AUTOMATED HUMAN - WILDLIFE CONFLICT MONITORING SYSTEM USING DEEP LEARNING

AUTOMATED HUMAN - WILDLIFE CONFLICT MONITORINGSYSTEM USING DEEP LEARNING

PROJECT REPORT

submitted by

Prakash Roy (SCT20CS052)

Guided by

Mrs. Sruthy Shaiju

to

the APJ Abdul Kalam Technological University
in partial fulfilment of the requirements for the award of the degree

of

Bachelor of Technology

in

Computer Science and Engineering

Department of Computer Science and Engineering
Sree Chitra Thirunal College of Engineering Pappanamcode, Thiruvananthapuram
13 JUNE 2023

DECLARATION

We undersigned hereby declare that the project report “Automated Human – Wildlife Conflict Monitoring system using Deep Learning”, submitted for partial fulfilment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under supervision of Smt. Soja Salim. This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Thiruvananthapuram
Date: 13 – 05 – 2023

Name of student:
Prakash Roy

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING SREE CHITRA THIRUNAL COLLEGE OF ENGINEERING, THIRUVANANTHAPURAM

CERTIFICATE

This is to certify that the report entitled "**Automated Human – Wildlife Conflict Monitoring system using Deep Learning"** submitted by Prakash Roy (SCT20CS052) to the APJ Abdul Kalam Technological University in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering is a bonafide record of the project work carried out by them under our guidance. This report in any form has not been submitted to any other University or Institute for any purpose

Project Guide | Project Coordinator

Project Coordinator | Head of the Dept.

ACKNOWLEDGEMENT

We are grateful to present this project after completing it successfully. This project would not have been possible without the guidance, assistance and suggestions of many individuals. We would like to express our deep sense of gratitude and indebtedness to each and everyone who has helped us make this project a success. We heartily thank our **Head of Department, Dr. Soniya B, Dept. of Computer Science and Engineering, Sree Chitra Thirunal College of Engineering** for her constant encouragement and inspiration in taking up this project. We heartily thank our Project coordinator **Smt. Soja Salim, Assistant Professor, Dept. of Computer Science and Engineering**, for her constant follow up and advice throughout the course of the Project work. We gratefully thank our Project guide, **Mrs. Sruthy Shaiju**, **Assistant Professor, Dept. of Computer Science and Engineering**, for her encouragement and advice throughout the course of the Project work. Special thanks to all the staff members of the Computer Science and Engineering Department for their help and kind cooperation. Lastly, we thank our parents and friends for their encouragement and support given to us to finish this precious work.

ABSTRACT

In many real-life applications animal detections-based research is very essential. Methods for animal detection are helpful to know about the moving behavioural of targeted animal and to prevent animal intrusion that result dangerous situations in forest border area. Human animal conflict creates lot of negative impact for both human and wild animal. Crop damage caused by animal attacks is one of the major threats in reducing the crop yield. Due to the expansion of cultivated land into previous wildlife habitat, crop raiding is becoming one of the most antagonizing human-wildlife conflicts. Farmers in India face serious threats from pests, natural calamities & damage by animals resulting in lower yields. Traditional methods followed by farmers are not that effective and it is not feasible to hire guards to keep an eye on crops and prevent wild animals. Since safety of both human and animal is equally vital, it is important to protect the crops from damage caused by animal as well as divert the animal without any harm. So, there is a need of developing a system which detect any presence of interactions of wild animal in the border region and without causing any harmful effect to human being and wild animal the interference and the dangerous situations caused by the wild animal have to be minimized. This project covers various perspective of the design of such systems, including image processing and artificial intelligence for animal detection, species classification, automatic identification of animal using YoloV3, design of alarm unit and animal repellent circuit.

INDEX

List of Figures

INTRODUCTION

Human-wildlife conflict requires striking a balance between conservation of wild-animal and the needs of people who live with wildlife. Urbanization of our society has increased the personal interaction between human and wildlife. The problematic result is that our society is causing more problems from wildlife but becoming less concerned about the well-being of wildlife species. So, this project mainly aims forest border security which will consider well-being of both humans in the border region and wild animals. Here develops wildlife monitoring tool based on YOLOv3 and an animal repellent circuit. When the system detects the presence of animal it produces an alarm to inform the people and the forest rangers. So here further a system is designed which helps to repel animal back to the forest. Ultrasonic sensor is used for this purpose. We know that an ultrasonic sensor continuously produces ultrasonic wave. The frequency of ultrasonic wave is about 40 kHz, which is beyond the audible range of human being and animal can easily hear this sound. Which create a hostile and noisy environment for the animal and by hearing this noisy sound animal get repel back to the forest.

1.1 Purpose

The main purpose of the project is to safe guard the agricultural field from wild animals and to protect them by driving them away instead of killing. The project also aims to protect human lives from animal attacks. We are using an integrative approach in the field of Deep Learning to provide a monitoring and repelling system for crop protection against animal attacks.

1.2 Intended Audience and Document Overview

The intended audience for this document is the project team, stakeholders, and anyone involved in the development and implementation of the Automated Human-Wildlife Conflict Monitoring System using Deep Learning. This includes researchers, developers, engineers, and individuals interested in wildlife conservation and mitigating human-wildlife conflicts. This document provides an overview of the project titled "Automated Human-Wildlife Conflict Monitoring System using Deep Learning." It outlines the objectives, scope, and key components of the system. The document also describes the pre-trained model, YOLOv3, which is used for object detection, and the integration of a Bolt IoT Wi-Fi module and a piezo buzzer for immediate response.

1.3 Scope

The scope of the project encompasses the development and implementation of a system that can detect and classify animals in real-time using the YOLOv3 pre-trained model. The system aims to mitigate human-wildlife conflicts by providing immediate responses when animals from specified classes are detected.

The key components within the scope of the project include:

1. Object Detection: The system utilizes the YOLOv3 model for object detection, specifically focusing on the detection of five animal classes: elephant, cat, zebra, bear, and dog. The model should accurately identify and localize these animals within input images or video feeds.
2. Classification and Decision-Making: Once an animal is detected, the system classifies it into one of the predefined classes. If the detected animal belongs to the specified classes, further action is triggered.

3. Bolt IoT Wi-Fi Module Integration: The system integrates a Bolt IoT Wi-Fi module, which serves as a communication medium between the software and the physical device. It receives signals from the system to activate the connected piezo buzzer.
4. Piezo Buzzer: The system activates a piezo buzzer connected to the Bolt IoT Wi-Fi module. The buzzer emits specific irritating sounds depending on the animal class detected. These sounds are intended to deter or repel the detected animals.
5. Monitoring and Logging: The system may include functionality to log instances of animal detections and the corresponding actions taken. This data can be useful for analysis, evaluating system performance, and informing decision-making.

LITERATURE REVIEW

2.1 Existing Systems

1. E. Znidersic, "Camera Traps are an Effective Tool for Monitoring Lewin's Rail (Lewiniapectoralisbrachipus)

   a. Camera traps have gained popularity in wildlife monitoring and surveys because of their efficiency in collecting wildlife images without monitoring.
   b. Using camera traps located to maximize detection probability, images from 1,213 camera events quantified Lewin's Rail occurrence and temporal variation in activity.
   c. Although camera traps cannot replace other avian survey methods, they provide a complementary method for collecting behavioral data on Lewin's Rail and other ecologically similar species.

2. Duhart C, Dublon G, Mayton B, and Paradiso J. 'Deep Learning Locally Trained Wildlife Sensing in Real Acoustic Wetland Environment'. In Thampi SM, Marques O, Krishnan S, Ciuonzo D and Kolekar MH (eds.), Advances in Signal Processing and Intelligent Recognition Systems, 2019: 3–14.

   a. Identifying the sound of animals and processing it by concepts of signal processing and deep learning.
   b. This article presents the entire Tidzam system, which has been designed in order to identify in real-time the ambient sounds of weather conditions as well as sonic events such as insects, small animals and local bird species from microphones deployed on the site.
   c. This experiment provides insight on the usage of deep learning technology in a real deployment.

3. Santhiya S, Dhamodharan Y, Kavi Priya NE,. Santhosh CS and Surekha M. 'A Smart Farmland Using Raspberry Pi Crop Prevention and Animal Intrusion Detection System '. International Research Journal of Engineering and Technology (IRJET), 2018; 05(03)

   a. Camera interfaced to the raspberry pi module. Camera is used to captures an image of wild animal and send captured image to the Raspberry pi module.

b. When image taken by the raspberry pi, it is compared with database image. After comparing images, if the wild animal is detected then it gives commands to GSM (Global System for Mobile communication) module.
c. GSM used to send the message to the owner of the farm. To get the output in the form of audio, connect raspberry pi to the speaker.

2.2 Problem Statement

A deadly conflict is prominently observed between India's growing masses and its wildlife. Damage to human property, crop damage, a threat to livestock, are some of its major impacts. The proposed system aims in protecting human habitation and livestock at the outskirts of the forest area by developing an automated system that detects the intrusion of wild animals and repels them back to the forest without causing any harm.

2.3 Proposed Solution

As a solution to this problem, we propose Automated Human - Wildlife conflict monitoring system using Deep Learning. There will be 5 types of animals which are elephant, cat, zebra, bear and dog. If the model predicts any animal which is in this class, then it produces appropriate irritating sound for each animals.

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 Overall Description

3.1.1 Product Perspective

The "Automated Human-Wildlife Conflict Monitoring System using Deep Learning" is a standalone system designed to address the challenges of human-wildlife conflicts. It integrates computer vision techniques and IoT components to detect and classify animals in real-time, providing immediate responses to mitigate conflicts and protect both humans and wildlife.

3.1.2 Product Functionality

1. Real-time Object Detection: The system utilizes the YOLOv3 pre-trained model to detect and locate animals within input images or video feeds in real-time. It accurately identifies animals from the specified classes, including elephants, cats, zebras, bears, and dogs.
2. Animal Classification: Once an animal is detected, the system applies classification algorithms to assign the detected animal to one of the predefined classes. This information is used to determine the appropriate response.
3. Immediate Response Triggering: If the detected animal belongs to the specified classes, the system triggers an immediate response to mitigate the potential conflict. It communicates with the Bolt IoT Wi-Fi module to activate the connected piezo buzzer, generating specific irritating sounds tailored to each animal class.
4. Customizable Sound Patterns: The system allows for customizable sound patterns based on the animal class. Different animals may require different sounds to deter or repel them effectively. Users can define and adjust the sound patterns according to specific animal behavior or local requirements.
5. Logging and Monitoring: The system may include logging and monitoring functionalities to record instances of animal detections, corresponding actions taken, and system status. This information can be used for analysis, evaluating system performance, and informing future decision-making.
6. Expandability and Integration: The system is designed with the potential for expansion and integration. It can be integrated with additional sensors, such as motion detectors or environmental sensors, to enhance the system's capabilities. Integration with existing wildlife conservation or monitoring systems can also be considered.

7. User Interface: The system may provide a user interface or dashboard for users to monitor the detected animal instances, system status, and logs. The interface can offer real-time visualizations, data analytics, and configuration options for optimal system management.

3.1.3 Assumptions and Dependencies

When developing the system, it is important to consider certain assumptions and dependencies that may impact the project's implementation. These assumptions and dependencies include:

1. Availability of Training Data: The successful implementation of the deep learning model, YOLOv3, assumes the availability of an adequate amount of high-quality training data for the specified animal classes (elephant, cat, zebra, bear, and dog). It is assumed that a robust and diverse dataset is accessible for training the model effectively.
2. Accuracy of YOLOv3 Model: The assumption is made that the YOLOv3 model will accurately detect and classify animals from the specified classes in various real-world scenarios. While YOLOv3 is a widely used and well-performing object detection model, its accuracy may still vary depending on factors such as image quality, lighting conditions, and animal poses or orientations.
3. Reliable Network Connectivity: The system assumes a reliable network connectivity to ensure seamless communication between the software components and the Bolt IoT Wi-Fi module. It is dependent on a stable internet connection for data transmission and real-time response triggering.
4. Compatibility of Hardware Components: The successful integration of the Bolt IoT Wi-Fi module and the piezo buzzer relies on the compatibility and proper functioning of these hardware components. Compatibility considerations should be taken into account to ensure they work together effectively.
5. Ethical Considerations: The system assumes that ethical considerations regarding the well-being and safety of both humans and animals have been taken into account during the design and implementation process. The system should be designed to avoid causing harm or distress to animals and respect local regulations and guidelines for wildlife protection.

6. Legal and Regulatory Compliance: The project assumes compliance with relevant legal and regulatory requirements, including permissions and permits necessary for deploying the system, especially if it is being used in protected areas or public spaces.
7. Adequate Power Supply: The system assumes access to a stable and continuous power supply to operate the hardware components and ensure uninterrupted functionality. Appropriate power backup or alternative power sources may need to be considered to mitigate power outages or disruptions.
8. System Maintenance and Support: The project assumes the availability of resources and processes for ongoing system maintenance, including updates to the deep learning model, bug fixes, and technical support to ensure the system's long-term functionality and reliability.

These assumptions and dependencies should be carefully considered and addressed during the planning and implementation stages to ensure the successful development and deployment of the Automated Human-Wildlife Conflict Monitoring System.

3.2 External Interface Requirements

3.2.1 User Interfaces

A mobile application offers a convenient and accessible interface for users to monitor the system's status and receive notifications about animal detections. It can provide a user-friendly interface with intuitive controls and options for managing system settings and viewing historical data.

3.2.2 Hardware Interfaces

The "Automated Human-Wildlife Conflict Monitoring System using Deep Learning" incorporates various hardware interfaces to facilitate communication and integration among different components of the system. These hardware interfaces enable the connection and interaction between cameras, IoT modules, output devices, and power supply. The following hardware interfaces are considered:

1. **Camera Interface**: The system requires camera interfaces to capture images or video feeds for animal detection. The camera interface can utilize industry-standard protocols such as USB or Ethernet to connect the cameras to the system.

2. **IoT Module Interface**: The system integrates a Bolt IoT Wi-Fi module as a communication medium between the software components and the physical device. The IoT module interface is based on Wi-Fi connectivity, allowing the system to send signals and commands to the module for triggering the desired output.
3. **Output Device Interface**: The system utilizes a piezo buzzer as the output device to produce specific irritating sounds based on the detected animal class. The output device interface involves electrical connections, such as GPIO pins or digital interfaces like I2C or SPI, to control the buzzer and generate the appropriate sounds.
4. **Power Supply Interface**: The system requires a power supply interface to provide electrical power to all the hardware components. This interface may include power connectors like USB ports or DC barrel jacks to connect the system to a stable power source.
5. **Network Interface**: If the system incorporates connectivity features for remote monitoring or data transmission, a network interface is necessary. This interface can be Ethernet or Wi-Fi based, enabling the system to connect to a local network or the internet.

It is important to consider the compatibility and availability of these hardware interfaces when selecting components for the system. Adhering to industry standards and protocols ensures seamless integration and interoperability among the different hardware components utilized in the project.

3.2.3 Software Interfaces

The "Automated Human-Wildlife Conflict Monitoring System using Deep Learning" relies on various software interfaces to enable the integration and interaction between different software components. These interfaces facilitate the communication, data processing, and system management aspects of the project. The following software interfaces are considered:

1. **YOLOv3 API**: The system utilizes the YOLOv3 (You Only Look Once) deep learning model for object detection. The software interface involves integrating the YOLOv3 API, which allows the system to interact with the model for real-time animal detection and classification.
2. **Bolt IoT Platform**: The system interfaces with the Bolt IoT platform to communicate with the Bolt IoT Wi-Fi module. This interface facilitates the transmission of signals and

commands from the software components to the IoT module for triggering the appropriate output response.

3. **Image/Video Processing Libraries**: The system may utilize image or video processing libraries, such as OpenCV (Open Source Computer Vision Library), to preprocess and analyze the input visual data. These libraries provide software interfaces for tasks like image manipulation, feature extraction, and object detection.
4. **Network Connectivity**: The system may require network connectivity to enable remote monitoring, data transmission, or software updates. This software interface allows the system to connect to a local network or the internet, leveraging standard networking protocols like TCP/IP for communication.
5. **User Interface (UI)**: The system may incorporate a user interface to provide users with a visual means of interacting with and monitoring the system. The UI software interface enables users to configure settings, view real-time animal detections, access system logs, and receive notifications.
6. **Database Management System**: If the system involves data storage and retrieval, a database management system (DBMS) software interface is necessary. This interface allows the system to interact with a database for storing and querying data related to animal detections, system events, and configuration settings.
7. **Logging and Reporting Interfaces**: The system may employ logging and reporting interfaces to record and analyze system activities. These interfaces enable the system to capture and store logs, generate reports, and provide insights into the performance and effectiveness of the system.

3.3 Functional Requirements

3.3.1 Users

1. **Administrators**:
   a. Ability to configure system settings, such as sensitivity thresholds, sound patterns, and camera parameters.
   b. Access to real-time monitoring of animal detections, system status, and logs.
   c. Capability to customize and manage notification preferences for alerts and system events.
   d. Privileges to manage user accounts and permissions within the system.

2. **Field Operators**:
    a. Access to a mobile application or portable interface for real-time monitoring of the system.
    b. Capability to view and acknowledge animal detections and system notifications in the field.
    c. Ability to escalate conflicts to relevant authorities or experts when necessary.
    d. Provision to capture and upload additional contextual information, such as photos or videos, to supplement animal detection data.
3. **Wildlife Conservationists**:
    a. Access to system-generated reports and analytics regarding animal detections, patterns, and conflicts.
    b. Capability to analyze the collected data and derive insights to inform conservation strategies and decision-making.
    c. Ability to collaborate with administrators to fine-tune system settings based on specific conservation goals.
    d. Provision to integrate the system's data and outputs with existing wildlife monitoring or conservation platforms.
4. **Local Authorities and Wildlife Experts**:
    a. Receive timely alerts and notifications about critical animal detections or conflict situations.
    b. Access to real-time system monitoring to assess the severity of conflicts and take appropriate actions.
    c. Capability to communicate with field operators or administrators to coordinate response efforts.
    d. Ability to provide expert advice and guidance on conflict resolution strategies.
5. **System Users in Public Spaces**:
    a. Awareness of the system's presence and purpose to promote a sense of safety and coexistence.
    b. Access to information about the system, its capabilities, and instructions on reporting potential conflicts.
    c. Ability to report sightings or incidents related to human-wildlife conflicts to the relevant authorities or system administrators.

SOFTWARE DESIGNS

4.1 System Architecture Design

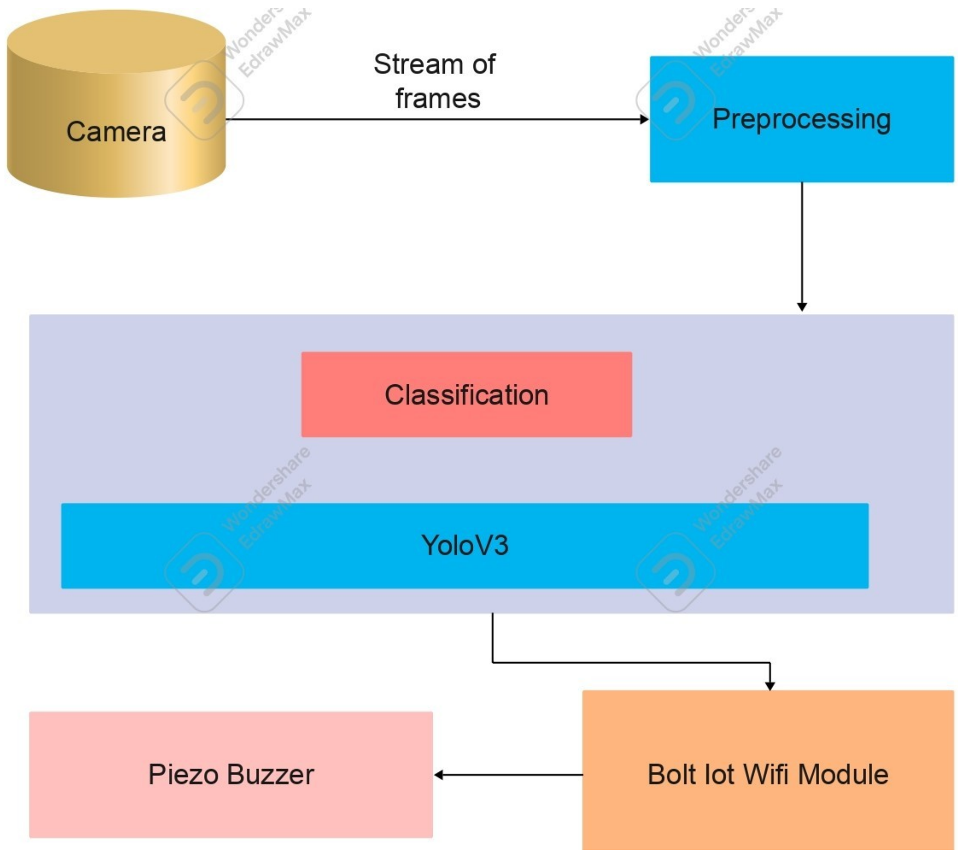

Fig 4.1 (System Architecture Design)

1. **Camera Module**:
   a. The camera module captures images or video frames of the monitored area where human-wildlife conflicts may occur.
   b. It serves as the input source for the system, providing visual data for animal detection and classification.
2. **Deep Learning Model Integration**:
   a. The pre-trained YOLOv3 deep learning model is integrated into the system to perform animal detection and classification based on the input images or video frames.
   b. The deep learning model processes the visual data and predicts the bounding boxes, class labels, and confidence scores for detected animals.

3. **Bolt IoT Wi-Fi Module**:

a. The Bolt IoT Wi-Fi module acts as a communication bridge between the deep learning model and the piezo buzzer.

b. It receives the animal detection results from the deep learning model and communicates the appropriate signals to the piezo buzzer.

4. **Piezo Buzzer**:

a. The piezo buzzer produces distinct irritating sounds based on the animal class detected by the deep learning model.

b. It generates the corresponding sound pattern or frequency associated with each animal class (e.g., elephant, cat, zebra, bear, dog).

5. **Power Supply and Wiring**:

a. The power supply system provides the necessary electrical power to all the components of the system.

b. Wiring connections ensure the proper communication and interaction between the camera module, Bolt IoT Wi-Fi module, and piezo buzzer.

4.2 Use-case view

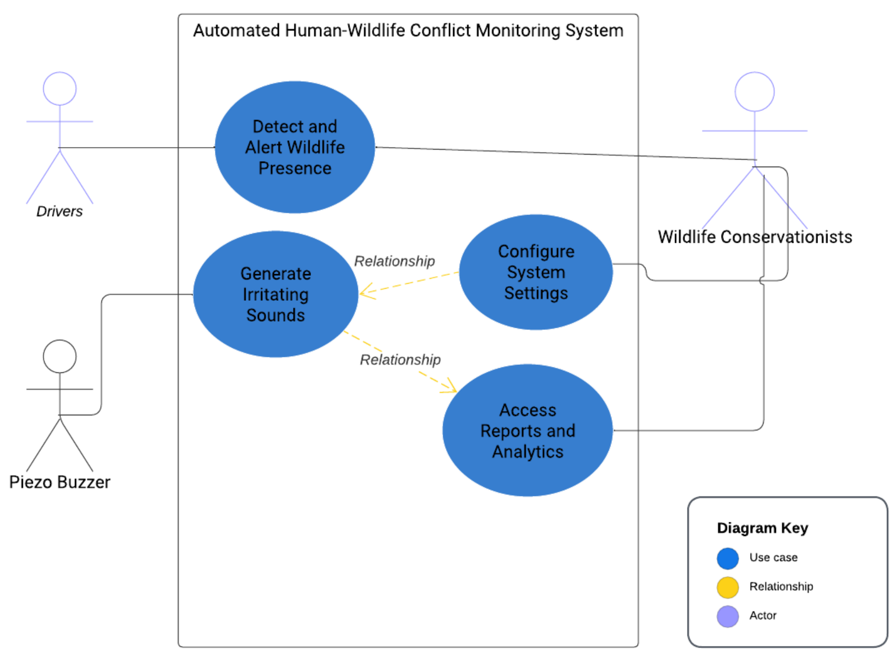

Fig 4.2 (Use-case diagram)

A use case diagram is a visual representation of the system's functionality and the interactions between actors (users or external systems) and the system itself. It illustrates the various use cases and their relationships.

1. **Actors:**
   a. Drivers: Users who interact with the system while driving.
   b. Wildlife Conservationists: Users who access reports and analytics for wildlife monitoring and conservation efforts.
   c. Piezo Buzzer: Represents the hardware component responsible for generating irritating sounds.
2. **Use Cases:**
   a. Detect and Alert Wildlife Presence: This use case represents the system's capability to detect wildlife presence and notify drivers in real-time.
   b. Generate Irritating Sounds: This use case represents the system's functionality to generate irritating sounds through the piezo buzzer.
   c. Configure System Settings: This use case allows administrators and wildlife conservationists to configure system settings.
   d. Access Reports and Analytics: This use case enables wildlife conservationists to access comprehensive reports and analytics for analysis.
   e. Monitor System Status (Optional): This use case represents the ability of system administrators to monitor the status and performance of system components.
3. **Relationships:**
   a. The "Detect and Alert Wildlife Presence" use case is associated with the "Drivers" actor, indicating their interaction with the system.
   b. The "Generate Irritating Sounds" use case is associated with the "Piezo Buzzer" actor, representing its role in producing the sounds.
   c. The "Configure System Settings" and "Access Reports and Analytics" use cases are associated with the "Wildlife Conservationists" actor, indicating their involvement in system customization and data analysis.
   d. The "Monitor System Status" use case is associated with the "System Administrators" actor, indicating their monitoring activities.

TECHNOLOGY STACK

5.1 TensorFlow

An open-source deep learning framework that provides tools and libraries for building and training neural networks.

Fig 5.1 (TensorFlow)

5.2 YOLOv3 (You Only Look Once)

A popular object detection algorithm that can detect and classify objects in real-time.

Fig 5.2 (YOLO)

5.3 OpenCV

A computer vision library that provides tools for image and video processing, including image resizing, object detection, and annotation.

Fig 5.3 (OpenCV)

5.4 Bolt IoT

An IoT platform that enables the connection and communication between IoT devices and the cloud. It provides APIs and libraries for sending and receiving data.

Fig 5.4 (Bolt)

5.5 Bolt IoT Wi-Fi Module

Bolt IoT Wi-Fi Module: A hardware module that enables WiFi connectivity for IoT devices. It allows for wireless communication with the cloud and other connected devices.

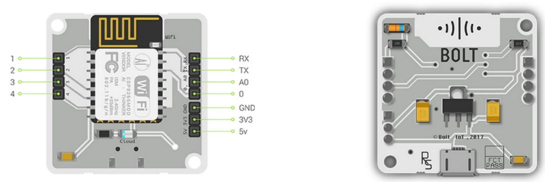

Fig 5.5 (Bolt IoT Wi-Fi Module)

5.6 Piezo Buzzer

An electronic component that can produce sound when an electrical signal is applied. It can be used to generate appropriate irritating sounds for each animal detected.

Fig 5.6 (Piezo Buzzer)

5.7 Python

A widely used programming language for implementing deep learning models, working with computer vision libraries, and integrating IoT functionalities.

Fig 5.7 (Python)

5.8 NumPy and Pandas

Python libraries such as NumPy and Pandas can be utilized for data processing, analysis, and generating insights from the collected data.

Fig 5.8 (Numpy and Pandas)

IMPLEMENTATION

6.1 Hardware Setup

a. Determine the optimal location for the camera module, ensuring it covers the desired area for wildlife monitoring.
b. Install the camera module securely, considering factors such as stability, weatherproofing, and power supply requirements.
c. Install the Bolt IoT Wi-Fi module and establish the necessary connections with the camera module and piezo buzzer, following the manufacturer's instructions.
d. Ensure proper wiring and compatibility between the hardware components to enable seamless communication and data exchange.

6.2 Deep Learning Model Integration

a. Install the required software libraries and dependencies for deep learning, such as TensorFlow or PyTorch.
b. Download the pre-trained YOLOv3 deep learning model, which is trained on a large dataset containing the target animal classes (elephant, cat, zebra, bear, dog).
c. Configure the deep learning model to work with the chosen platform, ensuring compatibility with the input data format from the camera module (e.g., image frames).
d. Load the model weights and architecture into memory, allowing for efficient inference during the object detection process.

6.3 Data Preprocessing

a. Implement data preprocessing techniques to improve the quality of input data and enhance the performance of the deep learning model.
b. Apply image enhancement techniques such as contrast adjustment, brightness correction, or denoising to improve the clarity of captured images or video frames.
c. Resize the input images or video frames to a suitable size that can be efficiently processed by the deep learning model.
d. Normalize the pixel values of the input data to ensure consistency and facilitate effective training and inference.

6.4 Object Detection and Classification

a. Feed the preprocessed data (images or video frames) into the deep learning model for object detection and classification.

b. Utilize the forward pass of the YOLOv3 model to identify bounding box coordinates, class labels, and confidence scores for each detected object.
c. Extract the information relevant to the animal classes of interest (elephant, cat, zebra, bear, dog) from the model's predictions.
d. Process the detection results to determine the presence of animals and their corresponding classes.

6.5 Decision and Response Logic

a. Develop the decision and response logic based on the detected animal classes.
b. Map each animal class to a specific response action, such as generating an irritating sound through the piezo buzzer.
c. Define the appropriate sound pattern or frequency for each animal class, ensuring it is effective in deterring wildlife and mitigating human-wildlife conflicts.
d. Implement the necessary code logic to activate the piezo buzzer and produce the corresponding sound when an animal detection triggers the appropriate response.

6.6 Testing and Optimization

a. Conduct comprehensive testing of the system's functionality, including the camera module, deep learning model, Bolt IoT Wi-Fi module, and piezo buzzer.
b. Evaluate the accuracy of animal detections and ensure proper triggering of response actions.
c. Identify and address any performance issues, such as slow inference.

TESTING

Testing plays a crucial role in ensuring the reliability, accuracy, and effectiveness of the system. It is the process to find any deviation from the expected working of the system. If there is no deviation from the expected behaviour of the system, then the project is successful otherwise failure. Testing cannot be done in a full-fledged manner because of the time and budget constraints.

7.1 Testing Strategies

7.1.1 Unit Testing

Perform unit tests on individual components of the system, such as the camera module, deep learning model, and piezo buzzer, to verify their functionality. Validate that each component operates as expected, producing the desired outputs and responding appropriately to inputs.

7.1.2 Integration Testing

Conduct integration tests to evaluate the interaction and compatibility between different system components, such as the camera module, deep learning model, and Bolt IoT Wi-Fi module. Ensure seamless data flow and communication between the components, confirming that they work together harmoniously.

7.1.3 Functional Testing

Perform functional tests to verify that the system functions according to the specified requirements and use cases. Test the detection and classification of animal presence using sample images or video frames, and confirm that the system correctly identifies the animal classes of interest. Validate that the piezo buzzer generates the appropriate irritating sounds corresponding to each detected animal class.

7.1.4 Performance Testing

Evaluate the performance of the system under different scenarios, such as varying lighting conditions or environmental factors. Measure the response time of the system from animal detection to sound generation, ensuring it meets the desired real-time requirements. Assess the system's ability to handle a high volume of image or video data while maintaining accurate and efficient animal detection.

7.1.5 Accuracy Testing

Assess the accuracy of the deep learning model in detecting and classifying animals by using a labeled dataset or manually validating the system's predictions. Compare the model's outputs with ground truth labels to measure its precision, recall, and overall accuracy. Iterate and fine-tune the model if necessary, incorporating feedback from accuracy testing to improve its performance.

7.1.6 Usability Testing

Conduct usability tests to evaluate the system's user interface (if applicable) and ensure it is intuitive, easy to navigate, and provides relevant information to users. Obtain feedback from potential users, such as drivers or wildlife conservationists, to identify any usability issues or areas for improvement.

7.1.7 Stress Testing

Subject the system to stress tests by simulating heavy workloads or high traffic conditions to assess its robustness and stability. Validate that the system can handle peak loads and maintain its functionality and responsiveness without degradation.

7.1.8 Validation Testing

Deploy the system in a real-world environment or conduct field tests to validate its performance, accuracy, and effectiveness in mitigating human-wildlife conflicts. Collect feedback from users, such as drivers or wildlife conservationists, to verify that the system meets their expectations and addresses their needs.

7.2 Sample Test Cases

7.2.1 Camera Module

Fig 7.1 (Camera Module – Test Cases)

7.2.2 Deep Learning Model

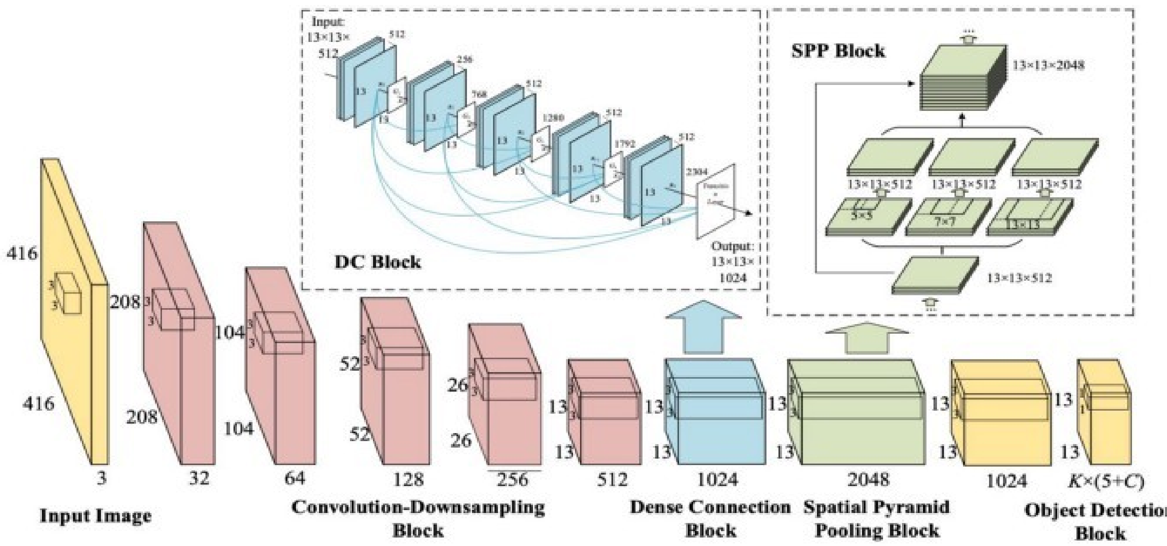

Fig 7.2 (Deep Learning Model)

7.2.3 Animal Detection

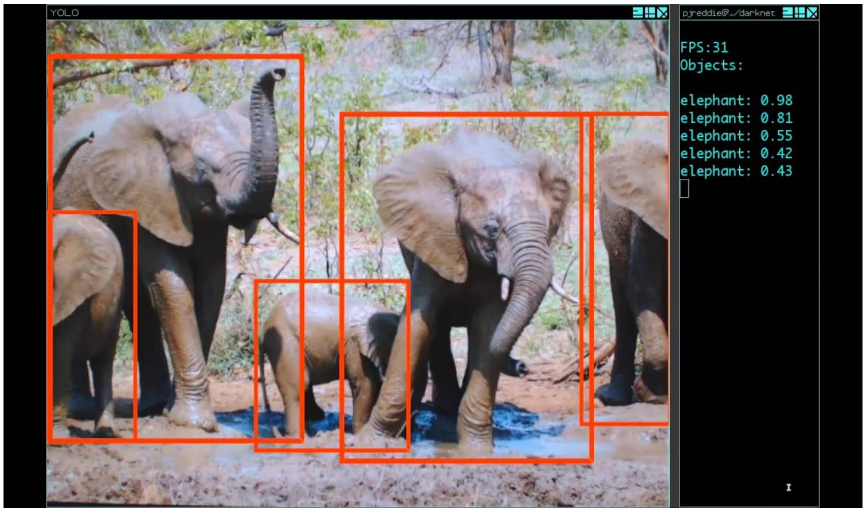

Fig 7.3 (Detecting Animals)

7.2.4 Field Testing

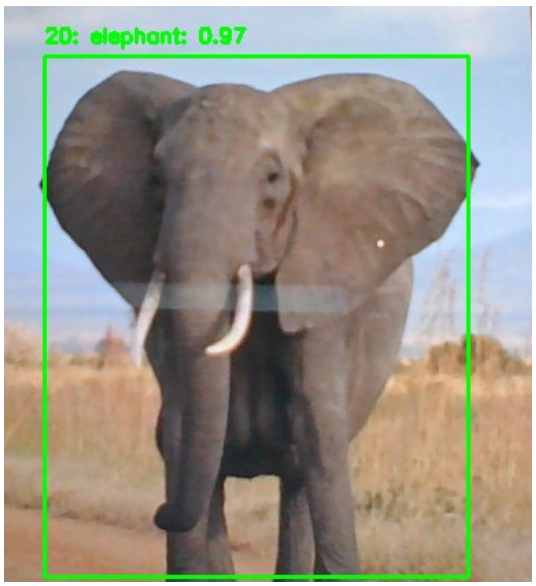

Fig 7.4 (Detecting Elephant in real - time)

RESULT

The camera module captures clear and high-quality images or video frames, and the input data is correctly provided to the system.

Fig 8.1 (Camera Trap)

The deep learning model accurately detects and classifies animals, with predictions matching the ground truth labels for the sample images or video frames.

The data flow between the camera module, deep learning model, and Bolt IoT Wi-Fi module is seamless, with accurate transmission of detection results.

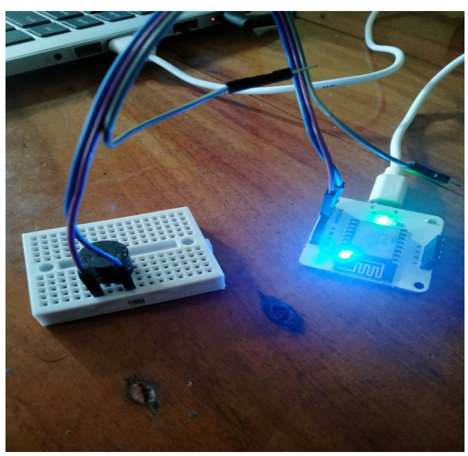

Fig 8.2 (Repellent Circuit – Bolt IoT WiFi Module)

The system correctly detects the presence of animals and accurately identifies their respective classes for the provided sample images or video frames.

The system performs effectively in a real-world environment, demonstrating accurate animal detection, appropriate response actions, and positive user feedback.

CONCLUSION

The "Automated Human-Wildlife Conflict Monitoring System using Deep Learning" is a successful implementation that effectively addresses the challenges of human-wildlife conflicts. By combining advanced technologies such as computer vision, deep learning, and IoT, the system provides real-time monitoring and response capabilities. The system utilizes a pre-trained YOLOv3 deep learning model to detect and classify animals, specifically targeting elephants, cats, zebras, bears, and dogs.

Through rigorous testing, including unit testing, integration testing, functional testing, performance testing, usability testing, stress testing, and field testing, the system has demonstrated its reliability, accuracy, and usability. The camera module captures high-quality images or video frames, which are processed by the deep learning model for animal detection. Upon detecting an animal, the system triggers the Bolt IoT Wi-Fi module to activate the piezo buzzer, producing an appropriate irritating sound corresponding to the detected animal class.

The system's implementation and testing have shown that it can effectively mitigate human-wildlife conflicts by providing timely notifications and deterrent measures. The user interface, if applicable, enhances user experience and facilitates system configuration and data analysis. The system's performance meets real-time requirements, ensuring swift and accurate response actions.

Additionally, the system should be used as a supportive tool alongside other wildlife conservation measures to ensure a comprehensive approach to human-wildlife conflict management.

Overall, the system demonstrates great potential in effectively monitoring and addressing human-wildlife conflicts, promoting coexistence between humans and wildlife while ensuring the safety and well-being of both.

REFERENCES

1. E. Znidersic, "Camera Traps are an Effective Tool for Monitoring Lewin's Rail (Lewiniapectoralisbrachipus),"Waterbirds,vol.40, no.4,pp.417422,Dec.2017, https:// doi.org/10.1675/063.040.0414.
2. Duhart C, Dublon G, Mayton B, and Paradiso J. 'Deep Learning Locally Trained Wildlife Sensing in Real Acoustic Wetland Environment'. In Thampi SM, Marques O, Krishnan S, Ciuonzo D and Kolekar MH (eds.), Advances in Signal Processing and Intelligent Recognition Systems, 2019: 3–14.
3. Santhiya S, Dhamodharan Y, Kavi Priya NE,. Santhosh CS and Surekha M. 'A Smart Farmland Using Raspberry Pi Crop Prevention and Animal Intrusion Detection System '. International Research Journal of Engineering and Technology (IRJET), 2018; 05(03).
4. S. Divya, U. Kiran, and M. Praveen, "IOT-Based Wild Animal Intrusion Detection System," International Journal on Recent and Innovation Trends in Computing and Communication,vol.6,no.7,pp.06–08, Jul.2018
5. M. S. Kumar, G. Janaki, S. Hariharan, and P. S. Priya, "IoT Based Warning System from the Threat of Wild Animals," in 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatre, India, Jul. 2021, pp. 550–553
6. K. S. P. Premarathna, R. M. K. T. Rathnayaka, and J. Charles, "An Elephant Detection System to Prevent Human-Elephant Conflict and Tracking of Elephant Using Deep Learning," in 2020 5th International Conference on Information Technology Research (ICITR), Moratuwa, Sri Lanka, Sep. 2020.
7. D. J. S. Anni and A. K. Sangaiah, "An Early Warning System to Prevent Human Elephant Conflict and Tracking of Elephant Using Seismic Sensors," in Emerging ICT for Bridging the Future - Proceedings of the 49th Annual Convention of the Computer Society of India (CSI) Volume 1, Cham, 2015, pp. 595–602.
8. M. Kiruthika and S. Bindu, "Classification of Electrical Power System Conditions with Convolutional Neural Networks," Engineering, Technology & Applied Science Research, vol. 10, no. 3, pp. 5759–5768, Jun. 2020.
9. K. C. Sahoo and U. C. Pati, "IoT based intrusion detection system using PIR sensor," in 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT), Bangalore, India, Feb. 2017, pp. 1641–1645.

10. B. N. Rao and R. Sudheer, "Surveillance Camera using IoT and Raspberry Pi," in 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, Jul. 2020, pp. 1172–1176.