A Novel Approach To Detect And Protect The Farm And Suitable Framework Using Iot Based

S. Thirisha[1], G. Narmatha[2], K. Kalaiyarasan[3]
[1, 2, 3]Dept of Computer Science And Engineering
[1, 2, 3] TRICHY ENGINEERING COLLEGE

Abstract- *Human – wildlife conflicts arising from habitat encroachment and deforestation have led to an alarming increase in crop raiding, causing substantial losses to farmers and posing risks to human safety. Conventional methods, ranging from lethal measures to non-lethal deterrents, have proven insufficient, often leading to environmental, high costs, and limited effectiveness. In response to these challenges, this project proposes novel Integrated Wildlife Management System that combines Computer Vision, leveraging Temporal Convolutional Networks (TCN), for precise animal species detection and recognition, with a targeted ultrasound emission technique for species -specific repelling. The system, driven by edge computing, ensures real-time responsiveness to mitigate crop raiding. The workflow commences with the activation of the camera by the edge computing device, triggering the deployment of an advanced Animal Intrusion Detection Model. By leveraging cutting-edge technology, the proposed solution seeks to strike a balance between protecting crops and minimizing environmental impact. This project contributes to the ongoing discourse on human-wildlife conflict resolution and highlights the potential of technology-driven solutions in fostering coexistence between agriculture and biodiversity.*

I. INTRODUCTION

Agriculture has seen many revolutions, whether the domestication of animals and plants a few thousand years ago, the systematic use of crop rotations and other improvements in farming practice a few hundred years ago, or the “green revolution” with systematic breeding and the widespread use of man-made fertilizers and pesticides a few decades ago.

Agriculture is undergoing a fourth revolution triggered by the exponentially increasing use of information and communication technology (ICT) in agriculture. Autonomous, robotic vehicles have been developed for farming purposes, such as mechanical weeding, application of fertilizer, or harvesting of fruits. The development of unmanned aerial vehicles with autonomous flight control, together with the now that allow farmers to differentiate between plant diseases based on optical information. Virtual fence technologies allow cattle herd management based on remote-sensing signals and sensors or actuators attached to the livestock.

Taken together, these technical improvements constitute a technical revolution that will generate disruptive changes in agricultural practices. This trend holds for farming not only in developed countries but also in developing countries, where deployments in ICT are being adopted at a rapid pace and could become the game-changers in the future.

1.1 PROBLEMS DEFINITION

Most farmers have challenges related to crop damage due to wildlife pests. Animal intrusion is a major threat to the productivity of the crops, which affects food security and reduces the profit to the farmers. Organic farmers have additional challenges because they cannot use chemical controls which are sometimes the most effective and efficient options. A need has been identified for alternative pest control appropriate for traditional and organic farmers.

Risks associated with faecal matter in the field are the highest. For even just one instance of faecal matter, the risk of contamination is moderate. Widespread evidence of faecal contamination is very high risk and would justify marking the contaminated area and creating a no-harvest buffer zone around the area where significant faeces was found. Animal activity on the farm can be a huge risk to food safety when growing fruits and vegetables, which is why preharvest wildlife scouting is so important. Existing methods like fencing can be an effective deterrent, but it may not be practical for larger farms; however, small portions of fencing may direct animals around high value or sensitive crops to other areas and electric fences are no longer efficient in solving such conflicts, to protect their crops from getting damaged because of animal intrusions, farmers have been using electric fences around their fields and areas where the fencing don’t prove efficient, farmers prefer to stay up all night and guard their fields from animal intrusions. Nuisance permits may be another option, but check with local Department of Environmental Conservation (DEC) or the National Resources Conservation Service (NRCS) before choosing this method. Practices like these have done more harm than good for us and in extreme cases; it has even coasted lives of both man and animals. So, we decided to come up with a smarter solution that could protect the crops from animal intrusions without causing any harm to the wildlife.

1.2 Artificial Intelligence

An AI is a computer system that is able to perform tasks that ordinarily require human intelligence. These artificial intelligence systems are powered by machine learning. Many of them are powered by machine learning, some of them are powered by specifically deep learning, and some of them are powered by very boring things like just rules.

1.2.1 Machine Learning in a Nutshell

Machine learning allows experts to “train” a machine by making it analyze massive datasets. The more data the machine analyzes, the more accurate results it can produce by making decisions and predictions for unseen events or scenarios. Machine learning models need structured data to make accurate predictions and decisions. If the data is not labeled and organized, machine learning models fail to comprehend it accurately, and it becomes a domain of deep learning.

1.2.2 Deep Learning in a Nutshell

The deep learning methodology designs a sophisticated learning model based on neural networks inspired by the human mind. These models have multiple layers of algorithms called neurons. They continue to improve without human intervention, like the cognitive mind that keeps improving and evolving with practice, revisits, and time. Deep learning models are mainly used for classification and feature extraction. For instance, deep models feed on a dataset in facial recognition. The model creates multidimensional matrices to memorize each facial feature as pixels. When you ask it to recognize a picture of a person it was not exposed to, it easily recognizes it by matching limited facial features.

1.2.3 Convolutional Neural Network

A Convolution Neural Network (Convent/CNN) is a Deep Learning algorithm that can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and be able to differentiate one from the other. The pre-processing required in a Convent is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, Convents have the ability to learn these filters/characteristics. The architecture of a Convent is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

1.2.4 Convolution Layer + Rely

This layer is the first layer that is used to extract the various features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size Mom. By sliding the filter over the input image, the dot product is taken between the filter and the parts of the input image with respect to the size of the filter (Mom).

The output is termed as the Feature map which gives us information about the image such as the corners and edges. Later, this feature map is fed to other layers to learn several other features of the input image.

1.2.5 Pooling Layer

In most cases, a Convolution Layer is followed by a Pooling Layer. The primary aim of this layer is to decrease the size of the convolved feature map to reduce computational costs. This is performed by decreasing the connections between layers and independently operating on each feature map. Depending upon the method used, there are several types of Pooling operations. In Max Pooling, the largest element is taken from the feature map. Average Pooling calculates the average of the elements in a predefined sized Image section. The total sum of the elements in the predefined section is computed in Sum Pooling.

1.2.6 Fully Connected Layer

The Fully Connected (FC) layer consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers. These layers are usually placed before the output layer and form the last few layers of a CNN Architecture. In this, the input image from the previous layers is flattened and fed to the FC layer. The flattened vector then undergoes a few more FC layers where mathematical operations usually take place. In this stage, the classification process begins to take place.

1.2.7 Dropout

Usually, when all the features are connected to the FC layer, it can cause over fitting in the training dataset. Over fitting occurs when a particular model works so well on the training data causing a negative impact on the model’s performance when used on new data. To overcome this problem, a dropout layer is utilized wherein a few neurons are dropped from the neural network during the training process resulting in reduced size of the model. On passing a dropout of 0.3, 30% of the nodes are dropped out randomly from the neural network.

1.3 Aim and Objective

The aim of the project is to develop an integrated wildlife defence system utilizing AI computer vision and ultrasound emissions technology to effectively detect, repel, and mitigate animal intrusions in agricultural environments, thereby reducing crop damage and increasing agricultural productivity.

- To implement AI computer vision algorithms for real-time animal detection and species recognition.
- To integrate species-specific ultrasound emissions to deter identified animals from agricultural fields.
- To design a user-friendly interface for farmers to monitor and manage the wildlife defense system.
- To develop alert mechanisms to notify farmers of potential wildlife intrusions via SMS or other means.
- To evaluate the effectiveness and efficiency of the integrated system through field trials and performance analysis.
- To provide scalable and adaptable solutions to address varying wildlife- related challenges in agriculture.

1.4 Scope

The project encompasses the development and implementation of AI- based animal detection algorithms, focusing on real-time identification and recognition of wildlife species using computer vision techniques. This includes the deployment of advanced machine learning models to analyze images or video streams captured from agricultural fields and accurately detect the presence of animals. Additionally, the scope extends to integrating species- specific ultrasound emission devices into the system architecture. These devices will be strategically deployed to emit ultrasound frequencies known to deter identified animals, thereby serving as a proactive defense mechanism against wildlife intrusions in agricultural areas. Moreover, the project involves the design and development of a user-friendly interface tailored for farmers. This interface will facilitate easy system configuration, monitoring of wildlife detection activities, and receiving alerts in case of potential intrusions. Implementing alert mechanisms, such as SMS notifications, will enable timely communication of detected animal intrusions to farmers, empowering them to take prompt action to protect their crops. The evaluation and testing phase will ensure thorough assessment of the integrated system's effectiveness, efficiency, and reliability under diverse environmental conditions and wildlife scenarios. Scalability and adaptability considerations will be addressed to ensure that the system can accommodate different agricultural settings, varying wildlife species, and evolving challenges over time.

II. LITERATURE REVIEW

2.1. Title: Creating Alert Messages Based on Wild Animal Activity Detection Using Hybrid Deep Neural Network

The issue of animal attacks is increasingly concerning for rural populations and forestry workers. To track the movement of wild animals, surveillance cameras and drones are often employed. However, an efficient model is required to detect the animal type, monitor its locomotion and provide its location information. Alert messages can then be sent to ensure the safety of people and foresters. While computer vision and machine learning-based approaches are frequently used for animal detection, they are often expensive and complex, making it difficult to achieve satisfactory results. This paper presents a Hybrid Visual Geometry Group (VGG) −19+ Bidirectional Long Short-Term Memory (Bi-LSTM) network to detect animals and generate alerts based on their activity. These alerts are sent to the local forest office as a Short Message Service (SMS) to allow for immediate response. The proposed model exhibits great improvements in model performance, with an average classification accuracy of 98%, a mean Average Precision (mAP) of 77.2%, and a Frame Per Second (FPS) of 170. The model was tested both qualitatively and quantitatively using 40,000 images from three different benchmark datasets with 25 classes and achieved a mean accuracy and precision of above 98%. This model is a reliable solution for providing accurate animal-based information and protecting human lives.

2.2. Title: Accurate and Fast Animal Species Detection System for Embedded Devices

Encounters between humans and wildlife often lead to injuries, especially in remote wilderness regions, and highways. Therefore, animal detection is a vital safety and wildlife conservation component that can mitigate the negative impacts of these encounters. Deep learning techniques have achieved the best results compared to other object detection techniques; however, they require many computations and parameters. A lightweight animal species detection model based on YOLOv2 was proposed. It was designed as a proof of concept of and as a first step to build a real-time mitigation system with embedded devices. Multi-level features merging is employed by adding a new pass- through layer to improve the feature extraction ability and accuracy of YOLOv2. Moreover, the two repeated 3 × 3 convolutional layers in the seventh block of the YOLOv2 architecture are removed to reduce computational complexity, and thus increase detection speed without reducing accuracy. Animal species detection methods based on regular Convolutional Neural Networks (CNNs) have been widely applied; however, these methods are difficult to adapt to geometric variations of animals in images. Thus, a modified YOLOv2 with the addition of deformable convolutional layers (DCLs) was proposed to resolve this issue. Our experimental results show that the proposed model outperforms the original YOLOv2 by 5.0% in accuracy and 12.0% in speed. Furthermore, our analysis shows that the modified YOLOv2 model is more suitable for deployment than YOLOv3 and YOLOv4 on embedded devices.

2.3. Title: Animal Intrusion Detection Using Deep Learning

One of the researcher's interests and challenges is animal identification techniques. There are several challenges that researchers in this field confront that limit detection effectiveness and efficiency, such as picture illumination variation, animal occlusion, color similarity of animal colors with backdrop environment, and so on. The purpose of this study is to detect and classify multi-label images of mammals, which we propose to do using the Single Shot Multi-Box Detector (SSD) and the Mobile Net v1 coco_2017 model. Another objective is to locate and identify several items (animals) from the Mammal category in digital photos. Based on deep learning technology, the recommended SSD is regarded as a more accurate, rapid, and efficient technique to recognize objects of various sizes. We utilized 2000 pictures in the network taken from standard datasets (such as Caltech 101) and the net in this proposal. The SSD framework enhances Convolution Neural Network (CNN) detection and identification operations. During the prediction phase, the network assigns scores to the existence of each object class and draws a box around each object in the picture. Each box includes a name that defines the kind of object, and the score denotes the likelihood of the object's association to that category. During the procedure, boxes are changed to get the best fit to the form of the item. The experimental findings of this work demonstrated the efficacy of identifying and detecting animals even when light, position, and occlusion were varied. The detection and classification accuracy can reach 98.7%.

Title: Design, Development and Evaluation of an Intelligent Animal Repelling System for Crop Protection Based on Embedded Edge-AI

In recent years, edge computing has become an essential technology for real- time application development by moving processing and storage capabilities close to end devices, thereby reducing latency, improving response time and ensuring secure data exchange. In this work, we focus on a Smart Agriculture application that aims to protect crops from ungulate attacks, and therefore to significantly reduce production losses, through the creation of virtual fences that take advantage of computer vision and ultrasound emission. Starting with an innovative device capable of generating ultrasound to drive away ungulates and thus protect crops from their attack, this work provides a comprehensive description of the design, development and assessment of an intelligent animal repulsion system that allows to detect and recognize the ungulates as well as generate ultrasonic signals tailored to each species of the ungulate. Taking into account the constraints coming from the rural environment in terms of energy supply and network connectivity, the proposed system is based on Iota platforms that provide a satisfactory compromise between performance, cost and energy consumption. More specifically, in this work, we deployed and evaluated various edge computing running real-time object detector with custom-trained models to identify the most suitable animal recognition HW/SW platform to be integrated with the ultrasound generator.

2.5. Title: Design and Analysis of Wireless Sensor Networks for Animal Tracking in Large Monitoring Polar Regions Using Phase-Type Distributions and Single Sensor Model

In recent years, edge computing has become an essential technology for real- time application development by moving processing and storage capabilities close to end devices, thereby reducing latency, improving response time and ensuring secure data exchange. In this work, we focus on a Smart Agriculture application that aims to protect crops from ungulate attacks, and therefore to significantly reduce production losses, through the creation of virtual fences that take advantage of computer vision and ultrasound emission. Starting with an innovative device capable of generating ultrasound to drive away ungulates and thus protect crops from their attack, this work provides a comprehensive description of the design, development and assessment of an intelligent animal repulsion system that allows to detect and recognize the ungulates as well as generate ultrasonic signals tailored to each species of the ungulate. Taking into account the constraints coming from the rural environment in terms of energy supply and network connectivity, the proposed system is based on Iota platforms that provide a satisfactory compromise between performance, cost and energy consumption.

III. SYSTEM ANALYSIS

3.1 Existing System

Wild animals are a special challenge for farmers throughout the world. Animals such as deer, wild boars, rabbits, moles, elephants, monkeys, and many others may cause serious damage to crops. They can damage the plants by feeding on plant parts or simply by running over the field and trampling over the crops. Therefore, wild animals may easily cause significant yield losses and provoke additional financial problems. Another aspect to consider is that wild animal crop protection requires a particularly cautious approach. In other words, while utilizing his crop production, every farmer should be aware and take into consideration the fact that animals are living beings and need to be protected from any potential suffering.

Farmers Traditional Approach

There are different existing approaches to address this problem which can be lethal and non-lethal firecrackers, bright lights, fire, beating drums, and dogs

Agricultural fences

Fencing is a popular wild animal protection practice for that can last for many years. Agricultural fences are quite an effective wild animal protection technology.

Wire fences

Constructed of metal wires woven together forming a physical barrier.

The fences are effective, long lasting, and require relatively little maintenance.

Plastic fences

Polypropylene fences are generally less expensive and easier to install and repair than other types. Additionally, these fences are widely acceptable and meet various regulations.

3.2 Proposed System

This project presents an integrated system aimed at addressing wildlife- related challenges in agriculture by combining advanced AI technologies with targeted ultrasound emissions and farmer alert mechanisms. The system utilizes Temporal Convolutional Network (TCN) and WildNet for accurate detection and recognition of animal species, coupled with species-specific ultrasound emissions for repelling identified animals. Additionally, the system incorporates an alert system to notify farmers via SMS when potential threats are detected.

AI Computer Vision Module

The TCN and WildNet form the core of the computer vision module, offering real-time video analysis for accurate detection and recognition of animal species. This component processes high-resolution imagery captured by cameras deployed in agricultural areas, enhancing the system's ability to identify and classify wildlife accurately.

Ultrasound Emission Module

The Ultrasound Emission module integrates species-specific ultrasound devices into the system. Upon identification of a threat by the computer vision module, the corresponding ultrasound emission is triggered, aiming to repel the detected species. Careful design ensures effectiveness against targeted species while minimizing impact on non-targeted entities.

Farmer Alert System

The alert system responds to potential threats by initiating notifications to farmers. Using SMS alerts, pre-registered farmers receive real-time information about the identified species and potential risks. This immediate and personalized communication enhances the farmers' ability to respond promptly to wildlife intrusions.

User Interface

The user interface provides a user-friendly platform for farmers to interact with the system. Farmers can configure parameters, monitor real-time data, and receive alerts through this interface. It includes options for adjusting sensitivity levels, updating species recognition models, and managing contact information for SMS alerts, empowering farmers with control and insights.

IV. SYSTEM SPECIFICATION

4.1 Hardware Requirements

Processor: Intel Core i5 or higher (or equivalent AMD processor)
RAM: 8 GB or more
Storage: SSD with at least 256 GB of storage capacity (for faster read/write speeds)
Graphics Card: Integrated graphics card or dedicated GPU (optional for GPU-accelerated tasks in TensorFlow)
Display: Monitor with a minimum resolution of 1920x1080 pixels.

4.2 Software Requirements

Operating System: Windows 10 (64-bit)
Programming: Python 3.8
Web Framework: Flask
IDE: PyCharm (or any other Python-compatible IDE)
Database Management System: MySQL
Local Server: WampServer
Web Browser: Google Chrome, Mozilla Firefox, or Microsoft Edge.
SMS API: pay4sms.in (for sending SMS notifications).
SMTP: Simple Mail Transfer Protocol for sending email notification.

V. SYSTEM ARCHITECTURE

Project Description

The Wildlife Defense Web App serves as a central hub for managing and monitoring wildlife defense activities in agricultural settings. Developed using Python with Flask as the web framework, MySQL for database management, and Bootstrap for a user-friendly interface, this application provides administrators and farmers with access to essential functionalities. It features secure user authentication, enabling role-based access control for different users. Administrators can train and build WildNet models using an intuitive interface, upload datasets, and deploy trained models to edge devices for real- time detection. Additionally, administrators can configure ultrasound repellents for each species and monitor wildlife activities across fields via a map interface. The app includes a robust alert system that notifies users in real-time upon detecting wildlife intrusion, with manual controls available for immediate response.The End User Interface comprises two main sections: the Admin Interface and the Farmer Interface. The Admin Interface allows administrators to securely log in, upload datasets, build and train Wild Net models, deploy models to edge devices, configure ultrasound repellents, and update models as needed. On the other hand, the Farmer Interface enables farmers to register, log in, access real-time monitoring dashboards displaying live camera feeds from their fields, view detailed reports on wildlife intrusion incidents, and utilize manual controls for immediate action. The Wild Net Model: Build and Train module encompasses dataset collection, pre-processing, animal segmentation, feature extraction, classification, and model deployment Convolution Neural Network (CNN) is then designed for classification, trained, and deployed within the Wildlife Defense Web App Firmware for real-time use. The Ultrasound Sound Integration module maintains an ultrasound sound library corresponding to different animal species.

Developing a system like the Wildlife Defense Web App involves several stages of software development. Below is a generalized roadmap for the development process:

Requirements Gathering

Collaborate with stakeholders (administrators, farmers, wildlife experts) to understand their needs and expectations. Define the features and functionalities required for each user interface (Admin and Farmer).

System Architecture Design

Design the overall system architecture, outlining the different modules and their interactions. Determine the technologies and frameworks to be used (Flask, MySQL, Bootstrap, etc.). Create a database schema to store user data, training data, and system configurations.

Frontend Development

Develop the user interfaces based on the design specifications. Implement responsive design using Bootstrap for a user-friendly experience across devices. Integrate user authentication mechanisms for secure access.

Backend Development

Set up the Flask web framework to handle server-side logic. Implement user authentication and authorization for Admin and Farmer roles. Develop functionalities for dataset upload, WildNet model training, deployment, and updates.

5.1 DATABASE IMPLEMENTATION

Set up a MySQL database to store user data, training datasets, WildNet models, and configuration settings. Implement database queries for data retrieval, insertion, and updates.

Ultrasound Sound Integration

Develop the Ultrasound Sound Library module, allowing admins to manage ultrasound emissions. Implement controls for frequency and intensity adjustments.

Alerts or Notification Module

Develop the module to generate alerts based on system events. Configure multiple channels for notifications (email, SMS, in-app). Include detailed information in alerts for situational awareness.

Setting up Flask Project Structure: Initialize a new Flask project. Create directories for templates, static files, and modules.

Database Setup:

Install MySQL and set up a database for storing user data, datasets, model configurations, etc.
Use a library like Flask-MySQLdb to interact with the MySQL database from Flask.

5.2 WILDNET MODEL MANAGEMENT

Create routes for uploading datasets, preprocessing images, training models, and deploying them. Use libraries like Tensor Flow or PyTorch for building and training the WildNet model. Store trained models in a secure location and update model configurations in the database.

Server Setup:

Choose a hosting provider and set up a server instance. Install necessary software like Python, Flask, MySQL, etc.

Deploy Flask Application:

Deploy the Flask application to the server using a WSGI server like Unicorn. Configure the server to serve static files and handle HTTPS requests.

Database Configuration:

Configure the MySQL database connection settings in the Flask application. Ensure proper security measures are in place, such as password protection and firewall rules.

WildNet Model: Build and Train

This module combines dataset collection, pre-processing, advanced image processing techniques, and CNN-based classification to build and train the WildNet model. The integration of the trained model into the Wildlife Defense Web App Firmware enhances the system's capability to detect and categorize wildlife, providing a robust defense mechanism for agricultural fields.

5.3 ANIMAL SEGMENTATION

Therefore, in this module, Region Proposal Network (RPN) generates RoIs by sliding windows on the feature map through anchors with different scales and different aspect ratios. Animal detection and segmentation method based on improved RPN. RPN is used to generate RoIs, and RoI Align faithfully preserves the exact spatial locations. These are responsible for providing a predefined set of bounding boxes of different sizes and ratios that are going to be used for reference when first predicting object locations for the RPN.

Animal Classification

DCNN algorithms were implemented to automatically detect and reject improper animal images during the classification process. This will ensure proper training and therefore the best possible performance.

Build and Train the Wild Net

The Wild Net model is constructed based on the defined CNN architecture. Parameters such as epochs, learning rates, and batch sizes are configured for optimal training. The model undergoes the training process, adjusting weights to minimize loss and maximize accuracy.

Deploy the Wild Net Model

Once training is successful, the trained WildNet model is deployed within the Wildlife Defense Web App Firmware. This ensures that the model is ready for real-time use in detecting and classifying wildlife in agricultural fields.

5.4 ULTRASOUND SOUND INTEGRATION

Ultrasound Sound Library

System maintains an ultrasound sound library that corresponds to different animal species. This library serves as a repository for various ultrasound emissions, each tailored to repel a specific type of wildlife. Admins can upload, update, and manage the ultrasound sound database through the configuration panel.

Frequency and Intensity Controls

Admins have granular control over the frequency and intensity of ultrasound emissions through the configuration panel. Fine-tuning these parameters allows for customization based on the specific sensitivities of different animal species, optimizing the deterrent effect.

Animal Intrusion Predictor

The Animal Intrusion Predictor module with Temporal Convolutional Networks (TCN) is designed to identify and predict the presence of animals in input images, live video streams, or recorded videos. TCN is a type of neural network architecture commonly used for sequence modeling, making it well- suited for tasks involving temporal dependencies, such as video analysis.

Temporal Convolutional Networks (TCN)

TCN is a crucial component for handling temporal dependencies in video data. It excels at capturing patterns and features across sequential frames, making it suitable for video-based tasks.

Animal Detection

The module is trained to detect and recognize various types of animals. This involves learning features and patterns from input data to make accurate predictions.

Prediction Output

The system outputs predictions regarding the presence of animals, providing information such as the type of animal detected and a confidence score.

Animal Identification

Along with detecting the presence of animals, the system identifies the specific type of animal in the input data. This could include common animals like dogs, cats, birds, etc.

Confidence Scores

For each prediction, the module provides a confidence score indicating the system's level of certainty regarding the detected animal. This score can be useful in decision-making processes.

Real-time Visualization

The predictions are dynamically visualized in real-time on the Wildlife Defence Web App or monitoring interface. This feature enables users to actively observe and respond to animal intrusions as they occur.

Monitoring and Visualizing

The system works in real time detect the animals in the field, in addition the farmers can access the view of their fields remotely. Type of animal and also the count can be given. The animal recognition module will share the data over the cloud regularly through a Wi-Fi connection.

Alerts or Notification

The Alerts or Notifications module is seamlessly integrated with the Ultrasound Emission and Animal Intrusion Predictor modules. This integration ensures that alerts are triggered in response to ultrasound emissions, wildlife intrusions, and other relevant events detected by the system. Notifications are delivered through multiple channels for accessibility.

5.5 ALGORITHM

1. Initialize the microcontroller / processor (Raspberry Pi / ESP32).
2. Initialize all connected sensors:
   - Passive Infrared (PIR) motion sensor
   - Ultrasonic distance sensor
   - Camera module
3. Initialize communication modules:
   - Wi-Fi / GSM / IoT cloud connection
4. Load the pre-trained Convolutional Neural Network (CNN) model for animal classification.
5. Initialize actuators:
   - Buzzer / speaker
   - LED floodlight
   - Optional deterrent devices (sprinkler, ultrasonic sound emitter).
6. Set system state to **Monitoring Mode**.

1) Step 2: Continuous Monitoring

- Continuously monitor the PIR sensor for motion detection.
- Read ultrasonic sensor values to measure distance and validate movement.
- If no motion is detected:
  - Continue monitoring (go to Step 7).

2) Step 3: Motion Verification

- If PIR sensor detects motion:
  - Verify motion using ultrasonic distance change.
- If motion is confirmed:
  - Trigger image acquisition (go to Step 12).
- If motion is not confirmed:
  - Ignore the event and return to monitoring mode.

3) Step 4: Image Acquisition

- Activate the camera module.
- Capture one or more image frames of the detected object.
- Store the captured image temporarily for processing.

4) Step 5: Image Preprocessing

- Resize the image to the input size required by the CNN model.
- Convert the image to grayscale (if required).
- Normalize pixel values to improve classification accuracy.
- Apply noise reduction and contrast enhancement if necessary.

5) Step 6: Animal Detection and Classification

- Feed the preprocessed image into the CNN model.
- Extract feature maps through convolution and pooling layers.
- Classify the object into one of the predefined classes:
  - Animal (e.g., elephant, deer, dog, cattle)
  - Human
  - Unknown / non-living object
- Obtain classification confidence score.

Step 7: Decision Making

- If the classified object is **human**:
  - Ignore intrusion and return to monitoring mode.
- If the classified object is **non-animal**:
  - Ignore and return to monitoring mode.
- If the classified object is **animal** and confidence ≥ threshold:
  - Confirm animal intrusion.
- If confidence < threshold:
  - Capture additional frames and repeat Steps 16–23.

6) Step 8: Intrusion Prevention Action

- Activate deterrent mechanisms:
  - Turn ON high-intensity LED lights.
  - Activate buzzer or ultrasonic sound generator.
  - Trigger water sprinkler (if available).
- Maintain deterrent action for a predefined duration.

7) Step 9: Alert Generation

- Generate an alert message containing:
  - Timestamp
  - Location
  - Type of animal detected
- Send alert to:
  - Farmer / user mobile application
  - Forest department / monitoring authority
  - Cloud dashboard

8) Step 10: Data Logging

- Store intrusion event data in cloud or local database:
  - Image
  - Animal type
  - Time and date
  - Action taken

9) Step 11: System Reset

- Deactivate deterrent mechanisms after timeout.
- Clear temporary image data.
- Reset system state to **Monitoring Mode**.
- Continue continuous monitoring (return to Step 7).

B. Pseudocode Representation

```
Initialize System
Load CNN Model
While (System ON)
    Read PIR Sensor
    Read Ultrasonic Sensor
    If Motion Detected Then
        Capture Image
        Preprocess Image
        Classify Image using CNN
```

```
        If Object == Animal And
Confidence ≥ Threshold Then
            Activate Deterrents
            Send Alert
            Log Data
        End If
    End If
End While
```

C. Key Features of the Algorithm

- Multi-stage verification reduces false alarms
- CNN-based classification improves accuracy
- Non-lethal and ethical prevention approach
- Real-time alerting and data logging
- Suitable for remote and rural deployment

APPENDIX

Packages
from flask import Flask, render_template, Response, redirect, request, session, abort, url_for
from camera import VideoCamera import argparse
import cv2
import pandas as pd import random import seaborn as sns
import matplotlib.pyplot as plt import mysql.connector
Training
Preprocessing
path_main = 'static/dataset'
for fname in os.listdir(path_main): dimg.append(fname)
#list_of_elements = os.listdir(os.path.join(path_main, folder)) #resize
'''img = cv2.imread('static/data/'+fname) rez = cv2.resize(img, (400, 300))
cv2.imwrite("static/dataset/"+fname, rez)''' img = cv2.imread('static/dataset/'+fname)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2.imwrite("static/trained/g_"+fname, gray)
##noice
img = cv2.imread('static/trained/g_'+fname)
dst = cv2.fastNlMeansDenoisingColored(img, None, 10, 10, 7, 15) fname2='ns_'+fname
cv2.imwrite("static/trained/"+fname2, dst) #Binarization
image = cv2.imread('static/dataset/'+fname) original = image.copy()
kmeans = kmeans_color_quantization(image, clusters=4)
# Convert to grayscale, Gaussian blur, adaptive threshold
gray = cv2.cvtColor(kmeans, cv2.COLOR_BGR2GRAY)
blur = cv2.GaussianBlur(gray, (3,3), 0)
thresh = cv2.adaptiveThreshold(blur,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY_INV,21,2)
# Draw largest enclosing circle onto a mask
mask = np.zeros(original.shape[:2], dtype=np.uint8)
cnts = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
cnts = cnts[0] if len(cnts) == 2 else cnts[1]
cnts = sorted(cnts, key=cv2.contourArea, reverse=True)
for c in cnts:
((x, y), r) = cv2.minEnclosingCircle(c) cv2.circle(image, (int(x), int(y)), int(r), (36, 255, 12), 2)
cv2.circle(mask, (int(x), int(y)), int(r), 255, -1) break
# Bitwise-and for result
result = cv2.bitwise_and(original, original, mask=mask)
result[mask==0] = (0,0,0)
cv2.imshow('thresh', thresh)
cv2.imshow('result', result) cv2.imshow('mask', mask)
cv2.imshow('kmeans', kmeans)
cv2.imwrite("static/trained/bb/bin_"+fname, thresh)
#RPN
img = cv2.imread('static/trained/g_'+fname)
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY) ret, thresh = cv2.threshold(gray,0,255,cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU) kernel = np.ones((3,3),np.uint8)
opening = cv2.morphologyEx(thresh,cv2.MORPH_OPEN, kernel, iterations = 2)
# sure background area
sure_bg = cv2.dilate(opening,kernel,iterations=3) # Finding sure foreground area
dist_transform = cv2.distanceTransform(opening,cv2.DIST_L2,5)
ret, sure_fg = cv2.threshold(dist_transform,1.5*dist_transform.max(),255,0) # Finding unknown region
sure_fg = np.uint8(sure_fg)
segment = cv2.subtract(sure_bg)

VI. CONCLUSION & FUTURE WORK

Agricultural farm security is widely needed technology nowadays. In order to accomplish this, a vision-based system is proposed and implemented using Python and OpenCV and developed an Animal Repellent System to blow out the animals. The implementation of the application required the design and development of a complex system for intelligent animal repulsion, which integrates newly developed software components and allows to recognize the presence and species of animals in real time and also to avoid crop damages caused by the animals. Based on the category of the animal detected, the edge computing device executes its TCN Animal Recognition model to identify the target, and if an animal is detected, it sends back a message to the Animal Repelling

Module including the type of ultrasound to be generated according to the category of the animal. The proposed TCN was evaluated on the created animal database. Future enhancements can improve the functionality, user experience, and effectiveness of the Animal Repellent system.

Integration with smart home systems

The Animal Repellent system can be integrated with existing smart home systems such as Google Home or Amazon Alexa to allow for voice commands and automated responses. This can enhance the user experience and provide more convenience.

Mobile app integration

The Animal Repellent system can have a mobile app that allows users to remotely monitor and control the system from their smartphones. The app can also provide notifications and alerts to users when the system detects animal species.

Integration with security systems

The Animal Repellent system can be integrated with existing security systems such as CCTV cameras or motion sensors to provide a more comprehensive security solution. This can enhance the system's overall effectiveness and provide more peace of mind to users.

VII. ACKNOWLEDGEMENT

First and foremost, we thank the almighty for giving me talents and opportunity to complete my project and my family for unwavering support.

I would like to express my sincere heartfelt thanks to our college chief Secretary, Er. SURIYA SUBRAMANIAM, for providing facility to getting more ideas related to this project work.

I would like to express my sincere gratitude to our college chief chairperson, Dr. SUJATHA SUBRAMANIAM, Ph.D., for providing large facilities in the institution for the completion of the project work.

I would like to express my sincere thanks and gratitude to our college principal, Dr. D.LOGANATHAN, M.Sc (Tech)., Ph.D., for his encouragement to do this innovative project.

I wish to express heartfelt thanks and sincere gratitude to our Head of the Department Mr.KALAIYARASAN.K, M.E.,(Ph.D)for her valuable advice to complete this work successfully and also for her enthusiastic encouragement to make my effort worthwhile and fruitful.

I am extremely indebted to my Project Supervisor Mrs. NARMATHA G M.E.,Assistant Professor, Department of Computer Science and Engineering, who extended his complete guidance and valuable advice which helped me to take further step into depth of my project.

REFERENCES

[1] A. A. van den Bossche et al., "Aerial Wildlife Image Repository (AWIR): an open dataset for animal monitoring with drones," *Database*, 2024.

[2] A. R. Dawson et al., "Drone-Based Thermal Imaging in the Detection of Wildlife: performance evaluation and detection probability," *PLoS ONE*, 2024.

[3] D. B. M. Yousefi, A. S. M. Rafie, S. A. R. Al-Haddad and S. Azrad, "A Systematic Literature Review on the Use of Deep Learning in Precision Livestock Detection and Localization Using Unmanned Aerial Vehicles," *IEEE Access*, 2022.

[4] G. J. Anstee et al., "An evaluation of factors affecting detection probability for poacher/target detection with thermal and RGB sensors," *Scientific Reports / PMC dataset analysis*, 2021.

[5] S. R. Bakana, "A lightweight and efficient wild animal recognition model," *Information Processing in Agriculture / ScienceDirect*, 2024.

[6] J. — (conference paper), "Real-time marine animal detection using YOLO-based deep learning networks in coral reef ecosystems," conference paper (YOLO marine detection), 2021–2022.

[7] "Detection of animal intrusion using CNN and image processing" — project/paper (image processing + CNN) — ResearchGate, 2022–2023.

[8] T. Manikandan et al., "IoT Based Animal Detection and Alert System for Farm Fields," conference paper / ResearchGate, 2024.

[9] S. C. Sajin, A. Joseph, A. M. U., J. Joseph, "Animal Guard: CNN-Driven System for Real-Time Animal Detection in Human Habitats," *J. Electrical Systems*, 2024.

[10] "AWIR — Aerial Wildlife Image Repository" (dataset & paper supporting drone/thermal animal detection), *Oxford Academic / Database*, 2024.

[11] "Snapshot Serengeti camera trap dataset and analysis" — dataset article / coverage describing large camera-trap collections used for ML in wildlife detection. *Wired / Scientific Data summary*.

[12] "AI cameras aim to track elephants, reduce man-animal conflict in Chhattisgarh (field deployment example)," *Times of India*, 2025 (news on AI/infrared camera deployments & siren/SMS alert integration).
[13] ."AI will eavesdrop on world's wildest places to track and help protect endangered wildlife" — AP News (acoustic monitoring & Microsoft Sparrow project), 2024–2025.
[14] "Automated Detection and Recognition of Wildlife Using Thermal Cameras," conference/paper.
[15] "Wildlife monitoring with drones: A survey of end users" — survey & review on drone usage for wildlife monitoring, 2023–2024.
[16] "Exploration of machine learning approaches for automated ecological monitoring" — review article on ML+D L methods for ecology & detection tasks, *ScienceDirect*, 2024.
[17] "CNN Based Wildlife Intrusion Detection and Alert System" — conference/paper (2024/2025) — implementation examples using CNN + alerts.
[18] "Sound-based wildlife protection with machine learning" — acoustic ML approach for wildlife/poacher detection (2025 preprints / conference).
[19] "AI-Enhanced IoT-Integrated Virtual Fencing: Proof-of-Concept for Camel Monitoring and Collision Mitigation," IoT + RL/AI virtual fence proof-of-concept, 2025.
[20] "WILDLIFE INTRUSION DETECTOR" — IRJMETS (case study / applied system using camera + CNN + alerts), Nov. 2024.