PREPRINT

PERTANIKA JOURNALS

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Development of Machine Learning Wildlife Camera Using ESP32 CAM for Small Mammals

Leu Mei Xin and Nik Fadzly N Rosely*

School of Biological Sciences, Universiti Sains Malaysia, 11800, USM Penang, Malaysia

ABSTRACT

Camera traps serve a pivotal role in the surveillance of wildlife populations within a given habitat. Nonetheless, commercially available camera traps tend to be relatively expensive, power-consuming, non-selective in their detection capabilities, and necessitate subsequent processing of the acquired images. In this study, we focus on developing a simple machine-learning wildlife camera using ESP32 CAM, a microcontroller that has an integrated camera, flash, and SD card storage, for detecting small mammals. The ESP32 CAM was designed to recognize and photograph squirrels in natural environments, employing an object detection model formulated via an accessible online machine-learning resource known as Edge Impulse. The efficacy of the models was systematically assessed, with training images derived from two distinct repositories, namely, the Google Images Repository and original photographs captured at the designated field site, compared for analysis. Model A, which utilized images from Google, achieved an accuracy rate of 61.5%, whereas Model B, which relied on local field photographs, attained an accuracy of 83.3%. We compared ESP32 CAM to Hawkray Digital trail camera to assess the model's accuracy in detecting squirrels. Results from the Chi-Square analysis reveal that the number of squirrel pictures taken by both cameras was the same. A thorough discussion of various factors influencing the model's accuracy, including background uniformity, size of the object, posture, and distance, was also undertaken. Furthermore, this study addressed the implementation of TinyML within the ESP32 CAM context, as well as the inherent limitations associated with the ESP32 CAM technology.

Keywords: Camera trap, edge impulse, ESP32 CAM, machine learning, object detection model, small mammal detection, training image

ARTICLE INFO

Article history:
Received: 23 September 2024
Accepted: 03 March 2025
Published: 11 June 2025

DOI: https://doi.org/10.47836/pjst.33.4.05

E-mail addresses:
meixin8663@student.usm.my (Leu Mei Xin)
nfadzly@usm.my (Nik Fadzly N Rosely)
* Corresponding author

e-ISSN: 2231-8526

INTRODUCTION

The use of camera traps is a well-known method for monitoring wildlife (Swann & Perkins, 2014). Camera traps, or trail cameras, typically denote devices that activate upon detecting animal movement within their operational range. They are advantageous for estimating population densities and conducting demographic studies, as they facilitate prolonged monitoring of a designated area without much human intrusion (Sreedhar et al., 2021; Swann & Perkins, 2014). Camera traps possess the capability to acquire high-resolution images and videos, which is essential for species monitoring and behavioral research (Bird et al., 2021; Blount et al., 2021; Zhu et al., 2021). Additionally, camera traps can be strategically employed to alleviate human-wildlife conflicts, including issues such as wildlife encroachment, crop destruction, and livestock predation, by monitoring and detecting wildlife activity and movement in an area so that the affected community and authorities can be alerted and appropriate actions can be planned and taken.

ESP32 CAM is a unique microcontroller that has a built-in integrated camera, flash, and SD card storage built into the module. The price is relatively cheap, ranging from five to seven US Dollars (around MYR20 to MYR30). Further technical details and a datasheet can be found at Ai Thinker's website (https://docs.ai-thinker.com/en/esp32-cam). Due to its cheap price and adaptability in terms of programming, it is usually used for object identification (Rai & Rehman, 2019; Vinod et al., 2023). Other uses include smart agricultural practices (Elhattab et al., 2023), wildlife observation (Linder & Olsson, 2022), and industrial automation (Vinod et al., 2023), where visual data is primarily used to enhance operational efficacy.

Machine Learning (ML) represents a domain within Artificial Intelligence (AI) wherein computational systems can discern patterns from datasets via algorithms, thereby executing tasks autonomously with or without direct human oversight (Das et al., 2015). ML methodologies are progressively being integrated into conservation efforts (Fernandes et al., 2020; Shivaprakash et al., 2022; Tuia et al., 2022). For instance, computer vision algorithms can be employed to filter and process images obtained from wildlife camera traps to screen out blank images and classify the present animal species (Sreedhar et al., 2021). The advent of Tiny Machine Learning (TinyML) facilitates the deployment of machine learning within resource-limited devices, such as embedded systems and Internet-of-Things (IoT) devices, by optimizing algorithms to be as compact as possible to conform to specific hardware and software limitations (Disabato & Roveri, 2022; Ray, 2022).

Small mammals are animals with a tiny body size and are less than five kilograms (Baharudin et al., 2023). Typical representatives of this group include rodents, moles, shrews and bats. The populations of small mammals face significant threats due to climate change (Szpunar et al., 2008), deforestation, urbanization, and the expansion of agricultural plantations, which are the principal contributors to habitat degradation

(Palmeirim et al., 2020). Global research has demonstrated that habitat fragmentation resulting from urbanization and agricultural growth has precipitated a decrease in small mammal populations (Gomes et al., 2011; Johnstone et al., 2014; Palmeirim et al., 2020).

A web-based artificial intelligence platform called Edge Impulse (https://edgeimpulse.com) uses datasets or sensor data to create machine learning models for edge devices. It is convenient because Edge Impulse has a user-friendly interface that creates models that can be exported and used with a variety of widely used programming languages, including Arduino, TensorFlow Lite, and Mbed. This research aims to develop a simple device that is efficient for detecting the presence of small mammals using Edge Impulse. This is followed by 3 research questions:

1. Which training set is best for training machine learning algorithms?
2. Can TinyML be utilized in ESP32 CAM for the recognition of small mammals?
3. Will ESP32 CAM be at par with its commercial counterparts?

This study specifically focuses on object detection, aiming to detect the presence of small mammals in cameras without classifying them into specific species. ESP32 CAM is expected to be triggered and capture an image as long as an object is detected as a small mammal. The emphasis is on presence detection rather than species classification. We train the model with pictures of rats and squirrels due to the frequent presence of these two groups of small mammals in the study site. This uses TinyML to be deployed on ESP32 CAM. The project emphasizes the potential of ESP32 CAM for detection and photographic documentation of small mammals through the application of machine learning techniques. It deliberately avoids intricate machine learning programming, opting instead for a user-friendly click-and-drag interface requiring only a basic understanding of coding.

Another main factor in choosing ESP32 CAM is its low cost. We aim to design the camera to be less than 50 MYR and can be customized based on the user's demand. Our primary goal is to evaluate the feasibility of using small, low-power, and cost-effective embedded cameras with onboard machine learning for small mammal detection in remote field settings. While effective for wildlife monitoring, traditional trail cameras often require substantial power, produce large volumes of data that necessitate manual post-processing, and may not support real-time edge-based classification. In contrast, embedded solutions, such as the ESP32-CAM combined with TinyML, offer a lightweight alternative capable of performing real-time inference directly on the device, reducing the need for cloud processing and data storage.

We selected ESP32 CAM due to its affordability, low power consumption, and ability to capture images autonomously in field conditions. TinyML was chosen as it enables efficient deployment of machine learning models on resource-constrained hardware, ensuring that inference can be performed locally without constant connectivity. Edge Impulse was utilized as it provides an accessible and optimized pipeline for training and deploying machine

learning models onto embedded devices, streamlining the integration of AI-based image recognition with minimal computational overhead. Furthermore, its no-code approach is easy even for non-coders to train computer vision models.

The small mammal detection model was trained using Edge Impulse, utilizing two distinct training datasets from the Google Images Repository and local field photographs obtained via wildlife camera traps within the same study locale. The accuracy of the model was evaluated between these two training data sets within Edge Impulse. The model exhibiting superior accuracy was subsequently selected for deployment in the ESP32 CAM. We tested the ESP32 CAM in the field for its effectiveness. We then compared the results to the commercial wildlife camera trap.

MATERIALS AND METHODS

Study Site

This study took place at the Eco Hub in the Durian Valley of Universiti Sains Malaysia (USM) between January and March 2024, located at coordinates 5°21'41.7"N 100°18'18.0" E (indicated by the red dot in Figure 1). The USM campus on Penang Island, Malaysia, has been recognized as a "University in the Garden" after the construction of the Eco Hub in 2001. Eco Hub represents the university's initiative to preserve campus greenery and protect diverse flora and fauna. Photographs of small mammals, subsequently utilized for training the machine learning model, were obtained prior to the experiment through a camera trap positioned at the location marked in Figure 1. During the preliminary trials conducted at the study site, the camera system only detected squirrels and rats. As a result, the small mammal detection model in this study was specifically designed to focus on these two

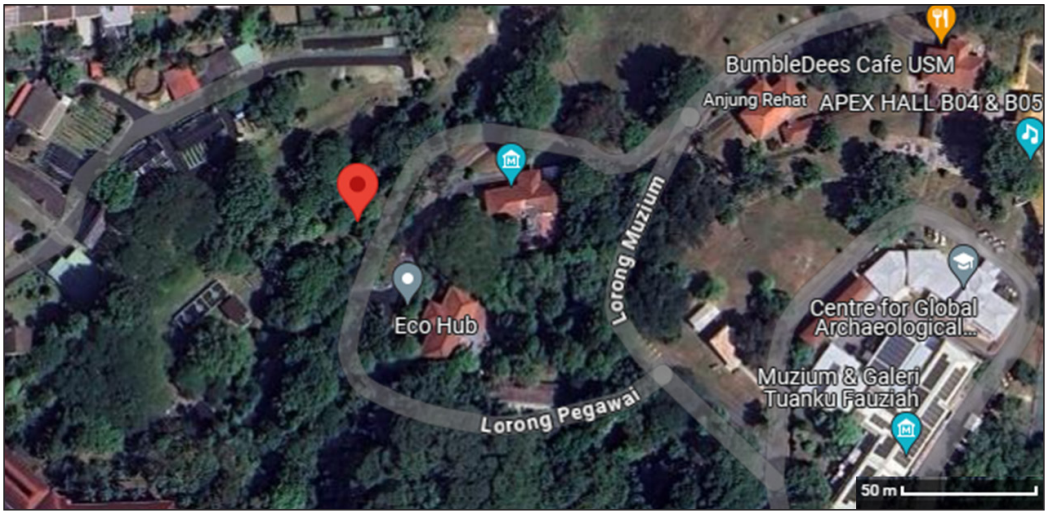

Figure 1. Location of experiment site (red marker) in Eco Hub USM (*Source:* Google Map)

categories of small mammals. This targeted approach ensures that the model is optimized for the most frequently observed small mammals within the study area. This same site was utilized for the deployment of both the ESP32 CAM and the camera trap throughout the experiment. Pieces of bananas were employed as bait to entice small mammals to the research area.

Hardware Components

The main hardware used in this research was the ESP32 CAM AI Thinker, as shown in Figure 2. The camera module is OV2640 (2-megapixel resolution). The built-in flash and SD card storage help reduce the additional components needed for the project. Although the board has WiFi, it was not used as the module was placed in the forest. General-purpose input/output (GPIO) pins enable us to add additional sensors if needed. There is also 4 MB of PseudoStatic Random Access Memory (PSRAM) for data streaming capabilities. This capability guarantees high image quality without leading to system instability in the ESP32. A basic blue housing produced via 3D printing by employing polylactic acid (PLA) filament was specifically engineered to secure and contain the ESP32 CAM (Figure 3).

The ESP32-CAM lacks an integrated USB interface. Consequently, an ESP32-CAM-MB programmer (Figure 4) is needed to successfully upload programmed code into the ESP32-CAM module.

Power was supplied to the ESP32 CAM through the soldering of pins from a four-slot 18650 lithium battery shield. This battery shield was interconnected with a UPS 4W solar panel (Figure 5) to enhance and prolong the operational longevity of the device when deployed in field conditions. This study used a 32GB micro-SD card to augment the storage capabilities of ESP32 CAM. With its ESP32 development board, the ESP32-CAM is compatible with programming via the Arduino Integrated Development Environment (IDE).

The Hawkray digital trail camera (Figure 6) was the camera trap employed for the preliminary acquisition of images of small mammals. We used the same camera trap again

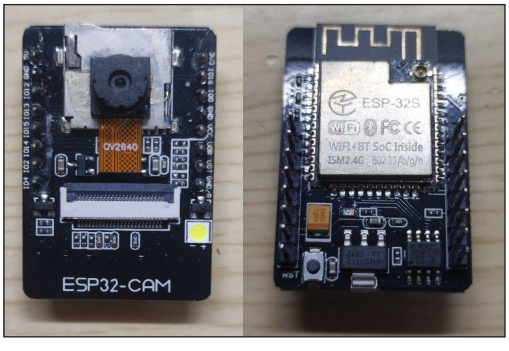

Figure 2. Front and rear view of ESP32 CAM

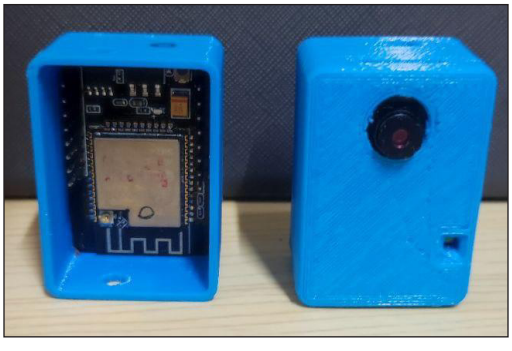

Figure 3. Front and rear view of a 3D printed casing for ESP32 CAM

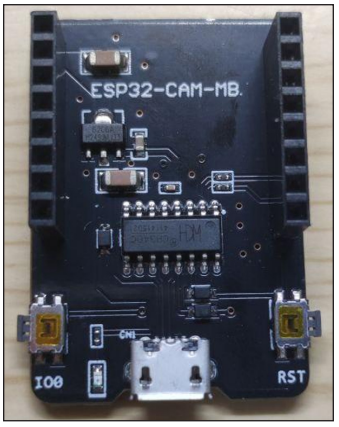

Figure 4. ESP32-CAM-MB programmer

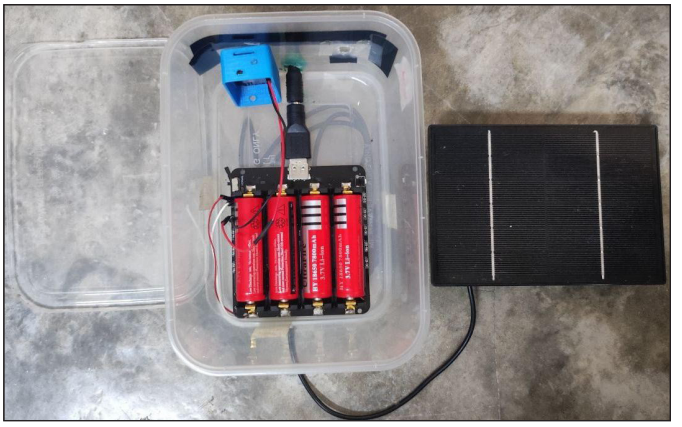

Figure 5. ESP32 CAM connected to a 18650 battery shield for a solar panel

in the field in conjunction with the ESP32 CAM AI Thinker. Four AA alkaline batteries supply the camera trap's operational power.

Figure 6. Front and rear view of the Hawkray digital trail camera

Both the cameras (Hawkray digital trail camera and the ESP32 CAM AI Thinker) function differently from each other. The Hawkray camera has a much higher-resolution image of 12 MP. It is also equipped with a Passive Infrared (PIR) sensor and InfraRed (IR) Light Emitting Diode (LED). When motion and heat are detected, the PIR sensor activates the camera. The image sensor captures the scene, using IR LEDs during low-light conditions, and saves the data to an SD card. The camera enters a power-saving mode when no motion or heat is detected, thus enabling it to function for a prolonged time. However, it is also prone to false positives, in which movements from the foliage or even leaves falling in front of the lens could trigger the camera. For our experiment, the camera was set to take 12 MP images when triggered. It was also programmed to capture the images in the first 5 minutes with a 10-second delay between each image.

The ESP32 CAM AI Thinker works by continuously watching the area. Since it is trained to recognize only specific objects (in our case, small mammals), it will ignore other movements or objects in its field of vision. Unfortunately, it is not equipped with any IR LEDs, thus rendering it useless during night or low-light situations. Therefore, it is programmed to be awake from 7 am until 8 pm. It will remain in deep sleep mode during the night. We programmed it to take pictures for 5 minutes with a 10-second delay between each image.

Machine Learning Model

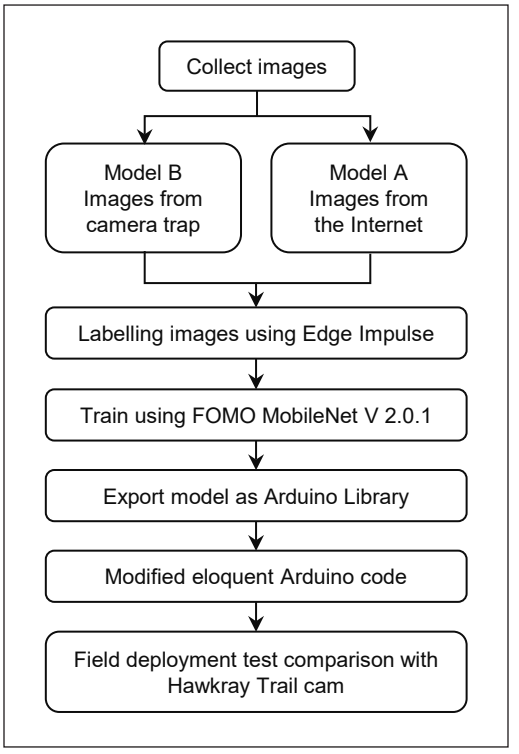

Figure 7. Flow chart of the entire project

Edge Impulse functioned as the training platform for this research initiative, enabling the formulation of the machine learning algorithm utilized by the ESP32 CAM for the recognition of small mammals. The training process was systematically divided into five distinct phases (Figure 7). Initially, images depicting the target subject, namely, small mammals, were collected. These images served as the training dataset within Edge Impulse, enabling the extraction of machine-recognizable features. For this study, the training images of small mammals were sourced from two primary channels: images retrieved from the Google Image Repository and local images obtained through a camera trap. In Model A, images of squirrels and rats were sourced from the Google Images Repository, culminating in the acquisition of 738 images of squirrels and 1,074 images of rats, which were subsequently uploaded to Edge Impulse. Examples of training images in Model A are shown in Figure 8. The imbalance ratio between images of squirrels and rats resulted from a manual selection process that prioritized images most relevant to local species and background settings. For Model B, local field images of squirrels and rats were obtained from a camera trap strategically positioned at the research site, EcoHub, to capture images of squirrels and rats. The final results of 244 images were documented, consisting of 201 images showcasing squirrels during daylight and 43 images depicting rats at night. Examples of training images in Model B are shown in Figure 9. Due to fewer images being captured, we created replicates for each image, producing 488 images to increase the dataset. Eight images were manually taken using the camera at the exact angle as a background dataset to help the model identify the background. These images were then uploaded to Edge Impulse. As this study focuses on small mammal detection rather than species classification, and squirrels and rats are within the targeted group, the difference in the number of images is not a primary concern.

The second phase involved annotating the images through the application of bounding boxes. Bounding boxes were precisely outlined around all observable squirrels and rats within the images, and corresponding labels were assigned. The designations employed

Figure 8. Labelled training image examples of the rat and squirrel in Model A

Figure 9. Labelled training image examples of the rat and squirrel in Model B

included ‘squirrel’ and ‘rat’. In instances where camera trap images contained no animals, no bounding boxes were labeled, and such images were systematically categorized as background images. In the Edge Impulse tutorial and DroneBot Workshop methodology (https://dronebotworkshop.com/esp32-cam-intro/), backgrounds are pictures without the object. We did not specify or rename it into ‘background’; rather, images that were not labeled (no objects within them) will be automatically considered ‘background’. Since we will be testing at the same place inside the forest, we chose our forest image (without any animals) as the background. We use the feature explorer in the Edge Impulse systems to describe the overall pattern of the dataset.

The third phase involves training the model to utilize the labelled images. The images were converted to grayscale to ensure that the model concentrated on the morphological features of small mammals, independent of color. The images were converted to grayscale to ensure that the model focused exclusively on the morphological features of small mammals, such as body shape, size, and structural details, independent of color variations that may not be relevant for classification. Morphological features are critical for distinguishing between small mammal species, as their physical structures often contain defining characteristics that are species-specific. This approach is particularly feasible because grayscale images reduce computational complexity and eliminate potential biases introduced by environmental lighting or fur coloration, which may vary within the same species. A good result here would be the model's ability to achieve high precision and recall in distinguishing between species based solely on morphology, demonstrating that the extracted grayscale features effectively capture the necessary species-specific patterns. Success would be indicated by accurate classification metrics, supporting the hypothesis that morphological features are sufficient for species differentiation in this context. Edge Impulse offers a pre-configured neural network capable of training the object detection model and producing an output. This process is also called Transfer Learning.

Transfer learning is a machine learning technique where a pre-trained model, initially trained on a large dataset, is adapted to a new but related task with a smaller dataset. Instead of training a model from scratch, transfer learning leverages the knowledge gained from a previously learned task, significantly reducing training time and improving performance, especially when labeled data is limited. The settings were left at default since we were using Edge Impulse's free account. Through trial and error, we found that increasing or modifying certain parameters tends to cause online training to fail. The number of training cycles/epoch were kept at 30. The learning rate was set at 0.01. The Training Processor is a CPU (GPU access is for paid accounts). The validation set data size is kept at 20%. This means that the images are sorted and selected randomly by Edge Impulse for training, testing and validation before training. The batch size is kept at 32, which is the default. Data augmentation is limited to grayscale only. We did not use the quantized model option, which will cause the model to crash during Arduino compilation later.

The available neural network architecture for the free account version of Edge Impulse is FOMO (Faster Object More Object) MobileNet V2.0.1. FOMO (Faster Objects, More Objects) MobileNet V2.0.1 represents an advanced object detection framework optimized for edge artificial intelligence applications within environments with limited resources. It builds upon the MobileNetV2 architecture, modified to facilitate rapid object detection with reduced computational demands.

In contrast to traditional object detection frameworks that rely on bounding boxes, FOMO employs a grid-based approach to object localization, rendering it exceptionally

efficient for applications such as counting, tracking, and real-time detection of minute objects. This streamlined model is particularly advantageous for implementation on microcontrollers and low-power edge devices, where conventional deep learning models may incur prohibitive computational costs. The dataset was partitioned into 75% for training and 25% for testing. In Edge Impulse, the platform automatically splits the dataset into training and validation sets during the training pipeline. The confusion matrix and performance metrics shown after training are derived from the validation set. This ensures the results reflect the model's capability to handle new data.

Edge Impulse employs several techniques to address overfitting during model training. One key method is data splitting, where the platform automatically divides the dataset into training, validation, and sometimes test sets. This ensures that the model is evaluated on unseen data, providing a more accurate measure of its generalization capabilities. Additionally, data augmentation can be enabled to introduce variability into the training data through methods such as noise addition, cropping, or flipping. This variability helps the model learn more robust patterns and reduces its reliance on specific data features. Lastly, Edge Impulse offers predefined machine learning architectures, such as convolutional neural networks, which are often optimized to balance model complexity with generalization, minimizing the likelihood of overfitting.

Based on the F1 score of accuracy (based on the validation dataset) that we derived from the confusion matrix, we selected the best model to be deployed to the ESP32 CAM. After the training, the fourth step is to upload the model that is compatible with the ESP32 CAM. Edge Impulse packages the model into a compact library that can be installed inside the Arduino IDE (Salerno, 2024). We adopted and modified codes from GitHub (https://github.com/eloquentarduino/EloquentEsp32cam). We further modify the FOMO example from the library to detect and save the images to the SD card. To reiterate a previous point, we cannot use an optimized quantized model due to the incompatibility with the Eloquent Arduino library.

The concluding phase involved executing the model on the ESP32 CAM to ascertain its operational efficacy. The code was transferred to the ESP32 CAM through a connection to a laptop utilizing the ESP32-CAM-MB programmer alongside the Arduino Integrated Development Environment (IDE). Subsequently, the camera was deployed in the field for empirical evaluation. The comprehensive code can be accessed at https://github.com/nroselnik/ESP32CAM-for-detecting-rats-and-squirrels.

Field Experiment

The experiment was conducted at the Eco Hub in Universiti Sains Malaysia. An ESP32 CAM and a Hawkray trail camera were deployed at the research location (Figure 1). Both devices were positioned at ground level with the lenses aimed towards a tree. Bananas

were distributed around the tree to serve as baits that attract small mammals. Given that the primary objective of this project is to assess the efficacy of the ESP32 CAM in detecting small mammals and capturing their images, the use of bait to lure the animals does not compromise the integrity of the study. The experiment was conducted over three distinct days to procure three replicates of the dataset. The cameras were operated continuously for 24 hours during each experimental session and were subsequently retrieved to transfer the images to a laptop. This procedure ensured that the cameras functioned effectively during deployment and helped conserve the memory capacity of the SD card, thereby preventing operational interruption due to full storage, which could impede the experiment's advancement. After the images were transferred to a laptop, we compared the ESP32 CAM and the Hawkray Digital Camera.

Data Analysis

In relation to the first research question, we compared the confusion matrix of Model A and Model B to find the best implementation model. This comparison assessed the difference in the accuracy of the developed models utilizing training images sourced from the Google Images Repository and from local field photographs captured by a camera trap at the identical location. Additionally, the analysis was done on the feature explorers shown by Edge Impulse for each model.

The captured images taken from both cameras were compared to address the second and third research questions. As ESP32 CAM does not come with an infrared (IR) sensor and is therefore incapable of detecting heat signatures, it depends solely on the machine learning model for image capture, only when the object of interest, a small mammal, enters the frame and is detected. As a result, the effectiveness of the ESP32-CAM in detecting small mammals is entirely dependent on the accuracy and performance of the machine learning-based detection algorithm. From the comparison with the Hawkray Trail Camera, which has an IR sensor, we can determine the effectiveness and performance of the detection model implemented in ESP32 CAM. The assessment relies on how many small mammals were accurately detected, as well as those incorrectly detected. Four classification metrics are involved in building a confusion matrix table:

- True Positive (TP): A small mammal appears, and an image is captured.
- False Positive (FP): A small mammal did not appear, yet an image was captured.
- True Negative (TN): Anything other than the small mammal appears, and no image is captured.
- False Negative (FN): A target small mammal appears, but the camera does not capture the image.

A chi-squared (Goodness of Fit) test was conducted to evaluate the differences between these cameras. The Significance level was set at 0.05. We calculated the Precision, Recall

and F1 values. The formulas for each metric are outlined as Equations 1, 2, and 3 (Chicco & Jurman, 2020).

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad [1]$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad [2]$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad [3]$$

RESULTS

Accuracies of Model A and Model B

Figures 10 and 11 show the Feature Explorer results for both models. In Model A, the features derived from labelled images of squirrels and rats exhibit a considerable degree of overlap. Conversely, for Model B, the features extracted from labelled squirrel images are distinctly separated from those of labelled rat images, exhibiting minimal overlap.

The Feature Explorer is automatically generated in Edge Impulse and represents a visualization of the extracted features from the training data. This is typically a pre-training step designed to give users an overview of how well the features separate the different classes in the feature space. At this stage, the model is not yet trained. The scatter plot is purely a visualization of the data in the feature space based on the extracted features (e.g., spectral, time-domain, or other custom features). The axes do not have predefined labels in this specific case because they are automatically generated based on dimensionality reduction techniques, such as principal component analysis (PCA) or t-distributed stochastic neighbor embedding (t-SNE). These techniques transform the high-dimensional feature space into two dimensions for visualization purposes, aiming to capture the maximum variance or class separability. As a result, the axes do not correspond directly to specific physical variables but instead represent combinations of features that best illustrate the class distributions in the dataset. It helps users assess whether the features are sufficiently discriminative for classification tasks before moving forward with model training.

The accuracy of the object detection model between two distinct image sources (the Google Images Repository and the camera trap) was compared. The confusion matrix in Edge Impulse offers valuable insights into the efficacy of the object detection model by comparing the predicted class and the actual class detected. Tables 1 and 2 illustrate the F1 score (based on the Validation dataset) and confusion matrix produced by Edge Impulse for Model A and Model B, respectively. The rows represent the predicted class, and the actual labels are the columns.

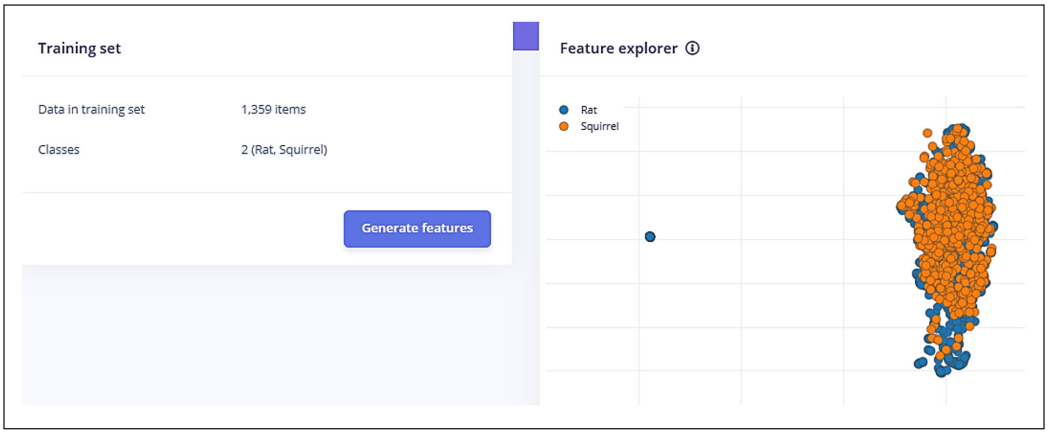

Figure 10. Training set and feature explorer of Model A in Edge Impulse

Figure 11. Training set and feature explorer of Model B in Edge Impulse

The matrix illustrates the number of instances that were inaccurately categorized. The F1 score serves as a metric to evaluate the efficacy of the predictive models. The F1 score attains its optimal value of 1, indicating perfect precision and recall, and its minimal value of 0.

The first table represents the performance of model A with an F1 score of 61.5%. The results indicate that the model performed exceptionally well in classifying the "Background" class, with 99.4% of the instances correctly identified and only 0.3% misclassified as either "rat" or "squirrel." This is further supported by the perfect F1 score of 1.00 for the "background" class, highlighting the model's reliability in detecting this class. For the "rat" class, the model correctly classified 58.7% of instances; however, a significant portion, 41.3%, was misclassified as "background," and no "rat" instances were

Table 1
F1 score and confusion matrix of Model A obtained from Edge Impulse

F1 Score 61.5%	Background	Rat	Squirrel
Background	99.4%	0.3%	0.3%
Rat	41.3%	58.7%	0%
Squirrel	18.4%	0%	81.6%
F1 Score	1.00	0.59	0.65

Table 2
F1 score and confusion matrix of Model B obtained from Edge Impulse

F1 Score 83.3%	Background	Rat	Squirrel
Background	100.0%	0%	0%
Rat	0%	100%	0%
Squirrel	31.4%	0%	68.6%
F1 Score	1.00	1.00	0.80

misclassified as "squirrel." This resulted in a lower F1 score of 0.59, reflecting challenges distinguishing the "rat" class, particularly due to confusion with "background." The "squirrel" class performed slightly better, with 81.6% of instances accurately identified, though 18.4% were also misclassified as "background." The F1 score for "squirrel" was 0.65, indicating moderate performance but room for improvement, especially in addressing the misclassification with "background."

The second table reflects the performance of Model B with an F1 score of 83.3%. In this iteration, the model achieved perfect classification for the "background" and "rat" classes, with 100% of instances correctly identified and without misclassifications. The F1 scores for both classes were 1.00, signifying excellent performance. For the "squirrel" class, the model correctly classified 68.6% of instances, an improvement over the first model. However, 31.4% of "squirrel" instances were still misclassified as "background," indicating that some challenges remain in distinguishing this class. Despite this, the F1 score for "squirrel" improved to 0.80, demonstrating a better balance between precision and recall compared to the first model.

Performance of ESP32 CAM

Hawkray Digital Trail Camera recorded 123 images, whereas the ESP32 CAM shows 117 images. Table 3 shows the number of images documented by each camera. Some of the images captured by both cameras are shown in Figures 12 and 13, with the small mammals circled in red.

The efficacy of the ESP32 CAM was systematically assessed. As the ESP32-CAM lacks night vision capabilities, it could not be evaluated for its effectiveness in detecting

nocturnal animals. Consequently, rats and bats were excluded from the results. Instead, the evaluation was exclusively conducted utilizing images depicting squirrels, as they were the only small mammals detected by the Hawkray camera and the ESP32-CAM during daylight. Table 4 shows the quantitative classification metrics. In this study, true positives refer to instances where the ESP32-CAM successfully captured images of squirrels that were also recorded by the Hawkray camera, which can be calculated as *29+51+18 = 98*. False positives occur when the ESP32-CAM captures images in the absence of squirrels,

Table 3
Number of images captured by Hawkray digital trail camera and ESP32 CAM in the 3 experiment days

Experiment Day	Hawkray digital trail camera	ESP32 CAM
1	30 images of squirrels 10 images of rats (night) 2 images of bats (night) 1 image of chicken	29 images of squirrels
2	54 images of squirrels	51 images of squirrels 15 blank images
3	24 images of squirrels 2 images of rats (night)	18 images of squirrels 4 blank images
Total	123 images	117 images

Figure 12. Images captured by ESP32 CAM showing the squirrel's appearance (red circle) in poor resolution (480 × 320 pixels)

Figure 13. Images captured by Hawkray trail camera showing the squirrel's appearance (red circle) in high resolution (1920 × 1080 pixels)

resulting in blank images. The total number of false positives is calculated as *15+4 = 19*. True negatives represent cases where non-small mammal species were present, and the ESP32-CAM was not triggered. This is evident from the Hawkray camera capturing an image of a chicken, while the ESP32-CAM did not, leading to 1 true negative instance. Bats and rats observed at night were excluded from this calculation, as the ESP32-CAM lacks night vision, rendering nighttime data invalid. False negatives occur when small mammals are present and captured by the Hawkray camera but are missed by the ESP32-CAM. This is determined by the difference between the number of squirrel images recorded by Hawkray and missed by the ESP32-CAM, calculated as *(30 - 29)+(54 - 51)+(24 - 18) = 10*. Based on the classification metrics in Table 4, the precision metric for the ESP32 CAM in the detection of squirrels is manually calculated as 0.84, while the calculated Recall metric is 0.91, and the F1 score is 0.87.

Figure 14 presents a comparative analysis of the ESP32 CAM and the Camera Trap regarding the quantity of squirrel images obtained. In general, the ESP32 CAM recorded a lower number of squirrel images than the Camera Trap. The difference in the quantity of captured squirrel images ranges from 1 to 6.

Table 4
Classification metrics of squirrel images captured by ESP32 CAM

	True Positive	**False Positive**	**True Negative**	**False Negative**
Number of squirrel images captured by ESP32 CAM	98	19	1	10

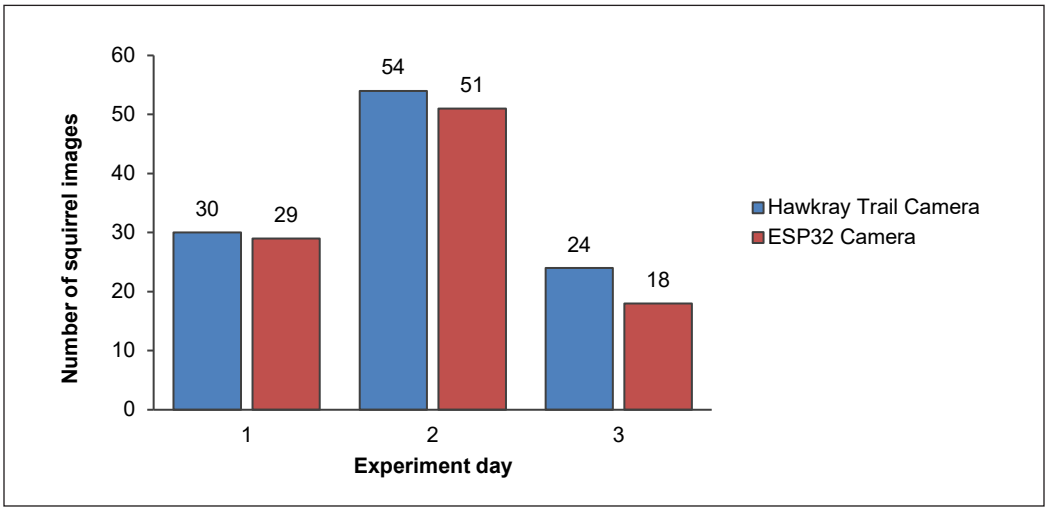

Figure 14. The number of squirrel images captured by the camera trap and ESP32 CAM

For the Chi-Square Goodness of Fit test, we used the number of images from the ESP32 CAM as the observed frequency and the number from the Hawkray Digital trail camera as the expected frequency. The results indicate no significant difference in the number of squirrel images obtained from the ESP32 CAM compared to those from the Hawkray trail camera, with X^2 (1, N = 98) = 0.853, p = 0.653.

DISCUSSION

Background Consistency and Object Size

From the findings, it is evident that, although both modeling frameworks attained an overall F1 score exceeding 50%, Model B, which utilized camera trap imagery, exhibited a superior F1 score and demonstrated heightened accuracy in the identification of small mammalian species in comparison to Model A, which relied on training images sourced from the Google repository. This discrepancy may be attributed to several variables, including the uniformity of the backgrounds in the images as well as the dimensions of the objects depicted therein. The images procured from the Google repository are characterized by diverse backgrounds, having been captured across a range of landscapes including arboreal canopies, forest undergrowth, and residential perimeters, employing various devices such as camera traps, mobile phone cameras, or professional-grade cameras, and in some instances, through artificial means utilizing artificial intelligence. This phenomenon is referred to as capture bias, as outlined by Tommasi et al. (2017). For example, the image background may occasionally be replete with dense vegetation or alternatively, be rendered plain or intentionally blurred to emphasize the animal's presence within the images (Figure 7). Such uncontrolled variables contribute to the inconsistency in the photographic backgrounds,

thereby introducing noise during the training process, which complicates the ability of machine learning algorithms to accurately discern the features of the target objects, namely squirrels and rats (Xiao et al., 2020).

A study conducted by Fadzly et al. (2021) posited that machine learning systems can misinterpret the image backgrounds as salient characteristics of the objects being analyzed. Conversely, the camera trap photographs captured at a singular field site generally exhibit a more consistent background (Figure 8) as the camera is oriented towards the same natural scenery, barring any human interventions or disturbances from wildlife. The homogeneity of the backgrounds in the camera trap photographs facilitates the machine learning model's capability to detect any anomalous activities or movements within the images, allowing it to visualize features that correspond with the morphological characteristics of squirrels and rats while disregarding any extraneous background components that are frequently present, such as foliage, vegetation, and fallen leaves (Nielsen et al., 2023; Wong & Fadzly, 2022).

The dimensions of objects and their angular representation in visual data are pivotal elements that affect the precision of the model (Feng & Lu, 2023; Sun et al., 2023). Figure 8 demonstrates the characteristics derived for squirrels and rats from online imagery in Model A, which exhibited significant overlap. This indicates that the Google Image Database model is inefficient in distinguishing squirrels and rats. The existence of overlapping characteristics may present difficulties in accurately differentiating between closely related species or in identifying individuals within intricate backgrounds.

The variability in the sizes of the squirrels and rats might be due to the differences in the focus areas. Some are too far, too close, or at a varied angle. This invariably confuses the model about the actual size of the trained/detected animals (Sun et al., 2023). This misinterpretation could fail to detect the object and the erroneous identification of non-relevant objects, compromising the model's accuracy in identifying the intended targets (Feng & Lu, 2023). In Model B, the features derived from the field photographs exhibit greater distinctiveness and separation (Figure 11). This might be due to the fixed angle, distance, and similar background. The uniformity in the size of the objects facilitates the model's capability to achieve a heightened level of precision in recognizing unique attributes of the small mammals, which is imperative for accurate species identification. Upon examination of both models, the images from the camera trap show that the object (rats/squirrels) is detected at nearly the same place each time. Squirrels are mostly in front of the tree trunk, and rats are mostly on the ground level. This consistency in the training datasets helps the model output compare between the object and background (Hao et al., 2022; Schneider et al., 2020; Xiao et al., 2020). Incorporating an additional step of collecting local background images from the study site is expected to improve model performance, since the similarity in the background has a profound effect on the model's effectiveness (Dujon & Schofield, 2019). This step would benefit the model rather than solely relying on

web-mined images, as it ensures better adaptation to the specific environmental conditions of the study area. It is crucial to consider these parameters in the formulation of training datasets prior to their application within a model, thereby facilitating the reduction of object detection inaccuracies and the optimization of its overall accuracy.

Application of TinyML in ESP32 CAM

The utilization of TinyML for the integration of object detection models within the ESP32 CAM framework is indeed feasible (Gotthard & Broström, 2023; Panda et al., 2022; Rahim & Zainal, 2024; Raiaan et al., 2023). These recent studies highlight the effectiveness of using ESP32 CAM for wildlife monitoring such as monkeys, dogs, rhinoceros and many other large endangered species. An analysis of the outcomes reveals that the camera successfully adhered to the algorithmic instructions by capturing images exclusively when small mammals were detected, while disregarding extraneous stimuli from alternative animal species. Results have shown no statistically significant differences in the frequency of squirrel detections for the two devices under consideration, namely the Hawkray trail camera and the ESP32 CAM. Nevertheless, the constraints imposed by the restricted quantity of training datasets may contribute to potential inaccuracies in the model's performance (Schneider et al., 2020) when implemented on the ESP32 CAM. The occurrence of blank images captured by the device could potentially be attributed to the misidentification of fallen leaves as small mammals. Despite its inability to detect certain instances of squirrels in comparison to the Hawkray trail camera, the findings substantiate the feasibility of employing TinyML within ESP32 CAM.

This is evidenced by the higher number of true positives compared to false positives and false negatives. The majority of research (Bagchi et al., 2022; Das & Halder, 2024; Kadhim et al., 2023) has predominantly investigated the application of ESP32 CAM for facial recognition systems, with a limited number of studies exploring its use in wildlife detection systems (Linder & Olsson, 2022). This disparity may stem from the wildlife detection model's exigency for a substantial and heterogeneous training dataset to attain a high level of accuracy in the automatic identification of various animal classes. Such extensive training requirements are likely impractical for most ecological research (Schneider et al., 2020). The more animal class is introduced in the algorithm, the more taxing it is for the system to perform. However, this investigation concentrated solely on a single group of animals, specifically small mammals, and was limited to only squirrels and rats. This targeted approach mitigates the necessity for a large training dataset. It facilitates the model's ability to learn from a more modest dataset, thereby achieving satisfactory accuracy in detecting squirrels. Given that TinyML is specifically engineered for operation under low power constraints, it is congruent with the ESP32 CAM, which is characterized by low power consumption and limited processing capabilities.

Furthermore, implementing TinyML in the ESP32 CAM demonstrated the usability of local on-device processing, minimizing processing costs (Schizas et al., 2022). A majority of online platforms, including AIDE, MegaDetector, and Wildlife Insights (Ahumada et al., 2020; Beery et al., 2019; Kellenberger et al., 2020), require the images to be uploaded before comparing the blank images and identifying the animals depicted in them. Through the mechanism of local processing, the camera is activated solely upon the presence of the selected species. Therefore, there would be no need to transfer and upload images for model learning first. This advancement holds significant potential for the prolonged deployment of cameras in remote regions that pose logistical challenges for accessibility. Future studies focused on population dynamics and the monitoring of animal species may greatly benefit from applying TinyML in compact devices such as the ESP32 CAM.

Limitations of ESP32 CAM

During the investigation, several limitations inherent to the ESP32 CAM were identified. In contrast to commercially available camera traps, the ESP32 CAM lacks an infrared sensor. An infrared sensor is a crucial component typically integrated into camera traps, utilized to detect IR or thermal signals from objects (Trolliet et al., 2014). These IR sensors help the camera to continue working even at night. ESP32 CAM lacks an IR sensor and relies only on environmental lighting (although this could be attached separately via the GPIO pins). This constraint renders the device impractical during nocturnal hours, as it is incapable of detecting any movement in complete darkness. Consequently, this represents the primary rationale for the inability to conduct rat detection using the ESP32 CAM in this study, given that the module lacks infrared technology or night vision capabilities, and considering that rats are predominantly nocturnal foragers. Researchers engaged in monitoring nocturnal fauna are advised to utilize an ESP32 CAM module equipped with an integrated infrared sensor.

Moreover, the image quality or resolution of the ESP32 CAM is suboptimal compared to that of the Hawkray trail camera. The ESP32 CAM is characterized by a 480 x 320 pixels resolution, whereas the Hawkray trail camera boasts a 1920 x 1080 pixels resolution. Pixels constitute the fundamental units that denote a singular point of color or grayscale, collectively forming a digital image. A higher pixel count correlates with enhanced detail within an image, thereby yielding superior image quality (Wong & Fadzly, 2022). The low resolution of the images from the ESP32 CAM results in a lack of clarity and detail, producing images that are often indistinct (Figure 12).

In contrast, the Hawkray trail camera, with its elevated resolution, generates notably sharper images, thereby offering a more detailed representation of squirrels (Figure 13). The ESP32 CAM is equipped with a diminutive and rudimentary camera lens, the OV2640, which possesses a restricted field of view in comparison to the high-performance lenses

employed by commercial camera traps. This rudimentary lens significantly constrains the capability of the ESP32 CAM to deliver images of satisfactory quality. While the ESP32 CAM may serve effectively for the singular purpose of wildlife detection, alerting the owner to the presence of wildlife, it is not an appropriate instrument for studies focused on characterizing or recognizing individual wildlife due to its inherent limitations in image resolution.

Another constraint of the ESP32 CAM pertains to its detection range. The fundamental camera integrated within the ESP32 CAM exhibits a restricted object detection distance of merely 4 meters, as highlighted in the research conducted by Budiharto et al. (2022). This signifies that only fauna that come within 4 meters of the camera will be quantified and detected by the ESP32 CAM. Fauna situated beyond this specified distance will remain undetected by the camera, culminating in a lost opportunity to ascertain the species and document the characteristics of the animal. Furthermore, the restricted detection range results in a limited coverage area for the camera. This limitation may present substantial challenges if the research site encompasses a large spatial area that a singular ESP32 CAM is incapable of adequately monitoring. The device may also necessitate strategic placement and orientation near locations where animals are likely to approach closely to enhance its detection efficacy and mitigate false negatives (the failure to detect wildlife).

Although this study highlights some of the weaknesses of ESP32 CAM, we have to point out that the original idea of detecting specific target animals using the module is feasible. Furthermore, ESP32 CAM is an open-source design, and additional electronics can be added through its GPIO pins. Sensors such as temperature, humidity, and a real-time clock (RTC) can be added to the module. The OV2640 camera that comes standard with the module can also be replaced with the OV5640, which is 5 Megapixels in quality. An IR LED can be soldered (replacing the normal LED on the board) to help in low-light conditions. However, this will invariably cause an increase in the budget for developing the camera as we aim for a device costing less than MYR50 (excluding the batteries). Our team works on several other microcontroller boards, such as the SeeedStudio Xiao series. Further development concerning these new microcontrollers is still in progress.

CONCLUSION

Although the ESP32 CAM demonstrates efficacy in identifying small mammals through an algorithm developed by Edge Impulse, it has constraints, such as a limited field of view, low-resolution images, and an absence of infrared capabilities necessary for nocturnal operations. For the training of the detection model, employing field photographs taken from the identical location is likely to yield superior accuracy in comparison to images sourced through web mining (e.g., Google image repositories). Despite successfully demonstrating the ESP32 CAM's potential as a machine-learning wildlife monitoring device, we have

some recommendations to enhance the study. Firstly, the volume of the training dataset could be augmented by acquiring a greater number of field photographs in advance. An expanded dataset could significantly enhance the model's efficacy (Shahinfar et al., 2020). Secondly, more focus could be directed at the different Learner AI models, such as YOLO v9 or FOMO. Enhancing the model's accuracy and performance may require adopting an advanced machine learning platform that offers superior processing capabilities. Thirdly, integrating infrared technology into the ESP32 CAM would enable its functionality in low-light conditions. Future research should optimize the ESP32 CAM as an AI-assisted Wildlife camera to detect additional wildlife species, thereby contributing to conservation and research initiatives.

ACKNOWLEDGEMENTS

This project is supported by R503-KR-FRG001-0006712190-K134, FRGS Malaysia Grant. We thank the EcoHub and the Biological Sciences School for their support and help.

REFERENCES

Ahumada, J. A., Fegraus, E., Birch, T., Flores, N., Kays, R., O'Brien, T. G., Palmer, J., Schuttler, S., Zhao, J. Y., Jetz, W., Kinnaird, M., Kulkarni, S., Lyet, A., Thau, D., Duong, M., Oliver, R., & Dancer, A. (2020). Wildlife insights: A platform to maximize the potential of camera trap and other passive sensor wildlife data for the planet. *Environmental Conservation, 47*(1), 1–6. https://doi.org/10.1017/S0376892919000298

Bagchi, T., Mahapatra, A., Yadav, D., Mishra, D., Pandey, A., Chandrasekhar, P., & Kumar, A. (2022). Intelligent security system based on face recognition and IoT. *Materials Today: Proceedings*, *62*(4), 2133-2137. https://doi.org/10.1016/j.matpr.2022.03.353

Baharudin, N. S., Tah, M. M. T. M., Zulkifli, S. Z., Ghani, N. I. A., Noor, H. M., & Sabar Sabal, N. H. (2023). Species diversity and distribution of non-volant small mammal between restoration, boundary, disturbed and undisturbed area in Cameron Highlands, Malaysia. *Tropical Life Sciences Research*, *34*(1), 151–183. https://doi.org/10.21315/tlsr2023.34.1.10

Beery, S., Morris, D., & Yang, S. (2019). Efficient pipeline for camera trap image review. *arXiv preprint arXiv.1907.06772*. https://doi.org/10.48550/arXiv.1907.06772

Bird, J. P., Fuller, R. A., Pascoe, P. P., & Shaw, J. D. S. (2022). Trialling camera traps to determine occupancy and breeding in burrowing seabirds. *Remote Sensing in Ecology and Conservation, 8(*2), 180-190. https://doi.org/10.1002/rse2.235

Blount, J. D., Chynoweth, M. W., Green, A. M., & Sekercioglu, C. H. (2021). Review: COVID-19 highlights the importance of camera traps for wildlife conservation research and management. *Biological Conservation, 256*, Article 108984. https://doi.org/10.1016/j.biocon.2021.108984

Budiharto, W., Irwansyah, E., Suroso, J. S., & Gunawan, A. A. S. (2022). Low-cost vision-based face recognition using Esp32-Cam for tracked robot. *ICIC Express Letters*, *13*(3), 321–327. https://doi.org/10.24507/icicelb.13.03.321

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics 21*, Article 6. https://doi.org/10.1186/s12864-019-6413-7

Das, B., & Halder, K. K. (2024). Face recognition using ESP32-Cam for real-time tracking and monitoring. In *2024 International Conference on Advances in Computing, Communication, Electrical, and Smart Systems (iCACCESS)* (pp. 01-06). IEEE Publishing. https://doi.org/10.1109/iCACCESS61735.2024.10499606

Das, S., Dey, A., Pal, A., & Roy, N. (2015). Applications of artificial intelligence in machine learning: Review and prospect. *International Journal of Computer Applications*, *115*(9), 31-41. https://doi.org/10.5120/20182-2402

Disabato, S., & Roveri, M. (2022). Tiny machine learning for concept drift. *IEEE Transactions on Neural Networks and Learning Systems, 35*(6), 8470-8481. https://doi.org/10.1109/TNNLS.2022.3229897

Dujon, A. M., & Schofield, G. (2019). Importance of machine learning for enhancing ecological studies using information-rich imagery. *Endangered Species Research, 39*, 91-104. https://doi.org/10.3354/esr00958

Elhattab, K., Abouelmehdi, K., & Elatar, S. (2023). New model to monitor plant growth remotely using ESP32-CAM and mobile application. In *2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM)* (pp. 1-6). IEEE Publishing. https://doi.org/10.1109/WINCOM59760.2023.10322939

Fadzly, N., Zuharah, W. F., & Wong, J. J. N. (2021). Can plants fool artificial intelligence? Using machine learning to compare between bee orchids and bees. *Plant Signaling & Behavior, 16*(10), Article 1935605. https://doi.org/10.1080/15592324.2021.1935605

Feng, S., & Lu, J. (2023). Research on image recognition based on neural network model learning algorithm. In *2023 4th International Conference on Computer Vision, Image and Deep Learning (CVIDL)* (pp. 21-24). IEEE Publishing. https://doi.org/10.1109/CVIDL58838.2023.10164845

Fernandes, A. C. M., Gonzalez, R. Q., Lenihan-Clarke, M. A., Trotter, E. F. L., & Arsanjani, J. J. (2020). Machine learning for conservation planning in a changing climate. *Sustainability*, *12*(18), Article 7657. https://doi.org/10.3390/su12187657

Gomes, V., Ribeiro, R., & Carretero, M. A. (2011). Effects of urban habitat fragmentation on common small mammals: Species versus communities. *Biodiversity and Conservation*, *20*, 3577–3590. https://doi.org/10.1007/s10531-011-0149-2

Gotthard, R., & Broström, M. (2023). *Edge machine learning for wildlife conservation – A part of the Ngulia Project* [Master thesis, Linköping University]. Linköping University Electronic Press. https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1788498&dswid=7533

Hao, Y., Pei, H., Lyu, Y., Yuan, Z., Rizzo, J., Wang, Y., & Fang, Y. (2022). Understanding the impact of image quality and distance of objects to object detection performance. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 11436-11442). IEEE Publishing. https://doi.org/10.48550/arXiv.2209.08237

Johnstone, C. P., Lill, A., & Reina, R. D. (2014). Habitat loss, fragmentation and degradation effects on small mammals: Analysis with conditional inference tree statistical modelling. *Biological Conservation, 176*, 80-98. https://doi.org/10.1016/j.biocon.2014.04.025

Kadhim, T. A., Hariri, W., Zghal, N. S., & Aissa, D. B. (2023). A face recognition application for Alzheimer's patients using ESP32-CAM and Raspberry Pi. *Journal of Real-Time Image Processing, 20*, Article 100. https://doi.org/10.1007/s11554-023-01357-w

Kellenberger, B., Tuia, D., & Morris, D. (2020). AIDE: Accelerating image-based ecological surveys with interactive machine learning. *Methods in Ecology and Evolution, 11*(12), 1716–1727. https://doi.org/10.1111/2041-210X.13489

Linder, J., & Olsson, O. (2022). *A smart surveillance system using edge-devices for wildlife preservation in animal sanctuaries* [Master dissertation, Linköping University]. Linköping University Diva-Portal. https://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-186261

Nielsen, I. E., Grundeland, E., Snedeker, J., Rasool, G., & Ramachandran, R. P. (2023). Targeted background removal creates interpretable feature visualizations. *arXiv preprint arXiv.2306.13178*. https://doi.org/10.48550/arXiv.2306.13178

Palmeirim, A. F., Santos-Filho, M., & Peres, C. A. (2020). Marked decline in forest dependent small mammals following habitat loss and fragmentation in an Amazonian deforestation frontier. *PLOS One, 15*(3), Article e0230209. https://doi.org/10.1371/journal.pone.0230209

Panda, P. K., Kumar, C. S., Vivek, B. S., Balachandra, M., & Dargar, S. K. (2022). Implementation of a wild animal intrusion detection model based on internet of things. In *2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)* (pp. 1256-1261). IEEE Publishing. https://doi.org/10.1109/ICAIS53314.2022.9742948

Rahim, A. M. A., & Zainal, M. S. (2024). An IoT based monkey deterrent for plantations with surveillance system: A method for damage control done by monkeys in plantations. *Progress in Engineering Application and Technology*, *5*(1), 114-121.

Rai, P., & Rehman, M. (2019). ESP32 based smart surveillance system. In *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)* (pp. 1-3). IEEE Publishing. https://doi.org/10.1109/ICOMET.2019.8673463

Raiaan, M. A. K., Fahad, N. M., Chowdhury, S., Sutradhar, D., Mihad, S. S., & Islam, M. M. (2023). IoT-based object-detection system to safeguard endangered animals and bolster agricultural farm security. *Future Internet*, *15*(12), Article 372. https://doi.org/10.3390/fi15120372

Ray, P. P. (2022). A review on TinyML: State-of-the-art and prospects. *Journal of King Saud University-Computer and Information Sciences*, *34*(4), 1595-1623. https://doi.org/10.1016/j.jksuci.2021.11.019

Salerno, S. (2024). *EloquentEsp32cam* (Version 2.6.0). https://www.arduino.cc/reference/en/libraries/eloquentesp32cam/

Schizas, N., Karras, A., Karras, C., & Sioutas, S. (2022) TinyML for ultra-low power AI and large scale IoT deployments: A systematic review. *Future Internet*, *14*(12), Article 363. https://doi.org/10.3390/fi14120363

Schneider, S., Greenberg, S., Taylor, G. W., & Kremer, S. C. (2020). Three critical factors affecting automated image species recognition performance for camera traps. *Ecology and Evolution, 10*(7), 3503–3517. https://doi.org/10.1002/ece3.6147

Shahinfar, S., Meek, P. D., & Falzon, G. (2020). "How many images do I need?" Understanding how sample size per class affects deep learning model performance metrics for balanced designs in autonomous wildlife monitoring. *Ecological Informatics, 57*, Article 101085. https://doi.org/10.1016/j.ecoinf.2020.101085

Shivaprakash, K. N., Swami, N., Mysorekar, S., Arora, R., Gangadharan, A., Vohra, K., Jadeyegowda, M., & Kiesecker, J. M. (2020). Potential for artificial intelligence (AI) and machine learning (ML) applications in biodiversity conservation, managing forests, and related services in India. *Sustainability*, *14*(12), Article 7154. https://doi.org/10.3390/su14127154

Sreedhar, S., Sandesh, S., Prarthana, P., Karthik, P. N., Prabhanjan, S., & Srinidhi, K. (2021). A literature survey on wildlife camera trap image processing using machine learning techniques. *Perspectives in Communication, Embedded-systems and Signal-processing – Pices*, *5*(2), 11–14. https://doi.org/10.5281/zenodo.4902949

Sun, G., Wang, S., & Xie, J. (2023). An image object detection model based on mixed attention mechanism optimized YOLOv5. *Electronics, 12*(7), Article 1515. https://doi.org/10.3390/electronics12071515

Swann, D., & Perkins, N. (2014). Camera trapping for animal monitoring and management: A review of applications. In P. Meek & P. Fleming (Eds.), *Camera Trapping: Wildlife Management and Research* (pp. 3-12). CSIRO PUBLISHING. https://doi.org/10.1071/9781486300402

Szpunar, G., Aloise, G., Mazzotti, S., Nieder, L., & Cristaldi, M. (2008). Effects of global climate change on terrestrial small mammal communities in Italy. *Fresenius Environmental Bulletin*, *17*(9b), 1526-1533.

Tommasi, T., Patricia, N., Caputo, B., & Tuytelaars, T. (2017). A deeper look at dataset bias. In G. Csurka (Ed.), *Domain Adaptation in Computer Vision Applications* (pp. 37-55). Springer. https://doi.org/10.48550/arXiv.1505.01257

Trolliet, F., Vermeulen, C., Huynen, M. C., & Hambuckers, A. (2014). Use of camera traps for wildlife studies: A review. *Biotechnologie, Agronomie, Société et Environnement, 18*(3), 446-454.

Tuia, D., Kellenberger, B., Beery, S., Costelloe, B. R., Zuffi, S., Rissie, B., Mathis, A., Mathis, M. W., van Langevelde, F., Burghardt, T., Kays, R., Klinck, H., Wikelski, M., Couzin, I. D., van Horn, G., Crofoot, M. C., Stewart, C. V., & Berger-Wolf, T. (2022). Perspectives in machine learning for wildlife conservation. *Nature Communications*, *13*, Article 792. https://doi.org/10.1038/s41467-022-27980-y

Vinod, S., Shakor, P., Sartipi, F., & Karakouzian, M. (2023). Object detection using ESP32 CAMeras for quality control of steel components in manufacturing structures. *Arabian Journal for Science and Engineering*, *48*, 12741–12758. https://doi.org/10.1007/s13369-022-07562-2

Wong, J. J. N., & Fadzly, N. (2022). Development of species recognition models using Google teachable machine on shorebirds and waterbirds. *Journal of Taibah University for Science, 16*(1), 1096–1111. https://doi.org/10.1080/16583655.2022.2143627

Xiao, K., Engstrom, L., Ilyas, A., & Madry, A. (2020). Noise or signal: The role of image backgrounds in object recognition. *arXiv preprint arXiv:2006.09994*. https://doi.org/10.48550/arXiv.2006.09994

Zhu, C., Li, W., Gregory, T., Wang, D., Ren, P., Zeng, D., Kang, Y., Ding, P., & Si, X. (2022). Arboreal camera trapping: a reliable tool to monitor plant-frugivore interactions in the trees on large scales. *Remote Sensing in Ecology and Conservation, 8*(1), 92-104. https://doi.org/10.1002/rse2.232