IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 8 May 2024, accepted 25 May 2024, date of publication 28 May 2024, date of current version 7 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3406585

RESEARCH ARTICLE

Edge AI in Sustainable Farming: Deep Learning-Driven IoT Framework to Safeguard Crops From Wildlife Threats

KONKALA VENKATESWARLU REDDY[1], B. S. KARTHIKEYA REDDY[1], VEERAPU GOUTHAM[1], (Member, IEEE), MIRIYALA MAHESH[2], (Member, IEEE), J. S. NISHA[1], GOPINATH PALANISAMY[1], MALLIKARJUNA GOLLA[3], SWETHA PURUSHOTHAMAN[1], KATANGURE RITHISHA REDDY[1], AND VARSHA RAMKUMAR[1]

[1]School of Electronics Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu 632014, India
[2]School of Electronics Engineering, VIT-AP University, Amaravati, Andhra Pradesh 522237, India
[3]School of Electrical Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu 632014, India

Corresponding author: Veerapu Goutham (gouthamveerapu@gmail.com)

This work was supported by the Vellore Institute of Technology, Vellore Campus, Tamil Nadu.

ABSTRACT The relentless global population growth and the ever-increasing food demand pose formidable challenges to the agricultural sector. Farmers grapple with the ongoing challenge of wildlife-induced crop damage and human loss, which not only impedes food production but also exacerbates supply and demand imbalances. However, the rise of TinyML enables Edge AI as a promising avenue for implementing resource-efficient deep learning techniques on low-end edge devices. In this paper, we introduce an innovative solution that harnesses the power of Edge AI using tinyML-based deep learning algorithms in conjunction with the Internet of Things (IoT) for animal intrusion detection and deterrence system. The proposed system is developed to create remotely managed defense system tailored to safeguard vast agricultural expanses. It integrates a laser detection system and an AI-CAM with light weight deep learning algorithms for animal intrusion detection and classification. This system also ensures efficient animal deterrence and real-time monitoring for farmers, enabling them to assess the situation with the assistance of an intelligent rover build using IoT. This work emphasizes on proposing a light-weight deep learning model named EvoNet for animal classification. Results reveal that the proposed model achieves the highest accuracy at 96.7%, surpassing other models presented in this paper. However, for edge devices where compact file sizes are crucial, the model also offers comparable accuracy with file sizes as low as 1.63MB with the help of pruning and quantization techniques. This conceptualized solution has the potential to revolutionize agricultural wildlife management, ushering in a new era of crop protection and economic resilience.

INDEX TERMS Animal intrusion detection, attention mechanism, deep learning, edge AI, EvoNet, intelligent rover, Internet of Things, sustainable farming, TinyML.

I. INTRODUCTION

In the evolving landscape of agriculture, transitioning from Agriculture 4.0 to Agriculture 5.0 is marked by a significant paradigm shift [1]. This transformation is characterized by the heightened integration of cutting-edge technologies such as the Internet of Things (IoT), sensor networks, cloud computing, edge AI, big data analytics, and the deployment of physical robots. The collective utilization of these advanced tools not only enhances productivity but also contributes

The associate editor coordinating the review of this manuscript and approving it for publication was Ramakrishnan Srinivasan.

TABLE 1. Estimated crop loss due to elephant depredation in south india(some places like Pachhedoddi and Kanchalli).

Name of crop	Economic loss (Rupees/acre)	
	Pachhedoddi	Kanchalli
Mazie	36,000	6000
Ragi	9000	3600
Turmeric	25,000	8330
Groundnut	6750	4700
Others	7330	3670

to substantial economic growth, thereby revolutionizing our daily lives.

The modernization of agriculture has proven to be instrumental in advancing various tasks through the integration of IoT and machine learning. One notable example is the application of machine learning for predicting soil types, soil taxonomy, output quality, and nutrition levels [2]. Employing algorithms such as k-Nearest Neighbors (kNN) and Logistic Regression (LogR), these technologies enable accurate and efficient predictions in the realm of agriculture. This amalgamation of technologies signifies a holistic approach towards precision farming, intelligent resource management, and sustainable agricultural practices, ushering in a new era of efficiency and innovation in the agricultural sector.

In the ever-changing landscape of agriculture, effectively handling interactions with external elements like wildlife remains a pertinent and ongoing challenge. In many regions, the delicate balance between human livelihoods and the survival of wildlife is facing a growing threat due to socio-economic activities encroaching upon natural habitats [3]. This encroachment has led to shrinking habitats and reduced food sources for wild animals, forcing them to venture into agricultural fields in search of sustenance. This leads to a conflict between human needs and the survival instincts of animals. Conservation and preservation efforts find themselves grappling with the complexities of this challenge, necessitating strategic interventions to mitigate the widespread phenomenon of crop raiding and foster harmonious coexistence. Mainly, crop raiding happens due to the invading of wild animals into the farmlands and causing damage to crops, presents a multifaceted challenge with extensive ecological, economic, and social consequences. This issue is especially prevalent in areas bordering forests. Furthermore, these animal intrusions occur throughout the various stages of crop growth, spanning from the planting of seeds to the final harvest, resulting in substantial agricultural losses. The implications of this issue extend beyond crop damage, encompassing problems such as vehicular accidents, disease transmission, environmental degradation, and harm to ornamental plants. The severity of crop damage is a global concern, as evidenced by a survey conducted in Kerala [4] and Karnataka [5], India, revealing substantial financial losses incurred by villages due to crop damage by wild animals as shown in Table 2 and Table 1. Even in the eastern coastal region of Odisha, wild animals, including elephants, wild boars, and deer, pose a formidable threat to agriculture, causing substantial crop damage, often ranging from 50-100%. In addition, incidents of human injuries reached a staggering total of 7,381 cases, incurring compensation expenses of Rs. 3.4 crores. These statistics underscore the critical need for effective animal intrusion detection systems to mitigate such human-wildlife conflicts and their associated human and economic tolls.

TABLE 2. Statistics of crop damage in Kerala.

Wild Animals	Crops	% crop damage
Elephant	Coconut, plantain, paddy	72
Gaur	Mulberry, sandal	62
Sambar	sapota	17
Wild boar	Tapioca, tubers, paddy	16

In this context, our research is to develop effective and sustainable solutions for animal intrusion detection and deterrence system using edge devices in agricultural fields. Accordingly, a novel system that leverages the power of the IoT and deep learning for detection and deterrence systems to mitigate the impact of wildlife incursions and to minimize crop damage, promote ecological conservation, and protect the livelihoods of farming communities. Our approach to animal intrusion detection revolves around Edge AI and TinyML, bringing computation closer to data sources within the agricultural environment. This strategy optimizes the deployment of modern IoT and AI applications for effective intrusion detection. Our technologically-driven crop protection system not only serves to safeguard crops but also promotes ecological conservation and protects the livelihoods of farming communities. By balancing the needs of human communities and wildlife, we strive to foster coexistence and advocate sustainable agricultural practices on a global scale. The main contributions of this works are:

1) A deep learning-based IoT framework and hardware system aimed at protecting crops from wildlife threats using edge devices has been developed. Our system remains in a dormant state until an early animal intrusion is detected to reduce power consumption.
2) In this framework, these edge devices utilize a tinyML based light weight deep learning model (EvoNet) for animal detection and classification.
3) Our system comprises both static and dynamic components. The static part is responsible for animal detection, while the dynamic aspect is geared towards tracking the animal with a intelligent rover. Additionally, the system empowers farmers with real-time assessment capabilities through the intelligent rover connecting through IoT.
4) Results reveal that the EvoNet model achieved the highest accuracy at 96.7%, compared to other models presented in this paper. However, for edge devices where compact file sizes are crucial, the model offers

comparable accuracy with file sizes as low as 1.63MB with the help of pruning and quantization techniques.

The rest of the paper is organized as follows: Section II presents the related work. Section III presents the animal intrusion system. Section IV describes the proposed model of the long-range surveillance intelligent rover. Section V provides the deep learning algorithms for animal classification. Section VI presents the results and discussions. The conclusion is presented in Section VII.

II. RELATED WORKS

In this section, a thorough overview of prior research efforts dedicated to the development of real-time animal detection systems is provided. These efforts encompass a diverse range of technologies and methodologies. Our aim is to present a comprehensive comparison that encapsulates the evolution of approaches in this domain.

In the literature, several research works were proposed innovative solutions using wireless sensor networks (WSN) and AI based systems to safeguard the crops from the wildlife threats. Mainly, the authors of [6] introduced a WSN-based system tailored for tracking wildlife in challenging environments. This system, designed according to IEEE 802.15.4 standards, strategically deploys video sensors connected to nodes for reliable target detection. Similarly, the authors in [7] introduced a WSN based system for crop protection against animal intrusions. It utilizes sensors and devices like PIR sensors, sound generators, and RF modules to detect intrusions and deter animals. In [8], the authors examined the wide range use of WSNs for agriculture monitoring which improves the quality and productivity of farming. With the help of sensors, data (i.e., humidity, carbon dioxide level, and temperature) is gathered in real-time scenarios. The convergence of Internet of Things (IoT) technology and image processing techniques has revolutionized agricultural security. In [9], authors explored a sophisticated system employing wireless sensors, deep learning algorithms (MCNN and AlexNet), and GSM modules. In [10], authors focused on the recent applications of WSNs in agriculture research as well as classifies and compares various wireless communication protocols, and energy harvesting techniques for WSNs used in the monitoring systems. The integration of IoT devices facilitates real-time monitoring, allowing for instant image capture and analysis. In [11], the authors examined a model that combines IoT and machine learning to address the problem of animal intrusion in agriculture. The model employs a Raspberry Pi, along with various hardware components, for surveillance and communication. In addition, the authors in [12] Smart Agriculture application integrates Edge Computing, IoT, and AI to safeguard crops from ungulate attacks using computer vision and species-specific ultrasound emission. Evaluated on diverse embedded platforms like NVIDIA Jetson Nano and Raspberry Pi with Intel Movidius NCS, the system, driven by YOLOv3, showcases superior detection accuracy. The study underscores the importance of tailored hardware-software integration, showcasing the potential of innovative technologies in Smart Agriculture. Similarly, A. V. Prabu et al. proposed an IoT-based Crop Field Protection System (ICFPS) that leverages deep learning techniques for feature extraction, disease detection, and field data monitoring [13]. In [14], the authors delved into assessing agricultural losses and considered the practical importance and economic feasibility of deploying acoustic equipment to deter wildlife from encroaching on farmlands. Given the complex and evolving nature of human-wildlife conflicts in agriculture, there is a compelling need for a technologically-driven crop protection system characterized by cost-effectiveness, robustness, reliability, and ease of adoption by farmers. If realized, such a system has the potential to strike a balance between the needs of human communities and wildlife, fostering coexistence and promoting sustainable agricultural practices on a global scale.

In the existing research works [7], [9], [11], [12], a comprehensive approaches is proposed for employing IoT, WSNs, and deep learning models are presented for safeguarding crops against animal intrusion. These methodologies, exhibit a limitation in affording complete flexibility to farmers in assessing the situation firsthand. This work mainly focusing on providing maximum flexibility to the farmers by introducing a intelligent rover equipped with the capability to provide a live video stream of the surrounding crop area to the farmer. This innovative addition allows the farmer to have real-time visibility and control over the situation. This intelligent rover can be remotely operated through a wireless IoT controller accessible from anywhere via the Internet. This feature not only bolsters the protection of crops but also empowers the farmer to actively monitor and assess potential threats, particularly in the event of animal intrusion. By leveraging the live video stream and remote control functionality, farmers gain a more dynamic and responsive means of safeguarding their crops.

Additionally, many of the suggested models prioritize accuracy, although at the expense of increased computational power and cost of the hardware system. Some models have incorporated complex detection mechanisms, further elevating the overall complexity of the model. While these approaches demonstrate commendable performance, their resource-intensive nature raises concerns. In contrast, our proposed model adopts a TinyML (Tiny Machine Learning) approach which allows for execution with minimal computational power, providing a balance between performance and efficiency. The inherent advantage of applying TinyML model lies in its ability to produce results in a shorter time frame without compromising on the quality of outcomes. However, TinyML model delivers satisfactory results when compared to the traditional deep learning algorithms while demanding less computational resources. This efficiency is particularly valuable in real-world applications where timely response to crop threats is crucial.

In addition, some of the existing models involve placing poles throughout the field and deploying sensors extensively to detect animal entry, incurring higher overall costs. In contrast, our approach utilizes a boundary laser-based system for detecting animal entry. This not only reduces the financial burden but also enhances the efficiency of the detection process. Moreover, a singular pole positioned at the center is employed, equipped with speakers and lights to deter animals. This alternative strategy proves to be more cost-effective while still effectively addressing the challenge of preventing animal intrusion.

III. SYSTEM MODEL

The proposed system model, depicted in Figure 1, is structured into three distinct modules: animal detection, animal classification, and animal deterrence. Each module serves a crucial role in addressing the challenge of managing animal intrusion in agricultural fields.

A. ANIMAL INTRUSION DETECTION AND DETERRENCE SYSTEM

The primary objective of the first module is to ascertain the presence of animals within the field. Since animals move unpredictably, several research works considered motion sensors, such as PIR sensors, for motion detection [15]. However, PIR sensors have limitations, including lower accuracy in motion detection, potential false triggers, limited range, and inoperability at temperatures above 35 degrees Celsius, which is common in many regions. To overcome these challenges, a laser based boundary system is introduced for early intrusion detection. We strategically place four boundary poles at the field's corners as shown in Fig. 2. Each boundary pole are equipped with AI-CAM, along with essential components like a laser diode and photo diode. The laser diode establishes a perimeter around the field, effectively covering all boundaries. Continuous signals are transmitted from the each laser to the respective adjacent photo diodes as shown in Fig. 2. This process forms a laser boundary around the field, connecting all poles like a rectangular security. In addition, the built-in camera of the microcontroller serves as a surveillance camera to monitor the environment. Likewise, total four cameras positioned around the field for capturing the animal intrusions. When no animals are present in the area, the laser line falls directly onto the photodiode, sending a constant value to the microcontroller. In this state, the camera remains inactive to conserve power, enabling the system to operate for an extended period with minimal power consumption. When an animal attempts to breach this perimeter and enter the field, a disruption of the laser barrier is detected by a microcontroller, confirming the presence of the early animal intrusion into the field. Simultaneously, the camera module captures images of the intruding animal, which are subsequently subjected to analysis through deep learning models to classify the type of animal. Furthermore, a central pole is strategically located at the center of the field, which serves as a decision-making device equipped with countermeasure options designed to deter animals from entering the field. The central pole is equipped with countermeasure devices, including a speaker and a light, controlled by the ESP32 microcontroller as shown in Fig. 4.

The microcontroller is equipped with a pre-trained deep learning model implemented using TensorFlow Lite [16]. The images captured by the camera are fed into the deep learning model for animal classification. Information about the predicted animal is then transmitted to the central pole using the ESP-NOW protocol, a communication protocol used by ESP32 modules.The ESP32 NOW protocol was chosen due to its low power consumption and integration into the microcontroller. ESP32 NOW operates in the sub-2.4GHz band and suits local network scenarios. It simplifies data transmission with an efficient API, crucial for real-time communication between boundary poles and a central pole in our system. This setup enables quick responses to animal intrusions, safeguarding the agricultural ecosystem. With antenna integration to ESP32 it provided a practical 200-meter range without packet loss, balancing cost-efficiency and functionality [17], [18]. Upon receiving the animal intrusion information, the central pole loud sounds using speakers, while the light is primarily used at night to alert the animal. To enhance the system's effectiveness, recorded sounds made by local farmers are used, as different animals respond differently to various sounds. It's worth noting that these sounds are carefully chosen to ensure they do not cause any harm to the animals. Based on the animal classification information, the microcontroller selects the appropriate sound to play on the speakers. Furthermore, the central pole's microcontroller is connected to Wi-Fi, enabling it to leverage IoT capabilities. Using this connectivity, the central pole sends instant notifications to the farmer by ringing the buzzer on the controller.

When the farmer receives a notification regarding an animal intrusion, they can activate the rover to assess the situation. The rover is equipped with a wireless controller that can be operated from anywhere, as long as it is connected to Wi-Fi. Positioned at the front of the rover is a camera that provides a live video stream to the farmer's mobile device. With the wireless controller in hand, the farmer gains control over the rover's movements and actions. They can survey the field in real-time using the live video stream to determine whether the animal has left the area. If it becomes evident that the animal has not yet departed, the farmer can employ high-pitched buzzers closer to the animals and encourage it to leave the field. The detailed description regarding the proposed model of the long-range surveillance intelligent rover using IoT is presented in Section IV. In cases where the situation escalates and becomes severe or beyond the control of remote measures, the farmer has the option to take necessary physical actions. This comprehensive approach significantly reduces crop damage resulting from animal intrusions.

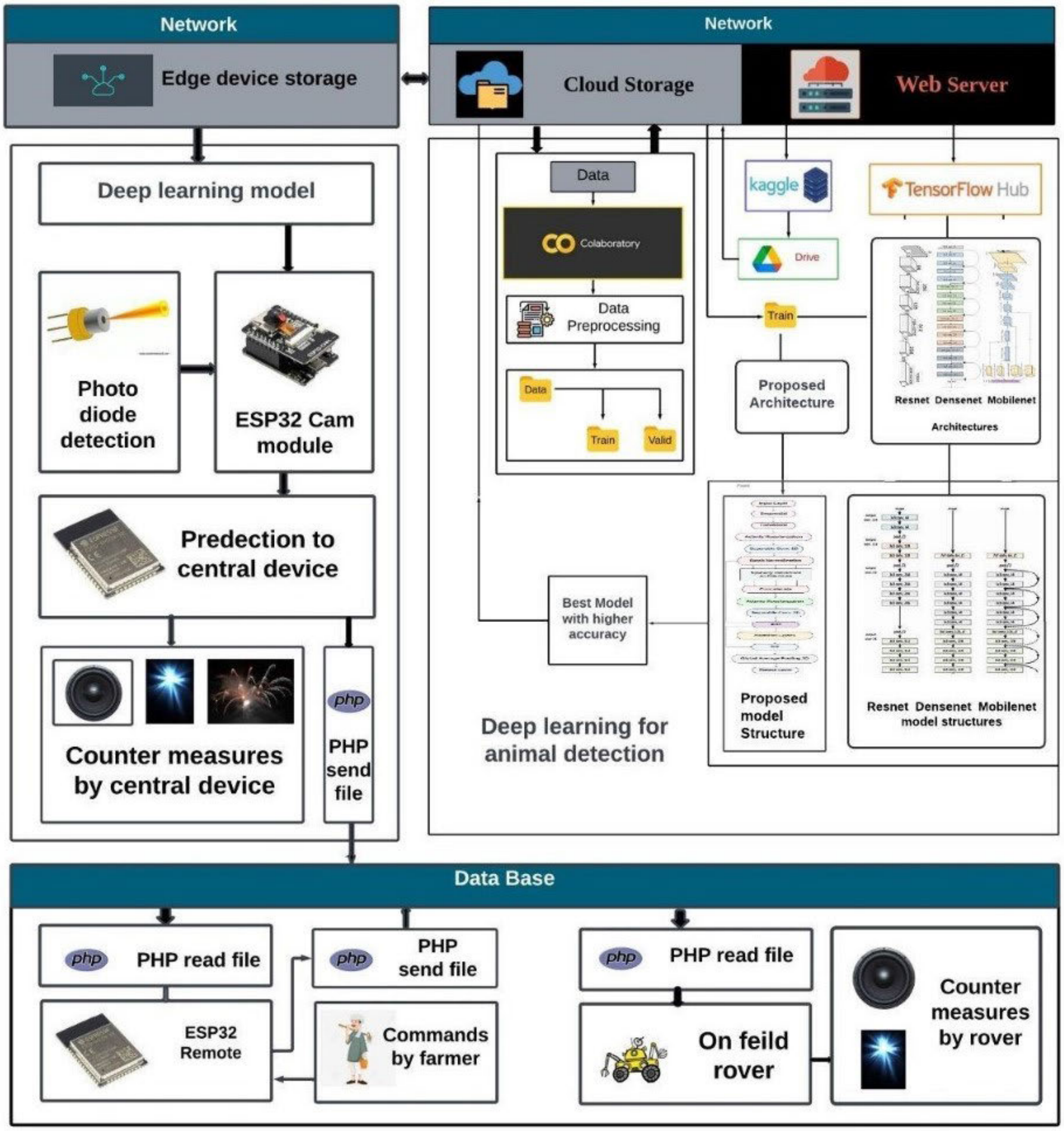


FIGURE 1. System model.

B. ANIMAL CLASSIFICATION

The second module of the system is focused on animal classification based on the images captured by the surveillance camera. This classification is made possible by utilizing a pre-trained tinyML based deep learning model, which is loaded onto the microcontroller. Given the potential similarities in the appearance of various animals, the chosen model must exhibit a high degree of intelligence to achieve accurate predictions. The process to attain high level of accuracy involves three pivotal steps: The initial step involves the selection of an extensive dataset that encompasses a diverse array of animal classifications, ensuring a comprehensive training set. The second step entails the careful choice of a deep learning model suited for the task. In this context, the proposed model (EvoNet) is compared with DesnseNet, ResNetV2-50, Inception, MobileNetV2, Efficientnet-B0 and EfficicentNet-B7. The selection among these models is informed by their performance in the specific task of animal type prediction. The third step involves the refinement of the selected model to enhance its ability to accurately classify the animal based on the input image. The more detailed description regarding the deep learning models is presented in Section V.

IV. PROPOSED MODEL OF THE LONG-RANGE SURVEILLANCE INTELLIGENT ROVER USING IOT

To tackle the dynamic environments, our system needs real-time assessment capabilities through the rover connected

FIGURE 2. Assembly at boundary poles.

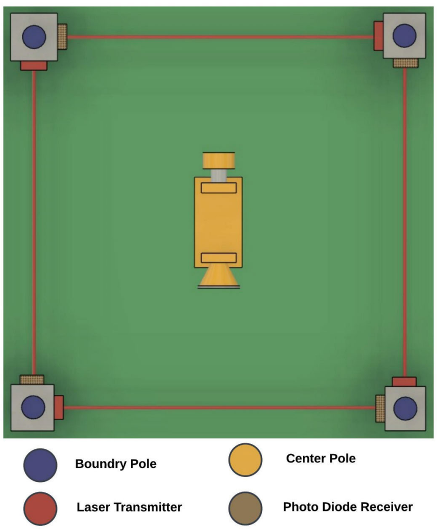


FIGURE 3. Overview of placements of the poles.

using IoT to the farmers. Thanks to the 5G connectivity for providing the seamless internet provision in the rural areas [19], [20]. Here, the rover and controller are seamlessly connected through the utilization of IoT capabilities offered by ESP32 modules. This connectivity empowers them to communicate effectively over extended distances, transcending the limitations of physical proximity. The controller initially connects to the internet, establishing a connection

FIGURE 4. Assembly at central poles.

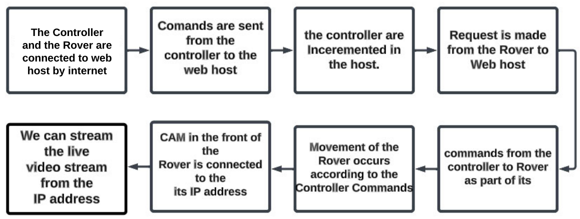


FIGURE 5. Flow diagram of the working of the long range surveillance rover.

to the database. Simultaneously, the intelligent rover is also internet-enabled and linked to the same database as shown in Fig. 5. As Bluetooth technology is not involved, there is no requirement for the controller and rover to be in close proximity. When a command is issued from one of the joysticks on the controller, it is captured by the ESP32 and transmitted to the controller's database. This transmission includes the corresponding ID number and a timestamp. Each new command generated by the joysticks is seamlessly appended to the database without overwriting previous commands. Meanwhile, the ESP32 on the rover, also internet-connected, becomes active and requests the latest commands from the database. The database responds by retrieving the commands with the lowest ID numbers and dispatches them to the rover's ESP32. The received command is then relayed to the motor driver, responsible for precision control of the rover's motors. The motors execute the command accurately, ensuring smooth and reliable mobility of the rover. Additionally, a camera is positioned on the front of the rover, mounted on top of a servo. This setup enables high-definition live video streaming from an assigned IP address. Consequently, users can observe the rover's

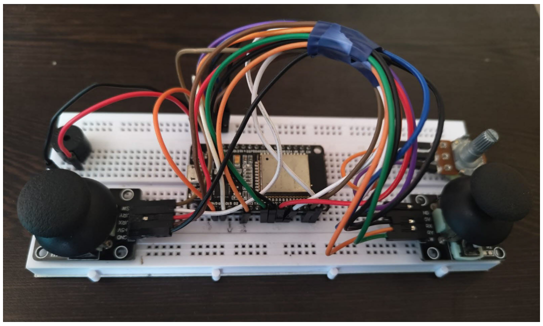


FIGURE 6. Controller to control the rover.

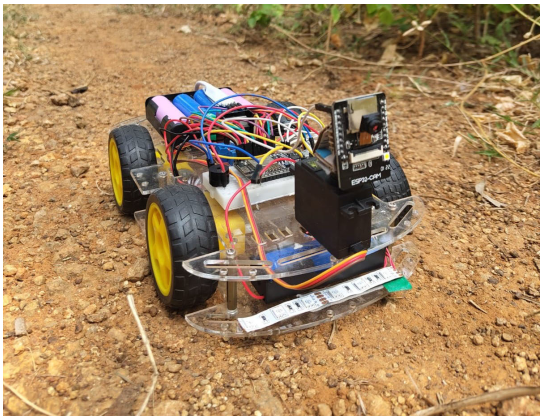


FIGURE 7. On field rover.

environment while controlling it, even when it is not within the visible range. The servo's design allows the camera to rotate freely by 360 degrees, facilitating comprehensive surveillance of the surroundings. Consequently, users can maintain uninterrupted access to the rover's operational environment.

A. SYSTEM OVERVIEW DIAGRAM OF ROVER

The controller is equipped with an ESP32 as its central component. Two 2-axis joysticks, a potentiometer and a buzzer are connected to the ESP32 as shown in Fig 6. The joysticks provide four different values for the movement of the rover: front, back, left, and right. The push switch on the joystick is used to enable the buzzers on the rover. The buzzer on the controller rings when the animal intrusion occurs in the field. The potentiometer enables the servo on the rover to rotate 360 degrees. Once the controller is connected to the internet, it sends the values from the joysticks and the potentiometer to the database. In the database, these six values are updated with an ID number and a timestamp. The ID number is automatically incremented when new commands are sent from the controller.

The rover is equipped with an ESP32 as its central component. Connected to the ESP32 are a motor driver and a servo. Four motors of the rover [21] are linked to the motor driver, which is connected to the primary component as shown in Fig. 7. Mounted on the servo is a AI-CAM for real-time video streaming as shown in Fig. 16 [22]. Additionally, buzzers and lights are present to deter animals. Initially, the rover is connected to the internet. When the ESP32 on the rover sends a request to the database for commands, it retrieves the commands with the lowest ID numbers first and transmits them to the ESP on the rover. The ESP processes the data and relays it to the motor driver, which then controls the motors based on the received commands. Simultaneously, access to the assigned IP address of the camera allows for streaming live video from the CAM.

V. DEEP LEARNING MODELS

In this work, we mainly focused on animal detection using various deep learning algorithms (such as EfficientNet-B0, EfficientNet-B7, and proposed model (EvoNet)) with Attention Mechanism.

A. EfficientNet

EfficientNet is a convolutional neural network architecture designed to achieve superior performance while maintaining computational efficiency. This is characterized by a compound scaling method that uniformly scales all dimensions of depth, width, and resolution. This scaling strategy allows EfficientNet to efficiently balance model size and accuracy across various resource constraints, making it highly versatile for deployment on different hardware platforms. By leveraging depth-wise and point-wise convolutions along with squeeze-and-excitation blocks, EfficientNet optimizes model capacity. Additionally, it incorporates techniques like dropout regularization and batch normalization to improve generalization and training stability. With its state-of-the-art performance on image classification tasks across different scales, EfficientNet has become a popular choice for applications where computational resources are limited or efficiency is paramount.

EfficientNet-B0 as the foundational model in the EfficientNet family, boasts an architecture renowned for its efficiency and impressive performance [23]. Its structure is defined by a series of convolutional and depth-wise convolution layers grouped into blocks. These blocks increase in number as the network's depth escalates, making EfficientNet-B0 an effective feature extractor. These convolutions optimize computational complexity by initially applying depth-wise convolutions for individual input channels. The reduction in parameters doesn't compromise the model's capacity to capture crucial features. With a base resolution of 224×224 pixels, this architecture efficiently processes a broad range of image-related tasks. It starts with a stem convolution operation that processes input images with 3×3 convolutions, ensuring that the subsequent blocks extract valuable information. EfficientNet-B0 culminates in a global average pooling layer, which reduces the feature maps to a 1D vector by aggregating the most important information. The final touch is a fully connected layer with a SoftMax activation function, generating a probability distribution across the output classes.

EfficientNet-B7 starts with a stem convolution operation, which processes input images using 3×3 convolutions, ensuring that vital information is extracted and preserved throughout the network. The model's base resolution of 224×224 pixels serves as a robust foundation for handling a wide range of image-related tasks. It culminates in a global average pooling layer, a pivotal component that transforms the feature maps into a 1D vector by aggregating essential information. The final layer is a fully connected layer featuring a SoftMax activation function, responsible for generating a probability distribution across the output classes. EfficientNet-B7's architecture is expertly designed to excel in applications demanding complexity and performance. Its extensive computational demands are balanced by its remarkable ability to extract intricate features, making it the ideal choice for tasks that require fine-grained detail recognition and image analysis.

For comparative analysis, DenseNet, Inception, MobileNetV2 and ResNetV2-50 models are used. DenseNet Model, introduced the concept of dense connectivity, where each layer is connected to every other layer in a feed-forward fashion. Inception, emphasizes multi-scale processing through the use of parallel convolutional pathways of different kernel sizes within the same layer. MobileNet, specifically designed for resource-constrained environments such as mobile and embedded devices. ResNetV2-50 [24], a variant of the ResNet architecture with 50 convolutional layers, takes the center stage in the animal type prediction module.

B. PROPOSED MODEL

EfficientNet models are excellent for image classification due to their compound scaling capability. However, one of the main drawbacks of the EfficientNet models is that they require high computational resources to train, although the requirements are considerably less compared to other models such as DenseNet and Inception. However, the trained deep learning models deployed on edge devices require low computational resources. Therefore, our proposed model aims to develop an edge device-friendly architecture by utilizing the pre-trained imagenet weights of EfficientNet-B0. The proposed model is divided into three parts. Firstly, the Base Model comprises pre-trained weights from ImageNet. Secondly, an custom CNN architecture is designed specifically for edge device-friendly layers. Lastly, an Attention mechanism is incorporated to enhance feature learning. This approach aims to leverage the strengths of pre-trained weights while tailoring the model architecture to suit the computational constraints of edge devices, thereby facilitating effective image classification.

1) BASE MODEL

In the proposed model architecture, the EfficientNet-B0 is utilized as the backbone or base of the model. Its primary function is to serve as a feature extractor by leveraging the pre-trained weights of ImageNet dataset. Setting the layers of the base model to non-trainable ensures that these learned features remain intact and unchanged during training. We want to preserve the knowledge encoded in these weights, as they capture general patterns in the dataset. The subsequent CNN architecture, composed of additional layers like SeparableConv2D and Dense layers, builds upon the features extracted by the base model. During training, the weights of these added layers are updated based on the images from the dataset. By employing this strategy, known as transfer learning, we benefit from the generalization power of the pre-trained base model while fine-tuning the model to perform well on the animal dataset. This approach is particularly effective when we have limited computational resources, as it allows us to leverage the knowledge learned from large-scale datasets like ImageNet.

2) CUSTOM CNN ARCHITECTURE

The three main layers used to build the CNN architecture are seprable convolutional layers, sparsely connected layers and Residual Connection.

In traditional Conv2D layers, which are commonly used in CNN architectures, each convolutional operation involves a substantial number of multiplications, making them computationally intensive. For example, if we aim to convert a feature map from $7 \times 7 \times 64$ to $7 \times 7 \times 128$ using a 3×3 kernel, the total number of calculations required would be $3 \times 3 \times 64 \times 5 \times 5 \times 128 = 1{,}843{,}200$. However, Separable Conv2D layers offer a more efficient alternative. These layers break down the convolution operation into two steps: depthwise convolution and pointwise convolution. Depthwise convolution applies a separate convolutional operation for each channel of the input feature map, reducing the computational load. In the given example, the number of multiplications required for depthwise convolution would be 64 (number of input channels) $\times$ 3 $\times$ 3 (kernel size) $\times$ 5 $\times$ 5 (number of moves) = 14,400. Following depthwise convolution, pointwise convolution combines the output channels from the depthwise step using 1×1 kernels. In our scenario, the number of multiplications for pointwise convolution would be 128 (number of output channels) $\times$ 64 (number of input channels) $\times$ 5 $\times$ 5 (number of moves) = 204,800. The total number of multiplications for both depthwise and pointwise convolutions sums up to 14,400 (depthwise) + 204,800 (pointwise) = 219,200, which is significantly lower compared to the traditional Conv2D approach. This makes the Separable Conv2D layers more computational efficient make it more compatible with for Edge devices.

As illustrated in Figure 8, the combination of three separate Separable Conv2D layers formed a sparsely connected layer. Such layers are valuable for preventing overfitting, especially in scenarios where limited training data is available. They enable us to increase both the depth and width of the model without significantly escalating computational requirements. Moreover, the inclusion of a Gaussian Dropout layer further

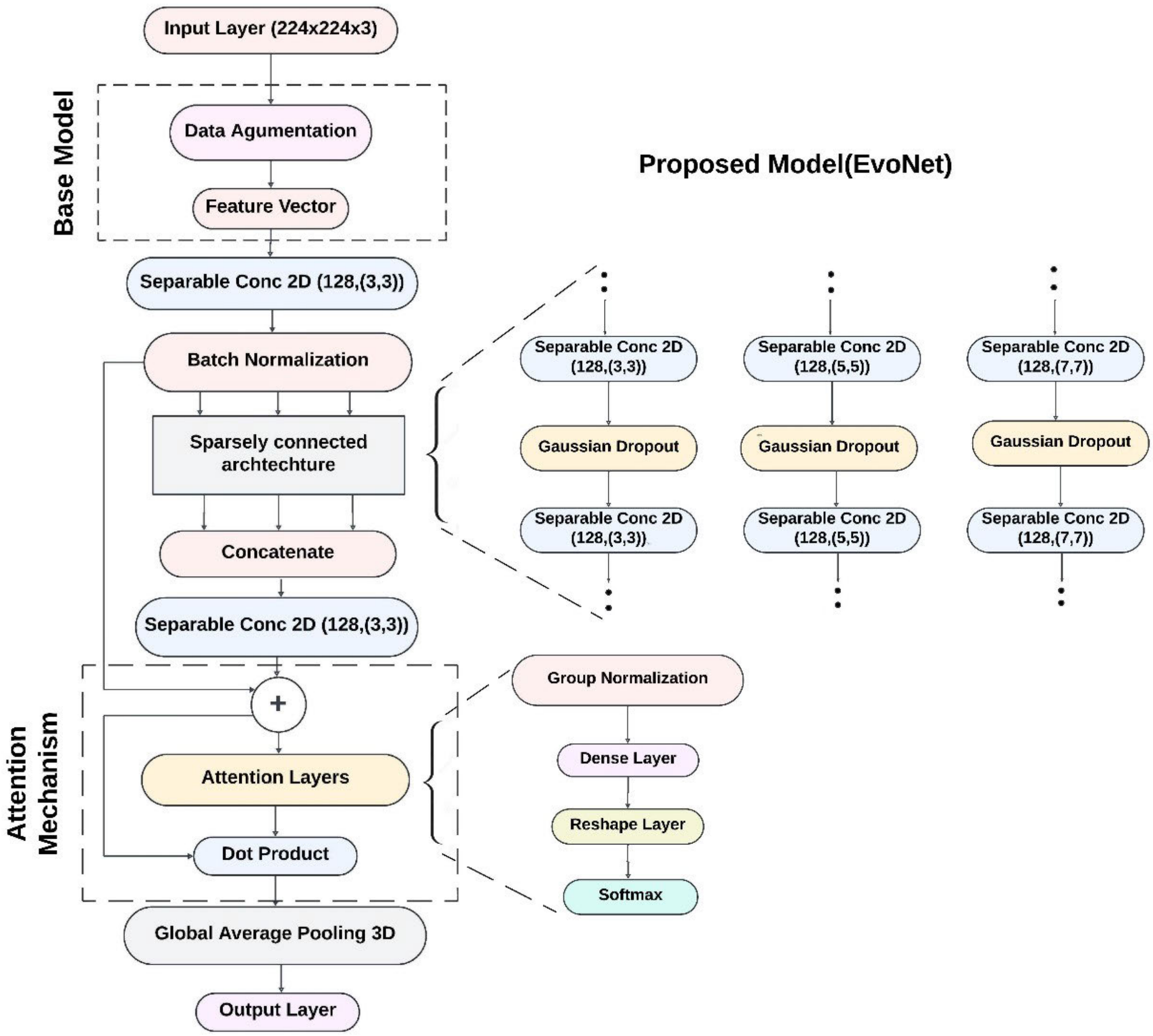


FIGURE 8. Architecture of EvoNet.

aids in mitigating overfitting by randomly dropping neurons and introducing noise to the input data.

In the proposed model architecture, the residual connection plays a pivotal role. By incorporating the output from the feature extractor into the sparsely connected layer's output, the residual connection allows the model to retain valuable information extracted by the base model from the input images. By preserving this information through the residual connection, the model can leverage both low-level and high-level features more effectively for classification tasks. Additionally, the residual connection serves as a form of regularization by providing an alternate path for the gradient to flow during training, thereby helping to prevent overfitting. This regularization contributes to maintaining a balance between model complexity and generalization performance, ultimately leading to improved model robustness and performance.

3) ATTENTION MECHANISM

The attention mechanism is inspired by human visual attention, where we selectively focus on certain regions of an image while ignoring others. Similarly, in neural networks, attention mechanisms allow the model to focus on relevant parts of the input data while downplaying less important regions. This can be particularly useful in tasks where different parts of the input contribute differently to the output. In the proposed model, attention weights are computed using a group normalization layer followed by a dense layer and softmax activation. Group normalization is applied to normalize the feature maps produced by the convolutional layers. This helps in stabilizing the training process and making the model less sensitive to variations in input statistics.

The normalized feature maps are then passed through a dense layer followed by a softmax activation function. The

TABLE 3. Image distribution between classes in the dataset.

Class	Training Data	Test data	Class	Training Data	Test Data
Bear	150	26	Leopard	150	37
Brown Bear	150	39	Lion	150	39
Cattle	150	38	Lynx	150	25
Cheetah	150	35	Mule	150	15
Deer	150	39	Pig	150	37
Elephant	150	32	Rabbit	150	37
Fox	150	29	Raccoon	150	35
Giraffe	150	23	Sheep	150	25
Goat	150	33	Tiger	150	25
Horse	150	34	Zebra	150	31

dense layer allows the model to learn the importance of different spatial locations within the feature maps. Softmax activation ensures that the attention weights sum up to 1, making them interpretable as probabilities. Once the attention weights are calculated, they are used to compute a weighted sum of the feature maps produced by the convolutional layers. This is achieved through a dot product operation between the attention weights and the feature maps. The resulting weighted sum represents a focused representation of the input data, where regions with higher attention weights contribute more to the final representation. This mechanism allows the network to focus on important regions of the input image while suppressing irrelevant areas, potentially improving classification performance.

C. DATASET

We imported a dataset that consists of different classes of animals, including Bear, Goat, Cattle, and 17 more species [25], totaling 4445 images as shown in Fig. 9 and the gathered information has been labeled. The objective is animal classification based on the images of the animals that may not always be in ideal environments.To achieve better results in prediction, a dataset was included containing images taken in various environments, including nighttime, broad daylight, and from different angles such as close-ups, long shots, and top views. This diverse dataset allows the model to train on a wide range of images under different conditions, improving its accuracy in animal classification. Given the similarity in appearance among many classes of animals, three different deep learning models were considered for animal classification: the "VGG" model, the "ResNet" model, and the "EfficientNet" model. This approach enables us to evaluate the results across different models and choose the best-performing one.

We divided the dataset into two parts: a training dataset containing 80% of the data and a test dataset containing the remaining 20% to remove the over fitting issue. The model is exclusively trained on the training data, which allows it to evaluate its performance on unseen data (the test data) during each epoch of training. This iterative process aims to enhance accuracy and ensure correct predictions when the model is applied in real time. When we test the model on unseen data, the images are not always in the correct orientation, and some may undergo distortion. In such cases, predicting the animal can be challenging. To address this issue, data preprocessing techniques, such as data augmentation are applied to the training data. Data augmentation involves modifying certain parameters of the training data images [26]. For example, some images may be horizontally or vertically flipped, while others may be distorted by resizing them to half their original size as shown in Fig. 10. This process allows the model to train on new and diverse data, contributing to the reduction of overfitting. As a result, the model becomes more capable of accurately predicting images in various conditions.

D. CHOOSING BEST LEARNING RATE

In deep learning, selecting an appropriate learning rate is a critical hyperparameter. It significantly impacts the model's training dynamics, convergence speed, and ultimately, its performance. An incorrect learning rate can lead to slow convergence, divergence, or suboptimal results. To address this, a learning rate scheduler is employed as a valuable tool in the training process. Its primary purpose is to adjust the learning rate dynamically during training to help the model converge efficiently and achieve optimal performance.

In the context of the deep learning architecture for animal detection, determining the best learning rate is paramount due to the model's depth and complexity. A learning rate scheduler assists in fine-tuning this hyperparameter by systematically varying the learning rate and monitoring the model's performance. The learning rate scheduler explores a range of learning rates during training. It starts with a range of potential learning rates and iteratively adjusts them based on the model's performance as shown in Fig. 11. This exploration allows the model to find the learning rate that optimally suits the task of animal type prediction.By gradually adapting the learning rate, the scheduler can overcome challenges like vanishing gradients or overshooting the optimal solution. This adaptability enhances the model's convergence speed and stability, which is especially critical in deep architectures like ResNet-50, EfficientNet.

E. CHOOSING HYPERPARAMETERS

The choice of hyper parameters plays a crucial role in the performance of our deep learning model. First, the image size was set to 224 × 224 pixels. This resolution was selected as it strikes a balance between computational efficiency and the model's ability to capture intricate features within the images. It ensures that the model processes a substantial amount of visual data while maintaining manageable computational requirements. For the batch size, we opt for 32. This batch size determines the number of images processed in each iteration during training. It was carefully chosen to optimize the balance between model convergence and computational efficiency. A batch size of 32 allows the model to learn multiple images simultaneously, thus enhancing training speed and efficiency.

$$\sigma(\overrightarrow{Z_i}) = \frac{e^{Z_i}}{\sum_{j=1}^{K} e^{Z_j}} \tag{1}$$

FIGURE 9. Sample images from the dataset.

FIGURE 10. Image before and after data augmentation.

Regarding activation functions, SoftMax, represented in Eq. (1), was utilized. SoftMax is a commonly employed activation function, especially in multiclass classification tasks. It converts the model's output into a probability distribution over multiple classes, making it well-suited for our animal classification problem. The choice of optimizer was Adam. Adam is an adaptive optimization algorithm known for its effectiveness in training deep neural networks. It dynamically adjusts the learning rate for each parameter, providing a powerful mechanism for efficient convergence during training. The Adam optimizer was selected to ensure that our model effectively learns and generalizes from the training data while minimizing the risk of overfitting. These hyperparameters collectively contribute to the overall performance and reliability of our deep learning model for animal intrusion detection. The Adam optimization

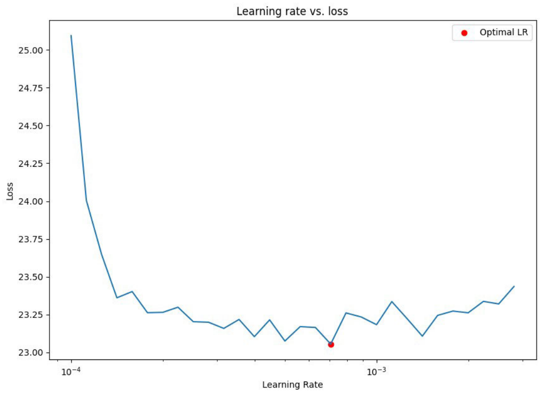


FIGURE 11. Finding best learning rate.

algorithm combines the advantages of two other optimization methods: RMSprop (Root Mean Square Propagation) and Momentum. It's known for its adaptive learning rate and efficient convergence properties. The formula for updating the weights using Adam optimization is as follows:

$$t = 0$$
$$m_0 = \text{vector of zeros for each parameter}$$
$$v_0 = \text{vector of zeros for each parameter}$$

Parameter Updates (for Each Iteration t):

Calculate the loss gradient with respect to the parameters: g_t

$$\begin{aligned} t &= t + 1 \\ m_t &= \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \\ v_t &= \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot (g_t^2) \\ m_t &= \frac{m_t}{1 - \beta_1^t} \\ v_t &= \frac{v_t}{1 - \beta_2^t} \end{aligned} \tag{2}$$

Update the parameters (weights) with the computed moment estimates:

$$\theta_t = \theta_{t-1} - \alpha \cdot \frac{m_t}{\sqrt{v_t} + \epsilon} \tag{3}$$

VI. RESULTS AND DISCUSSION

In our comparative analysis of deep learning models for animal classification, we observed distinctive performance characteristics among several prominent models: DenseNet, Inception, ResNet, MobileNetV2, EfficientNet-B0, EfficientNet-B7 and proposed model, EvoNet. Figure 12 illustrates that the proposed model performed well compared to all other models in terms of accuracy. EvoNet achieved a training accuracy of 96.7% and a validation accuracy of 91.4%, demonstrating that the custom architecture helps the model configure the weights to learn better features in an image. Furthermore, the proposed model's light weightiness is compared to others with only 1.4MB of trainable parameters. MobileNet and EfficientNet-B0 are lighter, but have lower validation accuracy, achieving only 82.49% and 80.73%, respectively as shown in Table 4. On the otherhand, DenseNet and EfficientNet-B7 are large models, and their extensive architecture often leads to overfitting due to limited training data availability. Finally, both Inception and ResNetV2-50 are moderately sized with training accuracy of 88.34% and 91.3%, respectively.

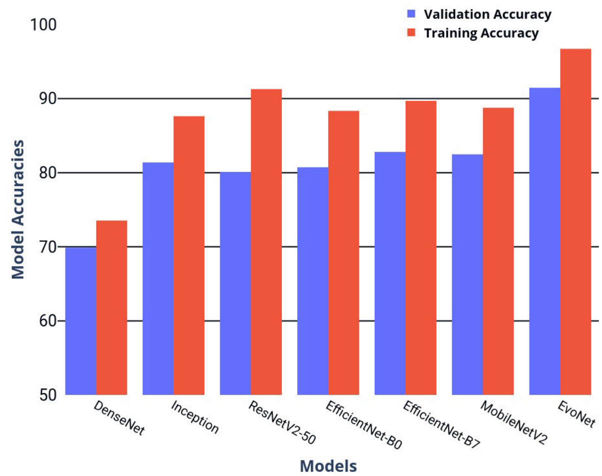


FIGURE 12. Accuracy of the models.

The choice of deep learning model significantly impacts the performance of animal classification tasks. While other models struggle to score accuracy, the proposed model presents a balanced option, offering a practical trade-off between accuracy and resource efficiency. But while choosing the best model for the application in the real-world scenarios, it is vital to consider a broader spectrum of metrics beyond accuracy alone. In practice, the effectiveness of a model is not solely determined by its overall accuracy, but also by its ability to maintain a balance between various performance indicators. Metrics like Precision, Recall, and F1 score are equally instrumental in evaluating a model's capability to distinguish between true and false positives, identify actual positive instances, and strike a harmonious balance between these factors.

Precision, often regarded as a pivotal metric, delves into the model's capacity to make precise positive predictions across different classes. In real-world applications, Precision is of paramount importance, particularly when false positive predictions can have substantial consequences. The Precision can be obtained by,

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

where TP and FP denote True Positive and False positive. Recall, on the other hand, focuses on the model's effectiveness in capturing actual positive instances. It measures the proportion of true positive predictions in relation to all actual positive instances. This metric is vital when it comes to ensuring that the system does not miss any relevant

TABLE 4. Performance metrics of all models.

Parameter	Precision		Recall		F1 Score		Accuracy		Average training time per epoch	Trainable Parameters Size
Model	Macro Average	Weighted Average	Macro Average	Weighted Average	Macro Average	Weighted Average	Training Accuracy	Validation Accuracy		
DenseNet	0.85	0.85	0.83	0.83	0.83	0.83	73.53%	69.87%	78s	367.7MB
Inception	0.85	0.87	0.87	0.85	0.86	0.87	87.63%	81.39%	72s	24.25MB
ResNetV2-50	0.06	0.08	0.04	0.06	0.03	0.04	91.3%	80.073%	61s	17.82MB
EfficientNet-B0	0.84	0.87	0.88	0.86	0.86	0.86	88.34%	80.73%	33s	3.5MB
EfficientNet-B7	0.85	0.88	0.88	0.87	0.86	0.88	89.7%	82.81%	73s	129.71MB
MobileNetV2	0.84	0.86	0.88	0.86	0.87	0.88	88.77%	82.49%	36s	3.52MB
Proposed Model (EvoNet)	0.89	0.88	0.88	0.89	0.88	0.88	95.02%	89.70%	23s	1.34MB
Proposed Model* (EvoNet with Attention Mechanism)	0.90	0.89	0.91	0.90	0.90	0.90	96.72%	91.47%	24s	1.40MB

* is the model that is picked for its best performance and light weightiness.

instances, as missing actual positive cases could have critical implications. The Recall can be calculated by,

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5}$$

where FN is False Negative. F1 score, often considered the balance between Precision and Recall, offers a more comprehensive perspective on the model's overall performance. It takes into account both Precision and Recall, seeking to strike a harmonious equilibrium between minimizing false positives and ensuring the identification of genuine positive instances. The F1 score is particularly useful when there is a need to manage a trade-off between Precision and Recall to optimize the model's efficiency in practical applications.

$$\text{F1 score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

When analyzing these metrics, it becomes clear that ResNetV2-50 is behind the rest of the models. It exhibits lower Precision, Recall, and F1 scores, indicating that it struggles not only in classifying animals accurately but also in minimizing false positives and capturing actual positive instances. Collectively, these metrics reflect the model's limited ability to make precise and comprehensive predictions. ResNet, although it achieved a high accuracy rate, surprisingly exhibits lower Precision, Recall, and F1 scores compared to other models. This discrepancy suggests that, while ResNet excels in classifying animals correctly, it faces challenges related to minimizing false positives and capturing true positives. This discrepancy may arise from the specific nuances of the ResNet model architecture, affecting its ability to strike a balance between Precision and Recall. The remaining models gave good metric scores with respect to their accuracies.

The proposed Model emerges as the top performers in terms of Precision, Recall, and F1 scores. This model not only achieve high accuracy but also excel in making precise positive predictions, capturing actual positive instances effectively, and maintaining a balanced F1 score. The balanced F1 score indicates that they manage the trade-off between Precision and Recall skillfully. This balance positions them as robust choices for animal classification tasks, highlighting the importance of their architecture in achieving well-rounded results. To further evaluate the model's performance, we tested the accuracy of each class with a total of 634 images, and the results are shown in Table 5. The table is formatted as follows. For example, in the class "bear", TP represents the number of images whose true label is "bear" and are detected as 'bear'. TN represents the number of images whose true label is not "bear" and are detected as not "bear". FP represents the number of images whose true label is not "bear" but are identified as "bear". FN represents the number of images whose true label is "bear" but identified as not "bear". TP, TN, FP, and FN have been calculated for all 20 classes to provide a more comprehensive representation of the model's performance.

TABLE 5. Accuracy of the each class for the proposed model.

Class	TP	TN	FP	FN	Acc*	Class	TP	TN	FP	FN	Acc*
Bear	24	602	2	6	0.987	Elephant	31	578	6	19	0.96
Brown bear	34	594	5	1	0.99	Fox	39	594	0	1	0.998
Cattle	31	590	7	6	0.979	Giraffe	25	609	0	0	1
Cheetah	19	593	16	6	0.965	Goat	15	611	0	8	0.987
Deer	36	593	3	2	0.992	Horse	36	594	1	3	0.993
Leopard	32	598	0	4	0.993	Rabbit	36	596	1	1	0.996
Lion	29	605	0	0	1	Raccoon	35	599	0	0	1
Lynx	21	610	2	1	0.995	Sheep	25	600	0	9	0.985
Mule	19	598	14	3	0.973	Tiger	22	609	3	0	0.995
Pig	22	598	12	2	0.977	Zebra	29	601	2	2	0.993

Acc* denotes accuracy.

A. TIME VS ACCURACY GRAPH

To test the efficiency of the model, it was evaluated using 634 test images. These test images comprise images with different orientations, grayscale, and only partial visibility of body parts, as shown in Figure 13. By conducting tests on these images, we can assess how well the model performs in real-world scenarios. As depicted in Figure 14, the proposed model achieved an accuracy of 91.17% within a mere 5 seconds. This demonstrates that the model has effectively learned the features of the images and exhibits high efficiency in real-world scenarios. Following closely in terms of time is MobileNet, which took 7 seconds but failed to provide satisfactory accuracy. EfficientNet-B0, on the other hand, performed commendably, delivering an accuracy of 87.43% within just 8 seconds. However, the poorest performing

FIGURE 13. Example of test images.

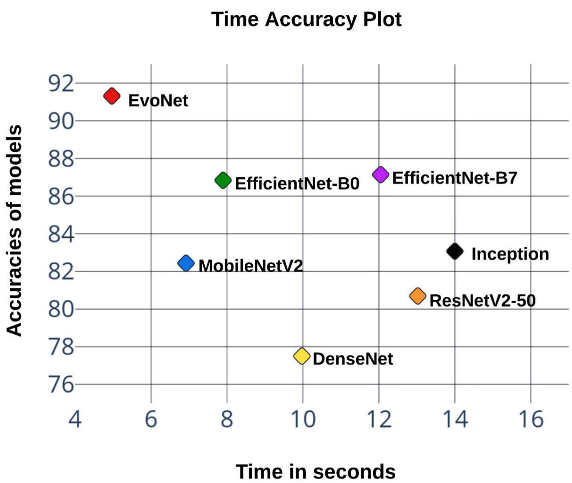


FIGURE 14. Testing image recognition by the CAM on the Boundary poles.

model among all is DenseNet, managing only an accuracy of 77.08% and requiring almost 10 seconds. This is attributed to the heavy parameters and depth of the model.

Overall, the proposed EvoNet model demonstrates strong performance, highlighting the intricate interplay between model architecture, fine-tuning strategies, and the inherent complexities of integrating attention mechanisms within CNN frameworks.

B. IMPLEMENTATION AND CHOOSING DEEP LEARNING MODEL FOR TINYML

TinyML represents a groundbreaking field in the world of machine learning. It acts as a bridge, allowing us to take powerful deep learning and machine learning models and use them in everyday devices. These devices range from small processors like microcontrollers to specialized chips known for their low power usage. TinyML has gained popularity because it's cost-effective, energy-efficient, reliable with data and respects user privacy [27]. Our decision to embrace TinyML in our hardware is all about deploying models on low-power gadgets. This approach ensures not only efficient operation but also reliable results in real-world scenarios, overcoming the limitations usually associated with deep learning models that require powerful GPUs. However, not all microcontrollers can handle TinyML models effectively. Specific requirements need to be met, as outlined in the table below. In our animal intrusion detection system, AI-CAM microcontrollers were utilized for our boundary poles.

TABLE 6. Comparison of basic requirements and AI-CAM specs.

	Architecture	Clock speed	RAM
Basic Requirements	32-bit	50MHz	100KB
AI-CAM	32-bit	240MHz	4MB

Among the deep learning models we evaluated in our research, EvoNet performed the best. The light weight and less computational power requirement with best accuracy makes the EvoNet model a practical choice for TinyML deployment, especially within the AI-CAM microcontroller environment. The operational efficiency of the AI-CAM matches well with the EvoNet model's requirements, ensuring a smooth process for animal intrusion detection while minimizing any unnecessary delays in image predictions. But while deploying the model in current state into the AI-CAM is not possible as the size of the model is around 16.4MB. To reduce the weight of the model it is converted into tensorflow lite version by using two techniques Pruning and Quantization. But the trade-off comes as change in model accuracy.

1) PRUNING

The main goal of the pruning is to eliminate unimportant parameters namely weakest weights at end of each training step. To prune the model TensorFlow Model Optimization Toolkit(tmof) was utilized. Pruning usually follows a schedule that determines when and how much to prune the neural network during training. The scheduler operates on a polynomial decay pattern. It requires an initial sparsity level, a final sparsity level, a starting step for pruning, an ending step for pruning, and the exponent for the polynomial decay. With each step, the tmof removes less important parameters to reach the specified sparsity level.

$$S = (S_e - S_0) \times \left(\frac{t - t_0}{t_e - t_0}\right)^{\alpha}$$

Here, S_0 and S_e denote the initial and final sparsity, respectively, while t_0 and t_e represent the starting and ending steps of pruning and the exponent of the polynomial decay (α). Before pruning begins, importance scores are computed for each parameter in the network. These scores quantify the importance of each parameter in contributing to the overall performance of the network. Common methods for

FIGURE 15. Testing image recognition by the CAM on the Boudary poles.

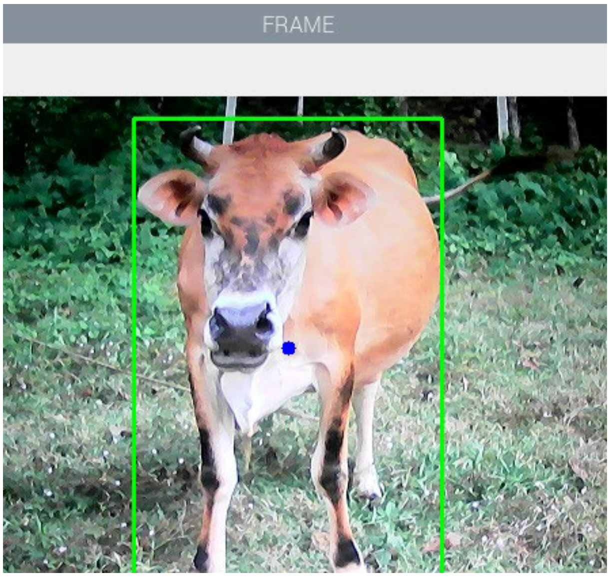


FIGURE 16. Stream from the AI CAM on the rover.

computing importance scores include weight magnitude, gradients, or activation values. Based on the computed importance scores and the pruning schedule, the parameters that are deemed less important are identified for pruning. Once the less important parameters are identified, they are pruned from the network. After pruning, the remaining parameters in the network may need to be adjusted to compensate for the removed connections.

2) QUANTIZATION

Quantization typically involves converting floating-point values to lower precision integer values. This can significantly reduce the model size by 4 times the original size and require less computational power, making it more efficient for deployment on hardware platforms with limited computational resources embedded systems. There are many types of Quantization techniques in that we opted for Post training dynamic range quantization. In this, the model is trained using floating-point precision, and then quantized to lower precision values after training is complete.

By utilizing both pruning and quantization, the model's size is reduced by approximately 10 times the original size, from 16.1 to 1.63MB. However, the trade-off is a reduction in the model's accuracy from 91.4% to 90.1%, which is quite common in the pruning process. Nevertheless, it still provided good performance in animal classification.

3) SHOWCASE OF SYSTEM RESULTS AND TinyML MODEL

As illustrated in Fig. 15, the image classification process using this model yielded highly accurate predictions with impressive percentages. This successful implementation of the TinyML model is a pivotal component of our animal intrusion detection system, which has proven to be highly effective. Fig. 16 provides a r instrumental in allowing users, such as farmers, to monitor their fields and surroundings remotely. The real-time glimpse into our system's capabilities by showcasing a live video stream captured by the rover. This feature is video stream provides valuable insights into the current field conditions, ensuring prompt response to any potential threats. Fig. 16 further emphasizes the system's proficiency by demonstrating the animal detection process via the rover's camera module while it patrols the field. These results collectively reinforce the robustness and efficiency of our integrated IoT and TinyML-based animal intrusion detection system, providing a reliable solution for safeguarding crops from wildlife threats in real-world agricultural scenarios.

VII. CONCLUSION AND FUTURE WORK

In this paper, an innovative system for animal intrusion detection is presented, harnessing the power of deep learning models within a TinyML framework. The results demonstrate that the proposed EvoNet model has performed better while maintaining its lightweight nature, which is important for TinyML models as they operate with limited computational resources. This balance between improved performance and efficient resource utilization is crucial for deploying models in resource constrained environments such as edge devices. Future work in this domain could explore the integration of additional sensors for more comprehensive intrusion detection, as well as the implementation of advanced machine learning techniques to handle various types of intrusions in critical situations. This could involve modifying the base models with different types of attention layers or changes in the backbone structure to efficiently improve the performance of the models. Additionally, the development of a user-friendly interface and mobile application to facilitate real-time monitoring and control is a promising avenue for further system enhancement.

REFERENCES

[1] V. Saiz-Rubio and F. Rovira-Más, "From smart farming towards agriculture 5.0: A review on crop data management," *Agronomy*, vol. 10, no. 2, p. 207, Feb. 2020.

[2] S. Condran, M. Bewong, M. Z. Islam, L. Maphosa, and L. Zheng, "Machine learning in precision agriculture: A survey on trends, applications and evaluations over two decades," *IEEE Access*, vol. 10, pp. 73786–73803, 2022.

[3] V. Krishnaswamy, N. Singh, M. Sharma, N. Verma, and A. Verma, "Application of CRISP-DM methodology for managing human-wildlife conflicts: An empirical case study in India," *J. Environ. Planning Manage.*, vol. 66, no. 11, pp. 2247–2273, Sep. 2023.
[4] A. Srivastava and D. K. Das, "A comprehensive review on the application of Internet of Thing (IoT) in smart agriculture," *Wireless Pers. Commun.*, vol. 122, no. 2, pp. 1807–1837, Jan. 2022.
[5] M. Negi, M. Goswami, and S. Nautiyal, "Agricultural loss due to wildlife: A case study on elephant depredation in a protected area of south India," *J. Social Econ. Develop.*, vol. 25, no. 2, pp. 350–364, Dec. 2023.
[6] A.-J. Garcia-Sanchez, F. Garcia-Sanchez, F. Losilla, P. Kulakowski, J. Garcia-Haro, A. Rodríguez, J.-V. López-Bao, and F. Palomares, "Wireless sensor network deployment for monitoring wildlife passages," *Sensors*, vol. 10, no. 8, pp. 7236–7262, Aug. 2010.
[7] V. Bapat, P. Kale, V. Shinde, N. Deshpande, and A. Shaligram, "WSN application for crop protection to divert animal intrusions in the agricultural land," *Comput. Electron. Agricult.*, vol. 133, pp. 88–96, Feb. 2017.
[8] R. Prodanović, D. Ranč ić, I. Vulić, N. Zorić, D. Bogić ević, G. Ostojić, S. Sarang, and S. Stankovski, "Wireless sensor network in agriculture: Model of cyber security," *Sensors*, vol. 20, no. 23, p. 6747, Nov. 2020.
[9] A. J. Simla, R. Chakravarthi, and L. M. Leo, "Agricultural intrusion detection (AID) based on the Internet of Things and deep learning with the enhanced lightweight M2M protocol," *Soft Comput.*, vol. 2023, pp. 1–12, Mar. 2023.
[10] H. Jawad, R. Nordin, S. Gharghan, A. Jawad, and M. Ismail, "Energy-efficient wireless sensor networks for precision agriculture: A review," *Sensors*, vol. 17, no. 8, p. 1781, Aug. 2017.
[11] K. Balakrishna, F. Mohammed, C. R. Ullas, C. M. Hema, and S. K. Sonakshi, "Application of IoT and machine learning in crop protection against animal intrusion," *Global Transitions Proc.*, vol. 2, no. 2, pp. 169–174, Nov. 2021.
[12] D. Adami, M. O. Ojo, and S. Giordano, "Design, development and evaluation of an intelligent animal repelling system for crop protection based on embedded edge-AI," *IEEE Access*, vol. 9, pp. 132125–132139, 2021.
[13] A. V. Prabu, G. S. Kumar, S. Rajasoundaran, P. P. Malla, S. Routray, and A. Mukherjee, "Internet of Things-based deeply proficient monitoring and protection system for crop field," *Expert Syst.*, vol. 39, no. 5, Jun. 2022, Art. no. e12876.
[14] S. Goel and R. Sharma, "Assessment of crop damage by wild animals and renewable energy interventions—A case study from coastal Odisha, India," in *Proc. 1st Odisha Int. Conf. Electr. Power Eng., Commun. Comput. Technol. (ODICON)*, Jan. 2021, pp. 1–5.
[15] H. Gami, "Movement direction and distance classification using a single PIR sensor," *IEEE Sensors Lett.*, vol. 2, no. 1, pp. 1–4, Mar. 2018.
[16] E. Manor and S. Greenberg, "Custom hardware inference accelerator for TensorFlow lite for microcontrollers," *IEEE Access*, vol. 10, pp. 73484–73493, 2022.
[17] D. Eridani, A. F. Rochim, and F. N. Cesara, "Comparative performance study of ESP-NOW, Wi-Fi, Bluetooth protocols based on range, transmission speed, latency, energy usage and barrier resistance," in *Proc. Int. Seminar Appl. Technol. Inf. Commun. (iSemantic)*, Sep. 2021, pp. 322–328.
[18] Espressif. (2020). *ESP-NOTW—ESP32—ESP-IDF Programming Guide*. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/network/esp_now.html#application-examples
[19] L. Chiaraviglio, N. Blefari-Melazzi, W. Liu, J. A. Gutierrez, J. van de Beek, R. Birke, L. Chen, F. Idzikowski, D. Kilper, P. Monti, A. Bagula, and J. Wu, "Bringing 5G into rural and low-income areas: Is it feasible?" *IEEE Commun. Standards Mag.*, vol. 1, no. 3, pp. 50–57, Sep. 2017.
[20] M. López, S. B. Damsgaard, I. Rodríguez, and P. Mogensen, "Connecting rural areas: An empirical assessment of 5G terrestrial-LEO satellite multi-connectivity," in *Proc. IEEE 97th Veh. Technol. Conf. (VTC-Spring)*, Jun. 2023, pp. 1–5.
[21] B. R. Babu, P. M. Afreed Khan, S. Vishnu, and K. Lova Raju, "Design and implementation of an IoT-enabled remote surveillance rover for versatile applications," in *Proc. IEEE Conf. Interdiscipl. Approaches Technol. Manage. Social Innov. (IATMSI)*, Dec. 2022, pp. 1–6.
[22] A. Kaur, A. Jadli, A. Sadhu, S. Goyal, and A. Mehra, "Cloud based surveillance using ESP32 CAM," in *Proc. Int. Conf. Intell. Technol., Syst. Service Internet Everything (ITSS-IoE)*, Nov. 2021, pp. 1–5.
[23] H. A. Shah, F. Saeed, S. Yun, J.-H. Park, A. Paul, and J.-M. Kang, "A robust approach for brain tumor detection in magnetic resonance images using finetuned EfficientNet," *IEEE Access*, vol. 10, pp. 65426–65438, 2022.
[24] J. Wang, X. He, S. Faming, G. Lu, H. Cong, and Q. Jiang, "A real-time bridge crack detection method based on an improved inception-ResNet-v2 structure," *IEEE Access*, vol. 9, pp. 93209–93223, 2021.
[25] Karthikeya. (2023). *Animal Dataset*. [Online]. Available: https://www.kaggle.com/dsv/7160855
[26] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.
[27] Y. Abadade, A. Temouden, H. Bamoumen, N. Benamar, Y. Chtouki, and A. S. Hafid, "A comprehensive survey on TinyML," *IEEE Access*, vol. 11, pp. 96892–96922, 2023.

KONKALA VENKATESWARLU REDDY is currently pursuing the B.Tech. degree in electronics and communication engineering with Vellore Institute of Technology, Vellore, India. His research interests include robotics and embedded technology, with a specific focus on developing new and innovative devices to improve agricultural robots and drones.

B. S. KARTHIKEYA REDDY is currently pursuing the bachelor's degree in electronics and communication engineering with Vellore Institute of Technology, Vellore, India. He has successfully completing various projects in the fields of embedded systems and deep learning. His research interests include robotics, embedded systems, deep learning algorithms, and the automation of systems.

VEERAPU GOUTHAM (Member, IEEE) received the bachelor's degree in electronics and communication engineering from Sri Krishnadevaraya University, Anantapur, the master's degree in digital communications from the National Institute of Technology Bhopal, India, and the Ph.D. degree from the National Institute of Technology Puducherry, in 2022. He is currently an Assistant Professor Senior Grade 1 with the Department of Communication Engineering, School of Electronics Engineering, Vellore Institute of Technology, Vellore, India. He has more than ten international publications and one granted patent to his credit. His research interests include non-orthogonal multiple access, 5G and beyond networks, intelligent reflecting surfaces, wireless communications, and underwater acoustic sensor networks. He is a Reviewer of many SCI/SCOPUS indexed international journals, such as IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE COMMUNICATIONS LETTERS, IEEE INTERNET OF THINGS JOURNAL.

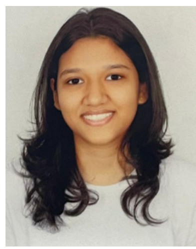
MIRIYALA MAHESH (Member, IEEE) received the B.Tech. degree in electronics and communication engineering from JNTU Hyderabad, the M.Tech. degree in embedded systems from JNTU Anantapur, and the Ph.D. degree in electronics and communication engineering from the National Institute of Technology Puducherry, Karaikal. He has published several articles in peer-reviewed journals, such as IEEE Communications Letters, IEEE Networking Letters, and *Computer Communications*. He has also been granted three patents for his work on IEEE 802.11ah-based wireless local area networks. His research interests include embedded systems, wireless communications, the Internet of Things, 5G networks, NOMA, IEEE 802.11ah, and more.

J. S. NISHA received the Ph.D. degree in electronics and communication engineering from the National Institute of Technology Trichy, India. She is currently an Assistant Professor (Grade 1) with the Department of Sensor and Biomedical Technology, School of Electronics Engineering, Vellore Institute of Technology, Vellore. Her research interests include medical image processing, machine learning, and deep learning. More specifically, cancer detection from colonoscopy images.

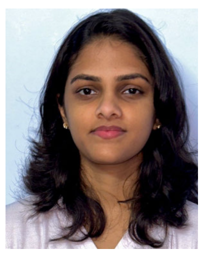
GOPINATH PALANISAMY received the Ph.D. degree from the National Institute of Technology, Tiruchirappalli, India. He is currently an Assistant Professor with the School of Electronics, Vellore Institute of Technology, Vellore. His research interests include signal and image processing. He is particularly working on fundus image analysis and brain tumor detection.

MALLIKARJUNA GOLLA was born in Nellore, Andhra Pradesh, India. He received the B.Tech. and M.Tech. degrees in electrical and electronics engineering from Jawaharlal Nehru Technological University Hyderabad (JNTUH), in 2010 and 2014, respectively, and the Ph.D. degree from the National Institute of Technology Puducherry (NITPY), Karaikal, in 2022. From 2013 to 2017, he was an Assistant Professor with the TKR College of Engineering Technology, Hyderabad. He has about five years of research experience as a JRF, a SRF, and a Postdoctoral Research Associate for DST-funded projects with NITPY and IIT Roorkee. He is currently a Senior Assistant Professor with the School of Electrical Engineering, Vellore Institute of Technology (VIT), Vellore Campus, Tamil Nadu, India. His research interests include grid integration of renewable energy sources, active power filters, artificial neural networks, microgrids, smart grids, and power conditions with power flow management.

SWETHA PURUSHOTHAMAN is currently pursuing the B.Tech. degree in electronics and communication engineering with Vellore Institute of Technology, Vellore, India. Her research interests include machine learning and communication, with a specific focus on developing new and innovative devices to improve agricultural productivity and communication field.



KATANGURE RITHISHA REDDY is currently pursuing the bachelor's degree in electronics and communication engineering with Vellore Institute of Technology, Vellore, India. Her research interests include sensor technology, the Internet of Things (IoT), as well as embedded systems.



VARSHA RAMKUMAR is currently pursuing the B.Tech. degree in electronics and communication engineering with specialization in biomedical engineering with Vellore Institute of Technology, Vellore, India. Her research interests include medical and agricultural instrumentation, with a specific focus on developing new and innovative devices to improve patient care and agricultural productivity.