Evaluation of the Robustness-Runtime Efficiency Trade-Off of Edge AI Models in UXO Localisation and Classification

Gheorghe-Marian Craioveanu
Automation and Industrial Informatics
University Politehnica of Bucharest
Bucharest, Romania
gheorghe.craioveanu@stud.acs.upb.ro

Grigore Stamatescu
Automation and Industrial Informatics
University Politehnica of Bucharest
Bucharest, Romania
grigore.stamatescu@upb.ro

***Abstract*—Real time localisation and classification of Unexploded Ordnance (UXO) can significantly benefit from advanced new model compression and quantization techniques towards embedded deployment on resource constrained fixed or mobile hardware platforms. This can extend the applicability, usefulness and adoption by first responders of such methods in real-world scenarios with significant social and environmental benefits. The proposed methodology considers the emergence of multiple frameworks and tools that have now become available to automate the comparative assessment of state-of-the-art image classification edge AI model. As main results, we present a quantitative evaluation of the robustness-runtime efficiency trade-off for representative CNN-based vision model and a parametrization discussion on a reference public UXO dataset. The approach is validated through deployment and experiments using a reference embedded GPU development board i.e. the Nvidia Xavier NX.**

***Index Terms*—image processing, neural networks, edge computing, uxo, optimization.**

I. Introduction

The 2003 Protocol on Explosive Remnants of War [1], as part of the Geneva Conventions, defines Unexploded Ordnance (UXO) as *"... explosive ordnance that has been primed, fused, armed, or otherwise prepared for use and used in an armed conflict. It may have been fired, dropped, launched or projected and should have exploded but failed to do so."*. UXO pose significant risks to personal health and safety and states have the post-conflict obligation to assure the recording, storage and release of information regarding UXO. New intelligent systems can thus support cost-effective and large scale UXO detection, localisation and classification, with wide reaching humanitarian impact across the globe.

The localisation and classification of unexploded ordnance (UXO), such as mortar bombs, grenades, and projectiles from past armed conflicts, has become a topic of significant interdisciplinary interest within the scientific community, for the immediate advantage of finding suitable methods: saving human lifes. This subject presents a series of challenges arising from the complexity of the scene, the diverse nature of the munitions, and the limited mobility of intelligent systems. The complexity of the scene and munitions is reflected in the variety and conditions in which these munitions can be found: oxidized, incomplete, or improvised. Our recent research [2] has demonstrated that a granular, multi-model approach integrated within a two-step deep learning methodology can address these challenges by handling exceptions and solving identified problems. In terms of mobility limitations, to define the necessity of portable hardware platform, it is important to note that UXOs are often located in hard-to-access areas. In the current study, we further refine this methodology by deploying intelligent systems on an ultra-portable AArch64 device, which includes CUDA compatibility for rapid parallel computation making an additional iteration in our study by eliminating the need for a server to perform inference for complex models. Models and the software framework were chosen in direct connection to the State-of-the-Art (SoA), while hyperparameter tuning was carried out specifically for this dataset and task. The proposed goal is supported through the use of a complete and representative dataset developed in collaboration with field experts [3].

The objectives of the current study include the testing of modern arhitectures on the updated open UXO dataset, applying post-quantization and other advanced techniques for optimizing models in edge-deployment e.g. through the experimental hardware evaluation on Jetson Xavier NX embedded development modules. The aim of the scientific work is to provide new capabilities and perspectives for UXO detection and identification, complementing and enhancing existing research dedicated to mitigation of risks associated with unexploded ordnances.

The key contributions of this study for novel AI-based embedded UXO localisation and classification are considered to be as follows:

- Fine-tuning of the RetinaNet with FocalLoss edge artificial intelligence model with multiple hyperparameters on a recent open specialised dataset for UXO localisation and classification (CTX-UXO);
- Testing, optimization and post-quantization of the trained models on multiple GPU capabilities: A100, T4, RTX4090 and NVidia Jetson device;

- Quantitative performance metric evaluation and benchmarking enabling replicable analysis and implementation of the proposed training and inference pipeline.

Referring to the structure of the paper, Section II reviews the specialized literature on inference using embedded systems with models trained or fine-tuned for detecting munitions or similar objects, limitations and approaches, as well as the characteristics of the embedded system utilized and selection of the frameworks involved. Based on these insights, Section III introduces the proposed model to be used, optimization methods and used frameworks, all included in a well defined and reproductible methodology. Following the methodology outlined in Section III, in Section IV the training and validation losses are presented, thereby discussing the diversity of the dataset and the significance of the utilized model. Two categories of performance metrics are considered and evaluated for this purpose: primary metrics (Mean Average Precision, Recall, F1-Score) and secondary metrics in terms of memory load and inference time (as how many frames per second can be processed) to identify the tradeoffs between robustness and runtime efficiency, getting in the scene optimization techniques, testing and deployment on edge devices.

II. Related Work

In previous work [2], we developed a robust system for detecting and identifying unexploded ordnance using a two-step deep learning methodology. The approach integrated domain-specific knowledge with advanced image processing techniques to address the challenges associated with UXO detection. Specifically, we proposed a pipeline with multiple models, leveraging the YOLOv8X architecture, the most performant and the largest, in terms of parameters and layers, model from the v8-series. This approach demonstrated promising results, the bottleneck of that moment being the envinroment constrained computational resources. Starting from this bottleneck, we adress the issues by proposing the use of an embedded system designed for AI inference in the form of the Nvidia Jetson Xavier NX family of embedded GPU development boards.

The Nvidia Jetson Xavier NX is equipped with a 6-core Carmel ARM v8.2 64-bit CPU operating at up to 2.2 GHz, and a 384-core Volta GPU featuring 48 Tensor Cores. It also includes 8GB of LPDDR4x memory and supports up to 21 TOPS (Tera Operations Per Second) of AI performance. The portable platform enables real-time execution of optimized deep learning models, making it suitable for applications in resource-constrained environments such as field deployments for UXO detection. Its compact size and the power consumption (5w-15w, critical 15-20W) further enhance its viability for on-site processing.

Regarding the model selection stage, the focus has been on finding a model able to handle hard negative examples [4] while keeping a good balance on primary metrics. Based on several research results published in [5], RetinaNet is a widely used object detection model due to its balanced architecture that delivers both accuracy and computational efficiency. Additionally, RetinaNet has demonstrated very good optimization capabilities and implementation for real-time solutions, such as face detection. RetinaNet employs Focal Loss [6], which addresses class imbalance by down-weighting negatives sampling and focusing more on hard-to-classify examples. On the dedicated embedded Nvidia Jetson platforms, this ensures robust performance without requiring excessively complex models.

In related work based on perfomance comparison on the edges, the authors of [7] conducted a detailed performance evaluation of several machine learning frameworks, including TensorFlow, Caffe2, PyTorch, MXNet, and TensorFlow Lite, across various edge devices. Notably, their experiments used CNNs as benchmark models to evaluate latency, memory footprint, and energy consumption on platforms such as the Nvidia Jetson devices. PyTorch demonstrated superior inference time on the Jetson devices, achieving an average inference latency of 1.35 seconds for 100 runs. In contrast, TensorFlow exhibited higher latency, emphasizing the need for optimization when using this framework. On smaller models like SqueezeNet, Caffe2 and MXNet outperformed other frameworks, with an inference latency of only 0.375 seconds per run, showcasing its efficiency in handling lightweight architectures. The Jetson device showed a wide range of memory utilization across frameworks. TensorFlow consumed up to 3000MB, while PyTorch utilized only 1301MB for the same model, indicating a significant variation in resource demands that must be considered during deployment. Caffe2 demonstrated the lowest memory usage for low-size models. Energy consumption represents another relevant aspect for our study, because the edge device with the results from the next section can be leveraged by deploying the resulting models on mobile platforms such as UAV systems or explosive ordnance disposal (EOD) robots that can access hard-to-reach old warzones in difficult terrain conditions. Correlated strongly with latency, PyTorch and MXNet are showing the best energy efficiency. The study highlighted that model loading often consumed more energy than inference, suggesting optimization opportunities in pre-processing and loading stages.

From these findings, we draw several actionable strategies for our study:

- PyTorch represents the most reliable framework in the specific case to be used for training/fine-tuning and later for the inference on Jetson devices;
- Pytorch, as the other frameworks, lacks in pre-processing stage and, depending on model and hardware arhitecture, in inference time. For adressing this issue, this paper will introduce optimization techniques by leveraging TensorRT framework.

Regarding TensorRT, it is a high-performance deep learning inference optimizer and runtime library developed by NVIDIA, suitable for Jetson Devices [8], with real-time capabilities [9]. It works by performing several optimizations [8], such as combining multiple neural network layers into one to reduce computation overhead, precision calibration

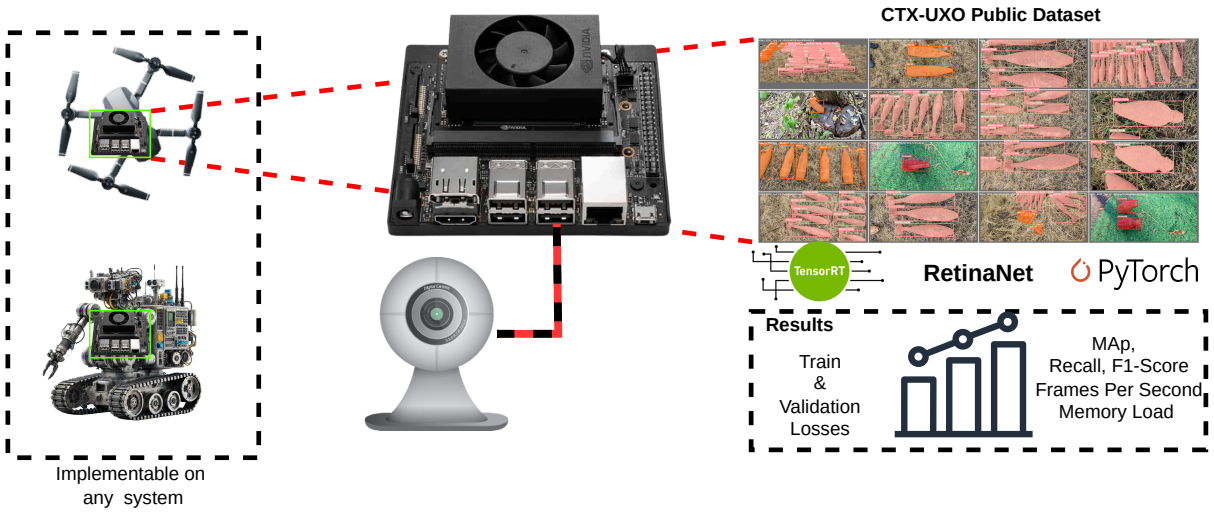

Fig. 1. The implementation of UXO detection models optimized for Jetson devices provides new capabilities for the safe operation of unmanned systems

(from FP32 to FP16 or Int8 if supported), selecting the most efficient kernels for that specific target hardware, optimizing memory usage, all in a high-level implementation. The advantages of TensorRT include reduced inference latency, higher throughput, and lower memory consumption. [10] However, the optimizations, especially the floating point precision calibration, can affect primary metrics, such as mean average precision, particularly when using lower precision like FP16. Nevertheless, careful calibration and tuning can mitigate these effects, ensuring that the models perform optimally in both remote server and edge environments.

The past approaches in UXO detection, stemming from [11], [12] and [13], were focused particularly on algorithms and finding suitable datasets, with limited real word implementation aspects and constraints. [11] developed a detection pipeline leveraging advanced computer vision techniques. The core methodology involved using Convolutional Neural Networks (CNNs) to detect specific cluster submunitions within aerial imagery. Their approach emphasized high detection accuracy, achieved by training the models on a dataset with augmentation techniques to account for various environmental and lighting conditions. The results indicated an accuracy of over 95%, showcasing the potential of CNNs in UXO detection. Still, the dataset contains similarities by including just one type of submunitions. The main purpose in our paper is to identify multiple classes, despite the type and manufacturer, instead of a specific subclass. Also, the reliance on high computational resources for processing large imagery datasets highlighted a critical limitation for deployment in resource-constrained environments such as edge devices.

It can be highlighted that UAVs [12] are a viable option, the immediate advantage of unmaned systems being that there is no risk for the EOD operator. Still, the approach of the researchers [12] is not real-time, inference and image processing take place later on a GPU server. The advantage of the Jetson device's portability is that it can be deployed on any type of unmanned device, as shown in Figure 1, so the results of the current study can be leveraged.

The increasing adoption of edge machine learning models for UXO detection necessitates a careful examination of the trade-offs between robustness and runtime efficiency. While high-performance models such as very deep Convolutional Neural Networks/Vision Transformers achieve superior detection precision, they often demand substantial computational power and memory, posing challenges for real-time and on-device processing. Conversely, lightweight models optimized for edge deployment may sacrifice accuracy/precision for efficiency and capabilities of implementation, requiring strategies such as model quantization, pruning, and knowledge distillation to mitigate performance losses. The high accuracy achieved by CNNs and multispectral imaging systems highlights the importance of robust model training and data preprocessing. In the same, the limitations associated with computational demands emphasize the need for optimizing models for edge deployment. This balance is central to our investigation, guiding the design and evaluation of edge neural networks models able of reliable UXO detection in resource-constrained environment. Based on these aspects, the Results Section includes an analysis from both primary and secondary metric perspectives to determine how model performance is influenced by the optimization techniques used, ensuring a balanced trade-off between robustness and runtime efficiency, with capabilities of real-time implementation.

A very important aspect to highlight is that by implementing high-performance models on Jetson devices, a new path toward portability is opened through the possibility of scaling the developed modules in UAV systems, EOD robots, and other hardware systems. This provides a safety lever for rescuers, as they no longer need to get too close to UXOs for identification.

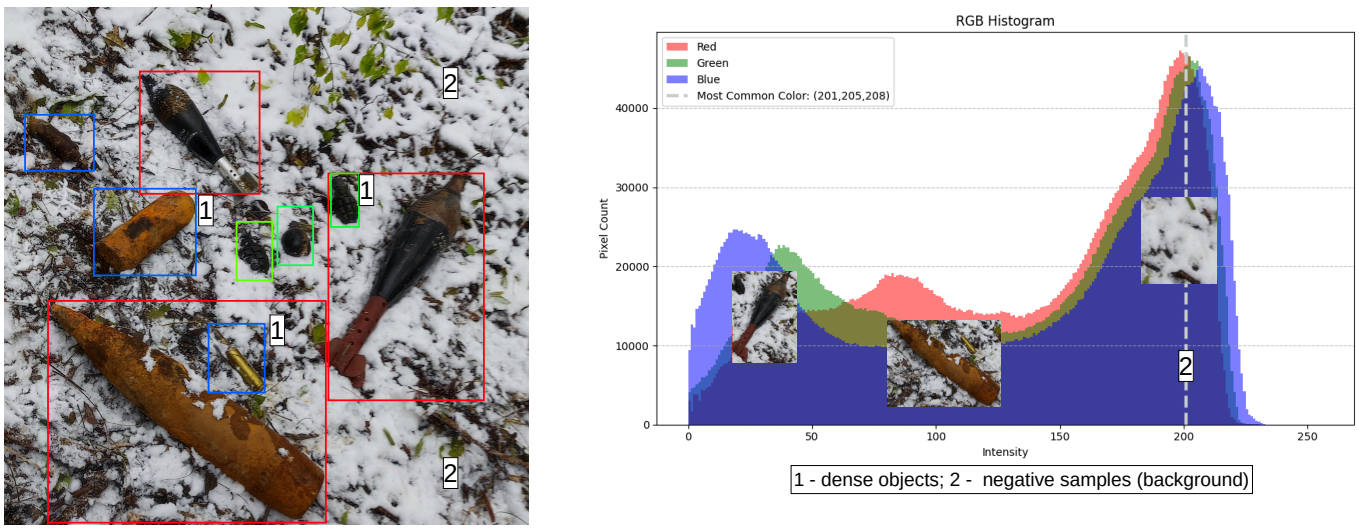

Fig. 2. Overlapping boundary boxes (marked as "1") and the number of negative samples (marked as "2") can pose a challenge in UXO detection.

III. Methodology

In addressing the issues raised, and by correlating with the state of the art, the present study will perform fine-tuning on the RetinaNet image classification model. RetinaNet stands out as a highly suitable architecture for the task of unexploded ordnance (UXO) detection, owing to its ability to combine high precision with computational efficiency (one-stage detection arhitecture) while addressing the challenges presented by unexploded ordnance detection. The detection of UXOs involves handling datasets that are often imbalanced, where the majority of the data represents background clutter, as it is exemplified in Figure 2, and only a small fraction corresponds to actual ordnance. In Figure 3 an image from the CTX-UXO dataset is illustrated where more UXOs are overlapping, so the risk is that even the bounding boxes used for labeling overlap, which means another instance can effectively be introduced as noise. Additionally, these datasets include objects of varying sizes and shapes, adding further complexity to the detection process. RetinaNet is specifically designed to address these challenges effectively, while keeping a low memory load and good inference speed, this being the reason why is suitable in our edge-deployment research.

Fig. 3. Dense Instances Image Sample from CTX-UXO Dataset

At the core of RetinaNet's suitability for UXO detection is its use of focal loss [14], a loss function designed to mitigate the effects of class imbalance and dense object detection, as an updated version of CrossEntropy Loss, while keeping a good inference speed and memory load:

$$FL(y, \hat{y}) = -\sum_{i=1}^{C} \alpha_i (1 - \hat{y}_i)^{\gamma} y_i \log(\hat{y}_i) \tag{1}$$

where:

- C is the total number of classes;
- $y_i \in \{0, 1\}$ is the ground truth label for class i;
- $\hat{y}_i$ is the predicted probability for class i;
- $(1-\hat{y}_i)^{\gamma}$ is the modulating factor, with $\gamma \geq 0$ controlling the focus on hard-to-classify examples. In the current work, γ is 2;
- α_i is a weighting factor to address class imbalance for class i.

The used loss for boundary boxes regression is the following:

$$L_{\text{bbox}}(t, t^*) = \sum_{i \in \{x,y,w,h\}} \text{smooth}_{L_1}(t_i - t_i^*) \tag{2}$$

smooth L1 being:

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1, \\ |x| - 0.5 & \text{otherwise.} \end{cases} \tag{3}$$

where:

- $t = (t_x, t_y, t_w, t_h)$ are the predicted bounding box;
- $t^* = (t_x^*, t_y^*, t_w^*, t_h^*)$ are the ground truth bounding box;

In UXO datasets, the overwhelming number of background samples (negative samples) can lead traditional detection models to focus more on these easy examples, at the expense of missing the rarer and more critical ordnance examples. Focal loss dynamically reduces the impact of well-classified

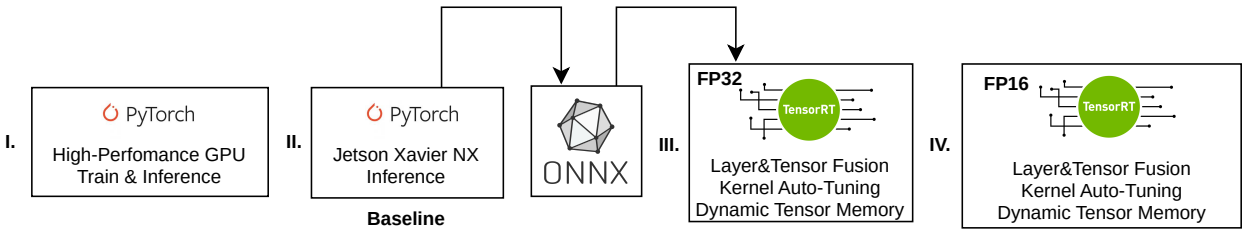

Fig. 4. Methodology to measure the perfomances for edge deployment for UXO Detection on Jetson Xavier NX

examples, ensuring that the model concentrates on harder-to-classify UXOs. This makes RetinaNet particularly adept at identifying rare and difficult-to-detect UXOs, even when they are small or partially obscured, as shown in Figure 3.

Another significant advantage of RetinaNet is its one-stage detection design [5], which streamlines the process by integrating object classification and bounding box regression into a single step. This design not only reduces inference time compared to two-stage detectors like Faster R-CNN but also ensures a better trade-off between speed and accuracy.

For UXO detection, especially in edge-computing environments where real-time processing is crucial, this efficiency is very important, being the opportunity to scale the results into a bigger project - like UAVs integration. UXOs are often located in visually complex and noisy backgrounds, such as fields, forests. RetinaNet's multi-scale feature extraction ensures it can differentiate UXOs from the surrounding noise, maintaining reliable performance under challenging conditions [14]. To evaluate these architectures, they will be deployed and run in the cloud on A100 GPU, RTX 4090 GPU, T4 GPU as well as on Nvidia Jetson device.

The used dataset is CTX-UXO: A Comprehensive Dataset for Detection and Identification of UneXploded Ordnances [3], meticulously curated to include real munitions and replicas in diverse orientations and physical conditions, on different seasons, ensuring robustness and applicability across a wide range of scenarios. It includes 3 classes (mortar bomb, grenade, projectiles), 12 543 positive instances with an input size for fine-tuning, validation and testing of 800 px x 800 px. Considering that some images may contain more instances than others, stratified sampling is necessary. Stratified sampling [15] is a technique used to ensure that each class is proportionally represented in the sampled subsets, preserving the original class distribution of the dataset. This method increase prediction accuracy as it was demonstrated by the scientific community. PyTorch represents the main framework used for training, validation and testing, while TensorRT will be used for edge-optimization techniques and testing on Nvidia Jetson device. To optimize the Unexploded Ordnance (UXO) detection model on the Jetson Xavier NX platform, the methodology from Figure 4 was implemented. The pre-trained models were fine-tuned using a custom UXO dataset and then exported to the Open Neural Network Exchange (ONNX) format to ensure compatibility with TensorRT. The ONNX model was further optimized using TensorRT, employing Half-precision floating-point (FP16). For comparison, PyTorch float32 initial model will be tested too on Nvidia Jetson. TensorRT automatically fused model layers and selected the most efficient CUDA kernels for the Jetson Xavier NX hardware. This process enhanced the utilization of GPU resources and reduced latency.

IV. Results

The fine-tuning was done using PyTorch on Nvidia A100 GPU, batch size 32, with an input size of 800 x 800, Ampere Arhitecture, 40GB VRam, base RetinaNet model. For consistency, the dataset split is stratified [15] per classes: 70% training data, 20% validation data, 10% testing data. No augmentations were used for training data, the dataset being contextual representative.

The loss results are composed from the sum of two losses: classification loss and regression loss (boundary boxes). For the purpose of visualizing and understanding the model capabilities. Figure 5 displays the plotted loss values for classification and separately for boundary box regression - train.

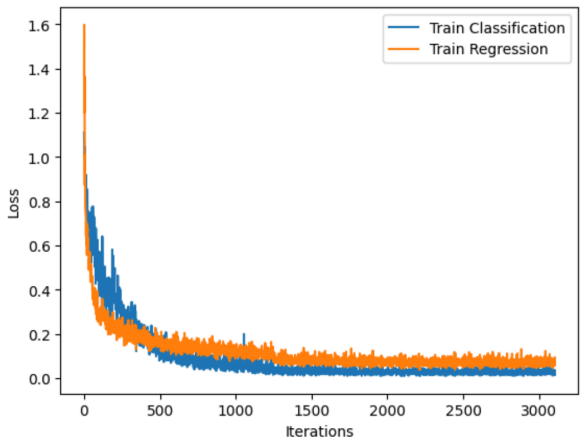

Fig. 5. Train losses for classification and regression - boundary boxes.

In Figure 6, the loss for train and validation is presented.

For ensuring the training efficiency and robustness validation, by searching for the convergence without overfitting on the training data, we employed a dynamic learning rate scheduler, Reduce Learning Rate on Plateau. This scheduler

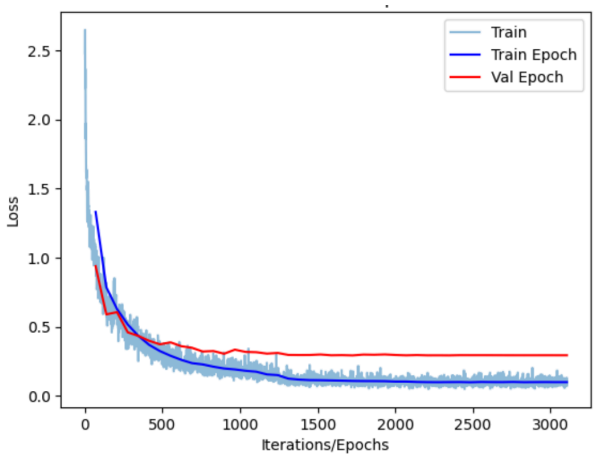

Fig. 6. Train and validation converged losses

adaptively lowers the learning rate when the monitored performance metric (being the validation loss value) stagnates over successive training epochs. The coefficient γ for Focal Loss was set to 2 after multiple successive fine-tuning iterations.

Regarding the primary metrics, Mean Average Precision, calculated at an Intersection over Union (IoU) threshold of 0.50 (mAP50) and as average at varying IoU between 0.50 and 0.95 (mAP50-95), Recall and F1-Score will be utilized. For secondary metrics, the average of the memory load and Frames per Second (FPS) rate will be used. All metrics in Table 1 are calculated on the Cloud platform (A100, RTX 4090, T4) and on Jetson Xavier NX.

TABLE I
PERFORMANCE OF RESULTED MODELS. RECALL AND F1-SCORED ARE CALCULATED FOR A STRICT IOU 50-95. F1-SCORE IS CALCULATED BASED ON MAP50-95 AND RECALL

Device	Framework	mAP50-95	mAP50	Recall	F1
Cloud	PyTorch	60.2 %	76.3%	73%	65.9%
Jetson	**PyTorch**	**60.2%**	**76.3%**	**73%**	**65.9%**
Jetson	TensorRT FP32	59.8%	76%	72.5%	66.93%
Jetson	TensorRT FP16	58.2%	75.2%	71.7%	65.16%

In Table 2, the secondary metrics are centralized, corresponding to each tested hardware configuration and used software framework.

TABLE II
PERFORMANCE OF FINE-TUNED MODELS IN TERMS OF MEMORY LOAD AND FRAMES PER SECOND.

Device	Framework	Memory Load	FPS
A100	PyTorch	0.3GB (0.75%)	38
RTX 4090	PyTorch	0.3GB (1.25%)	48
T4	PyTorch	0.3GB (1.87%)	13
Jetson	**PyTorch**	**0.3GB (3.75%)**	**2**
Jetson	TensorRT FP32	0.28GB (3.5%)	15
Jetson	TensorRT FP16	0.14GB (1.75%)	28

V. CONCLUSIONS

In this paper, we examined and demonstrated how a model can be selected and fine-tuned to have unexploded ordnance localisation and classification capabilities and how it can be refined to achieve robust results on edge devices. By employing a well-defined methodology, a contextual dataset and modern frameworks such as PyTorch and TensorRT, we achieved real-time performance on Nvidia Jetson devices, with a recall of 71.7% for UXO detection and up to 28 frames per second for the TensorRT FP16 model. The results are very promising and can be easily scaled by integrating them into existing portable hardware systems, such as unmanned aerial vehicles (UAVs) and Explosive Ordnance Disposal (EOD) robots, thereby reducing to zero the risks associated with the localisation and classification of munitions. As future steps, we aim to increase the number of classes in the dataset, studying ensemble learning on models with reduced parameters, while implementation capability on edge devices remains viable.

REFERENCES

[1] L. Maresca, "A new protocol on explosive remnants of war: The history and negotiation of protocol v to the 1980 convention on certain conventional weapons," *International Review of the Red Cross*, vol. 86, no. 856, pp. 815–835, 2004.

[2] M. G. Craioveanu and G. Stamatescu, "Detection and identification of unexploded ordnance using a two-step deep learning methodology," in *2024 32nd Mediterranean Conference on Control and Automation (MED)*. IEEE, 2024, pp. 257–262.

[3] G. M. Craioveanu and G. Stamatescu, "Ctx-uxo: A comprehensive dataset for detection and identification of unexploded ordnances," 2024. [Online]. Available: https://dx.doi.org/10.21227/cwnm-de53

[4] H. Xuan, A. Stylianou, X. Liu, and R. Pless, "Hard negative examples are hard, but useful," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK*. Springer, 2020, pp. 126–142.

[5] B. S. Reddy, A. M. Reddy, M. Sradda, T. Mounika, S. Mounika, and K. Meghana, "A comparative study on object detection using retinanet," in *2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon)*. IEEE, 2022, pp. 1–6.

[6] Z. Ren, X. Liu, J. Xu, Y. Zhang, and M. Fang, "Littlefacenet: A small-sized face recognition method based on retinaface and adaface," *Journal of Imaging*, vol. 11, no. 1, p. 24, 2025.

[7] X. Zhang, Y. Wang, and W. Shi, "pcamp: Performance comparison of machine learning packages on the edges," in *USENIX workshop on hot topics in edge computing (HotEdge 18)*, 2018.

[8] E. Jeong, J. Kim, and S. Ha, "Tensorrt-based framework and optimization methodology for deep learning inference on jetson boards," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 21, no. 5, pp. 1–26, 2022.

[9] Y. Zhou and K. Yang, "Improved real-time deep learning inference by exploiting tensorrt," *Available at SSRN 4529548*, 2023.

[10] P. Gopiktrishna, A. Rishekeeshan *et al.*, "Accelerating native inference model performance in edge devices using tensorrt," in *2024 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, 2024, pp. 1–7.

[11] A. Harvey and E. LeBrun, "Computer vision detection of explosive ordnance: A high-performance 9n235/9n210 cluster submunition detector," *The Journal of Conventional Weapons Destruction*, vol. 27, no. 2, 2023.

[12] J. Baur, G. Steinberg, A. Nikulin, K. Chiu, and T. S. de Smet, "Applying deep learning to automate uav-based detection of scatterable landmines," *Remote Sensing*, vol. 12, no. 5, p. 859, 2020.

[13] Z. Qi, X. Li, H. Li, and W. Liu, "First results from drone-based transient electromagnetic survey to map and detect unexploded ordnance," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, pp. 2055–2059, 2020.

[14] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," 2018. [Online]. Available: https://arxiv.org/abs/1708.02002

[15] J. Sadaiyandi, P. Arumugam, A. K. Sangaiah, and C. Zhang, "Stratified sampling-based deep learning approach to increase prediction accuracy of unbalanced dataset," *Electronics*, vol. 12, no. 21, p. 4423, 2023.