A study on edge devices for image classification of the Tasmanian devil (*Sarcophilus harrisii*) for vaccine delivery

Prithul Chaturvedi[1], Andrew S. Flies[1*], William M. Connelly[2]

[1] Menzies Institute for Medical Research, College of Health and Medicine, University of Tasmania, Hobart, TAS 7000, Australia

[2] Tasmanian School of Medicine, College of Health and Medicine, University of Tasmania, Hobart, TAS 7000, Australia

* Corresponding author: Andrew S. Flies (andy.flies@utas.edu.au)

Acknowledgements: We acknowledge the traditional custodians of the lands on which we live, and pay our respects to elders past and present. We thank the Department of Natural Resources and Environment Tasmania for providing trail camera images. We also thank S. Caren Han for her guidance in image classification model development. This study was funded by the Australian Research Council Linkage Projects grant LP210301148 awarded to Andrew S. Flies.

Data availability: Data will be made available upon request.

Conflict of interest statement: The authors declare no competing interests.

Author contribution: PC, ASF and WMC conceptualised the approach of using image classification on an edge device for target-specific bait delivery. ASF enabled the procurement of trail camera images for model development. PC and WMC developed the models and conducted experiments on the edge devices. PC led the writing of the manuscript. All authors contributed critically to the drafts and gave final approval for publication.

ABSTRACT

1. A target-specific bait dispenser is required for oral bait vaccination of the endangered Tasmanian devil (*Sarcophilus harrisii*) against the devil facial tumour disease. Development of a dispenser that can be extended to other species would be a beneficial conservation tool. A camera-based edge device with an onboard deep learning model can be used to identify the target species using image classification.
2. It is important to choose a suitable edge device for the dispenser's effective application in power constrained environments. In this study, we evaluated four edge devices for the smart bait dispenser— ArduinoPro Nicla Vision, ArduinoPro Portenta H7 with Vision Shield (LoRa), Raspberry Pi 3 Model B+ and Raspberry Pi Zero 2 W. Two simple convolutional neural networks (CNN1 and 2), and four fine-tuned pretrained models (MobileNetV2, MobileNetV3Small, ResNet50V2, and ResNet152V2) were trained on trail camera images of devil and non-devil species. These models were evaluated across four metrics and post-training quantised for deployment. The edge devices were assessed on inference latency of each model in seconds (s) and power consumption in watt (W).
3. We found that a simple CNN based image classification model yielded the best overall result of ~96% across all metrics. Raspberry Pi 3 Model B+ and Raspberry Pi Zero 2 W could run all the six models whereas ArduinoPro Portenta H7 with Vision Shield failed to run the two ResNet models and ArduinoPro Nicla Vision failed to run all four fine-tuned pretrained models. All edge devices were found to be quick enough for dispenser application (average inference latency of 0.718 s). ArduinoPro Portenta H7 Vision Shield consumed the lowest power during inference (0.19 W), idle (0.185 W), and light sleep (0.045 W) states whereas ArduinoPro Nicla Vision consumed the lowest power during deep sleep state (0.002 W).

4. Overall, we conclude that a simple CNN image classifier, CNN1, deployed on ArduinoPro Portenta H7 with Vision Shield (LoRa) is the most suitable edgeAI application for a smart bait dispenser. We suggest integration of this system with a sleep-mode efficient microcontroller unit like ATtiny85 to minimise power consumption of the dispenser for multiday field application. This method can be readily adapted for other species, such as eastern quoll and racoons, for projects like supplemental feeding and vaccination.

Keywords: Bait dispenser, EdgeAI, Image classification, Microcontroller unit, Oral bait vaccination, Single-board computer, Tasmanian devil, TinyML, Wildlife conservation

1 INTRODUCTION

The Tasmanian devil (*Sarcophilus harrisii*; hereafter devil) is the largest extant carnivorous marsupial and the native apex predator of Tasmania, Australia. The species has been listed as endangered by the International Union for Conservation of Nature and Natural Resources (IUCN) since 2008 (Hawkins, 2008), largely due to the emergence of two transmissible cancers— devil facial tumour 1 (DFT1) and DFT2. Each of the tumours can independently cause devil facial tumour disease (DFTD). The disease has caused regional declines of up to 82% with nearly 100% case fatality rates (Cunningham et al., 2021).

Oral bait vaccines (OBVs) have been widely used as a wildlife disease management strategy for rabies (reviewed in (Rupprecht et al., 2024)). A vaccine against DFTD that could be delivered in edible baits would be a powerful management and conservation tool (Flies et al., 2020). These baits can be hand distributed or dropped from airplanes and helicopters in target population areas. Whilst these approaches are effective at a broad scale, they can be costly due to consumption by non-target species and consumption of multiple baits by a single individual. Maximising bait uptake by devils while minimising uptake by sympatric species would be a key for economical and effective control of DFTD.

In ecological systems where non-target competition is a significant challenge, bait dispensers have been used (Bastille-Rousseau et al., 2024; Smyser et al., 2015; Whisson & Salmon, 2009). Initial trials of ground distribution of placebo baits in Tasmania resulted in devils consuming around 7% of the baits; eastern quolls (*Dasyurus viverrinus*), brushtail possums (*Trichosurus vulpecula*), and Tasmanian pademelons (*Thylogale billardierii*) consumed approximately 75% of the baits (Dempsey et al., 2022). Use of a timer-based bait dispenser developed for raccoons (*Procyon lotor*) in North America (Smyser et al., 2015) was trialled in Tasmania and resulted in a nearly tenfold increase in bait uptake by devils (Dempsey et al., 2022). However, eastern quolls often removed the baits from the dispenser before devils arrived. Furthermore, the

relatively small home range of eastern quolls led to individuals frequently returning to the dispenser to retrieve additional baits

A dispenser that implemented species recognition technology could reduce bait consumption by non-target species and better limit the number of baits consumed by an individual. Designing a dispenser that could be rapidly adapted to target alternative species would be a broadly useful conservation tool. This would have to balance multifunctionality with the increased cost of adding features and potential reductions in durability.

Deep learning (DL) is a sub-field of machine learning (ML) that extracts higher-level features from data (e.g. images) with limited feature engineering (Choudhary et al., 2022; LeCun et al., 2015). DL methods are applicable in a wide range of domains due to its scalability, generalisability, and robustness (Alzubaidi et al., 2021). However, they are computationally demanding and are typically deployed over high-performance infrastructures or distributed clusters in data centres (Guo, 2018). Here, input data is sent to another location for processing via a network. This requires a secure and private connection for deployment, which may not be feasible in field applications. Alternatively, it is possible to run scaled-down DL models on smaller and cheaper edge devices like single board computers (SBCs) and microcontroller unit (MCU) boards (reviewed in (Kong et al., 2022)). Here, input data is captured and processed in the same location, enabling secure real-time processing (Warden & Situnayake, 2019) and allowing decisions to be made in the field. SBCs and MCU boards, coupled with a camera, are used to deploy DL methods such as object detection and image classification (reviewed in (Iqbal et al., 2024) and (Maheepala et al., 2021)). This is also referred to as edge artificial intelligence (edgeAI).

EdgeAI applications in camera traps and acoustics have been used in conservation for species recognition to remotely monitor wildlife populations and to detect poaching and illegal deforestation (Fergus et al., 2024; Schwartz et al., 2021). A five-bait dispenser prototype was

successful in targeting feral dogs (*Canis familiaris*) based on real-time camera trap images using object detection method (Charlton et al., 2023). This system is based on Raspberry Pi 3 Model B+ that requires a large battery to provide power for multiday field applications, making execution costly and setup bulky. Raspberry Pi Zero 2 W is a smaller and more power efficient alternative but has comparatively limited computing power. Alternatively, using a high-end microcontroller unit device that offers comparable computational capabilities for image-based species classification can run the dispenser longer on smaller batteries. Overall, balancing edge device capabilities with the power requirements is crucial for field applications.

In this study, we compare four edge devices for a smart bait dispenser— two Arduino-based microcontroller unit devices, ArduinoPro Nicla Vision and ArduinoPro Portenta H7 with Vision Shield (LoRa), and two Raspberry Pi-based single board computer devices, Raspberry Pi 3 Model B+, and Raspberry Pi Zero 2 W. We conducted a preliminary assessment of 31 convolutional neural network (CNN) models to identify the most suitable models for devil classification on edge devices. Here, we discuss six of the best performing models based on accuracy, precision, recall, and F1-score metrics. The edge devices were scrutinised on inference latency and power consumption, as these are the most important factors for successfully implementing the bait dispenser in energy deficit remote locations.

2 METHODS

2.1 Dataset

The images used in this study were provided by the Department of Natural Resources and Environment Tasmania. These 1920×1080 pixels photos were collected during routine wildlife monitoring activities in Tasmania between 2015 and 2020 using Reconyx HC500 trail cameras. The dataset contained images of relevant species: devil, eastern quoll, spotted tail quoll (*Dasyurus maculatus*) wombat (*Vombatus ursinus*), brushtail possum (*Trichosurus vulpecula*), Bennett's wallaby (*Notamacropus rufogriseus*), Tasmanian pademelon, feral cat (*Felis catus*), bandicoots (*Isoodon obesulus* and *Perameles gunnii* species), unidentified small mammals like rodent (presumable *Rattus spp.* and *Mus spp.*), snakes, cattle, birds, and empty background (Figure 1).

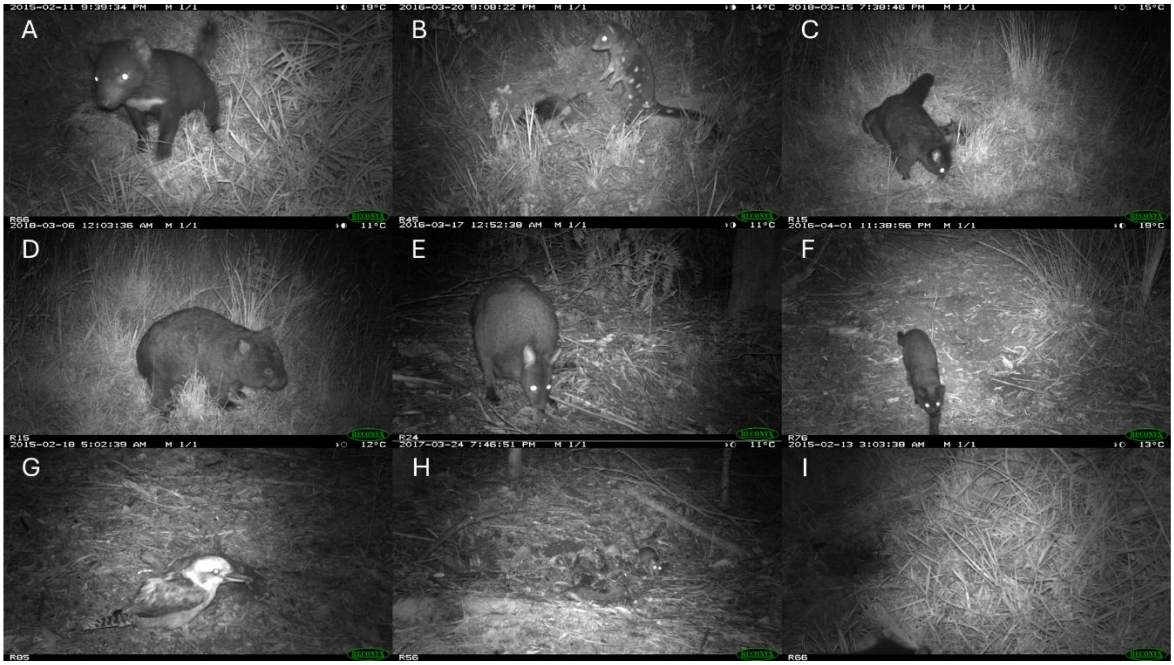

Figure 1. Example images used in training image classification models. (A) Tasmanian devil, (B) spotted tail quoll, (C) brushtail possum, (D) wombat, (E) wallaby, (F) feral cat, (G) kookaburra, (H) rodent, and (I) background.

A total of 9315 images were manually annotated as devil and non-devil. The devil class consisted of images where clearly distinctive features of a Tasmanian devil were visible face (eyes, ears, hairless snout, DFTD tumour, mouth/teeth), tail, and white markings on chest, arm, or back. During initial experiments, we found that partial images of wombats and possums generated false-positive predictions across all models. To minimise this, we created a "devil without distinctive features" subclass under non-devil class. The other subclasses under non-devil class were cat, wombat, macropods, quolls, possum, and miscellaneous. The miscellaneous class was created to include images of species like rodent, bandicoot, birds, and cattle that are seldom caught on trail cameras, and empty background.

The dataset was randomly split into training (80%, including validation) and testing (20%) subsets. To ensure that our data contained a representative distribution of species, stratified randomisation was used on the non-devil class. To improve intra-class variability, each image was passed through a 'data augmentation layer' consisting of random horizontal flips and random rotation at factor = 0.05 before feeding into models. The images were rescaled to 54×54 pixels for the classification algorithms to meet the memory limitations of the MCUs.

2.2 Model training

All computations were carried out using Python 3 on JupyterLab hosted on the Australian Research Data Commons' Nectar Research Cloud. Keras application programming interface (API) was used in model development (Chollet, 2015). Seaborn and Matplotlib were used to generate visualisations (Hunter, 2007; Waskom, 2021).

2.2.1 Models

For this study, we considered 16 custom trained convolutional neural networks (CNN) and 15 pretrained models for transfer learning. This was to experimentally identify the most suitable candidates for binary image classification on edge devices. Here we focus on the best two CNNs and the best four pretrained models. The architectures of the chosen CNNs are in Table 1. The top performing pretrained models were ResNet50V2, ResNet152V2 (both architectures in (He et al., 2016)), MobileNetV2, MobileNetV3 (architectures in (Howard, 2017) and (Howard et al., 2019), respectively).

Table 1. Architectures of models (A) CNN1 and (B) CNN2.

A. Model CNN1

Layer Name	Type	Output shape	Parameters
1.1	Conv2D	(54, 54, 32)	896
1.2	MaxPooling2D	(27, 27, 32)	0
2.1	Conv2D	(27, 27, 32)	9,248
2.2	MaxPooling2D	(13, 13, 32)	0
3.1	Conv2D	(13, 13, 16)	2,064
3.2	MaxPooling2D	(6, 6, 16)	0
4	Dropout (0.25)	(6, 6, 16)	0
5	Flatten	(576)	0
6	Dense	(1)	577

Total parameters: 12,785

B. Model CNN2

Layer Name	Type	Output Shape	Parameters
1.1	Conv2D	(54, 54, 32)	896
1.2	MaxPooling2D	(27, 27, 32)	0
2.1	Conv2D	(27, 27, 32)	9,248

Layer Name	Type	Output Shape	Parameters
2.2	MaxPooling2D	(13, 13, 32)	0
3.1	Conv2D	(13, 13, 16)	2,064
3.2	MaxPooling2D	(6, 6, 16)	0
4.1	Conv2D	(6, 6, 16)	1,040
4.2	MaxPooling2D	(3, 3, 16)	0
5	Dropout (0.25)	(3, 3, 16)	0
6	Flatten	(144)	0
7	Dense	(1)	145

Total parameters: 13,393

2.2.2 Model training hyperparameters

a. Image size: 54×54 (dimensions) ×3 (channels) pixels
b. Batch size: 128 images
c. Loss function: Binary cross-entropy
d. Optimiser: Adaptive Moment Estimation (Adam)
e. Learning rate:
    i. For CNNs: 10^{-3}
    ii. For pretrained models: 10^{-3} for training the top layer and 10^{-5} for fine tuning
f. Epochs:
    i. For CNNs: 300
    ii. For pretrained models: 50 for training the top layer and 250 for fine-tuning

2.2.3 Post-training quantisation

All full-size models were quantised to reduce model size and, consequently, latency on edge devices at the cost of an acceptable dip in model performance using the following three techniques:

1. Dynamic range Float32 (FP32) quantisation: Model weights and activations are converted to 8-bit integers (INT8), shrinking to approximately one quarter of its original size; integers are dequantized back to FP32 values during inference. F32 models can only run on SBCs.
2. Float16 (F16) quantisation: Model weights and activations are converted to F16 values, shrinking model size to half. F16 models can only run on SBCs.
3. Full integer quantisation: Model weights and activations are converted to INT8, shrinking model size to quarter. INT8 models can run on both MCUs and SBCs.

The intent behind exploring three types of post-training quantisation methods was to thoroughly assess the performance metrics of full models against smaller counterparts for edge devices. Since our goal is to evaluate two SBC and two MCU, running selected I8Q models provides head-to-head comparison. TensorFlow Lite (also known as LiteRT) was used for post-training quantisation (Developers, 2025).

2.2.4. Model evaluation

The optimal classification threshold for each original and quantised model was determined by the maximum Youden's J statistic on the receiver operating characteristic (ROC) curve. If fpr and tpr are false positive and true positive rates respectively, then the optimal threshold t is characterised by:

$$J_{max} = \max_t[fpr(t) - tpr(t)]$$

We chose accuracy, precision, recall, and F_1-score as the metrics to assess each model's predictive performance on the test dataset. If F_P, F_N, T_P, and T_N denote false positive, false negative, true positive, and true negative respectively, then:

a. $\text{Accuracy} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N}$

b. $\text{Precision} = \frac{T_P}{T_P + F_P}$

c. $\text{Recall} = \frac{T_P}{T_P + F_N}$

d. $F_1\text{-score} = 2\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

Scikit-learn was used to evaluate all full-size and quantised models (Pedregosa et al., 2011).

2.3 Edge devices

The following devices were considered for use in the smart dispenser in this study:

1. ArduinoPro Nicla Vision (hereafter Nicla Vision) (Arduino) with GC2145 onboard camera (GalaxyCore).
2. ArduinoPro Portenta H7 (Arduino) with Vision Shield (LoRa) (hereafter Portenta) (Arduino) with HM01B0 onboard camera (Himax).
3. Raspberry Pi 3 Model B+ (hereafter RPi 3B+) (Pi) with Waveshare OV5647 camera module (Sparkfun; Waveshare).
4. Raspberry Pi Zero 2 W (hereafter RPi Z2W) (Pi) with Waveshare OV5647 camera module (Sparkfun; Waveshare).

2.3.1 Firmware/operating system

OpenMV firmware v4.7 was used for microcontroller boards Nicla Vision and Portenta. Raspberry Pi OS Legacy 32-bit Lite Debian v11 Bullseye was used for single board computers RPi 3B+ and RPi Z2W.

2.3.2 Inference latency

Inference latency data was collected by simulating the process of executing the six models on images captured by the respective camera of each edge device. Studying inference response time for the smart dispenser were independent of the content captured in these images and corresponding predictions. Algorithm 1 describes our approach of computing average inference latency (10 images) over ten iterations for each model across the four edge devices. Tensorflow Lite (tflite-runtime) on Raspberry Pi SBCs and OpenMV's Machine Learning module (ml) on the Arduino MCU devices were used to run inference. Python/MicroPython's high-resolution timer time.time() was used to capture inference latency on all SBC and MCU devices. Model, images and result objects were deleted after computational use to clear volatile memory for subsequent iterations.

Algorithm 1: Simulating inference for smart bait dispenser

Input: Image of dimension 54×54 pixels *Im* from camera of edge device
Output: Average lapsed inference time T_{avg}

```
D, T, T_avg ← {}, {}, {}                    // Initialise dataset, time and average time arrays
for each m ∈ models do
    Load m
    for i from 1 to 10 do
        for image from 1 to 10 do
            D[m][i][image] ← D[m][i][image] + Im                //Accumulate images
            Delete Im
        end for
```

```
        T_start ← Timer()                                   // Invoke start timer
        R ← m(D)                                            // Invoke inference
        T_stop ← Timer()                                    // Invoke stop timer
        T[m][i] ← T[m][i] +(T_stop-T_start)                 // Accumulate elapsed time
        Delete D, T_start, T_stop, R
    end for
    T_avg[model] ← Average(T[model])        // Accumulate average elapsed time for m
    Delete model
    Delete T
end for
return T_avg
```

2.3.3 Power consumption

We assessed the power consumption of each edge device in watt (W) across four operational states or modes— idle (or standby, not executing any task from program), capturing images and performing inference, deep sleep, and light sleep. Deep and light sleep are power-saving modes in MCUs that can be woken up with a timer or external interrupt to general-purpose input/output (GPIO) pins. Deep sleep minimises power consumption by powering down most internal modules, resulting in loss of volatile memory state. Light sleep retains volatile memory state by maintaining power to key internal modules. Thus, light sleep can resume operations from last operation executed before entering sleep mode. Since deep and light sleep modes are featured in only MCU, the "sleep" mode for SBCs was simulated with idle mode.

FNIRSI FNB38 universal serial bus meter (FNIRSI, 2020) was used to measure voltage in volt (V) and current drawn (I) in ampere (A), respectively, by the edge devices when running processes. The power consumption (P) was calculated by $P = V \times I$. When the current draw was lower than one milliampere, a shunt resistor of 2.7 ohm was connected in series between ground and voltage input pins of the edge devices. The voltage drop was measured and used in calculation of current drawn by this formula: $I = V/R$.

3 RESULTS

3.1 Image classification models

7452 training images and 1863 testing images were respectively used in training and assessing the binary classification models to identify Tasmanian devils from other species commonly found in Tasmania. Model sizes and test performance metrics across all full and quantised models is given in Table 2. CNN 1 was the smallest in size, followed by CNN2, MobileNetV3Small, MobileNetV2, ResNet50V2, and ResNet152V2, across all quantisation methods. As expected due to being simple models, CNN 1 and 2 were substantially smaller in size than pretrained models. A general dip in performance was observed in all quantised models compared to their respective original models; MobileNetV2 had a considerable improvement across all performance metrics (Figure 2). Quantising models typically reduces model performance, though opposite effects have been noted previously (example in (Van Baalen et al., 2023)). Custom trained CNN 1 achieved the highest accuracy, recall, and F1-score across all architectures and models. Although full sized CNN 1 had a higher precision than full sized CNN2, the latter achieved best precision of all the quantised counterparts.

Table 2. Comparison of size (in kilobyte) and test performance of original and post-training quantised models across metrics. Best model size (smallest) and metric scores (highest) per quantisation group are **bolded**, second-best are <u>underlined</u>.

Model	Quantisation	Size (KB)	Accuracy	Precision	Recall	F1-score
CNN 1	Original	**192**	**0.9703**	**0.9656**	**0.9580**	**0.9618**
	INT8	**66**	**0.9643**	<u>0.9471</u>	**0.9619**	**0.9545**
	FP32	**63**	**<u>0.9643</u>**	<u>0.9403</u>	**0.9697**	**0.9548**
	FP16	**119**	**0.9646**	<u>0.9421</u>	**0.9688**	**0.9552**
CNN 2	Original	<u>206</u>	<u>0.9688</u>	<u>0.9654</u>	<u>0.9541</u>	<u>0.9597</u>
	INT8	<u>76</u>	<u>0.9627</u>	**0.9639**	<u>0.9395</u>	<u>0.9515</u>

Model	Quantisation	Size (KB)	Accuracy	Precision	Recall	F1-score
	FP32	73	**0.9643**	**0.9506**	0.9580	0.9543
	FP16	136	0.9639	**0.9523**	0.9551	0.9537
MobileNetV2	Original	17,502	0.7106	0.5936	0.8145	0.6867
	INT8	2,648	0.8601	0.7752	0.9023	0.8339
	FP32	2,449	0.8897	0.8300	0.9014	0.8642
	FP16	4,362	0.8802	0.8038	0.9160	0.8562
MobileNetV3Small	Original	6,630	0.9038	0.8427	0.9258	0.8823
	INT8	1,190	0.5878	0.4752	0.5615	0.5148
	FP32	1,086	0.7049	0.6142	0.6514	0.6322
	FP16	1,860	0.7221	0.6121	0.7813	0.6864
ResNet152V2	Original	364,048	0.9510	0.9534	0.9189	0.9359
	INT8	58,857	0.9228	0.9298	0.8672	0.8974
	FP32	58,330	0.9361	0.8934	0.9492	0.9205
	FP16	113,791	0.9426	0.9217	0.9316	0.9267
ResNet50V2	Original	154,238	0.9452	0.9239	0.9365	0.9302
	INT8	23,680	0.9114	0.8707	0.9072	0.8886
	FP32	23,489	0.9270	0.8796	0.9414	0.9094
	FP16	45,990	0.9274	0.8832	0.9375	0.9095

Since our goal for training an image classification model is to use it in a smart bait dispenser for devils with potential of easy downstream adaptation for other target species, it is important that the model correctly identifies the target species, i.e., is high in sensitivity. This is because we would rather the dispenser deliver a bait to a non-target species occasionally than fail to drop a bait when a devil is present. Hence, recall (ratio of correctly predicted devils out of all devils in the dataset) should be considered as the primary metric of model evaluation. Additionally, as a complementary metric and indicator for holistic performance, we suggest that using F1-score to compare models. This metric provides a balance between precision and recall and is more insightful than accuracy when class imbalance is present in dataset, which is expected in real wildlife conservation scenarios. Since MCU boards can only support INT8

quantised models, we narrowed our focus on INT8 models only for head-to-head comparison across both MCU and SBC boards. INT8 quantised CNN 1 achieved the best results in both recall (96.19%) and F1-score (95.45%), making it the most suitable model for the smart bait dispenser based on test data evaluation.

3.2 Inference on edge devices

3.2.1 Latency

Observed inference latency is given in Table 3. The two SBCs— RPi 3B+ and RPi Z2W were able to run all the models, opposed to the MCU boards where Portenta could not run the two models of the ResNet family and Nicla Vision could not run any of the four pretrained models. The fastest average inference time of 0.069 seconds was observed on the Portenta running MobileNetV3Small, followed by 0.091 seconds on RPi Z2W running MobileNetV3Small, and 0.149 seconds on Nicla Vision running CNN 1. The slowest average inference time recorded was 2.836 seconds, when RPi 3B+ was running ResNet152V2. From these results, we consider inference times across all edge device to be quick enough for the bait dispenser. Hence, we must give a higher preference to INT8 performance metrics to identify the most suitable model for field deployment.

Table 3. Comparison of inference latency (in seconds) across different models. Lowest latencies under each model are **bolded**, second-lowest are <u>underlined</u>. N/A indicates data not available because model could not run on device.

Device	CNN 1	CNN 2	MobileNet V2	MobileNet V3Small	ResNet50 V2	ResNet152 V2
ArduinoPro Nicla Vision	0.149	**0.150**	N/A	N/A	N/A	N/A
ArduinoPro Portenta H7 + Vision Shield (LoRa)	0.253	0.255	0.158	**0.069**	N/A	N/A
Raspberry Pi 3 Model B+	1.155	1.937	2.461	0.669	2.800	2.836
Raspberry Pi Zero2 W	**0.124**	0.187	**0.251**	0.091	**0.257**	**0.262**

3.2.2 Power consumption

Minimal power consumption will be critical for bait dispensers deployed in remote wilderness areas. Data on power consumption by the four edge devices across essential activity states are presented in Table 4. As expected, MCU boards utilise much less power than SBCs.

We observed that Portenta is ~90% more efficient than Nicla Vision in both idle and inference modes, but the latter outperforms the former by 160% in light sleep and ~190% in deep sleep. Upon discussion with Arduino developers (forum discussion in (kwagyeman, 2025)), we realised this is expected since Portenta is not optimised for low power. Between the two SBCs, RPi Z2W consumed less power than RPi 3B+ by 72% and 45.4% during idle and inference modes, respectively. Since SBCs do not offer sleep modes, the percentage difference of idle mode between the most power efficient SBC (RPi Z2W, 1.6 W) and the most efficient MCU board (Portenta, 0.185 W) is 158.5%. RPi Z2W in idle mode consumed 198.8% and 199.5% more power than Nicla Vision's light sleep (0.005 W) and deep sleep (0.00187 W) modes, respectively.

Table 4. Power Consumption (in watt) by selected edge devices across different operating modes. Lowest consumptions under each mode are **bolded**, second-lowest are underlined. N/A indicates data not available because device does not have sleep modes.

Device	**Idle**	**Running Inference**	**Sleep mode**	
			Light Sleep	**Deep Sleep**
ArduinoPro Nicla Vision	0.500	0.500	0.005	**0.00187**
ArduinoPro Portenta H7 + Vision Shield (LoRa)	**0.185**	**0.190**	**0.045**	0.070
Raspberry Pi 3 Model B+	3.400	3.650	N/A	N/A
Raspberry Pi Zero2 W	1.600	2.300	N/A	N/A

4 DISCUSSION

In this study, we compared four edge devices to assess the feasibility of using them in a smart bait dispenser for vaccination of Tasmanian devils against DFTD. We short listed two generic and four pretrained CNN-based image classification models to deploy on the considered devices. We conclude CNN 1 to be the best performing model to classify devil from non-devil species based on test data evaluation on metrics recall and F1-score, and average inference latency across the four devices.

The image classification approach allows for rapidly adapting the models for alternate target species. For example, an eastern quoll translocation project is underway to help restore genetic diversity in declining populations in Tasmania {Hamer, 2023 #267}. Supplemental feeding has been used to provide food while the quolls navigate the new site. The bait dispenser could be repurposed to deliver the supplemental food only to quolls. In this case, our dataset already contains eastern quoll images, so reusing the species recognition models only requires defining eastern quoll as the target species and retraining. Similarly, existing camera trap images on any target species, like racoons in urban North America, can be leveraged for this application.

Vaccination campaigns are typically multiday field projects and hence the smart bait dispenser must be power conscious. ArduinoPro Portenta H7 Vision Shield (LoRa) yielded quick inference results and the lowest power consumption during idle state and inference. Having a sleep mode on the dispenser is essential because the system will remain inactive for majority of time in wait until a motion sensor triggers the process to start. An efficient sleep mode will extend the battery life of the dispenser and keep it running in field for days with low to no battery maintenance. As mentioned earlier, Portenta is not optimised for sleep and SBCs RPi 3B+ and RPi Z2W do not have sleep modes. Nicla Vision yielded the lowest deep and light sleep power consumption of 0.002 W and 0.005 W respectively. However, a low-end MCU like Atmel ATtiny85 can attain a much lower deep and light sleep power consumption of up to 6

μW and 30 μW respectively (Atmel, 2013). Hence, ATtiny85 in sleep mode can be used as a low power switch that turns on the primary edge device upon motion. Using this method with RPi 3B+ and RPi Z2W is not suitable for the dispenser because these SBCs have a substantially longer boot-up times than MCUs.

Between the two MCU boards in this study, the 28-pin Portenta has prototyping advantages over the 19-pin Nicla Vision. Portenta's broader input/output capabilities support more extensive external integrations with the bait dispenser. Although both the devices have dual-core processors, Portenta has a larger memory of 8 megabytes (MB) of synchronous dynamic random-access memory and 16 MB flash, whereas Nicla Vision only has 1 MB random-access memory and 2 MB flash. Additionally, Portenta's storage can be conveniently expanded via a memory card using the Vision Shield, streamlining data logging processes. In contrast, Nicla Vision requires a separate compatible memory card reader module to achieve similar functionality.

The GC2145 (2 megapixel) camera of Nicla Vision is superior to HM01B0 (320×240 pixels) camera of Portenta Vision Shield. However, GC2145 includes an infrared filter by default, making it unsuitable to capture images under infrared lights during dark. Hence, this camera inapt for the smart dispenser’s deployment for the nocturnal Tasmanian devil. In contrast, HM01B0 lacks an infrared filter and hence can be used in the dispenser. Recent iterations of the Portenta Vision Shield are manufactured with upgraded HM0360 camera module (640×480 resolution) which includes an infrared filter by default. However, a no infrared (NoIR) variant this camera can be sourced directly from the official manufacturer, Arducam.

The long-range (LoRa) radio communication variant of Portenta Vision Shield can enable field scientists to remotely monitor the bait dispenser via long-range wide area network (LoRaWAN). To achieve this with other edge devices considered in this study, we would have to use an external LoRa module. LoRaWAN is an attractive functionality addition to the

dispenser as it can be used to display live updates on motion triggers, species prediction, and number of baits dispensed on an online dashboard. Applications of LoRaWAN are found in agriculture and animal monitoring ((Boo et al., 2023; Haxhibeqiri et al., 2018; Hunan HKT Technology Co.) and reviewed in (Schulthess et al., 2023)) due to key features like security, low power consumption, ease of deployment and network coverage of over 15 km in clear line of sight in rural areas (Petajajarvi et al., 2015).

Altogether, we conclude that ArduinoPro Portenta H7 Vision Shield (LoRa) is the most competent edge device for development of a camera-based smart bait dispenser to target the Tasmanian devil for vaccination against DFTD. We found that a simple CNN-based image classification model, CNN1, was the best devil classifier in performance and deployment on Portenta. Additionally, we suggest using Atmel ATtiny85 microcontroller at achieve sleep mode on Portenta for multi-day field application.

5 REFERENCES

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1), 53. https://doi.org/10.1186/s40537-021-00444-8

Arduino. *Nicla Vision*. https://docs.arduino.cc/hardware/nicla-vision/

Arduino. *Portenta H7*. https://docs.arduino.cc/hardware/portenta-h7/#tech-specs

Arduino. *Portenta Vision Shield*. https://docs.arduino.cc/hardware/portenta-vision-shield/

Atmel. (2013). *ATtiny25/45/85 datasheet*. https://ww1.microchip.com/downloads/en/devicedoc/atmel-2586-avr-8-bit-microcontroller-attiny25-attiny45-attiny85_datasheet.pdf

Bastille-Rousseau, G., Gorman, N. T., McClure, K. M., Nituch, L., Buchanan, T., Chipman, R. B., Gilbert, A. T., & Pepin, K. M. (2024). Assessing the Efficiency of Local Rabies Vaccination Strategies for Raccoons (Procyon lotor) in an Urban Setting. *Journal of Wildlife Diseases*, *60*(1), 26-38. https://doi.org/10.7589/jwd-d-23-00059

Boo, J., Lim, H., Kim, H., Kim, N., Anderson, J. Q., Chin, W., Wang, M. Y., & Smith, A. H. (2023, 17-20 Oct. 2023). LoRaWAN Implementation for Acoustic Localization of Coyotes. 2023 23rd International Conference on Control, Automation and Systems (ICCAS),

Charlton, G., Falzon, G., Shepley, A., Fleming, P. J. S., Ballard, G., & Meek, P. D. (2023). The Sentinel Bait Station: an automated, intelligent design pest animal baiting system. *Wildlife Research*. https://doi.org/10.1071/wr22183

Choudhary, K., DeCost, B., Chen, C., Jain, A., Tavazza, F., Cohn, R., Park, C. W., Choudhary, A., Agrawal, A., Billinge, S. J. L., Holm, E., Ong, S. P., & Wolverton, C. (2022). Recent

advances and applications of deep learning methods in materials science. *npj Computational Materials*, *8*(1), 59. https://doi.org/10.1038/s41524-022-00734-6

Cunningham, C. X., Comte, S., McCallum, H., Hamilton, D. G., Hamede, R., Storfer, A., Hollings, T., Ruiz-Aravena, M., Kerlin, D. H., Brook, B. W., Hocking, G., & Jones, M. E. (2021). Quantifying 25 years of disease-caused declines in Tasmanian devil populations: host density drives spatial pathogen spread. *Ecology Letters*, *24*(5), 958-969. https://doi.org/10.1111/ele.13703

Dempsey, S., Pye, R. J., Gilbert, A. T., Fountain-Jones, N. M., Moffat, J. M., Benson-Amram, S., Smyser, T. J., & Flies, A. S. (2022). Evaluation of oral baits and distribution methods for Tasmanian devils (Sarcophilus harrisii). *Wildlife Research*.

Developers, G. A. f. (2025). *LiteRT*. https://ai.google.dev/edge/litert

Fergus, P., Chalmers, C., Longmore, S., & Wich, S. (2024). Harnessing Artificial Intelligence for Wildlife Conservation. *Conservation*, *4*(4), 685-702.

Flies, A. S., Flies, E. J., Fox, S., Gilbert, A., Johnson, S. R., Liu, G.-S., Lyons, A. B., Patchett, A. L., Pemberton, D., & Pye, R. J. (2020). An oral bait vaccination approach for the Tasmanian devil facial tumor diseases. *Expert Review of Vaccines*, *19*(1), 1-10. https://doi.org/10.1080/14760584.2020.1711058

FNIRSI. (2020). *FNB38 USB Tester Manual V1.2*. https://www.fnirsi.com/pages/manuals-firmwares?srsltid=AfmBOooK1wUgKcP61WjAGDqTNUbCD28NYTIHK4d0Bg-zaE839qF0IgcD

GalaxyCore. *GC2145 CSP Datasheet V1.0*. https://e2e.ti.com/cfs-file/__key/communityserver-discussions-components-files/968/GC2145-CSP-DataSheet-release-V1.0_5F00_20131201.pdf

Guo, T. (2018, 17-20 April 2018). Cloud-Based or On-Device: An Empirical Study of Mobile Deep Inference. 2018 IEEE International Conference on Cloud Engineering (IC2E),

Hawkins, C. E., McCallum, H., Mooney, N., Jones, M. & Holdsworth, M. (2008). *Sarcophilus harrisii*. The IUCN Red List of Threatened Species 2008: e.T40540A10331066. Retrieved 05 September from https://dx.doi.org/10.2305/IUCN.UK.2008.RLTS.T40540A10331066.en.

Haxhibeqiri, J., De Poorter, E., Moerman, I., & Hoebeke, J. (2018). A Survey of LoRaWAN for IoT: From Technology to Application. *Sensors*, *18*(11). https://doi.org/ARTN 399510.3390/s18113995

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity mappings in deep residual networks. Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14,

Himax. *HM01B0 Ultralow Power CIS*. https://www.himax.com.tw/products/cmos-image-sensor/always-on-vision-sensors/hm01b0/

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., & Vasudevan, V. (2019). Searching for mobilenetv3. Proceedings of the IEEE/CVF international conference on computer vision,

Howard, A. G. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Hunan HKT Technology Co., L. *LoRaWAN Bio-Capsule For Cow*. https://www.hktlora.com/product/lorawan-bio-capsule-for-animal/#:~:text=The%20BC%2D100%20is%20a,no%20harm%20to%20the%20animal.

Iqbal, U., Davies, T., & Perez, P. (2024). A Review of Recent Hardware and Software Advances in GPU-Accelerated Edge-Computing Single-Board Computers (SBCs) for Computer Vision. *Sensors*, *24*(15).

Kong, L., Tan, J., Huang, J., Chen, G., Wang, S., Jin, X., Zeng, P., Khan, M., & Das, S. K. (2022). Edge-computing-driven Internet of Things: A Survey. *ACM Comput. Surv.*, *55*(8), Article 174. https://doi.org/10.1145/3555308

kwagyeman, i., ;. (2025). Lightsleep v deepsleep on Portenta. In *Arduino Boards*. OpenMV Forum: OpenMV.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436-444. https://doi.org/10.1038/nature14539

Maheepala, M., Joordens, M. A., & Kouzani, A. Z. (2021). Low Power Processors and Image Sensors for Vision-Based IoT Devices: A Review. *IEEE Sensors Journal*, *21*(2), 1172-1186. https://doi.org/10.1109/JSEN.2020.3015932

Petajajarvi, J., Mikhaylov, K., Roivainen, A., Hanninen, T., & Pettissalo, M. (2015, 2-4 Dec. 2015). On the coverage of LPWANs: range evaluation and channel attenuation model for LoRa technology. 2015 14th International Conference on ITS Telecommunications (ITST),

Pi, R. *Documentation: Raspberry Pi hardware*. https://github.com/raspberrypi/documentation/blob/develop/documentation/asciidoc/computers/raspberry-pi/introduction.adoc

Rupprecht, C. E., Buchanan, T., Cliquet, F., King, R., Müller, T., Yakobson, B., & Yang, D. K. (2024). A Global Perspective on Oral Vaccination of Wildlife against Rabies. *J Wildl Dis*, *60*(2), 241-284. https://doi.org/10.7589/jwd-d-23-00078

Schulthess, L., Longchamp, F., Vogt, C., & Magno, M. (2023). *A Lora-Based and Maintenance-Free Cattle Monitoring System for Alpine Pastures and Remote Locations* Proceedings of the 11th International Workshop on Energy Harvesting & Energy-Neutral Sensing Systems, Istanbul, Turkiye. https://doi.org/10.1145/3628353.3628549

Schwartz, D., Selman, J. M. G., Wrege, P., & Paepcke, A. (2021, 13-16 Dec. 2021). Deployment of Embedded Edge-AI for Wildlife Monitoring in Remote Regions. 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA),

Smyser, T. J., Redding, J. V., Bevis, C. M., Page, L. K., & Swihart, R. K. (2015). Development of an Automated Dispenser for the Delivery of Medicinal or Vaccine-Laden Baits to Raccoons. *Journal of Wildlife Diseases*, *51*(2), 513-518. https://doi.org/10.7589/2014-08-211

Sparkfun. *OmniVision Technologies OV5647 Datasheet*. https://cdn.sparkfun.com/datasheets/Dev/RaspberryPi/ov5647_full.pdf

Van Baalen, M., Kuzmin, A., Nair, S. S., Ren, Y., Mahurin, E., Patel, C., Subramanian, S., Lee, S., Nagel, M., & Soriaga, J. (2023). FP8 versus INT8 for efficient deep learning inference. *arXiv preprint arXiv:2303.17951*.

Warden, P., & Situnayake, D. (2019). *Tinyml: Machine learning with tensorflow lite on arduino and ultra-low-power microcontrollers*. O'Reilly Media.

Waveshare. *RPi Camera OV5647*. https://www.waveshare.com/wiki/RPi_Camera_(F)

Whisson, D. A., & Salmon, T. P. (2009). Assessing the effectiveness of bait stations for controlling California ground squirrels. *Crop Protection*, *28*(8), 690-695. https://doi.org/10.1016/j.cropro.2009.04.002