Evaluating the Energy Efficiency of Few-Shot Learning for Object Detection in Industrial Settings

Georgios Tsoumplekas*, Vladislav Li†, Ilias Siniosoglou*‡, Vasileios Argyriou†, Sotirios K. Goudos§, Ioannis D. Moscholios¶, Panagiotis Radoglou-Grammatikis‡‖ and Panagiotis Sarigiannidis*‡

***Abstract*—In the ever-evolving era of Artificial Intelligence (AI), model performance has constituted a key metric driving innovation, leading to an exponential growth in model size and complexity. However, sustainability and energy efficiency have been critical requirements during deployment in contemporary industrial settings, necessitating the use of data-efficient approaches such as few-shot learning. In this paper, to alleviate the burden of lengthy model training and minimize energy consumption, a finetuning approach to adapt standard object detection models to downstream tasks is examined. Subsequently, a thorough case study and evaluation of the energy demands of the developed models, applied in object detection benchmark datasets from volatile industrial environments, is presented. Specifically, different finetuning strategies, as well as utilization of ancillary evaluation data during training, are examined, and the trade-off between performance and efficiency is highlighted in this low-data regime. Finally, this paper introduces a novel way to quantify this trade-off through a customized Efficiency Factor metric.**

***Index Terms*—Few-Shot Learning, Green AI, Deep Learning, Model Optimization, Object Detection, Industrial Image Data**

I. Introduction

Over the last few years, the global industrial ecosystem has increasingly incorporated AI capabilities for operational efficiency and innovation, particularly in areas like predictive maintenance, quality control, logistics, and supply chain optimization. Deep Learning (DL) models have proven to be essential for identifying outliers, anomalies, and irregularities in industrial environments, enabling decisive decision-making and proactive safety operations [1]. However, the innate nature of AI presents challenges, especially regarding energy consumption. As complex models process larger datasets, the energy required to train and apply them increases, raising environmental, sustainability, and scalability concerns. Therefore, the energy efficiency of AI systems is a critical area of focus, necessitating research to create more energy-efficient and cost-effective AI models without compromising performance and accuracy, which could lead to broader adoption of them in industrial applications.

Few-shot learning [2] (FSL) has recently emerged as a learning paradigm that enables models to learn from a limited amount of data, tackling the extensive resource demands of standard AI models. For typical learning tasks within the industrial ecosystem, such as object detection [3], data can be scarce or expensive to acquire due to privacy regulations. At the same time, hardware and bandwidth limitations inherent to edge devices used in these settings render model training from scratch prohibitively inefficient.

A simple yet effective FSL approach to handle these restrictions would be to follow a finetuning-based approach, where a pretrained model is adapted to the new limited data by retraining the model partially or in total with them. This approach is straightforward and can be effectively applied in a data-scarce regime. It also requires only a fraction of the resources needed compared to model training from scratch. As a result, it can potentially serve as a practical and efficient solution for tasks such as few-shot object detection in settings where energy efficiency is critical. However, quantifying and enhancing the energy efficiency of finetuning-based FSL models has been scarcely explored in the literature.

This paper explores the balance between the training performance and energy efficiency of finetuning-based methods for few-shot object detection in industrial settings. For this purpose, various YOLOv8 models are trained using different finetuning strategies in three benchmark image datasets from volatile industrial environments, and both detection performance and energy efficiency are systematically evaluated, providing insights into the trade-off between these two objectives. Finally, a novel metric is proposed to better capture the aforementioned trade-off. The overall contributions of this paper can be summarized as follows:

- Introduces a novel metric, Efficiency Factor, to quantify and correlate the energy consumption vs performance trade-off of FSL models.
- Presents a thorough evaluation of various finetuned models' performance against their energy efficiency.
- Creates an in-depth comparative study of the effect of different benchmark datasets on a widely used object detection model.
- Evaluates the efficacy of finetuning-based FSL as a prominent method to minimize training time and energy

[*] G. Tsoumplekas, I. Siniosoglou and P. Sarigiannidis are with the R&D Department, MetaMind Innovations P.C., Kozani, Greece - E-Mail: {gtsoumplekas, isiniosoglou, psarigiannidis}@metamind.gr

[†] V. Li and V. Argyriou are with the Department of Networks and Digital Media, Kingston University, Kingston upon Thames, United Kingdom - E-Mail: {v.li, vasileios.argyriou}@kingston.ac.uk

[‡] I. Siniosoglou, P. R. Grammatikis and P. Sarigiannidis are with the Department of Electrical and Computer Engineering, University of Western Macedonia, Kozani, Greece - E-Mail: {isiniosoglou, pradoglou, psarigiannidis}@uowm.gr

[§] S. K. Goudos is with the Physics Department, Aristotle University of Thessaloniki, Thessaloniki, Greece - E-Mail: sgoudo@physics.auth.gr

[¶] I. D. Moscholios is with the Department of Informatics and Telecommunications, University of Peloponnese, Tripoli, Greece - E-Mail: idm@uop.gr

[‖] P. R. Grammatikis is with the Department of Research and Development, K3Y Ltd., Sofia, 1000, Bulgaria - E-Mail: pradoglou@k3y.bg

979-8-3503-6334-0/24/$31.00 ©2024 IEEE
DOI 10.1109/RAGE62451.2024.00016

consumption in AI models.

The rest of this paper is organized as follows: the related work is discussed in Section II, followed by an overview of the methodology in Section III. Section IV provides a comprehensive analysis of the available data while measuring the energy efficiency and performance of the models. Section V offers concluding remarks.

II. Related Work

A. Object Detection

Object Detection has made great progress through the development of models such as YOLOv8, Mask RCNN, and Fast RCNN, each making distinct contributions to the area. YOLOv8, being a successor to YOLOv5, provides significant enhancements in terms of both accuracy and speed [4] However, the energy efficiency of YOLOv8 still remains a challenge, especially in edge computing situations. Mask RCNN is an extension of Faster RCNN [5] that allows for pixel-level segmentation [6], which means it can accurately locate objects and perform instance segmentation. It performs exceptionally well in situations that demand precise detection, such as medical picture analysis. Nevertheless, the intricate structure of the system results in increased computational expenses, hence affecting its energy efficiency in settings with limited resources. Fast RCNN, an antecedent to Faster RCNN, integrates selective search with a deep CNN, thereby diminishing the duplication in feature computing [7]. Although it represented a notable advancement in the efficiency of object identification, its dependence on external region proposal methods hinders its speed and energy efficiency in comparison to more integrated models such as Faster RCNN.

B. Few-Shot Object Detection

While most FSL research has traditionally focused on image classification, in recent years there is an increasing interest in the development of novel few-shot object detection (FSOD) algorithms. One of the first approaches towards that direction has proposed a reweighting module that transforms the extracted feature representation and is jointly trained with a YOLO detection model in a two-step training procedure [8]. A two-step training approach has also been explored in [9] which demonstrates the effectiveness of finetuning in FSOD without the need for external modules. More recently, recasting the object detection problem as an image classification problem and learning new object classes in an adversarial manner has also been proposed [10]. Additionally, meta-learning has been leveraged to enable learning task-specific and task-agnostic model parameters in the context of FSOD [11]. Finally, FSOD has also been extended to novel settings via its combination with incremental learning [12] and domain adaptation.

C. AI Model Energy Efficiency

The field of Green and Energy efficient AI is steadily evolving, with research currently focusing on mitigating the ecological consequences associated with training extensive ML models and aiming to quantize and subsequently tackle the energy needs of modern AI. In particular, the rise in model size and complexity has led to a large increase in energy usage and carbon emissions [13] and consequently, several methods have been suggested to reduce these impacts, such as calculating the carbon emissions caused by AI models and creating tools to assess the environmental impact of model training. Federated Learning (FL) has also been investigated as a means to mitigate energy usage [14]. However, constraints in computational capacity and the requirement for inter-device communication present obstacles. Although Green AI is crucial, there remains a dearth of research on employing FSL as a viable alternative. While meta-learning has achieved excellent results in terms of performance in FSL, its large computational complexity renders it inhibiting for energy efficient applications. However, recently, transfer learning methods, such as fine-tuning, have emerged as viable alternatives in this context, due to their high performance and low computational cost.

III. Methodology

A. Model Architecture

Given that our focus is towards models that adhere to the principles of Green AI[15] and at the same time demonstrate strong generalization performance, YOLOv8, the latest version of the You-Only-Look-Once (YOLO) object detection models family, is used. Compared to its predecessor, YOLOv5, YOLOv8 introduces a highly efficient anchor-free object detection approach that leads to increased performance.

Regarding its architecture, YOLOv8 consists of a backbone feature extractor, used to extract meaningful feature representations from the images, followed by the model's head network that produces the final predictions. As for the feature extractor, it is based on a modified version of CSPDarknet53[16]. Its structure follows that of feature pyramid networks (FPNs) [17], which enables the identification of objects of varying sizes and scales within an image by extracting features at multiple scales. On the other hand, the head network consists of a series of convolutional layers followed by three different detection modules whose inputs are features extracted from different levels of the FPN, allowing for multi-scaled object detection.

B. Few-shot Object Detection

The main objective of FSOD is training of models that are capable of quickly adapting to novel tasks given only a minimal number of training samples within each new task. In the context of FSOD, a standard approach is to consider two different sets of data with different object classes, the base classes and the novel classes. In this case, it is realistic to assume that the examined object detection model has been trained on a training set that consists of abundant data belonging to these base classes. More specifically, the training set can be denoted as:

$$D^{train} = \{(x_i, y_i)\}_{i=1}^{|D^{train}|} \tag{1}$$

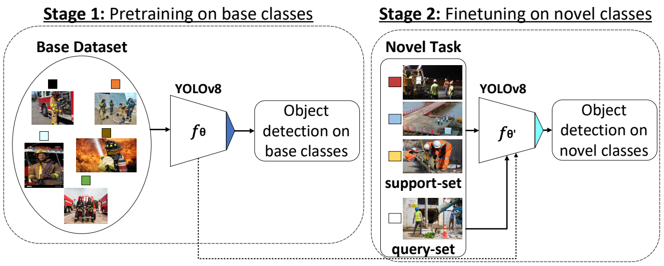

Stage 1: Pretraining on base classes
Stage 2: Finetuning on novel classes
Base Dataset
YOLOv8
f_θ
Object detection on base classes
Novel Task
YOLOv8
$f_{\theta'}$
Object detection on novel classes
support-set
query-set

Fig. 1: Illustration of the two-step training procedure based on base class pretraining and novel class finetuning

where $x_i \in \mathbb{R}^M$ is the i-th training image, $y_i \in \{0, 1, ..., N_B - 1\}$ is its label, and N_B is the total number of base classes. Consequently, a trained model f_θ is produced.

Following training of the model on D^{train}, the next step is its adaptation to the novel classes. In general, these novel classes can be formulated as part of a few-shot task that consists of only a small of number of novel class images available during adaptation and an arbitrary number of novel class images used for evaluation of the model. Specifically, for a given task τ with N_N novel classes, it can be split into a support set $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{|\mathcal{S}|}$ used for adaptation, and a query set $\mathcal{Q} = \{(x_j, y_j)\}_{j=1}^{|\mathcal{Q}|}$ used for model evaluation in this task. Adhering to the standard methodology of FSL, N-way K-shot tasks are examined, which consist of N novel classes and there are K training samples for each novel class in the task's support set (as a result $|\mathcal{S}| = NK$).

To adapt to task τ that contains the novel class samples, the trained model f_θ is further trained on τ's support set $\mathcal{S}$, producing an adapted model $f_{\theta'}$ which is then evaluated on τ's query set $\mathcal{Q}$. Finally, the model's overall performance is its mean performance across all tasks $\{\tau_i\}_{i=1}^{T}$, where T is the total number of tasks, reported along with the corresponding standard deviation.

C. Few-Shot Learning via Model Finetuning

Based on the aforementioned standard methodology of FSOD, one question that typically arises in these settings is how to effectively and efficiently obtain $f_{\theta'}$ from f_θ using $\mathcal{S}$. One of the simplest approaches in this case, which has recently achieved competitive results compared to more complex solutions in various settings, is to simply finetune the trained model f_θ in $\mathcal{S}$. Overall, this finetuning approach can be seen as a two-step procedure. In the first step, the model is trained on D^{train}, emulating a form of model pretraining and producing f_θ. In the second step, the pretrained model is then finetuned on each task's support set $\mathcal{S}$, producing the final finetuned model $f_{\theta'}$. It is also worth mentioning that a new output layer is introduced for each task due to the different classes included in each of them. Figure 1 also illustrates this two-step approach.

A common consideration when finetuning is used to adapt a model in the context of FSL is deciding on a full vs partial finetuning approach. To further examine these two different approaches both in terms of downstream task performance as well as computational efficiency during training, we employ three different model finetuning variations, leveraging YOLOv8's internal structure: (a) full finetuning of the whole model, including both backbone and head, (b) partial finetuning including the model's head only, and (c) partial finetuning including the model's detection modules only.

IV. Experimental Results

A. Experimental Configuration

To allow for a fair comparison across the three aforementioned finetuning approaches, model pretraining is fixed in all cases and a pretrained version of YOLOv8 trained on the MSCOCO dataset is used, specifically YOLOv8n which consists of 3.2M parameters. In the case of full model finetuning all 3.2M parameters are adapted, while detection modules finetuning involves the adaptation of ≈750K parameters, and head finetuning involves the adaptation of ≈1.7M parameters. In our experiments, full finetuned models are denoted as $full$, models with finetuned heads are denoted with $head$, and models with finetuned detection modules are denoted with det.

Additionally, during model finetuning, object classes have only K support set samples, with $K \in \{1, 2, 3, 5, 10, 30\}$. The number of finetuning epochs is also adapted based on K. For $K = 1$, the number of epochs is 10, for $K \in \{2, 3, 5\}$ it is 30, and finally for $K \in \{10, 30\}$ it is 200. The use of a validation set to measure model performance after each finetuning epoch is also examined (despite potential computational overhead) to facilitate selecting the best-performing model to be used during model testing. In our experiments, models using a validation set are denoted as $best$, while the rest are denoted as $last$. To ensure robust comparisons, each model is evaluated in three different downstream tasks, with the mean value and standard deviation reported for each metric across these tasks.

Finally, the optimized AI models are tested on an edge ecosystem, considering resource limitations. A mid-range resource-constrained edge device with a 12th Gen Intel i7 CPU, 16GB memory, integrated graphics, and Ubuntu 22.04 was used.

B. Datasets

For the following experiments, three different datasets related to object detection of objects commonly found in industrial settings were utilized to finetune the object detection models, in an effort to establish recognition and proactiveness in safety upkeeping and robust decision-making:

- **Personal Protective Equipment Detection [1]:** Aims to help AI detectors locate and identify various PPE used by first responders for enhanced safety.
- **Construction Safety Detection [18]:** Trains AI models to identify PPE presence or absence in industrial/construction settings.
- **Fire Detection:** Focuses on training AI to locate, recognize, and classify fires for proactive safety measures

The abovementioned datasets were utilized to finetune the leveraged AI models of this work in the premise of evaluating

Fig. 2: Dataset Preview. (a) PPE, (b) Construction Safety, (c) Fire Detection

the final models' performance vs energy efficiency for object detection. The attributes of these datasets are summarized in Table I, while Figure 2 shows some examples of these datasets' images.

C. Evaluation Metrics

For the evaluation of the examined models, both performance and efficiency should be taken into consideration. In this setting, model performance refers to the typical evaluation metrics used to measure the generalization capabilities of a model on a given dataset and, consequently, can be quantified using Mean Average Precision (mAP):

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{2}$$

where AP_i is the average precision for class i, and N is the total number of classes in the dataset.

On the other hand, model efficiency refers to a measure of how energy-efficient the examined model is. Consequently, energy consumption, measured in watt-hours (Wh), is calculated during the model's finetuning step, using Python's CodeCarbon library.

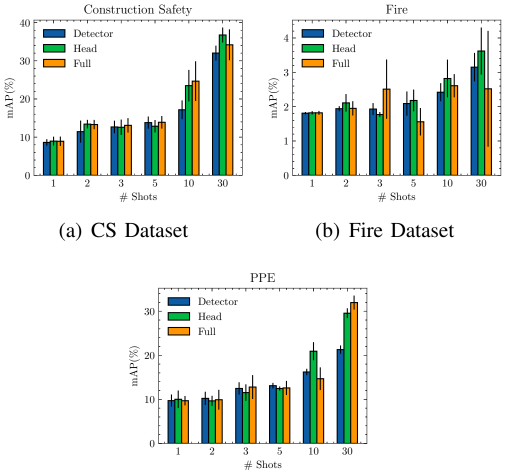
Finally, to account for both model performance and efficiency, a novel metric, Efficiency Factor (EF), is introduced that takes into consideration both mAP and energy consumption. Specifically, optimal models should demonstrate high performance (high mAP) as well as high efficiency (low energy consumption). As a result, EF can be formulated as:

$$EF = \frac{mAP}{1 + EC} \tag{3}$$

where $mAP \in [0, 100]$ is in its percentage form, and $EC \in (0, +\infty)$ is the model's energy consumption measured in Wh. A constant value is also included in the denominator so that EF is bounded in $[0, 100)$, with higher EF values given to models that achieve both high mAP values during testing and low energy consumption during training.

D. Experimental Results

Model Performance. Table II shows the performance of all six different finetuning combinations in terms of mAP for the three examined datasets. It is evident that in almost all cases, using a validation set during finetuning leads to consistently better-performing models, outlining the importance of carefully selecting the number of finetuning epochs. At the same time, increasing the number of shots also leads to increased model performance, which is reasonable considering that there are more available training data in the task's support set. Finally, model performance seems to be less sensitive in the selection of the finetuning strategy with all three strategies leading to comparable results for the same number of shots. This is also illustrated in Figure 3, where the performance of models using evaluation sets during finetuning can be seen for different finetuning strategies and different numbers of shots.



(a) CS Dataset

(b) Fire Dataset

(c) PPE Dataset

Fig. 3: Mean mAP and standard deviation of *best* models for varying finetuning strategies and numbers of shots.

Model Efficiency. Table III shows the energy consumption of the evaluated models during finetuning, clearly demonstrating that finetuning only the detection modules without using a validation set consistently outperforms the rest of the models in all three datasets and for each number of shots. Overall, models that do not use a validation set demonstrate better energy efficiency compared to the corresponding ones that use, which is logical given that finetuning becomes less computationally expensive without the use of the validation set in each finetuning epoch. Additionally, reducing the number of model parameters that are finetuned also leads to reduced energy consumption since the finetuning procedure becomes less computationally heavy.

Effect of finetuning. Figure 4 illustrates model performance with respect to the energy consumed during training for each of the examined datasets. Overall, optimal models in terms of performance and energy efficiency should achieve high mAP combined with low energy consumption. As a result, models that lie on the upper left area of each scatter plot are considered best. However, there is a clear trade-off between performance and efficiency since better-performing models consume more energy during finetuning. Finally, it is

TABLE I: Main characteristics of the examined industrial object detection datasets.

Dataset Name	Total Samples	Classes		Train	Test	Validation
PPE Detection Dataset	342	Helmet, Gloves, Mask, Cloth	Localize and identify PPE for first responders	280	31	31
Worker-Safety Computer Vision Project	3200	Helmets, Vests, other PPE	Identify presence/absence of PPE in industrial settings	2991	90	119
Fire Detection Dataset	3677	Fire scenes	Locate, recognize, and classify fires	3527	-	150

TABLE II: Few-shot detection performance (mean mAP and standard deviation) on each of the three examined datasets.

Dataset	Model	Shots					
		1	2	3	5	10	30
PPE	y8-det-last	9.68±1.43	6.79±2.35	10.49±1.94	10.93±1.10	16.50±1.89	19.11±1.12
	y8-det-best	9.71±1.39	**10.22±1.51**	12.47±1.41	**13.10±0.64**	16.21±0.72	21.29±0.94
	y8-head-last	9.97±2.04	6.84±2.32	10.95±1.66	10.91±1.88	19.63±4.09	29.88±1.49
	y8-head-best	**10.01±1.99**	9.66±1.14	11.53±1.88	12.46±0.47	**20.92±2.05**	29.55±1.12
	y8-full-last	9.69±1.07	8.84±2.20	12.76±3.06	10.07±3.78	13.09±2.07	29.87±1.52
	y8-full-best	9.69±1.07	9.91±2.26	**12.79±2.71**	12.59±1.60	14.68±2.58	**31.96±1.59**
Fire	y8-det-last	1.63±0.24	1.99±0.49	1.75±0.15	1.74±0.42	1.54±0.49	1.63±0.53
	y8-det-best	1.81±0.03	1.94±0.07	1.93±0.17	2.09±0.35	2.42±0.26	3.15±0.42
	y8-head-last	1.69±0.20	2.04±0.27	1.95±0.27	1.73±0.36	0.79±0.34	1.12±0.76
	y8-head-best	**1.82±0.05**	**2.11±0.26**	1.77±0.07	**2.18±0.32**	**2.82±0.55**	**3.62±0.69**
	y8-full-last	1.61±0.15	1.39±0.27	1.75±0.98	0.86±0.38	0.73±0.40	0.84±0.69
	y8-full-best	**1.82±0.06**	1.95±0.21	**2.51±0.86**	1.56±0.40	2.61±0.34	2.52±1.69
CS	y8-det-last	8.54±0.98	10.30±3.41	10.88±2.64	12.79±2.70	19.48±3.03	31.64±0.74
	y8-det-best	8.62±0.85	11.44±2.91	12.67±1.65	13.81±1.60	17.17±2.46	32.02±1.95
	y8-head-last	8.77±1.43	11.92±0.76	9.57±3.16	11.31±1.94	**26.16±0.42**	36.28±1.74
	y8-head-best	**8.96±1.17**	**13.41±1.13**	12.59±2.02	12.86±1.63	23.49±4.12	**36.76±1.97**
	y8-full-last	8.81±1.37	11.70±0.80	10.15±2.77	13.00±2.57	24.76±5.39	34.54±3.91
	y8-full-best	8.92±1.23	13.30±1.22	**13.09±1.89**	**13.89±1.66**	24.67±5.21	34.19±4.06

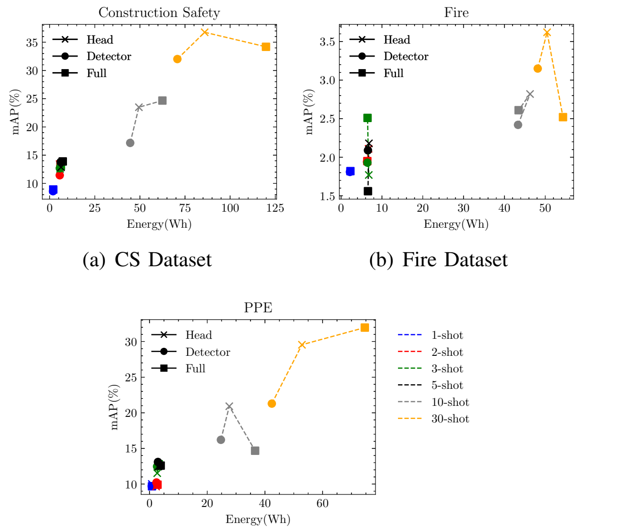
TABLE III: Energy consumption (mean Wh and standard deviation) on each of the three examined datasets.

Dataset	Model	Shots					
		1	2	3	5	10	30
PPE	y8-det-last	**0.271±0.003**	**0.639±0.036**	**0.843±0.007**	**1.194±0.060**	**12.435±0.268**	**30.010±0.658**
	y8-det-best	0.774±0.020	2.400±0.016	2.607±0.007	2.916±0.053	24.738±0.442	42.395±1.153
	y8-head-last	0.282±0.002	0.720±0.032	0.994±0.001	1.464±0.066	16.821±0.359	41.395±0.715
	y8-head-best	0.779±0.015	2.280±0.045	2.668±0.017	3.070±0.047	27.647±0.185	52.831±0.691
	y8-full-last	0.360±0.004	1.095±0.068	1.517±0.014	2.196±0.096	25.169±0.290	63.614±0.919
	y8-full-best	0.852±0.003	2.762±0.020	3.190±0.017	3.866±0.077	36.599±0.224	74.630±1.195
Fire	y8-det-last	**0.487±0.010**	**0.601±0.005**	**0.660±0.007**	**0.734±0.004**	**3.772±0.040**	**10.294±0.062**
	y8-det-best	2.161±0.041	6.341±0.015	6.448±0.036	6.573±0.011	43.335±0.470	48.161±0.434
	y8-head-last	0.545±0.009	0.741±0.012	0.783±0.009	0.897±0.013	5.709±0.021	13.889±0.110
	y8-head-best	2.316±0.069	6.734±0.020	6.775±0.032	6.809±0.013	46.277±0.760	50.452±1.361
	y8-full-last	0.532±0.012	0.782±0.014	0.852±0.009	1.018±0.013	7.235±0.018	20.388±0.504
	y8-full-best	2.273±0.014	6.421±0.044	6.472±0.039	6.576±0.113	43.477±0.259	54.348±1.413
CS	y8-det-last	**0.443±0.007**	**0.857±0.011**	**1.095±0.025**	**1.677±0.028**	**15.681±0.088**	**43.325±0.673**
	y8-det-best	1.855±0.077	5.588±0.031	5.620±0.053	6.111±0.160	44.621±0.703	70.761±1.686
	y8-head-last	0.495±0.006	1.032±0.008	1.365±0.041	2.033±0.013	20.858±0.384	57.974±0.673
	y8-head-best	1.801±0.045	5.506±0.025	5.778±0.037	6.300±0.102	49.440±0.402	85.889±0.524
	y8-full-last	0.616±0.009	1.468±0.018	1.913±0.067	3.093±0.031	33.436±0.776	88.492±0.635
	y8-full-best	1.986±0.097	5.977±0.045	6.222±0.040	7.263±0.079	62.464±1.427	119.86±2.998

also worth noticing that finetuning the model's head seems to constitute a good compromise between these two conflicting objectives, especially for an increased number of shots.

Effect of validation set. To illustrate how model performance with respect to energy consumption is affected by the use of a validation set during finetuning the related results from the PPE dataset are displayed in Figure 5. Similar to Figure 4, there seems to be a clear trade-off between model performance and energy consumption, underlining the challenges in developing FSOD models that are both sustainable and effective. Additionally, in terms of mAP, the use of a validation set becomes less important as the number of shots increases. However, its use leads to a significant increase in energy consumption, rendering it less useful in these scenarios.

EF as a consolidated performance-efficiency measure. While performance and training efficiency seem to be two conflicting objectives in the context of FSOD, it is important to be able to compare models in a unified way, taking into consideration both of these desired properties. Table IV shows the EF values of the examined models in all three datasets for a varying number of shots. It is manifest that not using a validation set during finetuning leads to increased EF values due to the reduced energy consumption of these models compared to the corresponding ones that use validation sets. Additionally, while adapting less parameters during finetuning shrinks energy consumption, it does not necessarily lead to increased EF values, since it might also be accompanied by a loss in performance. However, the number of shots directly affects EF, with a smaller number of shots leading to increased EF values. This can be attributed to the disproportionate increase in energy consumption as the number of shots increases because of the corresponding escalation of

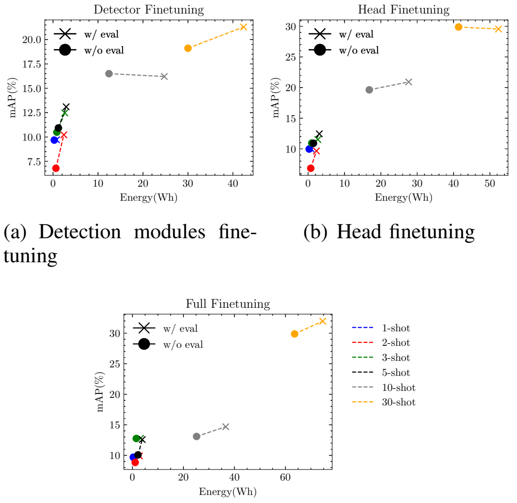


(a) CS Dataset

(b) Fire Dataset

(c) PPE Dataset

Fig. 4: mAP with respect to energy consumption for different finetuning strategies and numbers of shots.



(a) Detection modules finetuning

(b) Head finetuning

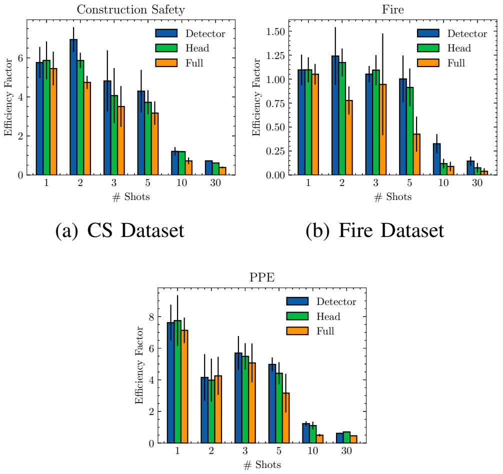
(c) Full finetuning

Fig. 5: Model performance with respect to energy consumption in the PPE dataset.

TABLE IV: Model performance per units of energy efficiency (mean EF and standard deviation) on each of the three examined datasets.

Dataset	**Model**	**Shots**					
		1	**2**	**3**	**5**	**10**	**30**
PPE	y8-det-last	7.618±1.141	4.151±1.475	**5.696±1.077**	**4.977±0.439**	**1.230±0.150**	0.617±0.044
	y8-det-best	5.484±0.828	3.008±0.453	3.456±0.385	3.347±0.208	0.630±0.026	0.491±0.032
	y8-head-last	**7.784±1.605**	3.982±1.363	5.488±0.832	4.417±0.698	1.106±0.248	**0.705±0.033**
	y8-head-best	5.635±1.154	2.948±0.374	3.146±0.519	3.060±0.079	0.730±0.071	0.549±0.027
	y8-full-last	7.132±0.808	**4.255±1.204**	5.071±1.226	3.160±1.228	0.500±0.079	0.462±0.024
	y8-full-best	5.235±0.575	2.637±0.605	3.052±0.647	2.590±0.348	0.390±0.067	0.423±0.027
Fire	y8-det-last	1.096±0.159	**1.241±0.301**	1.052±0.085	**1.002±0.244**	**0.326±0.101**	**0.145±0.047**
	y8-det-best	0.573±0.003	0.265±0.010	0.260±0.023	0.276±0.047	0.055±0.006	0.064±0.008
	y8-head-last	**1.097±0.131**	1.173±0.146	**1.094±0.157**	0.913±0.197	0.119±0.050	0.075±0.050
	y8-head-best	0.549±0.007	0.272±0.033	0.227±0.010	0.280±0.041	0.060±0.013	0.070±0.013
	y8-full-last	1.051±0.108	0.777±0.147	0.945±0.531	0.426±0.183	0.089±0.049	0.039±0.032
	y8-full-best	0.556±0.019	0.263±0.028	0.336±0.114	0.206±0.054	0.059±0.008	0.046±0.030
CS	y8-det-last	5.762±0.801	**6.936±0.637**	**4.819±1.566**	**4.295±1.090**	**1.211±0.215**	**0.726±0.015**
	y8-det-best	3.009±0.272	2.065±0.234	1.866±0.287	1.836±0.289	0.397±0.063	0.452±0.026
	y8-head-last	**5.868±0.975**	5.866±0.398	4.067±1.415	3.724±0.620	1.197±0.034	0.615±0.023
	y8-head-best	3.195±0.367	2.061±0.176	1.855±0.287	1.759±0.205	0.466±0.085	0.423±0.023
	y8-full-last	5.452±0.864	4.742±0.338	3.508±1.046	3.171±0.602	0.723±0.171	0.386±0.046
	y8-full-best	2.978±0.322	1.907±0.185	1.813±0.273	1.679±0.184	0.391±0.089	0.282±0.026

the finetuning epochs. The aforementioned observations are also illustrated in Figure 6, where the EF values of models finetuned without using a validation set are displayed.



(a) CS Dataset

(b) Fire Dataset

(c) PPE Fataset

Fig. 6: Mean EF and standard deviation of $last$ models for varying finetuning strategies and numbers of shots.

V. Conclusions

While modern machine and deep learning approaches have led to tremendous strides in various industrial settings and applications, their performance and energy efficiency evaluation in realistic volatile and data-scarce industrial environments has been underexplored. In this paper, we examine how finetuning-based FSL methods can be leveraged to produce models that demonstrate both high performance and energy efficiency in downstream detection tasks with limited samples in industrial settings. An empirical study based on three different industrial datasets is conducted, demonstrating the balance between energy efficiency and the training performance involved in finetuning-based FSL approaches while also examining how these variables are affected by different finetuning strategies. Finally, a novel metric, Efficiency Factor, is introduced to help quantify the interaction of model performance and efficiency in a consolidated way. Overall, this work could serve as an initial step towards a broader investigation of FSL's energy efficiency, which could be expanded to include standard techniques such as meta-learning and data augmentation.

Acknowledgement

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101070181 (TALON).

References

[1] A. Sesis, I. Siniosoglou, Y. Spyridis, G. Efstathopoulos, T. Lagkas, V. Argyriou, and P. Sarigiannidis, "A robust deep learning architecture for firefighter ppes detection," in *2022 IEEE 8th World Forum on Internet of Things (WF-IoT)*, 2022, pp. 1–6.

[2] G. Tsoumplekas, V. Li, V. Argyriou, A. Lytos, E. Fountoukidis, S. K. Goudos, I. D. Moscholios, and P. Sarigiannidis, "Toward green and human-like artificial intelligence: A complete survey on contemporary few-shot learning approaches," *arXiv preprint arXiv:2402.03017*, 2024.

[3] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A survey on performance metrics for object-detection algorithms," in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2020, pp. 237–242.

[4] F. M. Talaat and H. ZainEldin, "An improved fire detection approach based on YOLO-v8 for smart cities," *Neural Computing and Applications*, vol. 35, no. 28, pp. 20 939–20 954, Oct. 2023.

[5] N. Rane, "Yolo and faster r-cnn object detection for smart industry 4.0 and industry 5.0: applications, challenges, and opportunities," *SSRN Electronic Journal*, 01 2023.

[6] K. Sujatha, K. Amrutha, and N. Veeranjaneyulu, "Enhancing object detection with mask r-cnn: A deep learning perspective," in *2023 International Conference on Network, Multimedia and Information Technology (NMITCON)*, 2023, pp. 1–6.

[7] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimed. Tools Appl.*, vol. 82, no. 6, pp. 9243–9275, 2023.

[8] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell, "Few-shot object detection via feature reweighting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8420–8429.

[9] X. Wang, T. Huang, T. Darrell, J. Gonzalez, and F. Yu, "Frustratingly simple few-shot object detection. arxiv 2020," *arXiv preprint arXiv:2003.06957*.

[10] B. Li, B. Yang, C. Liu, F. Liu, R. Ji, and Q. Ye, "Beyond max-margin: Class margin equilibrium for few-shot object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 7363–7372.

[11] Y.-X. Wang, D. Ramanan, and M. Hebert, "Meta-learning to detect rare objects," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9925–9934.

[12] L. Yin, J. M. Perez-Rua, and K. J. Liang, "Sylph: A hypernetwork framework for incremental few-shot object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9035–9045.

[13] P. Henderson, J. Hu, J. Romoff, E. Brunskill, D. Jurafsky, and J. Pineau, "Towards the systematic reporting of the energy and carbon footprints of machine learning," *J. Mach. Learn. Res.*, vol. 21, no. 1, jan 2020.

[14] X. Qiu, T. Parcollet, J. Fernandez-Marques, P. P. de Gusmao, Y. Gao, D. J. Beutel, T. Topal, A. Mathur, and N. D. Lane, "A first look into the carbon footprint of federated learning." *J. Mach. Learn. Res.*, vol. 24, pp. 129–1, 2023.

[15] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green ai," *Communications of the ACM*, vol. 63, no. 12, pp. 54–63, 2020.

[16] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[17] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[18] computer vision, "Worker-safety dataset," https://universe.roboflow.com/computer-vision/worker-safety, jul 2022, visited on 2024-01-23. [Online]. Available: https://universe.roboflow.com/computer-vision/worker-safety