# tinyML-03: Efficient and High-performance Solution for Wildlife Monitoring


Sudharshan N
sudharshann106@gmail.com
Chennai, Tamil Nadu, India

Saran Sundar S
saransrimathi590@gmail.com
Chennai, Tamil Nadu, India

Rohit S
rohitsarav14@gmail.com
Chennai, Tamil Nadu, India

CH Leela Sri Chowdary
leelasric@gmail.com
Chennai, Tamil Nadu, India

Jayaprakash M
jpkash2001@gmail.com
Chennai, Tamil Nadu, India



## ABSTRACT

This paper presents the development of an efficient, low-cost, and low-power TinyML solution for Wild Life Monitoring. The objective is to reliably identify and classify various animals using a model optimized for deployment on resource-constrained devices. The paper concludes with insights into potential improvements with enterprise accounts and extended training, and the GitHub repository is provided for further exploration and implementation. The proposed solution opens avenues for scalable and high-performance TinyML applications in Wild life monitoring.



## KEYWORDS

Wild Life monitoring, FOMO (Faster Objects, More Objects), TinyML, ESP32-S3, CNN, EdgeAI, Machine Learning


## 1 INTRODUCTION

Building upon the primary objective of developing efficient, low-cost, and low-power solutions for wildlife monitoring, the research aims to pioneer a paradigm shift in the intersection of technology and biodiversity conservation. Recognizing the urgency of preserving and understanding diverse wildlife species, our focus extends beyond mere identification to encompass reliable classification. This commitment to classification is crucial for gaining insights into the ecological dynamics of different species and implementing targeted conservation strategies. The emphasis on scalability is a forward-looking aspect of our research, acknowledging the vast diversity of environments and wildlife species. By designing solutions that can adapt to various ecosystems and accommodate a multitude of species, we lay the groundwork for a versatile tool that transcends geographical and ecological constraints. Scalability is not just a technical consideration but a fundamental principle that ensures the relevance and applicability of our solutions in dynamic and varied landscapes where wildlife conservation is paramount.

In essence, our research aspires to be a catalyst for a new era in wildlife monitoring, one where technology converges with conservation goals seamlessly. Through the lens of efficiency, cost-effectiveness, and scalability, we aim to empower a global community of stakeholders to actively contribute to the preservation of our planet's rich biodiversity.

## 2 METHODOLOGY

The methodology encompasses defining objectives for an efficient and cost-effective wildlife monitoring system, selecting hardware (XIAO ESP32S3 Sense), and acquiring a diverse dataset. A custom TinyML model is trained and optimized on Edge Impulse for hardware constraints and accuracy. The model is quantized and integrated into the hardware prototype, including components like the OV2640 camera and Lithium-Ion battery. Performance evaluation involves testing for accuracy, latency, and resource utilization, with overall project cost transparency. Documentation and a GitHub repository facilitate sharing the streamlined process and outcomes, emphasizing reliability, efficiency, and cost-effectiveness in the development of a scalable TinyML solution for Wild Life Monitoring.

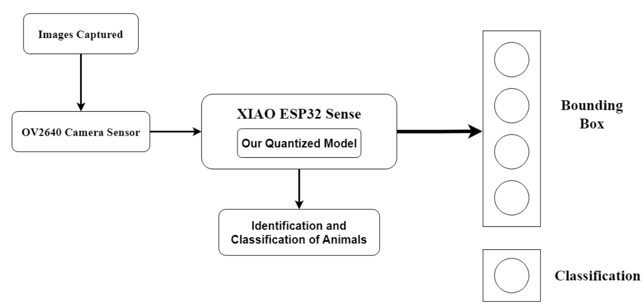


**Figure 1: Flow chart representing the entire process**

### 2.1 Experimental Setup

List of components used

- **Microcontroller Unit (MCU):** `XIAO ESP32-S3 Sense`with the ESP32-S3R8 32-bit Processor, 2.4GHz WiFi, low-power Bluetooth, Xtensa processor chip (240 MHz), 8MB PSRAM, and 8MB FLASH.
- **Camera Sensor:** Detachable OV2640 camera sensor for capturing images at flexible resolutions.
- **Power Source:** M-STAR 3.7v -18650 -4300 mAh Lithium-Ion Battery for providing power to the system.
- **3D Printed Case for XIAO-ESP32S3-Sense:** The 3D printed case for the XIAO ESP32-S3 Sense serves as a protective and functional housing, enhancing the overall robustness and aesthetic appeal of the hardware setup in our wild life monitoring system.

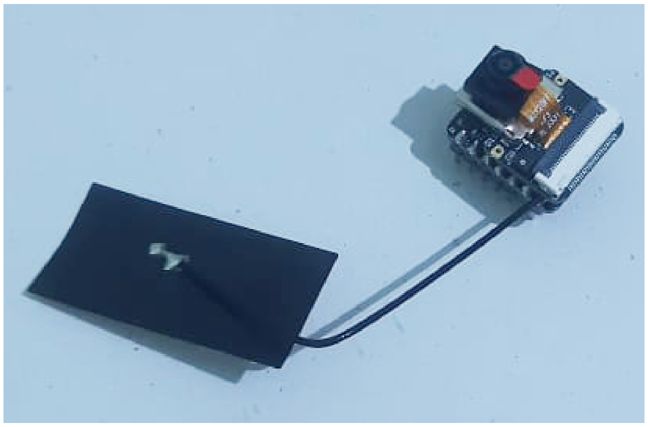

Figure 2: ESP32 XIAO Sense board S3

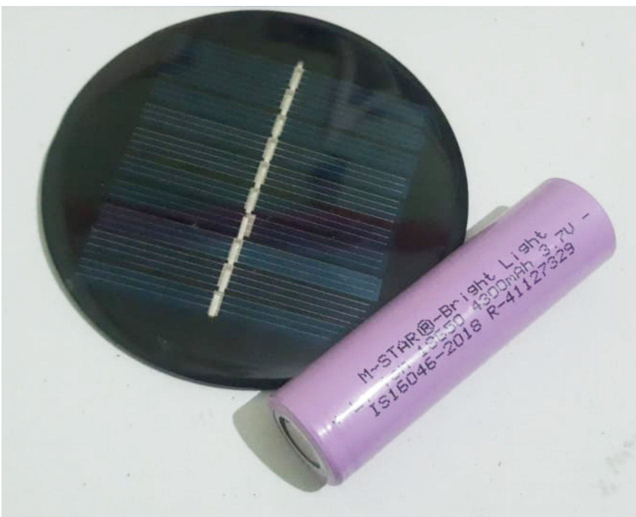


Figure 3: M-STAR Battery with Solar Panel

Table 1: Cost Estimation

| Components | Quantity | Price (INR) |
|---|---|---|
| XIAO ESP32-S3 Sense | 1 | 1499 |
| BATTERY-Li-ion 18650 4300mah | 2 | 155 each |
| 3D Printed Case | 1 | 20 |

## 3 COST ESTIMATIONS

The below table consists of the detailed estimated budget for the project

## 4 DATASET

We've utilized the dataset given in the data source section of the competition https://cthulhu.dyn.wildme.io/public/datasets/wild.tar.gzTo address imbalances inherent in the dataset, we strategically removed heavily undersampled classes, ensuring a more equitable

Figure 4: 3D model for casing

representation of various wild animals. This preprocessing step is essential for preventing model bias towards more frequently occurring classes and improving the overall robustness of the machine learning model. To further enhance the model's generalization and mitigate overfitting, we leveraged advanced techniques available in Edge Impulse. The platform's Data Augmentation feature was employed to introduce variations into the training data, thereby exposing the model to a more diverse range of scenarios. Additionally, class re-weighting was applied to account for the imbalances in the remaining classes, a crucial step to prevent the model from favoring the majority class during training. In contrast, the initial model underwent training using both testing and training data, allowing us to explore the model's potential in different scenarios. This strategic approach to dataset utilization and preprocessing underscores our commitment to developing a robust and unbiased wildlife detection model. The integration of cutting-edge techniques from Edge Impulse enriches the dataset preprocessing pipeline, contributing to the model's accuracy and reliability in identifying and classifying wild animals.

## 5 ML MODEL

Our chosen model, the MobileNetV2-based FOMO (Faster Objects, More Objects) from Edge Impulse, is crafted for image detection, effectively distinguishing between background and objects of interest in a grid-based manner. This design promotes efficiency, with the model intentionally kept under 100KB in size, making it suitable for resource-constrained devices. Operating with less than 200K of RAM further ensures compatibility with devices where memory is limited. The decision to train and deploy the entire model on Edge Impulse stems from challenges encountered when attempting to deploy a custom model on the Edge device. Despite these challenges, the initial model showcased commendable accuracy and compactness, aligning with our goal of creating an effective and efficient solution for wildlife monitoring. Detailed model specifications and results can be explored in the Links section of our provided GitHub link.

The Whole model was trained on Edge Impulse as we had trouble deploying our custom model in the Edge device. The First model we trained was very good in terms of accuracy and size(We'll attach the .ipynb file, a quantized model with the results in a miscellaneous section of the GitHub link we'll provide).

**Table 2: Unoptimized float32 model**

| Properties | Image | Classifier | Total |
|---|---|---|---|
| Latency | 15 ms | 4647 ms | 4662 ms |
| Ram | 4.0 K | 887.1 K | 887.1 K |
| Flash | - | 104.0 K | - |

**Table 3: Quantised Int8 Model**

| Properties | Image | Classifier | Total |
|---|---|---|---|
| Latency | 15 ms | 1499 ms | 1514 ms |
| Ram | 4.0 K | 239.5 K | 239.5 K |
| Flash | - | 78.7 K | - |

## 6 ASSEMBLY

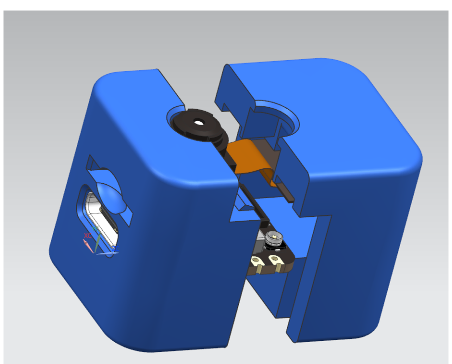

**Figure 5: 3D Model of casing, ESP32, Camera assembled**

## 7 POWER CONSUMPTION

**Webcam Web application:**

1. Type-C Average power consumption: 5V/138mA
2. Type-C Photo moment: 5V/341mA
3. Battery Average power consumption: 3.8V/154mA
4. Battery Photo moment: 3.8V/304mA
5. Charging battery current: 100mA

Average Consumption during Inference is 1.74 watts, with 2 lithium-ion batteries (4300 mah) this system will last for 16.5 hours.

**Figure 6: 3D Printed Case for XIAO-ESP32S3-Sense**

## 8 RESULTS

we have summarized our results as follows:

**Model-1 (Not Used):** The initial MobileNetV2-based model, equipped with both a classification head for identifying types of animals and a regression head for bounding box prediction, showcased promising results during the training phase. After 20 epochs, the model achieved approximately 70% accuracy on both the training and test sets for bounding box predictions, as measured by Mean Squared Error. The categorical accuracy for classifying wildlife species reached around 72% on both sets, evaluated using Categorical Cross Entropy for One-Hot Encoded Labels (OHE). Despite these encouraging metrics, the deployment of this model onto the XIAO ESP32-S3 posed challenges. Attempts to quantize the model for efficient deployment on the ESP32-S3 were hindered by various errors, preventing its practical application in the intended environment.

**Model-2 (Used):** In response to the deployment challenges encountered with Model-1, we opted for a different approach, adopting a FOMO model from EdgeImpulse. This model, trained for 7 epochs due to constraints imposed by the non-enterprise Edge Impulse account (limited to 20 minutes and 4 GB of data), demonstrated a commendable performance. The precision, recall, and F1 scores stood at 66% 37% and 50% respectively. These metrics, while indicative of a functional model, acknowledge the potential for further improvement. It is noted that utilizing an enterprise account for extended training sessions could enhance the accuracy of the model, particularly in terms of precision and recall.

**XIAO ESP32-S3 Scalability:** The physical dimensions of the XIAO ESP32-S3, measuring approximately 21 x 17.5mm, underscore its suitability for discreet placement in various environments. This compact scale not only allows for integration into wildlife hideouts but also positions the device as a scalable solution adaptable to a variety of ecosystems. The small form factor aligns with the project's goal of creating unobtrusive and versatile monitoring devices for wildlife conservation.

In conclusion, the transition from Model-1 to Model-2 reflects a pragmatic approach to overcoming deployment challenges, showcasing the team's adaptability and commitment to delivering a functional solution. While Model-2 exhibits satisfactory performance, acknowledging potential improvements through extended training sessions with enterprise accounts underscores a forward-looking perspective in refining the deployed solution for enhanced precision and recall. The scalability of the XIAO ESP32-S3 adds a layer of practicality, envisioning widespread adoption across diverse wildlife monitoring scenarios

# 9 GITHUB REP LINK

## 9.1 GitHub link

https://github.com/sudharshan2001/tinyML-03-Wildlife-Monitoring

## 9.2 Demo Video

https://drive.google.com/drive/folders/1dIig4nRRTvORxRpIEhXgVZB78YVJXF1V?usp=sharing

# REFERENCES


[1] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for Hyper-Parameter Optimization. In *Proceedings of the 24th International Conference on Neural Information Processing Systems* (Granada, Spain) *(NIPS'11)*. Curran Associates Inc., Red Hook, NY, USA, 2546–2554.

[2] Robert David, Jared Duke, Advait Jain, Vijay Janapa Reddi, Nat Jeffries, Jian Li, Nick Kreeger, Ian Nappier, Meghna Natraj, Shlomi Regev, Rocky Rhodes, Tiezhen Wang, and Pete Warden. 2021. TensorFlow Lite Micro: Embedded Machine Learning on TinyML Systems. arXiv:2010.08678

[3] FOMO: Object detection for constrained devices. 2023. *Edge Impulse*. https://docs.edgeimpulse.com/docs/edge-impulse-studio/learning-blocks/object-detection/fomo-object-detection-for-constrained-devices.

[4] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. 2021. A Survey of Quantization Methods for Efficient Neural Network Inference. arXiv:2103.13630

[5] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv:1704.04861

[6] Xijie Huang, Zechun Liu, Shih-Yang Liu, and Kwang-Ting Cheng. 2023. Efficient Quantization-aware Training with Adaptive Coreset Selection. arXiv:2306.07215

[7] Shawn Hymel, Colby Banbury, Daniel Situnayake, Alex Elium, Carl Ward, Mat Kelcey, Mathijs Baaijens, Mateusz Majchrzycki, Jenny Plunkett, David Tischler, Alessandro Grande, Louis Moreau, Dmitry Maslov, Artie Beavis, Jan Jongboom, and Vijay Janapa Reddi. 2023. Edge Impulse: An MLOps Platform for Tiny Machine Learning. arXiv:2212.03332

[8] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2017. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. arXiv:1712.05877

[9] Andrey Kuzmin, Markus Nagel, Mart van Baalen, Arash Behboodi, and Tijmen Blankevoort. 2023. Pruning vs Quantization: Which is Better? arXiv:2307.02973

[10] Hengfui Liau, Nimmagadda Yamini, and YengLiong Wong. 2018. Fire SSD: Wide Fire Modules based Single Shot Detector on Edge Device. arXiv:1806.05363

[11] Ji Lin, Wei-Ming Chen, Yujun Lin, John Cohn, Chuang Gan, and Song Han. 2020. MCUNet: Tiny Deep Learning on IoT Devices. arXiv:2007.10319

[12] Shihan Liu, Junlin Zha, Jian Sun, Zhuo Li, and Gang Wang. 2023. EdgeYOLO: An Edge-Real-Time Object Detector. arXiv:2302.07483

[13] Simon Narduzzi, Engin Türetken, Jean-Philippe Thiran, and L. Andrea Dunbar. 2022. Adaptation of MobileNetV2 for Face Detection on Ultra-Low Power Platform. arXiv:2208.11011

[14] Shvetank Prakash, Matthew Stewart, Colby Banbury, Mark Mazumder, Pete Warden, Brian Plancher, and Vijay Janapa Reddi. 2023. Is TinyML Sustainable? Assessing the Environmental Impacts of Machine Learning on Microcontrollers. arXiv:2301.11899

[15] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. arXiv:1506.02640

[16] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2019. MobileNetV2: Inverted Residuals and Linear Bottlenecks. arXiv:1801.04381

[17] Mingzhu Shen, Feng Liang, Ruihao Gong, Yuhang Li, Chuming Li, Chen Lin, Fengwei Yu, Junjie Yan, and Wanli Ouyang. 2021. Once Quantization-Aware Training: High Performance Extremely Low-bit Architecture Search. arXiv:2010.04354

[18] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556

[19] Juan Terven and Diana Cordova-Esparza. 2023. A Comprehensive Review of YOLO: From YOLOv1 and Beyond. arXiv:2304.00501

[20] Suorong Yang, Weikang Xiao, Mengchen Zhang, Suhan Guo, Jian Zhao, and Furao Shen. 2023. Image Data Augmentation for Deep Learning: A Survey. arXiv:2204.08610

[21] Yiwu Yao, Weiqiang Yang, and Haoqi Zhu. 2019. Creating Lightweight Object Detectors with Model Compression for Deployment on Edge Devices. arXiv:1905.01787