha$, a quantizer with a target bit width of b bits induces $R \approx 2^b \cdot \Delta$. Assuming a sinusoidal signal that spans the full range of $[\alpha, \beta]$, the signal power $\mathbf{p}_s$ is $R^2/8$. We define the signal-to-quantization-error ratio (SQER) for a pixel p in decibels (dB) based on the signal-to-noise ratio formula as:

$$\begin{aligned} \text{SQER}_p &= 10 \log_{10} \frac{\mathbf{p}_s}{\mathbf{p}_e} = 10 \log_{10} \frac{R^2/8}{\Delta^2/12} \\ &\approx 10 \log_{10} \frac{R^2/8}{R^2/(12 \cdot 2^{2b})} \\ &= 6.02b + 1.76. \end{aligned} \quad (4)$$

For an object *ob* represented by N_{ob} pixels, without loss of generality, we assume the quantization error for *ob* is averaged over N_{ob} independent pixels. In this way, for each pixel, $\mathbf{p}_s$ remains the same, whereas $\mathbf{p}_e$ is divided by a factor of N_{ob} (by law of large numbers). Therefore, the per-pixel SQER for object *ob* is

$$\begin{aligned} \text{SQER}_{ob} &= 10 \log_{10} \frac{\mathbf{p}_s}{\mathbf{p}_e/N_{ob}} \\ &\approx 6.02b + 1.76 + 10 \log_{10} N_{ob}, \end{aligned} \quad (5)$$

which clearly demonstrates that smaller objects suffer more from quantization error introduced by QAT, as fewer N_{ob} leads to lower SQER given the same bit width b. In other words, quantization acts as a scale-dependent filter that degrades the representation quality for objects with a small spatial area. Considering the fact that small objects are the primary targets to detect in most remote sensing and earth observation tasks, their vulnerability to quantization error poses a fundamental barrier to maintaining detection performance using QAT.

3.3 Object-Scale-Aware Regularization

To address the scale-dependent quantization vulnerability identified in Equation 5, we propose an object-scale-aware (OSA) regularization term that dynamically adjusts the penalty for quantization-induced representation distortion based on object size. Drawing on knowledge distillation principles, we encourage the quantized student model to preserve feature similarity with the full-precision teacher model, with emphasis on smaller objects that exhibit lower SQER. Concretely, the standard detection loss $\mathcal{L}_{det}$ is augmented as

$$\mathcal{L} = \mathcal{L}_{det} + \lambda \cdot \sum_{P \in \mathcal{P}} \sum_{ob \in O} \frac{1}{A_{ob}} \cdot \left\| F_{ob}^{P} - \hat{F_{ob}^{P}} \right\|^2, \qquad (6)$$

where F_{ob}^{P} and $\hat{F_{ob}^{P}}$ are respective feature maps of ob extracted from the full-precision and quantized models at the pyramid level P in the FPN, $\|\cdot\|^2$ denotes the Euclidean norm, $\mathcal{P}$ defines the set of all selected P, O encompasses all objects in the current mini-batch, and λ controls the regularization strength. The derived $\mathcal{L}$ serves as the loss function for QAT.

Scale-Dependent Weighting. The inverse weighting by object area A_{ob} implements our core insight: smaller objects receive exponentially larger gradient contributions and thus stronger supervision signals during backpropagation. The Euclidean norm of feature map differences is weighted inversely by the true object area A_{ob} to prioritize optimization on smaller objects, as a smaller A_{ob} derives a larger regularization value. This regularization term dynamically adjusts according to A_{ob}, thereby making it object-scale-aware (OSA). For instance, an object with A_{ob} = 100 pixels receives 100× stronger regularization than an object with A_{ob} = 10,000 pixels. This design directly counteracts the SQER degradation pattern in Equation 5, where SQER decreases by $10 \log_{10} N_{ob}$ for smaller N_{ob}.

Cosine-Scheduled Regularization Strength. To prevent excessively strong regularization from impeding convergence during early training phases when the student quantized model is still adapting to quantization constraints, we implement a cosine-scheduled strength parameter:

$$\lambda(t) = \frac{\lambda_{max}}{2} \left(1 - \cos\left(\frac{\pi t}{T} \right) \right) \qquad (7)$$

where t denotes the current training step, T represents the total training steps, calculated as the product of the number of mini-batches and training epochs, and λ_{max} is the maximum regularization strength. We set $\lambda_{max} = 0.2$ in our experiments. This schedule gradually increases regularization strength to enable the model to establish reasonable quantized representations before strict feature alignment through OSA regularization.

Computational Overhead Considerations. Small objects are typically detected on high-resolution, low-stride feature maps. Therefore, it is sufficient to focus only on low pyramid levels by restricting $\mathcal{P} = [P_2, P_3]$ with $s = 4$ and $s = 8$, respectively, to reduce overhead while maintaining effectiveness. Meanwhile, since full-precision weights are maintained throughout QAT for gradient computation (Jacob et al., 2018, Krishnamoorthi, 2018), as illustrated in Figure 2, calculating the Euclidean norm between F_{ob}^{P} and $\hat{F_{ob}^{P}}$ introduces negligible computational overhead, while the memory footprint also remains manageable as P_2 and P_3 feature maps constitute only a fraction of the total feature pyramid hierarchy. Meanwhile, these additional costs only occur during training. Hence, quantized models for deployment require no additional computation during inference.

3.4 Model Deployment on FPGAs

FPGAs for Model Acceleration. On-board model efficiency stems from not only software-level optimization by model quantization, but also from effective utilization of hardware acceleration. FPGAs provide substantial acceleration for model inference through their inherently parallel architecture and customizable computational pipelines. Unlike CPUs, which execute instructions sequentially, or GPUs relying on single-instruction multiple-data architectures with limited flexibility, FPGAs enable the construction of customizable hardware accelerators tailored to the computational graph of a particular neural network. This customization allows for the instantiation of numerous dedicated multiply-accumulate units, which can be configured into optimized systolic arrays (Blaiech et al., 2019). These arrays maximize data reuse through local, high-bandwidth connections between processing elements, minimizing costly off-chip memory accesses and dramatically improving throughput (Nechi et al., 2023). Furthermore, the reconfigurable nature of FPGAs permits layer-specific optimizations, such as custom dataflow patterns and precision-adaptive arithmetic units, which collectively enable superior energy efficiency and inference speed compared to other platforms, particularly for quantized models where reduced-precision arithmetic can be fully exploited through hardware specialization.

Implementation. To leverage these architectural advantages in our experiments, we employ the Vitis AI toolchain (Kathail, 2020) for model deployment on Xilinx Kria KV260 FPGAs. The Vitis AI workflow begins with converting the quantized models into the ONNX representation. The Vitis AI quantizer then validates and optimizes the quantization parameters to ensure compatibility with the target FPGA architecture. Subsequently, the Vitis AI compiler transforms the quantized model into the `xmodel` format, i.e., a highly optimized binary representation designed for execution on Deep Learning Processing Units (DPUs), which are programmable and optimized for convolutional neural networks on Xilinx FPGAs. During compilation, the Vitis AI compiler performs extensive optimizations including operator fusion, memory allocation strategies, and instruction scheduling, to maximize hardware utilization. The resulting `xmodel` is then deployed on the FPGA, where the DPU executes inference operations with direct access to on-chip memory and optimized data movement patterns.

Synergy of QAT and FPGA Deployment. In this way, deploying a quantized model on FPGA can further improve model efficiency compared to relying solely on QAT. For instance, quantizing a model from `FP32` to `INT8` yields quarter-sized data representations for most operations, theoretically inducing up to 4× and 16× throughput for memory/computation-bounded layers, respectively. In practice, although the actual through-

Model	Type	AP (%) @0.5:0.95	Throughput (FPS)
YOLOX-Nano	T1	16.91	0.050
	T2	16.02 (-5.3%)	0.192 (3.8×)
	T3	16.02 (-5.3%)	2.510 (**50.2**×)
YOLOX-Tiny	T1	21.15	0.010
	T2	20.39 (-3.6%)	0.041 (4.1×)
	T3	20.39 (-3.6%)	0.493 (**49.3**×)

Table 1. Detection performance and throughput comparison of the Case Study 1: Bird detection at airports.

put gain is an observed measure that cannot be predicted precisely, the upper bound of less than 16× is evident. However, our experimental results demonstrate that deploying quantized models on FPGAs achieves up to 50.2× acceleration compared to direct execution of full-precision models on FPGAs without Vitis AI compilation, substantially exceeding theoretical 4×/16× speedup through quantization alone. This gap underscores that on-board model efficiency demands synergistic optimization across both software and hardware levels. While QAT effectively reduces computational load and memory footprint, FPGA deployment exploits these reductions through customized parallel architectures and optimized dataflow patterns.

4. Case Studies

We present two real-world case studies to evaluate our approach. Both studies share the following experimental settings.

Model & Training. YOLOX-Nano and YOLOX-Tiny models (Ge et al., 2021) are selected for domain adaptation via QAT, with merely 0.91 and 5.06 million parameters, respectively. We employ the model weights pre-trained on the COCO dataset (Lin et al., 2014). All models employ stochastic gradient descent with Nesterov momentum as the optimizer (Sutskever et al., 2013). Models are trained for 70 epochs with 10-epoch warmup. Model weights and activations are quantized from `FP32` to `INT8`. The learning rate is scheduled using cosine annealing (Loshchilov and Hutter, 2017) with an initial value of 0.01. The batch size is set to 8 due to high image resolution.

Evaluation. Models are deployed on Xilinx Kria KV260 FPGAs for evaluation. The detection performance is evaluated using the standard metric AP@0.5:0.95, which is the average precision across 10 intersection-over-union thresholds ranging from 0.5 to 0.95 (Everingham et al., 2010, Lin et al., 2014). Model efficiency is quantified by inference throughput, measured in frames per second (FPS). Performance comparisons are conducted across the following three model types:

- `T1` Full-precision models (`FP32`): to serve as the baseline.
- `T2` Quantized models (`INT8`): to present the performance gain attributed solely to model quantization.
- `T3` Compiled quantized models (`xmodel`): to assess the synergistic effect of QAT and FPGA-enabled acceleration.

Note that `T2` and `T3` employ identical quantized model weights, differing only in deployment configuration. Hence, `T2` and `T3` would achieve identical detection performance, with throughput differences attributable solely to hardware acceleration.

Figure 3. The bird detection demonstrator deployed at Airport Oberpfaffenhofen, Germany, with detection examples. The case is designed to accommodate the FPGA board with waterproof sealing, power supply, cable management, and air cooling.

4.1 Case Study 1: Bird Detection at Airports

The first case study implements real-time bird detection on FPGAs deployed at airports for bird strike prevention. To evaluate our method in a real-world scenario, we built a demonstrator as shown in Figure 3 and deployed it at Airport Oberpfaffenhofen, Germany. The FPGA board connects to an IP camera for video streaming through the RTSP protocol (Schulzrinne et al., 1998).

Dataset. For domain adaptation, we utilize the MVA2023 small bird detection dataset with 47,260 images for training and 9,759 images for validation (Kondo et al., 2023). All images are in 4K resolution (2176 × 3840 pixels) where bird sizes are relatively small, typically ranging from 20 to 200 pixels in width, which is suitable to examine the effectiveness of the OSA regularization.

Results & Discussion. Table 1 presents detection performance and throughput across three model types. Quantized YOLOX-Nano (`T2`, `T3`) achieves 16.02% AP@0.5:0.95 with a 5.3% relative performance degradation compared to the full-precision `T1`. The compiled quantized model (`T3`) demonstrates 50.2× throughput acceleration relative to `T1`, substantially exceeding the 3.8× speedup of `T2`. YOLOX-Tiny exhibits similar patterns, where `T2` and `T3` achieve 20.39% AP@0.5:0.95 (3.6% degradation from `T1`), and `T3` delivers 49.3× acceleration, significantly outperforming `T2` (4.1×) in throughput compared to `T1`.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-2-2026-697-2026 | © Author(s) 2026. CC BY 4.0 License.

Model	Type	AP (%) @0.5:0.95	Throughput (FPS)
YOLOX-Nano	T1	21.38	0.071
	T2	20.15 (-5.8%)	0.281 (4.0×)
	T3	20.15 (-5.8%)	3.331 (**46.9**×)
YOLOX-Tiny	T1	30.02	0.013
	T2	28.66 (-4.5%)	0.047 (3.6×)
	T3	28.66 (-4.5%)	0.602 (**46.3**×)

Table 2. Detection performance and throughput comparison of the Case Study 2: Aerial-view building detection.

These results reveal that quantization alone may be insufficient to enhance throughput for latency-sensitive applications. This case study is a representative example. While T2 achieves approximately $4\times$ throughput compared to T1, the resulting throughput remains impractical. Merely 0.192/0.041 FPS for YOLOX-Nano/Tiny means that processing a single 4K frame requires roughly 5/24 seconds, rendering real-time bird tracking impossible and invalidating on-site deployment.

In comparison, benefiting from FPGA-specific optimizations, e.g., custom dataflow, operator fusion, and parallel pipelines, T3 achieves 2.510 FPS, demonstrating that hardware-software co-optimization is not merely optional but essential for operational viability. This additional speedup beyond quantization alone transforms an impractical system into an operational solution, enabling fast frame-by-frame analysis for tracking bird trajectories. Therefore, effective edge deployment requires both model compression and utilization of hardware acceleration capabilities to unlock the full potential of edge platforms.

4.2 Case Study 2: Aerial-view Building Detection

The second case study addresses building detection from aerial orthophotos, which presents distinct challenges compared to bird detection in case study 1, including higher object density, more diverse object scales, and complex urban backgrounds.

Dataset. We use the Bavaria Building dataset (BBD) (Werner et al., 2023) for domain adaptation. BBD contains 18,205 orthophotos with a resolution of 40 cm and segmentation masks from OpenStreetMap and governmental building footprints. Images are cropped to 2496×2560 pixels for QAT and evaluation.

Results & Discussion. Similar to the bird detection case, results from Table 2 confirm that our QAT pipeline with OSA regularization maintains consistent optimization patterns in diverse detection scenarios. The quantized YOLOX-Nano (T2, T3) achieves 20.15% AP@0.5:0.95 with a 5.8% relative degradation from T1. Despite this modest accuracy trade-off, T3 delivers 46.9× throughput improvement, reaching 3.331 FPS compared to 0.071 FPS for T1. Similarly, YOLOX-Tiny (T2, T3) demonstrates 28.66% AP@0.5:0.95 (4.5% degradation from 30.02%), while T3 achieves 46.3× throughput compared to T1.

The higher absolute AP values compared to the bird detection case reflect the more favorable characteristics for detection in orthophotos, where buildings present more regular patterns with identical orientations (aerial view). Notably, acceleration factors remain consistent across all experiments, indicating that the optimization pipeline saturates the computational resources on FPGAs and scales effectively regardless of model capacity.

Figure 4. Comparison of the ground truth (above) and detected building footprints (below). Several small cabins in white rectangles are newly detected and absent from the BBD.

Figure 4 depicts the ground truth and the detection result of a sample image from BBD, revealing an additional benefit of the OSA regularization: several small buildings absent from the ground truth, highlighted in white rectangles, can be newly identified. This is because the OSA regularization guides models to enhance their sensitivity to detect small objects that may have been overlooked during dataset annotation. This capability has practical implications for cadastral mapping and urban planning, where exhaustive manual annotation is infeasible.

4.3 Ablation Study

For both case studies, we also perform QAT without the OSA regularization to validate its effectiveness. Table 3 demonstrates the critical contribution of OSA regularization to maintaining detection accuracy during QAT. Without OSA regularization, YOLOX-Nano experiences substantial performance degradation: 5.0% relative decrease for bird detection and 5.6% for building detection. YOLOX-Tiny exhibits even greater sensitivity to quantization without OSA, with degradations of 5.9% and 7.1% for the respective tasks. The more pronounced impact on YOLOX-Tiny suggests that larger models, despite their greater capacity, require stronger supervision to maintain feature representations under quantization constraints. The consistent performance improvements across both architectures and datasets validate that OSA regularization effectively addresses the scale-dependent vulnerability identified in Equation 5, where the inverse area weighting $1/A_{ob}$ provides crucial gradient signals for preserving small object representations during QAT.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-2-2026-697-2026 | © Author(s) 2026. CC BY 4.0 License.

Model	Setting	AP@0.5:0.95 (%)
Case study 1: bird detection at airports		
YOLOX-Nano	w/ OSA	16.02
	w/o OSA	15.22 (-5.0%)
YOLOX-Tiny	w/ OSA	20.39
	w/o OSA	19.19 (-5.9%)
Case study 2: aerial-view building detection		
YOLOX-Nano	w/ OSA	20.15
	w/o OSA	19.02 (-5.6%)
YOLOX-Tiny	w/ OSA	28.66
	w/o OSA	26.63 (-7.1%)

Table 3. Detection performance comparison among models (T2) with (w/) and without (w/o) OSA regularization.

5. Conclusion

This paper addresses the challenge of compute-intensive object detection model deployment in resource-constrained environments. Compared to PTQ, QAT provides a more succinct development pipeline to compress models without multi-stage domain data dependencies. We demonstrate that QAT particularly affects small object detection and propose OSA regularization to mitigate this issue by dynamically prioritizing feature fidelity for smaller objects during training. Experimental results from two case studies illustrate that our method can significantly accelerate object detection model execution on FPGAs with minimal detection performance degradation compared to full-precision counterparts, establishing a concise and effective model deployment workflow for object detection tasks in geoscience and remote sensing applications.

These findings establish QAT with OSA regularization as a practical pathway for edge deployment of object detection models, particularly for remote sensing applications where small object prevalence and computational constraints coincide. Future work could explore adaptive mixed-precision quantization schemes that dynamically adjust bit-widths based on object scale distributions, potentially achieving superior accuracy-efficiency trade-offs for heterogeneous detection scenarios.

References

Bengio, Y., Léonard, N., Courville, A., 2013. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation. *arXiv preprint arXiv:1308.3432*.

Blaiech, A. G., Ben Khalifa, K., Valderrama, C., Fernandes, M. A., Bedoui, M. H., 2019. A Survey and Taxonomy of FPGA-based Deep Learning Accelerators. *Journal of Systems Architecture*, 98, 331-345.

Caricchio, C., Mendonça, L. F., Lentini, C. A. D., Lima, A. T. C., Silva, D. O., Meirelles e Góes, P. H., 2025. YOLOv8 Neural Network Application for Noncollaborative Vessel Detection Using Sentinel-1 SAR Data: A Case Study. *IEEE Geoscience and Remote Sensing Letters*, 22, 1-5.

Chen, W., Wang, H., Li, H., Li, Q., Yang, Y., Yang, K., 2021. Real-time garbage object detection with data augmentation and feature fusion using SUAV low-altitude remote sensing images. *IEEE Geoscience and Remote Sensing Letters*, 19, 1–5.

Deng, L., Li, G., Han, S., Shi, L., Xie, Y., 2020. Model Compression and Hardware Acceleration for Neural Networks: A Comprehensive Survey. *Proceedings of the IEEE*, 108(4), 485-532.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., Zisserman, A., 2010. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2), 303-338.

Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J., 2021. YOLOX: Exceeding YOLO Series in 2021. *arXiv preprint arXiv:2107.08430*.

Girshick, R., 2015. Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Han, S., Mao, H., Dally, W. J., 2016. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *International Conference on Learning Representations (ICLR)*.

Horowitz, M., 2014. 1.1 computing’s energy problem (and what we can do about it). *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, IEEE, 10–14.

Ijaz, H., Ahmad, R., Ahmed, R., Ahmed, W., Kai, Y., Jun, W., 2023. A UAV-Assisted Edge Framework for Real-Time Disaster Management. *IEEE Transactions on Geoscience and Remote Sensing*, 61, 1-13.

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D., 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kathail, V., 2020. Xilinx vitis unified software platform. *FPGA ’20: Proceedings of the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, Association for Computing Machinery, New York, NY, USA, 173–174.

Kondo, Y., Ukita, N., Yamaguchi, T., Hou, H.-Y., Shen, M.-Y., Hsu, C.-C., Huang, E.-M., Huang, Y.-C., Xia, Y.-C., Wang, C.-Y., Lee, C.-Y., Huo, D., Kastner, M. A., Liu, T., Kawanishi, Y., Hirayama, T., Komamizu, T., Ide, I., Shinya, Y., Liu, X., Liang, G., Yasui, S., 2023. MVA2023 Small Object Detection Challenge for Spotting Birds: Dataset, Methods, and Results. *2023 18th International Conference on Machine Vision and Applications (MVA)*.

Krishnamoorthi, R., 2018. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*.

Law, H., Deng, J., 2018. CornerNet: Detecting objects as paired keypoints. *Proceedings of the European Conference on Computer Vision (ECCV)*.

Li, E., Zeng, L., Zhou, Z., Chen, X., 2020. Edge AI: On-Demand Accelerating Deep Neural Network Inference via Edge Computing. *IEEE Transactions on Wireless Communications*, 19(1), 447-457.

Li, H., Deuser, F., Yin, W., Luo, X., Walther, P., Mai, G., Huang, W., Werner, M., 2025. Cross-view geolocalization and disaster mapping with street-view and VHR satellite imagery: A case study of Hurricane IAN. *ISPRS Journal of Photogrammetry and Remote Sensing*, 220, 841-854.

Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., Sun, J., 2018. DetNet: Design Backbone for Object Detection. *Proceedings of the European Conference on Computer Vision (ECCV)*.

Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature Pyramid Networks for Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C. L., 2014. Microsoft COCO: Common Objects in Context. *Computer Vision – ECCV 2014*, Springer International Publishing, Cham, 740–755.

Loshchilov, I., Hutter, F., 2017. SGDR: Stochastic Gradient Descent with Warm Restarts. *International Conference on Learning Representations*.

Luo, X., Walther, P., Mansour, W., Teuscher, B., Zollner, J. M., Li, H., Werner, M., 2023. Exploring GeoAI Methods for Supraglacial Lake Mapping on Greenland Ice Sheet. *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, SIGSPATIAL ’23, Association for Computing Machinery, New York, NY, USA.

Macias, R., Bernabé, S., Báscones, D., González, C., 2022. FPGA Implementation of a Hardware Optimized Automatic Target Detection and Classification Algorithm for Hyperspectral Image Analysis. *IEEE Geoscience and Remote Sensing Letters*, 19, 1-5.

Martone, M., Villano, M., Younis, M., Krieger, G., 2019. Efficient Onboard Quantization for Multichannel SAR Systems. *IEEE Geoscience and Remote Sensing Letters*, 16(12), 1859-1863.

Nechi, A., Groth, L., Mulhem, S., Merchant, F., Buchty, R., Berekovic, M., 2023. FPGA-based Deep Learning Inference Accelerators: Where Are We Standing? *ACM Trans. Reconfigurable Technol. Syst.*, 16(4). https://doi.org/10.1145/3613963.

Nguyen, T., Williams, S., Siracusa, M., MacLean, C., Doerfler, D., Wright, N. J., 2020. The Performance and Energy Efficiency Potential of FPGAs in Scientific Computing. *2020 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*, 8–19.

Nie, J., Sun, H., Sun, X., Ni, L., Gao, L., 2024. Cross-Modal Feature Fusion and Interaction Strategy for CNN-Transformer-Based Object Detection in Visual and Infrared Remote Sensing Imagery. *IEEE Geoscience and Remote Sensing Letters*, 21, 1-5.

Qi, S., Song, X., Shang, T., Hu, X., Han, K., 2024. MSFE-YOLO: An Improved YOLOv8 Network for Object Detection on Drone View. *IEEE Geoscience and Remote Sensing Letters*.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Schulzrinne, H., Rao, A., Lanphier, R., 1998. Real time streaming protocol (RTSP). Technical report, RealNetworks, Netscape, and Columbia University.

Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., Sun, J., 2019. Objects365: A Large-Scale, High-Quality Dataset for Object Detection. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Sutskever, I., Martens, J., Dahl, G., Hinton, G., 2013. On the Importance of Initialization and Momentum in Deep Learning. S. Dasgupta, D. McAllester (eds), *Proceedings of the 30th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 28, PMLR, Atlanta, Georgia, USA, 1139–1147.

Sze, V., Chen, Y.-H., Yang, T.-J., Emer, J. S., 2017. Efficient Processing of Deep Neural Networks: A Tutorial and Survey. *Proceedings of the IEEE*, 105(12), 2295-2329.

Tian, Z., Shen, C., Chen, H., He, T., 2019. FCOS: Fully Convolutional One-Stage Object Detection. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Wang, H., Cao, H., Kai, Y., Bai, H., Chen, X., Yang, Y., Xing, L., Zhou, C., 2022. Multi-Source Remote Sensing Intelligent Characterization Technique-Based Disaster Regions Detection in High-Altitude Mountain Forest Areas. *IEEE Geoscience and Remote Sensing Letters*, 19, 1-5.

Wang, K., Liu, Z., Lin, Y., Lin, J., Han, S., 2019. HAQ: Hardware-Aware Automated Quantization with Mixed Precision. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wei, X., Chen, H., Liu, W., Xie, Y., 2021. Mixed-Precision Quantization for CNN-Based Remote Sensing Scene Classification. *IEEE Geoscience and Remote Sensing Letters*, 18(10), 1721-1725.

Werner, M., Li, H., Zollner, J. M., Teuscher, B., Deuser, F., 2023. Bavaria Buildings - A Novel Dataset for Building Footprint Extraction, Instance Segmentation, and Data Quality Estimation. *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, SIGSPATIAL ’23, Association for Computing Machinery, New York, NY, USA.

Wu, W., Fan, X., Qu, H., Yang, X., Tjahjadi, T., 2022. TCD-Net: Tree Crown Detection From UAV Optical Images Using Uncertainty-Aware One-Stage Network. *IEEE Geoscience and Remote Sensing Letters*, 19, 1-5.

Xu, C., Jiang, S., Luo, G., Sun, G., An, N., Huang, G., Liu, X., 2022. The Case for FPGA-Based Edge Computing. *IEEE Transactions on Mobile Computing*, 21(7), 2610-2619.

Yan, H., Zhang, E., Wang, J., Leng, C., Peng, J., 2022. MTFFN: Multimodal Transfer Feature Fusion Network for Hyperspectral Image Classification. *IEEE Geoscience and Remote Sensing Letters*, 19, 1-5.

Zhang, R., Jiang, X., An, J., Cui, T., 2022. Data-Free Low-Bit Quantization for Remote Sensing Object Detection. *IEEE Geoscience and Remote Sensing Letters*, 19, 1-5.

Zou, Z., Chen, K., Shi, Z., Guo, Y., Ye, J., 2023. Object Detection in 20 Years: A Survey. *Proceedings of the IEEE*, 111(3), 257-276.