ETA-DyNN: An Energy-Efficient Task Allocation Scheme for Varroa Detection on Edge AI Devices via Dynamic Neural Networks

Davide Bazzana
Sapienza University of Rome
Rome, Italy
d.bazzana@diag.uniroma1.it

Roberto Beraldi
Sapienza University of Rome
Rome, Italy
beraldi@diag.uniroma1.it

Marco Angelini
Link University of Rome
Rome, Italy
m.angelini@unilink.it

***Abstract*—Varroa destructor is a parasitic mite that poses a major threat to honeybee colonies by compromising hive health. Although recent computer vision approaches based on Vision Transformers (ViTs) can achieve high recognition accuracy, their computational complexity and energy requirements limit their deployability in off-grid beehive settings. Moreover, a cloud-only solution would require sustained uplink bandwidth, which is often unavailable in rural environments.**

To address these limitations, we propose an environment-aware hybrid edge/cloud approach that dynamically allocates the classification task based on both system conditions and sample difficulty. Given a video sequence of a bee, the method extracts key frames and decides whether to perform inference locally using an early-exit lightweight Convolutional Neural Network (CNN) or selectively offload uncertain cases to a cloud-based ViT service. The decision accounts for current and predicted energy and memory availability, as well as the estimated classification difficulty. Experimental results indicate that the proposed method achieves accuracy comparable to state-of-the-art solutions while significantly improving energy efficiency. To the best of our knowledge, this is the first architecture combining difficulty-aware edge/cloud partitioning with energy-aware decision-making in photovoltaic-powered edge deployments, and it can generalize to other domains characterized by limited connectivity and strict energy constraints.

***Index Terms*—Edge Computing, Artificial Intelligence, Task Allocation.**

I. Introduction

Bees are central contributors to biodiversity and environmental preservation. One major threat to bee health is the Varroa Destructor mite [8]; therefore, it is fundamental to be capable of detecting it promptly in beehives. Current solutions apply computer vision-based methods [7, 1] to this task but incur high computational costs, creating a trade-off between accuracy and sustainability. Beyond negative environmental impacts from using energy-demanding cloud-based models, guaranteeing service continuity in rural areas, the common scenario for beehives, requires local computation on energy-constrained devices. Building on these issues, this work focuses on an edge-oriented solution tailored to on-field conditions that achieves classification performance comparable to much more capable Neural Network (NN) models at a much lower energy cost. To bridge the gap between NN models and their sustainable field applicability, we present a decision-making module that leverages the flexibility of Dynamic Neural Networks (DyNN) to account for system and environmental conditions. Instead of a one-fits-all approach, we propose a two-tier edge/cloud architecture that makes use of locally harvested energy to power the computation of classification tasks where the data is being produced, while using a cloud-hosted classification service running a more capable classification model for all those cases where the smaller, edge-based network is incapable of providing an accurate answer or the edge device has insufficient resources.

This work was conducted in the context of the Italian national research project "Edge Vision against Varroa (EV2): Edge computing in defence of bees", url: https://alcorlab.diag.uniroma1.it/projects/ev2. Grant No. 202277WMAE.

II. Related Work

Beehive health is fundamental to biodiversity, as bees serve as the main pollinators of crops [12]. For detecting parasites, specifically the Varroa destructor mite, several studies propose assessing bee infestation through automated visual inspection. CNNs are well-suited to this problem, given their success in image classification. Most solutions employ CNNs successfully, either by defining custom architectures [2] or by using state-of-the-art architectures, such as ResNet, GoogLeNet, and DenseNet [14, 1]. Finally, other methods propose using features beyond the visual characteristics of the parasite and the bee, or even beyond the visible spectrum. For instance, Giovannesi et al. [7] use Vision Transformers to detect infested bees based on movement, while Duma et al. [4] use clustering algorithms applied to hyperspectral images to distinguish the parasite from the bee. Although effective, some of the aforementioned solutions have practical limitations when deployed in the field, for instance, the direct involvement of beekeepers in the process [2], or the use of costly hardware, such as hyperspectral cameras [4]. Moreover, researchers have assessed such solutions only in limited ways in realistic deployment settings. While Berkaya et al. [1] measure the inference time, none of the others do so, nor does anyone assess the energy effectiveness of their systems. Effective energy use is crucial

to the deployment settings of such activity. Indeed, beehives are commonly found in rural areas where access to energy is scarce or nonexistent (off-grid locations), requiring the system to be self-sustaining by harvesting energy from its surroundings, such as solar irradiance using photovoltaic (PV) modules. We address these limitations by using an Early-Exiting Dynamic Neural Network (DyNN) [9, 13] as the classifier, along with a decision-making module that integrates environmental factors into the early-exiting process to improve sustainability and energy efficiency while maintaining comparable accuracy [3]. Our work contributes to the field of efficient NNs by fragmenting a model through an early-exiting strategy that is driven by, and dynamically adapts to, the system's resources and environmental conditions, as well as the task's difficulty, thereby enabling a more granular and efficient decision-making process.

III. The ETA-DyNN Approach

We consider the problem of classifying honey bees entering and leaving beehives. We define a *task* as the classification of a bee as infested or free of the Varroa Destructor mite. The classification system should maximize (i) the classification accuracy, (ii) the usage of locally harvested energy (at each hive), and (iii) the number of classification tasks completed.

To achieve this, we propose a two-tier edge/cloud approach that involves a set of *Energy-Harvesting (EH) Edge Devices*, devoted to the collection and elaboration of videos of bees entering and leaving a hive and capable of running lightweight classification algorithms, and a *Remote Node*, capable of running expensive classification algorithms on the classification tasks delegated from the edge devices. In the following, we present a model that describes the decision-making process for allocating classification tasks, either locally to the edge device (where the task is generated) or upstream to a remote node. When enough energy is available, the system takes a video from the buffer to preprocess it. The preprocessing tracks the bee and extracts the key frames from the videos using OpenCV tracking methods and Wolf's [16] key frame extraction algorithm. The preprocessing stage provides, for each video, a batch of frames that the NN model uses to classify the bee. Given the energy constraints of the proposed approach, the choice for the NN model architecture is the Early-Exit Dynamic Neural Network with a static early-exit policy [13] based on the confidence of the early-exit branches. In particular, the proposed solution uses MobileNetV3 large [10] as the backbone of the architecture, with a single side branch in the middle, specifically, at the eighth inverted residual block. Each early-exit branch has upper and lower confidence thresholds corresponding to positive and negative answers. For each exit, the network compares the predicted probabilities (one per frame) against the thresholds and returns an answer if at least one prediction is valid. If multiple predictions are valid, the model returns a positive answer if at least one prediction is positive, and a negative answer otherwise. This heuristic favors recall over precision, prioritizing early detection and minimizing the risk of missing true-positive cases, even at the cost of additional false positives, to better prevent the parasite's spread.

A. Minimization problem

We formulate the problem as a minimization problem with discrete decisions and constant-duration time slots indexed by $t \in \mathbb{N}$. The state of the system comprises: the current stage $t_s \in [0, t_S]$ of the EE architecture (with t_S being the maximum number of stages), the memory usage $m \in [0, 1]$ (meaning 1 that the memory is fully in use), and the battery State Of Charge (SOC) $b \in [0, 1]$ (meaning 1 that the battery is fully charged). At each timestamp, the possible actions are either to *locally elaborate* or to *delegate* the current task to the remote node.

1) Cost Functions: In what follows, we define the costs of the two possible decisions.

a) Cost of Delegation: The *delegation cost* function estimates the cost of delegating (*off-loading*) the task in terms of the memory currently in use m and the current stage t_s:

$$c_d(t) = \gamma_{d1} \cdot (1 - m(t)) + \gamma_{d2} \cdot e^{t_s(t) - t_S} \tag{1}$$

b) Costs of Local Stage: The *stage cost* function estimates the cost of the EE stage in terms of the confidence trend and the current state:

$$c_s(t) = \gamma_{s1} \cdot e^{-z'(t_s(t))} + \gamma_{s2} \cdot \frac{2}{\pi} arctan\left(\frac{m(t)}{b(t)}\right) \tag{2}$$

with the confidence z defined as:

$$z = 2 \cdot max_{i=1,\dots,I}(|s_i - 0.5|) \tag{3}$$

where I is the number of keyframes and s_i is the output of the current exit in the EE-CNN about the i-th keyframe.

2) Discount: Next, we introduce the *discount* function. It accounts for the condition the model expects to find in the future. The discount function has two terms, the *battery discount* d_b and the *memory discount* d_m:

$$d(t) = d_b(t) \cdot (1 - d_m(t)) \tag{4}$$

We define the battery discount as:

$$d_b(t) = \begin{cases} 1 & \text{if } x(t) \leq L(t) \\ \dfrac{x_{\text{MAX}} - x(t)}{x_{\text{MAX}} - L(t)} & \text{if } L(t) < x(t) \leq x_{MAX} \\ 0 & \text{if } x(t) > x_{MAX} \end{cases} \tag{5}$$

where $x(t)$ is the number of tasks expected in a future time window and $L(t)$ is the locally tractable amount of tasks with the battery available at time t. We define the memory discount as:

$$d_m(t) = min\left(\frac{m_t(t) + x(t)}{m_{MAX}}, 1\right) \tag{6}$$

where $m_t(t)$ is the number of tasks in memory at time t, and m_{MAX} is the maximum number of tasks that the memory can handle. Finally, we use logarithmic barrier functions to keep

TABLE I: Performance comparison between the proposed system (with edge and remote nodes) and the baseline (remote node only).

	System (Pre)	Baseline (Pre)	System (Inf)	Baseline (Inf)	System (Transf)	System (Tot)	Baseline (Tot)	Savings
Duration	2404s	1229s	227s	125s	34s	2665s	1355s	**-96.74%**
Tot Energy	11.71 Wh	50.73 Wh	0.56 Wh	7.06 Wh	0.05 Wh	12.32 Wh	57.79 Wh	**78.68%**
CPU Energy	8.47 Wh	23.89 Wh	0.22 Wh	2.42 Wh	0.02 Wh	8.71 Wh	26.31 Wh	**66.90%**
GPU Energy	0.68 Wh	13.87 Wh	0.16 Wh	3.33 Wh	0.00 Wh	0.83 Wh	17.20 Wh	**95.16%**
RAM Energy	2.57 Wh	12.97 Wh	0.19 Wh	1.31 Wh	0.03 Wh	2.78 Wh	14.23 Wh	**80.51%**

TABLE II: Classification performance comparison between the proposed system and the baseline.

	Accuracy	Precision	Recall	F1 Score	% ViT
System	0.958	0.980	0.907	0.942	**2.8%**
Baseline	0.986	1.000	0.963	0.981	100%

the battery within its optimal operating range [5]. Outside the optimal range, the barrier function $b_l(t)$ dominates the decision in favor of delegation when $SOC < 0.4$, while the barrier function $b_r(t)$ dominates the decision in favor of local computation when $SOC > 0.8$.

3) Minimization function: The best action a at time t is the one providing the minimum cost:

$$a_t^* = \text{argmin}_a \; a\big[(1 - d(x(t))) \cdot c_s(t) \cdot b_l(t)\big] + (1-a)\big[d(x(t)) \cdot c_d(t) \cdot b_r(t)\big] \tag{7}$$

subject to: =

$$b(t) - [a \cdot C_l + (1-a) \cdot C_d] \geq 0 \tag{8}$$

where C_l and C_d are the energy costs of a local stage of computation and delegation, respectively. The possible actions are: $a = 1$, i.e., a local stage choice, and $a = 0$, i.e., a delegation. Finally, Equation 8 expresses that the battery level should be adequate to perform the chosen action at all times.

In the following, we choose $\gamma_{s1} = 0.402$, $\gamma_{s2} = 0.598$, $\gamma_{d1} = 0.322$, $\gamma_{d2} = 0.678$ as these values provide the least imbalance between the additive components T of each cost function (Eqs. 1 and 2).[1]

IV. Evaluation

We conduct the evaluation using real-world environmental data, combined to emulate realistic usage scenarios. We collected environmental data from the Solcast API [15] and implemented an emulation environment that replicates real PV panel power outputs, inspired by the PV-GIS tool [6]. The code and evaluation results are available at: https://github.com/XAIber-lab/ETA-DyNN.

In order to evaluate the proposed sysyem, we use standard classification metrics along with CPU, GPU, and RAM energy consumptions (Wh), and the percentage of *dropped tasks* [2]. We compared the performance of the EE-CNN model against the ViT4V [7] model on the same dataset, the Varroa Destructor Video Dataset.

Finally, in the evaluations that follow, we will assume $lt_0 = 0.1$, $ut_0 = 0.9$, $lt_1 = 0.2$, and $ut_1 = 0.8$ for the lower and upper confidence thresholds of the first and second exit, respectively, of the EE-CNN model, balancing between classification performance and energy savings. We adopt an experimental setup that closely reflects a real deployment scenario: an Nvidia Jetson Orin Nano as the edge device, running Ubuntu 22.04.5 LTS, and a remote node with an Intel Core i9-7900X CPU and an Nvidia GV100 Titan V GPU, running Ubuntu 24.04.4 LTS.

V. Results

A. EE-CNN performance evaluation

Table I shows an exhaustive comparison of the energy performance, where *System* refers to the above setup (edge and remote node), and *Baseline* refers to a setup where every component of the system runs on the remote node. The proposed system saves 78.68% of the energy used by the baseline. The CPU and GPU energy consumptions are reduced by more than 66% and 95%, respectively, given the lower resource footprint required, and the RAM energy consumption is reduced by more than 80%, given that the EE-CNN model uses just 5 frames (on average) to draw its inference, instead of the entire video. However, the duration almost doubles (-96.74%) because of the edge device's lower computational power. Conversely, Table II shows that the classification performance remains almost untouched (e.g., 0.958 accuracy instead of 0.986), even though the ViT contributes to just 2.8% of the answers. Overall, these results show that some of the ViT model's higher computational costs are unjustified and can be avoided, since answer validity is largely preserved.

B. Environment-Aware Task Allocation System: Daily Performance Comparison

To show consistent behavior over an extended time horizon, we simulated its use under the environmental conditions of 2024 (see Figure 1). In the following, we assume a hardware configuration with a PV module of $40\ W$ Max Power Output, and battery and memory capacities of $200\ Ah$ and 2000 *number of tasks*, respectively.

[1] With imbalance defined as:

$$\text{imb}(\gamma_1) = \frac{\left|\gamma_1 \overline{T_1} - (1-\gamma_1)\overline{T_2}\right|}{\gamma_1 \overline{T_1} + (1-\gamma_1)\overline{T_2}} \in [0,1],$$

with $\overline{T} = \sqrt{\mathbb{E}[T^2]}$, the root mean square value of the component.

[2] We define a task as *dropped* if, at the time a new task is collected (i.e., when a bee passes in front of the cameras), the system is either in recovery mode or the memory buffer is full

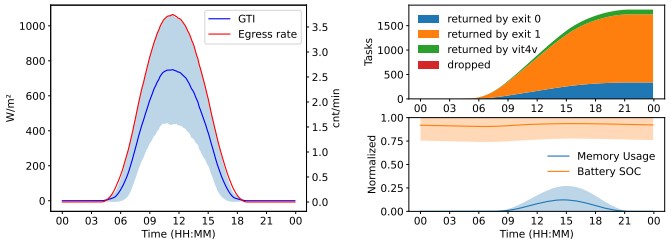


Fig. 1: Average daily performance during 2024: GTI and egress rate (left), proportion of tasks returned by each component of the EE-CNN model (top-right), and battery and memory conditions (bottom-right).

The system's decisions effectively promoted mindful resource use. Until around 09:00, the system delegates only those tasks for which the local CNN has low confidence. As midday approaches, the system correctly begins delegating a fraction of tasks, including some that the local CNN model could solve. Around 15:00, the GHI starts to fade, and the system correctly resumes solving tasks locally, exploiting local resources. No tasks are dropped: the battery never drops below the cut-off limit (20% SOC), and memory usage is also limited, with all tasks completed within each day. The percentage of answers provided by the remote node oscillates between 2.83% and 19.24%, since under high GHI conditions, the system harvests a large amount of energy, enabling it to delegate only those tasks for which the local CNN has low confidence, as opposed to lower GHI conditions, where the system harvests less energy and therefore has to use it more parsimoniously.

VI. Conclusions

In this work, we develop a CNN-based Early-Exit method for image classification to detect Varroa mites in beehives. Depending on the device and environmental conditions, the proposed solution allocates inference tasks to the local edge device and delegates them to a remote service only when needed, maximizing energy savings while preserving classification performance. Experimental evaluation shows up to 78.68% energy savings while achieving 0.958 accuracy. A simulation over an entire year of usage demonstrates the effectiveness of the environmentally aware task allocation algorithm, promoting parsimonious use of computational resources. As future work, we plan to extend the analysis to individual seasons and hardware dimensioning and deploy the system in the field to further validate its performance and sustainability under actual operating conditions. Finally, although the proposed methodology has been tested on a specific application domain, its generality is broader, contributing to the field of sustainable AI [11].

References

[1] Selcan Kaplan Berkaya et al. "Deep learning-based classification models for beehive monitoring". In: *Ecological Informatics* 64 (2021), p. 101353.

[2] Kim Bjerge et al. "A computer vision system to monitor the infestation level of Varroa destructor in a honeybee colony". In: *Computers and Electronics in Agriculture* 164 (2019), p. 104898.

[3] Marcello Bullo et al. "Energy-Aware Dynamic Neural Inference". In: *arXiv preprint arXiv:2411.02471* (2024).

[4] Zina-Sabrina Duma et al. "Varroa destructor detection on honey bees using hyperspectral imagery". In: *Computers and Electronics in Agriculture* 224 (2024), p. 109219.

[5] Rasha Elazab et al. "A novel holistic metric for sustainability assessment of photovoltaic/battery systems". In: *Scientific Reports* 15.1 (2025), p. 30018.

[6] European Commission, Joint Research Centre (JRC). *Photovoltaic Geographical Information System (PVGIS) — PV Tools*. https://re.jrc.ec.europa.eu/pvg_tools/en/. Accessed 2025; online solar resource and PV performance tools by JRC. 2025.

[7] Luca Giovannesi et al. "Vit4V: a Video Classification Method for the Detection of Varroa Destructor from Honeybees". In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 5343–5351.

[8] Dave Goulson et al. "The canary in the coalmine; bee declines as an indicator of environmental health". In: *Science Progress* 99.3 (2016), pp. 312–326.

[9] Yizeng Han et al. "Dynamic neural networks: A survey". In: *IEEE transactions on pattern analysis and machine intelligence* 44.11 (2021), pp. 7436–7456.

[10] Andrew Howard et al. "Searching for mobilenetv3". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 1314–1324.

[11] Barbara Pernici et al. "Sustainable Quality in Data Preparation". In: *J. Data and Information Quality* 17.4 (Dec. 2025). ISSN: 1936-1955. DOI: 10.1145/3769120. URL: https://doi.org/10.1145/3769120.

[12] Romina Rader et al. "Non-bee insects are important contributors to global crop pollination". In: *Proceedings of the National Academy of Sciences* 113.1 (2016), pp. 146–151.

[13] Haseena Rahmath P et al. "Early-exit deep neural network-a comprehensive survey". In: *ACM Computing Surveys* 57.3 (2024), pp. 1–37.

[14] Stefan Schurischuster et al. "Image-based classification of honeybees". In: *2020 Tenth International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE. 2020, pp. 1–6.

[15] Solcast. *Global solar irradiance data and PV system power output data*. https://solcast.com/. Accessed 2025. 2025.

[16] Wayne Wolf. "Key frame selection by motion analysis". In: *1996 IEEE international conference on acoustics, speech, and signal processing conference proceedings*. Vol. 2. IEEE. 1996, pp. 1228–1231.