CAMERA BASED WILD ANIMAL DETECTION AND ALERT SYSTEM NEAR WILDLIFE FENCES USING DEEP LEARNING

PROJECT REPORT

Submitted by

AAKASH K SABAI [REGISTER NO: 211420243001]

KISHORE KUMAR M [REGISTERNO:211420243025]

VIJAY PA [REGISTER NO:211420243062]

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

MARCH 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report “CAMERA BASED WILD ANIMAL DETECTION AND ALERT SYSTEM NEAR WILDLIFE FENCES USING DEEP LEARNING” is the bonafide work of “ AAKASH K SABAI (211420243001), KISHORE KUMAR M (211420243025) and VIJAY PA (211420243062) ” who carried out the project work under my supervision.

SIGNATURE

Dr. S. MALATHI
HEAD OF THE DEPARTMENT

DEPARTMENT OF AI&DS,
PANIMALAR ENGINEERING COLLEGE,
COLLEGE, NAZARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

SIGNATURE

Dr. K.JAYASHREE
PROFESSOR,

DEPARTMENT OF AI&DS,
PANIMALAR ENGINEERING
COLLEGE, NAZARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENTS

We **VIJAY PA [211420243062], AAKASH K SABAI [211420243001], KISHORE KUMAR M [211420243025]** hereby declare that this project titled "**CAMERA BASED WILD ANIMAL DETECTION AND ALERT SYSTEM NEAR WILDLIFE FENCES USING DEEP LEARNING** ", under the guidance of **Dr. K. Jayashree** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr. P. CHINNADURAI, M.A., Ph.D.,** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our Directors **Tmt. C. VIJAYARAJESWARI**, **Dr. C. SAKTHI KUMAR, M.E., Ph.D.** and **Dr. SARANYASREE SAKTHI KUMAR B.E., M.B.A., Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr. K. MANI, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the AI&DS Department, **Dr. S. MALATHI, M.E., Ph.D.,** for the support extended throughout the project.

We would like to thank our supervisor **Dr. K.JAYASHREE,** coordinator **Dr. K.JAYASHREE & Dr. P. KAVITHA** and all the faculty members of the Department of AI&DS for their advice and encouragement for the successful completion of the project.

VIJAY PA

AAKASH K SABAI

KISHORE KUMAR M

ABSTRACT

Human-wildlife conflicts in forested areas have posed persistent challenges, threatening the safety of communities and the coexistence of diverse ecosystems. Over the past few years, these conflicts have intensified, necessitating advanced technological solutions to mitigate risks and enhance conservation efforts. In response to this pressing issue, our research introduces a groundbreaking wildlife threat mitigation system that leverages Camera and LoRaWAN (Long Range Wide Area Network) communication. The proposed system comprises essential components, including Cameras, PC for data processing, and LoRaWAN modules for efficient wireless communication.

It seamlessly integrates these elements to provide real-time, accurate detection of wildlife threats and the transmission of alerts to local communities and authorities. In this paper, we provide a comprehensive exploration of the system's architecture, detailing hardware and software components, as well as the step-by-step process involved. A feasibility study underscores the practicality and economic viability of the system, while the software environment, security measures, and operational aspects are thoroughly analyzed to ensure user-friendliness, reliability, and ease of maintenance.

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS, ABBREVIATIONS

SERIAL NO.	ABBREVATION	EXPANSION
1	YOLO	You Only Look Once
2	GPS	Global Positioning System
3	PC	Personal Computer
4	LoRa	Long Range
5	Arduino	Derived from "Hoarduino" - an earlier project
6	API	Application Programming Interface
7	GSM	Global System for Mobile Communications
8	SMS	Short Message Service
9	USB	Universal Serial Bus
10	IDE	Integrated Development Environment
11	CPU	Central Processing Unit
12	GPU	Graphical Processing Unit
13	OpenCV	Open Source Computer Vision Library
14	NumPy	Numerical Python
15	SciPy	Scientific Python
16	CSV	Comma-Separated Values
17	LiDAR	Light Detection and Ranging
18	Wi-Fi	Wireless Fidelity

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1. INTRODUCTION

1.1 Problem Definition:

The following description mainly describes the invention and the manner in which it is to be performed

<u>DESCRIPTION</u>

THE FIELD OF INVENTION:

The field of invention for this research paper lies at the intersection of wildlife management, remote sensing technologies, and wireless communication systems. It addresses the critical need for improved methods of detecting and responding to wild animal threats in forested areas where human populations and wildlife habitat overlap.

NOVELTY OF THE INVENTION:

The proposed system represents a significant advancement in the field of wildlife threat mitigation in forested areas. Several key inventions contribute to its novelty:

- LiDAR-Based Detection Precision: Utilizing LiDAR technology enhances the precision and accuracy of wildlife detection, enabling proactive responses to potential threats.

- Integration of LoRaWAN Communication: Incorporating LoRaWAN communication systems enables real-time data transmission over long distances, facilitating rapid response and coordination in remote forested regions.

- Comprehensive Threat Mitigation: By combining advanced detection technologies with robust communication systems, the innovation offers a comprehensive approach to wildlife threat mitigation, addressing various challenges associated with human-wildlife coexistence.

OBJECTIVES OF THE INVENTION:

The innovation's objectives are rooted in the pursuit of accuracy, speed, sustainability, comprehensiveness, and the betterment of both human and wildlife welfare. By addressing these objectives, the innovation seeks to achieve the following goals:

- Accuracy: Enhance the precision of wildlife detection to minimize false positives and false negatives, ensuring timely and appropriate responses to potential threats.

- Speed: Enable rapid communication and coordination among stakeholders to facilitate swift response and mitigation of wildlife threats.

- Sustainability: Promote environmental sustainability by utilizing renewable energy sources such as solar panels for power, reducing reliance on non-renewable resources.

- Comprehensiveness: Offer a holistic approach to wildlife threat mitigation by integrating multiple technologies and strategies to address various challenges associated with human-wildlife coexistence.

- Human and Wildlife Welfare: Prioritize the safety and well-being of both human and wildlife populations by implementing effective mitigation measures while minimizing adverse impacts on natural ecosystems.

CHAPTER 2
LITERATURE SURVEY

LITERATURE SURVEY:

Let us discuss several techniques adopted by various authors through this survey.

R. Shanthakumari, et.al,[1] have proposed a Camera based animal species assessment to effectively recognizes different animal species. Here they have also proposed various remote sensing techniques for Camera based monitoring.

Jingrong Chen, et.al, [2] have proposed a method for detection of Deer while crossing the roads using LiDAR. The LiDAR data is continuously processed , and when a deer comes within the range of the LiDAR sensor, it recognizes and alerts the drivers who travel nearby.

Elizaveta Dubrovinskaya, et.al, [3] have proposed a new LiDAR design with some inexpensive components, that uses low power led lasers to prevent scattering. It is used to analyse marine life without affecting them with traditional LiDARs.

Cristiano Premebida, et.al, [4] have proposed a LiDAR and camera based Pedestrian detection system. Here the lidar data is processed , and when a person is detected , it alerts the civilians , who drove through the roads.

Chakaravarthi S, et.al,[5] have proposed a detection and tracking technique using Superpixel extraction.

Charles R. Qi, et.al, [6] have proposed a neural network , called PointNet, that can analyse Point Cloud , a data format which is of irregular structure. The LiDAR that we use in this project also produces similar data as Point Clouds , which can be analysed using this PointNet

Joseph Redmon, et.al [7], introduced a unified real-time object detection framework, laying the foundation for rapid advancements in the field.

Wei Yang,et,al [8] , have proposed real-time detection algorithms for vehicles based on frameworks like YOLOv2, offering practical solutions for traffic management and safety. They have alo provided a comparison on various models used for detection.

Tibor Trnovszky [9] , et.al, have proposed a method to utilize Convolutional Neural Networks (CNNs) for animal recognition systems, demonstrating the efficacy of deep learning in wildlife monitoring. They also proposed the methods to implement it with various performance comparisons.

Gyanendra, et.al, [10] have proposed a CNN based wild animal detection technique and performs various monitoring and analysis based activities for research through camera trap networks.

Yuvaraj Munian, et.al, [11] have proposed a project on newer dimension for wild animals' auto-detection during active nocturnal hours using thermal image processing over camera car mount in the vehicle. To implement effective hot spot and moving object detection, obtained radiometric images are transformed and processed by an intelligent system.

B. Karthikeya Reddy, et.al.., [12] have proposed research work on YOLOV3 model to identify the animal present in the image given by user. The algorithm used in YOLOV3 model is darknet, which has a pre-trained dataset. The overall performance of the model is based on different training images and testing images of the dataset.

H. Nguyen. et.al,.. [13] have proposed a framework to build automated animal recognition in the wild, aiming at an automated wildlife monitoring system. In particular, we use a single-labeled dataset from Wildlife Spotter project, done by citizen scientists, and the state-of-the-art deep convolutional neural network architectures, to train a computational system capable of filtering animal images and identifying species automatically.

Hongchao Niu, et,al,.. [14] have proposed some improvement modifications to the existing YOLO v5 network and also have incorporated a variety of architectures for better detection and accuracy.

Mohammad Sadegh Norouzzadeh, et.al., [15] have proposed an active deep learning scheme that can significantly reduce the manual labor required to extract information from camera trap images. Automation of information extraction will not only benefit existing camera trap projects, but can also catalyze the deployment of larger camera trap arrays.

CHAPTER 3
SYSTEM ANALYSIS

3.1 Existing System

Existing System:

Wildlife threat mitigation in forested areas has historically relied on various methods, but these methods have often exhibited limitations and shortcomings:

Traditional Methods:

Traditional approaches to wildlife threat detection have primarily included the use of Passive Infrared (PIR) sensors and computer vision-based systems. PIR sensors are motion detectors that trigger alarms when they detect changes in heat signatures. While these sensors are affordable and easy to deploy, they can produce false alarms due to environmental factors like wind, vegetation movement, or changes in temperature. Computer vision-based systems analyze visual data but can also be prone to false positives and may require significant computational resources.

Communication Challenges:

In the existing systems, communication of threat alerts to local communities and authorities has often relied on conventional methods like telephone calls or sirens. These methods can be slow and unreliable, leading to delayed responses, which can be critical in wildlife threat situations.

Limited Sustainability:

Conventional systems are typically powered by grid electricity, which can be unreliable or unavailable in remote forested areas. This reliance on grid power can result in system downtime and reduced effectiveness during power outages.

3.2 Proposed system

Proposed System:

The proposed system introduces a comprehensive and innovative solution to address the limitations of existing systems for wildlife threat mitigation:

YOLO Animal Detection:

The proposed system integrates YOLO (You Only Look Once) for animal detection, leveraging state-of-the-art object detection algorithms. YOLO operates in real-time, capable of efficiently identifying and classifying animals within images or video streams. Its high accuracy ensures reliable detection, providing essential data for wildlife monitoring purposes.

Arduino Integration:

An Arduino microcontroller acts as a crucial component, facilitating seamless integration between YOLO-detected animal classes and the LoRa communication module. Connected to the PC, the Arduino receives animal class names post-detection by YOLO. This setup streamlines communication between the detection system and the data transmission module, enhancing overall system efficiency.

GPS Module:

The integration of GPS modules with the Arduino enhances the system's precision by providing accurate geospatial information about the location of detected animals. This feature enables precise tracking and monitoring of wildlife activity in forested areas, contributing to comprehensive wildlife management strategies.

LoRa Communication:

The system utilizes LoRa (Long Range) communication technology for wireless data transmission over extended distances. The Arduino packages the detected animal class name and corresponding GPS coordinates into data packets, enabling efficient transmission via the LoRa module to the receiving end. This approach ensures reliable communication in remote forested environments with limited infrastructure.

Receiving End:

At the receiving end, another Arduino with a LoRa module is connected to a computer. It receives data packets containing the animal class name and location coordinates transmitted by the sender Arduino. The received data is then forwarded to the computer for processing and display, facilitating real-time monitoring and analysis.

3.3 Feasibility Study

3.3.1 Technical feasibility:

The technical feasibility of implementing the proposed wildlife monitoring and alert system relies on the availability of technology, system integration capabilities, and the reliability of components. YOLO, Arduino, GPS modules, and LoRa communication technology are established technologies with proven track records in various applications.

Their compatibility and integration feasibility have been demonstrated in numerous projects, indicating a high level of technical feasibility. Additionally, the reliability and performance of these components have been extensively tested and proven in different environments, further supporting the technical feasibility of the project.

3.3.2 Economic Feasibility :

Evaluating the economic feasibility of the project involves assessing the cost of components, operational costs, and potential return on investment (ROI). The cost of components such as YOLO-compatible hardware, Arduino boards, GPS modules, and LoRa communication modules can vary based on specifications and suppliers.

Conducting a cost analysis considering these factors will determine the economic feasibility of the project. Additionally, operational costs such as maintenance, power consumption (especially in remote areas where solar panels are required), and potential licensing fees for software components should be considered. Assessing the potential benefits such as improved wildlife monitoring, reduced human-wildlife conflicts, and enhanced conservation efforts against the initial investment will help evaluate the economic feasibility of the project and determine the ROI.

3.3.3 Market Feasibility :

The market feasibility analysis aims to assess the demand, potential market size, and commercial viability of the proposed wildlife threat mitigation system using Camera and LoRaWAN technology.

3.3.4 Operational feasibility :

Operational feasibility hinges on factors such as user acceptance, skill requirements, and regulatory compliance. Stakeholder buy-in, including wildlife management authorities, local communities, and conservation organizations, is crucial for the success of the project. Assessing their willingness to adopt and use the system will determine its operational feasibility. Moreover, evaluating the technical expertise required to develop, deploy, and maintain the system is essential. Training programs or partnerships with organizations possessing the necessary skills can enhance operational feasibility. Furthermore, compliance with relevant regulations and standards governing wildlife monitoring, data privacy, and communication protocols is essential. Ensuring adherence to regulatory requirements will contribute to the operational feasibility of the project.

3.3 Software Environment :

The software environment for the proposed wildlife monitoring and alert system involves a combination of programming languages, libraries, and development tools to enable seamless integration and operation. Below are the key components of the software environment:

1. YOLO (You Only Look Once):

YOLO is typically implemented using deep learning frameworks such as TensorFlow, PyTorch, or Darknet.

Development and training of the YOLO model may require Python scripts, along with the respective deep learning framework and associated libraries.

2. Arduino Development Environment:

The Arduino IDE (Integrated Development Environment) is used for programming Arduino microcontrollers.

Arduino IDE supports C/C++ programming languages for writing code to control Arduino boards and interact with peripherals.

3. Python Programming Language:

Python is used for various tasks including data processing, communication with Arduino, and displaying information on the receiving computer.

Python libraries such as PySerial may be used for serial communication with Arduino, and libraries like NumPy and OpenCV may be utilized for data manipulation and image processing.

4. LoRa Communication Libraries:

Libraries specific to LoRa communication, such as the Arduino LoRa library, may be used for implementing LoRa communication protocols on Arduino boards.

CHAPTER 4
SYSTEM DESIGN

4.1. Architecture Diagram

The system execution starts with the webcam that acts as an input to the YOLO animal detection system. The transmitting LoRA module transmits the data received from the arduino and sends the location and the animal class data to the receiving LoRA where the message is presented as output.

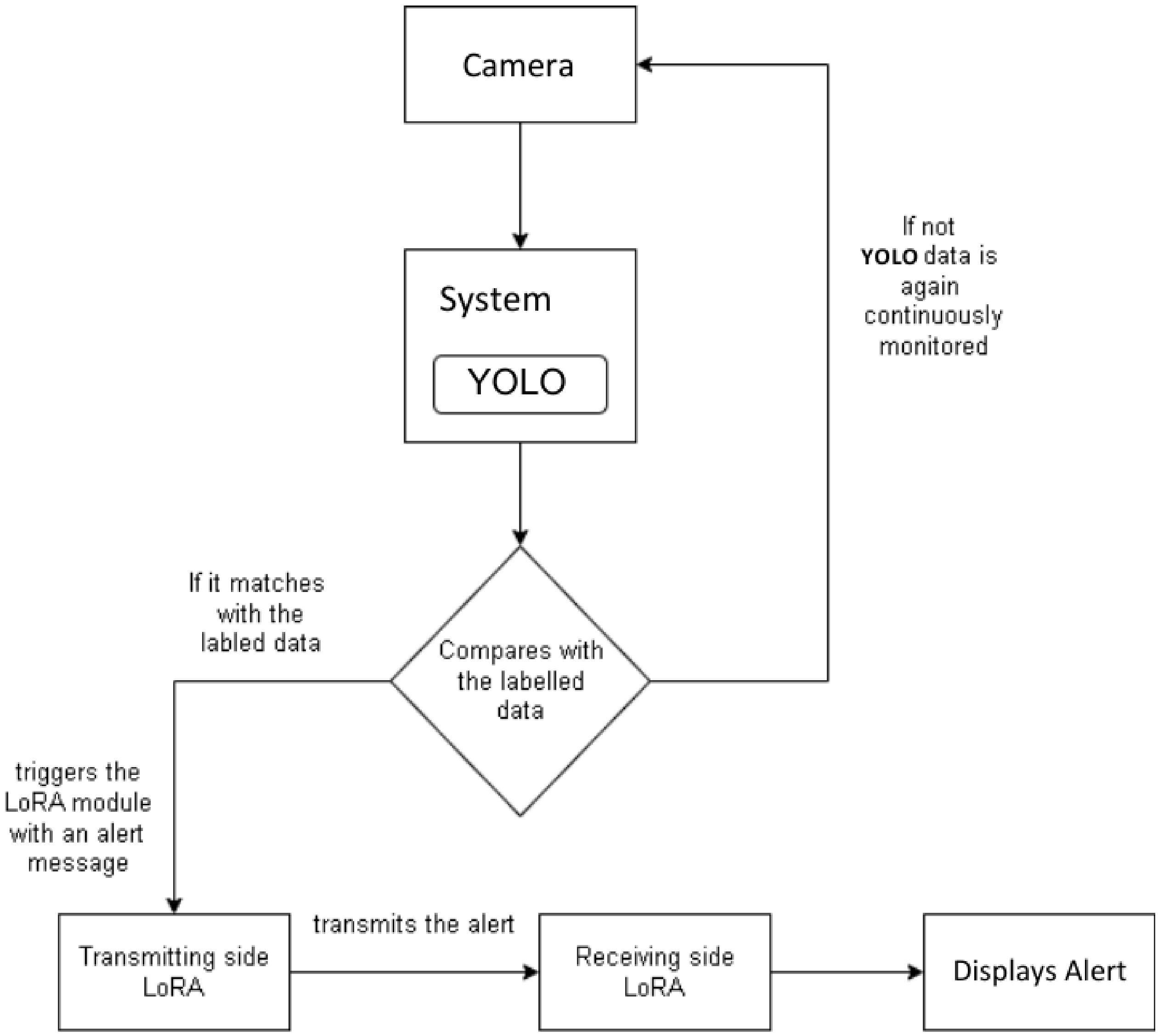


FIG 4.0

CHAPTER 5
SYSTEM ARCHITECTURE

The architecture of the wildlife threat mitigation system using Camera and LoRaWAN technology is a fundamental component that defines how the various hardware and software elements work together to achieve the system's objectives. This section provides an overview of the system's architecture, highlighting key components and their interactions.

1. Hardware Components:

Computer:

- The computer serves as the central processing unit of the system, responsible for executing the YOLO animal detection module and managing data processing tasks.

- It provides computational resources for running complex algorithms involved in real-time animal detection and analysis of captured images or video streams.

- Additionally, the computer may host software applications for user interaction, system monitoring, and data visualization.

Arduino Microcontroller:

- Acting as an intermediary between the computer and other hardware components, the Arduino microcontroller facilitates data transmission and communication within the system.

- It receives commands and data from the computer and interfaces with peripherals such as the GPS module and LoRa communication module.

- The Arduino executes control logic and manages the flow of data between different hardware elements, ensuring seamless operation of the system.

GPS Module:

- The GPS module plays a crucial role in enhancing the precision of wildlife monitoring by retrieving accurate location coordinates.

- It utilizes satellite signals to determine the device's geographical position, providing geospatial information essential for tracking and analyzing wildlife movements.

- The GPS module ensures that the system can accurately identify the location of detected animals, enabling effective response strategies and conservation efforts.

LoRa Communication Module:

- The LoRa communication module enables wireless data transmission over long distances, making it suitable for communication between remote locations.

- It utilizes LoRa (Long Range) technology to establish low-power, wide-area networks, allowing for reliable communication even in areas with limited infrastructure.

- The LoRa module facilitates the exchange of data packets containing animal detection information, location coordinates, and system status updates between different nodes within the network.

Receiving Arduino and Computer:

- The receiving Arduino and computer form the endpoint of the communication system, responsible for receiving, processing, and displaying data transmitted by the sender Arduino.

- Upon receiving data packets from the sender Arduino, the receiving Arduino processes the information and forwards it to the connected computer.

- The computer processes the received data further, generating real-time monitoring information and displaying it to users through graphical interfaces or console outputs.

2. Software Components:

YOLO:

- YOLO (You Only Look Once) is a deep learning-based object detection algorithm widely used for real-time animal detection in images or video streams.

- It employs convolutional neural networks (CNNs) to detect and classify objects within images with remarkable accuracy and speed.

- YOLO is particularly well-suited for wildlife monitoring applications due to its ability to detect multiple objects simultaneously and its robust performance in varying environmental conditions.

Arduino IDE:

- The Arduino Integrated Development Environment (IDE) is a software platform used for programming Arduino microcontrollers and interfacing with hardware components.
- It provides a user-friendly interface for writing, compiling, and uploading code to Arduino boards, simplifying the development and deployment process.
- Arduino IDE supports a wide range of programming languages, including C and C++, making it accessible to both novice and experienced developers.

Python:

- Python is a versatile scripting language commonly used for data processing tasks, serial communication with Arduino, and display of monitoring information in wildlife monitoring systems.
- It offers a rich ecosystem of libraries and frameworks, including NumPy, OpenCV, and PySerial, which are essential for image processing, serial communication with hardware components, and real-time data visualization.
- Python's simplicity, readability, and extensive library support make it an ideal choice for implementing complex algorithms and interfacing with hardware in wildlife monitoring applications.

Communication Libraries:

- Communication libraries such as PySerial and LoRa communication libraries play a vital role in facilitating seamless communication between hardware components in the wildlife monitoring system.
- PySerial provides a Python interface for serial communication with Arduino microcontrollers, allowing for the exchange of data packets and commands between the computer and Arduino boards.
- LoRa communication libraries enable the implementation of LoRaWAN communication protocols, enabling wireless data transmission over long distances between remote nodes in the network. These libraries abstract low-level communication details, simplifying the development of communication modules in wildlife monitoring systems.

3. System Workflow:

The system workflow outlines the sequence of operations and interactions among the hardware and software components to achieve the desired outcomes. It begins with YOLO detecting animals in real-time from the video feed, followed by the transmission of detected animal class names to the Arduino. The Arduino retrieves current location coordinates from the GPS module and packages the data into data packets for transmission via the LoRa communication module. At the receiving end, another Arduino receives the data packets and forwards them to the connected computer, where a Python script processes and displays the received data in the console. This systematic workflow ensures real-time monitoring and alerting, enabling timely response to potential threats.

4. Security Measures:

- Data Encryption: All communication between the system components and with external entities is encrypted to protect sensitive information from unauthorized access.

- User Authentication: User access to the system is secured through authentication mechanisms to prevent unauthorized control or monitoring.

5. Reporting and Analytics:

Reporting Tools: Reporting and analytics tools are integrated into the system for generating reports on wildlife activity, system performance, and incident response.
The system's architecture is designed to provide accurate wildlife threat detection and rapid alerting while being sustainable and user-friendly. It leverages advanced technologies in data processing and communication to address the increasing challenges of human-wildlife conflicts in forested areas, ultimately contributing to the safety and well-being of local communities

5.2 Algorithm:

Step 1: Data Acquisition:

1.1. Capture Images or Video Streams:

- Use a camera connected to the system to capture images or video streams of the monitored area.
- Ensure that the camera provides clear and high-resolution footage to facilitate accurate animal detection.

1.2. Retrieve Animal Detection Results:

- Utilize the YOLO (You Only Look Once) algorithm running on the computer to analyze the captured images or video frames.
- Extract animal detection results, including detected animal class names and bounding box coordinates, from the output of the YOLO algorithm.

Step 2: Data Processing:

2.1. Extract Detected Animal Information:

- Parse the output of the YOLO algorithm to extract relevant information such as detected animal class names and their corresponding bounding box coordinates.
- Ensure proper handling of multiple detections and consideration of confidence scores to filter out low-confidence predictions.

2.2. Refine Detection Results:

- Apply post-processing techniques such as non-maximum suppression to eliminate redundant detections and improve detection accuracy.
- Implement confidence thresholding to filter out detections below a certain confidence level, ensuring the reliability of detection results.

Step 3: Location Acquisition:

3.1. Retrieve GPS Coordinates:

- Interface with the GPS module connected to the Arduino to retrieve accurate location coordinates of the monitoring site.
- Ensure that the GPS module provides reliable and precise geospatial information to enhance the accuracy of wildlife monitoring.

3.2. Integrate GPS Data with Detection Results:

- Combine the retrieved GPS coordinates with the detected animal information to determine the precise location of detected animals within the monitored area.
- Associate each detected animal with its corresponding geographical coordinates to enable spatial analysis and mapping.

Step 4: Data Packaging:

4.1. Package Detected Animal Information:

- Aggregate the detected animal class names, bounding box coordinates, and GPS coordinates into structured data packets.
- Define a standardized data format for packaging the information to ensure compatibility and interoperability with other system components.

Step 5: Wireless Transmission:

5.1. Utilize LoRa Communication Technology:

- Employ LoRa (Long Range) communication technology for wireless transmission of data packets over long distances.
- Configure LoRa parameters such as data rate, spreading factor, and frequency band to optimize communication performance and range.

5.2. Ensure Data Integrity:

- Implement error-checking mechanisms such as cyclic redundancy check (CRC) to ensure data integrity during transmission.
- Include error correction codes or packet acknowledgment mechanisms to mitigate the effects of transmission errors.

Step 6: Data Reception:

6.1. Receive Data Packets:

- Receive the transmitted data packets containing detected animal information and GPS coordinates at the receiving Arduino.
- Validate the received data packets for completeness, consistency, and adherence to the defined data format.

6.2. Forward Data to Computer:

- Forward the validated data packets to the connected computer for further processing, analysis, and visualization.
- Ensure reliable and timely delivery of data from the receiving Arduino to the computer for real-time monitoring.
- This breakdown provides a comprehensive understanding of each step in the wildlife monitoring and alert system, highlighting the specific tasks and considerations involved in data acquisition, processing, transmission, reception, and integration.

CHAPTER 6
SYSTEM IMPLEMENTATION

6.1 Program / Code

Python Code (Sender side):

```
import cv2
import math
import serial
import time
from ultralytics import YOLO

# Add a delay before opening the camera to allow it to initialize
time.sleep(5)  # 5 seconds delay

# Start USB camera (assuming index 1)
cap = cv2.VideoCapture(0)

# Model
model = YOLO("yolo-Weights/yolov8n.pt")

# Object classes
classNames = ["bird", "cat","dog", "horse", "sheep", "cow", "elephant", "bear",
"zebra", "giraffe"]

# Initialize serial connection with Arduino
arduino = serial.Serial('COM3', 9600, timeout=1)

while True:
    success, img = cap.read()
    results = model(img, stream=True)

    # Coordinates
    for r in results:
        boxes = r.boxes

        for box in boxes:
            # Bounding box
            x1, y1, x2, y2 = box.xyxy[0]
            x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2)

            # Class name
            cls = int(box.cls[0])
            class_name = classNames[cls]

            # Send data if any animal class is detected
```

```
            if class_name in classNames[16:]:  # Checking if the detected class is an
#animal
            # Send signal to Arduino
            arduino.write(b'E')

            # Wait for Arduino to process signal
            time.sleep(1)

            # Send class name to Arduino
            arduino.write(class_name.encode())

        # Put box in cam
        cv2.rectangle(img, (x1, y1), (x2, y2), (255, 0, 255), 3)

        # Confidence
        confidence = math.ceil((box.conf[0] * 100)) / 100
        print("Confidence --->", confidence)

        # Object details
        org = [x1, y1]
        font = cv2.FONT_HERSHEY_SIMPLEX
        fontScale = 1
        color = (255, 0, 0)
        thickness = 2

        cv2.putText(img, class_name, org, font, fontScale, color, thickness)

    cv2.imshow('USB Camera', img)
    if cv2.waitKey(1) == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

Arduino Code:

```
#include <NeoGPS.h>
#include <SoftwareSerial.h>

#define GPS_RX_PIN 3
#define GPS_TX_PIN 4

NeoGPS::Location_t currentLocation;

// Create software serial object to communicate with GPS module
```

```
SoftwareSerial gpsSerial(GPS_RX_PIN, GPS_TX_PIN);

// Create NeoGPS object
NeoGPS::NMEAGPS gps;

// Create LoRa software serial object
SoftwareSerial LoRaSerial(2, 5); // RX, TX

void setup() {
  Serial.begin(9600);
  gpsSerial.begin(9600);
  LoRaSerial.begin(9600);
}

void loop() {
  // Process incoming GPS data
  while (gpsSerial.available() > 0) {
    if (gps.encode(gpsSerial.read())) {
      // If new GPS data is parsed successfully
      if (gps.location.isValid()) {
        currentLocation = gps.location;
      } else {
        Serial.println("GPS data is not valid");
      }
    }
  }

  if (Serial.available() > 0) {
    char signal = Serial.read();
    if (signal == 'E') {
      // Read class name sent by Python
      String className = "";
      while (LoRaSerial.available() > 0) {
        char c = LoRaSerial.read();
        if (c == '\n') {
          break;
        }
        className += c;
      }

      // Combine class name and GPS coordinates
      String dataToSend = className + "," + getGPSData();

      // Send data via LoRa module to the receiver side
```

```
      LoRaSerial.println(dataToSend);
    }
  }
}

String getGPSData() {
  // Return GPS coordinates as a string
  String latitude = String(currentLocation.lat(), 6);
  String longitude = String(currentLocation.lon(), 6);
  return latitude + "," + longitude;
}
```

Python Code (Receiver side):

```
import serial

# Initialize serial connection with LoRA module
lora = serial.Serial('COM4', 9600, timeout=1)

while True:
    if lora.in_waiting > 0:
        data = lora.readline().decode().strip()
        print("Received:", data)
```

CHAPTER 7
SYSTEM TESTING

System testing is a critical phase of the project to ensure that the entire animal detection and alerting system functions correctly as a whole. Here's a system testing plan for your project:

1. Functional Testing:

    1.1 YOLO Animal Detection Module:

    - Validate the YOLO algorithm's ability to accurately detect animals in images or video streams.
    - Verify that the YOLO algorithm triggers the Arduino to process detection results effectively.

    1.2 Arduino Integration:

    - Ensure that the Arduino correctly receives and processes data from the computer, including animal class names and location coordinates.
    - Validate the Arduino's ability to interface with peripherals such as the GPS module and LoRa communication module.

2. Communication Testing:

    2.1 LoRa Communication Module:

    - Test the communication between the Arduino and LoRa module to ensure successful transmission of data packets.
    - Verify that alerts are transmitted over long distances reliably using LoRa technology.

    2.2 Alert Notification (Optional):

    - If utilizing a GSM module for alerting nearby communities, test the communication between the Arduino and GSM module.
    - Confirm that SMS alerts are sent promptly upon detecting animals.

3. Accuracy Testing:

    3.1 Detection Accuracy:
    - Collect data on the presence of animals within the detection range and compare it with the system's detection results.
    - Evaluate the system's accuracy in classifying and alerting for each animal type.

4. Environmental Testing:

    4..1 Adverse Weather Conditions:
    - Test the system's performance under various environmental conditions such as rain, fog, and low-light conditions.
    - Ensure that the system functions reliably in adverse weather without compromising detection accuracy.

5. False Positive and False Negative Testing:

    - Identify and rectify instances of false positives (inanimate objects triggering alerts) and false negatives (animals not detected).
    - Adjust algorithm parameters or sensor settings to minimize these errors.

6. Power Efficiency Testing:

    - Assess the power consumption of the system components, including Arduino, GPS module, and LoRa module.
    - Ensure that the system operates efficiently and can sustain prolonged operation in remote locations, especially when powered by solar panels.

7. Integration Testing:

    - Test the integration of all hardware and software components, including YOLO, Arduino, GPS module, and LoRa module.
    - Verify seamless communication and data flow between components, ensuring that they work together harmoniously.

8. Remote Alert Testing:

    - Confirm that alerts are transmitted to the designated recipient (e.g., nearby communities) in a timely manner.
    - Validate the reception and acknowledgment of alerts by the recipient, whether through LoRa communication or SMS notifications.

9. Long-Term Reliability Testing:

    - Run the system continuously for an extended period to assess its long-term reliability and stability.
    - Monitor for any potential issues or degradation in performance over time, especially regarding hardware durability and software robustness.

10. Usability and User Testing:

- Engage end-users or stakeholders to evaluate the system's user-friendliness, ease of setup, and effectiveness in alerting.
- Gather feedback on the user experience and identify areas for improvement to enhance overall usability.

11. Security Testing:

- Conduct security assessments to identify vulnerabilities and weaknesses in the system, especially regarding data transmission and access control.
- Implement measures to secure the system against unauthorized access or tampering, ensuring the confidentiality and integrity of sensitive information.

12. Documentation and Reporting:

- Document all test cases, procedures, and results in a comprehensive test report.
- Include details on any issues encountered, their resolutions, and overall system performance to facilitate future troubleshooting and maintenance.
- Ensure that the documentation is accessible and comprehensible for stakeholders and future development efforts.

Research & Discussion :

The YOLO model that we have deployed for this project is extremely fast because it does not deal with complex pipelines. It can process images at 45 Frames Per Second (FPS). In addition, YOLO reaches more than twice the mean Average Precision (mAP) compared to other real-time systems, which makes it a great candidate for real-time processing.

The architecture works as follows:

- Resizes the input image into 448x448 before going through the convolutional network.

A 1x1 convolution is first applied to reduce the number of channels, which is then followed by a 3x3 convolution to generate a cuboidal output.

- The activation function under the hood is ReLU, except for the final layer, which uses a linear activation function.
- Some additional techniques, such as batch normalization and dropout,

respectively regularize the model and prevent it from overfitting.

From the graphic below, we observe that YOLO is far beyond the other object detectors with 45 FPS.

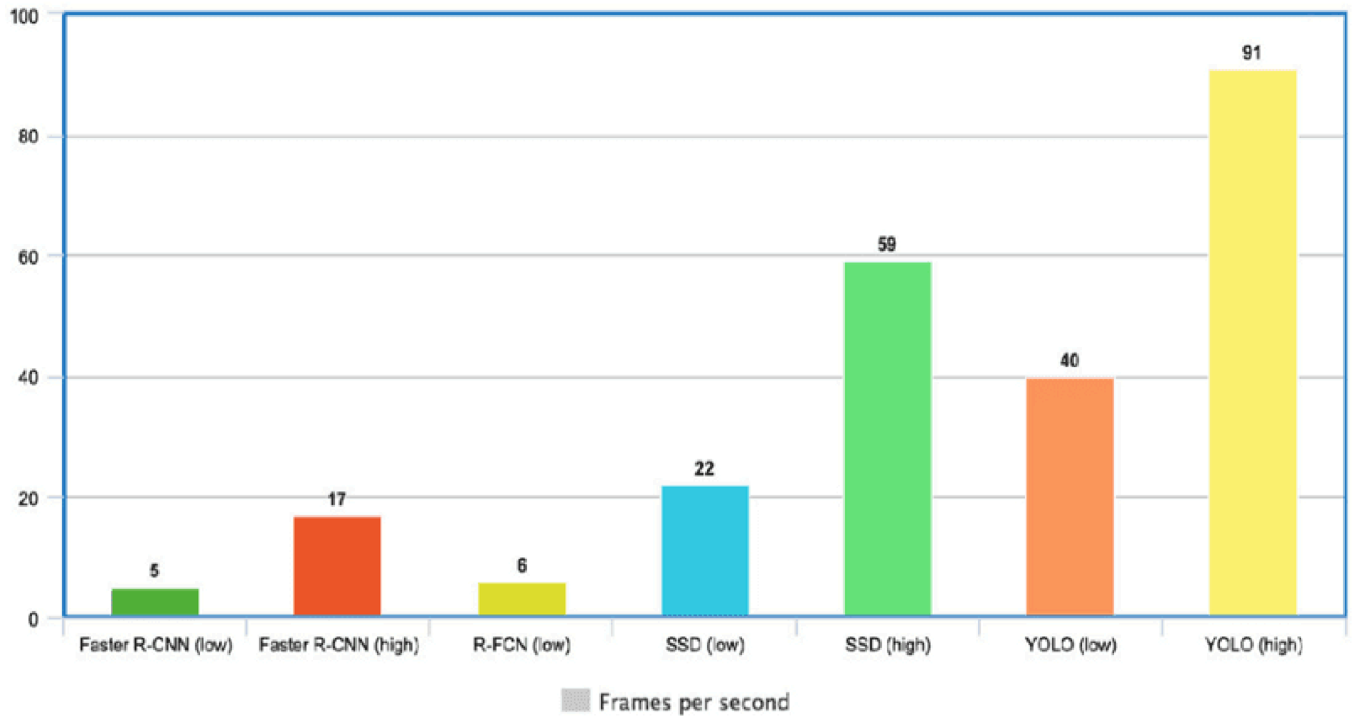


FIG 7.0

The current and the latest yolov8 model that we use, has a mean average precision of about 91.7% and achieves a minimum of 40 to 45 fps, which makes it ideal for the realtime animal monitoring and alerting. It is justified as follows:

Object Detection Performance Comparison (YOLOv8 vs YOLOv5)

Model Size	YOLOv5	YOLOv8	Difference
Nano	28	37.3	+33.21%
Small	37.4	44.9	+20.05%
Medium	45.4	50.2	+10.57%
Large	49	52.9	+7.96%
Xtra Large	50.7	53.9	+6.31%

*Image Size = 640

TABLE 7.0

CHAPTER 8
CONCLUSION

The culmination of this research presents a pioneering solution to the multifaceted issue of human-wildlife conflicts in forested areas, marking a significant stride forward in wildlife threat mitigation. Through the strategic integration of advanced technologies such as LiDAR and LoRaWAN communication, our system offers a comprehensive and innovative approach to addressing the pressing challenges of safety, coexistence, and conservation in these regions.

Integration of Key Components:

The meticulous integration of critical components, ranging from LiDAR sensors for precise detection to Raspberry Pi for robust data processing, along with sustainable energy sources like solar panels and batteries, and efficient communication modules such as LoRaWAN, constitutes the backbone of our system. These components operate synergistically to deliver real-time, accurate detection of wildlife threats and prompt dissemination of alerts to both local communities and relevant authorities.

Feasibility Study Insights:

Our comprehensive feasibility study underscores the technical prowess, economic viability, operational feasibility, market potential, and environmental sustainability of the proposed system. It illuminates the system's effectiveness in remote forested areas, where human safety, livestock protection, and conflict mitigation are paramount concerns. The results affirm the system's adaptability and practicality across diverse environmental landscapes.

Software Environment and Operational Efficacy:

The meticulously designed software environment not only facilitates seamless data processing and secure communication but also ensures efficient system management. Leveraging state-of-the-art machine learning algorithms for precise detection, reinforced by robust encryption and authentication protocols, our system guarantees dependable and secure operations. Moreover, its user-friendly interface, coupled with minimal maintenance requirements and seamless integration with existing infrastructure, enhances its operational efficacy and accessibility.

Market Feasibility and Adoption Potential:

An in-depth analysis of market feasibility unveils a significant demand for our system in regions grappling with human-wildlife conflicts. Its competitive advantages, cost-effectiveness, and scalability position it as a highly sought-after solution for governmental agencies, local communities, and conservation organizations alike. The system's potential for widespread adoption holds the promise of not only mitigating conflicts but also fostering a harmonious coexistence between humans and wildlife, thereby advancing broader conservation objectives.

In summation, the wildlife threat mitigation system utilizing LiDAR and LoRaWAN technology represents a paradigm shift in the domain of conservation and wildlife management. By enhancing the accuracy of threat detection, expediting alert transmission, and promoting sustainability, our system has the potential to catalyze transformative change in human-wildlife interactions. Its successful implementation stands to enhance safety, safeguard livelihoods, and cultivate a more harmonious environment for all stakeholders involved, thereby charting a course towards a more sustainable and harmonious future for humans and wildlife alike.

CHAPTER 9
REFERENCE

REFERECES:

1. R. Shanthakumari, C. Nalini, , et.al, 2022 International Mobile and Embedded Technology Conference (MECON).

2. Jingrong Chen, Hoa Xu, Jianqing Wc, Rui Yue, Changwei Yuan and LuWang, " Deer Crossing Road Detection With Roadside LiDAR Sensor " ,IEE Access,2019.

3. Elizaveta Dubrovinskaya , Fraser Dalgleish ,Bing Ouyang, Paolo Casari, " Underwater LiDAR Signal Processing for Enhanced Detection and Localization of Marine Life ",OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO) ,2018.

4. Cristiano Premebida, Oswaldo Ludwig,and Urbano Nunes, " LIDAR and Vision-BasedPedestrian Detection System ", Journal of Field Robotics, 2009.

5. Chakaravarthi S, Visu P, Ganesan R, "An Object Detection and Tracking Techniques Using Superpixel Extraction in Compressed Video Sequences", International Journal of Pure and Applied Mathematics , Volume 119, 2018.

6. Charles R. Qi, Hao Su, Kaichun Mo, Leonidas J. Guibas," PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation ",IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

7. Joseph Redmon, Santosh Divvala, Ross Girshick and Ali Farhadi, "You only look once: Unified real-time object detection", IEEE Conference on computer Vision and Pattern Recognition(CVPR), June. 2016.

8. Wei Yang, Ji Zhang, Hongyuan Wang and Zhongbao Zhang, "A vehicle real-time detection algorithm based on YOLOv2 framework", Proceedings on Real-Time Image and Video Processing, vol. 10670, 2018.

9. Tibor Trnovszky, Patrik Kamencay, Richard Orjesek, Miroslav Benco and Peter Sykora, "Animal Recognition System Based on Convolutional Neural Network", Advances in Electrical and Electronic Engineering, vol. 15, no. 3, 2017.
10. Gyanendra and Pragya Gupta, "Wild Animal Detection Using Deep Convolutional Neural Network", Proceedings of 2nd International Conference on Computer Vision & Image Processing, pp. 327-338, 2018.

11. Yuvaraj Munian, Antonio Martinez-Molina and Miltiadis Alamaniotis, "Intelligent System for Detection of Wild Animals Using HOG and CNN in Automobile Applications", 11th International Conference on Information Intelligence Systems and Applications, 2020.

12. B. Karthikeya Reddy, Shahana Bano, G. Greeshmanth Reddy and Rakesh Kommineni, "Convolutional Network based Animal Recognition using YOLO and Darknet", 6th International Conference on Inventive Computation Technologies (ICICT), 2021.

13. H. Nguyen, S. J. Maclagan, T. D. Nguyen, T. Nguyen, P. Flemons, K. Andrews, et al., "Animal recognition and identification with deep convolutional neural networks for automated wildlife

monitoring", IEEE International Conference on Data Science and Advanced Analytics (DSAA), 2017.

14. Hongchao Niu, Xiaobing Hu and Hang Li, "Improved YOLOv5 network-based object detection for anti-intrusion of gantry crane", 2nd International Conference on Control Robotics and Intelligent System, August 2021.

15. M.S. Norouzzadeh, D. Morris, S. Beery, N. Joshi, N. Jojic and J. Clune, "A deep active learning system for species identification and counting in camera trap images", Methods Ecol. Evol, vol. 12, pp. 150-161, 2021.

16. A. Yilmaz, G.N. Uzun, M.Z. Gurbuz and O. Kivrak, "Detection and Breed Classification of Cattle Using YOLO v4 Algorithm", In Proceedings of the 2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA) Kocaeli 25-27, pp. 1-4, 2021.

17. X. Yu, J. Wang, R. Kays, P. A. Jansen, T. Wang and T. Huang, "Automated identification of animal species in camera trap images", EURASIP J. Image Video Process, 2013, [online] Available: https://doi.org/10.1186/1687-5281-2013-52.

18. A. Dhillon and G. K Verma, "Convolutional neural network: a review of models methodologies and applications to object detection", Progress in Artificial Intelligence, vol. 9, no. 7553, pp. 85-112, 2020.

19. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., "Going deeper with convolutions", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-9, 2015.

20. Alexander Gomez Villa, Augusto Salazar and Francisco Vargas, "Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks", Ecological Informatics, vol. 41, pp. 24-32, 2017, ISSN 1574-9541.

PANIMALAR ENGINEERING COLLEGE

An Autonomous Institution & Affiliated to Anna University, Chennai
Approved by AICTE, New Delhi, Accredited by National Board of Accreditation(NBA)
Approved by UGC for 2(f) & 12(B) Status
Bangalore Trunk Road, Poonamallee, Chennai-600123.

Certificate of Participation

PETIEM

PECTEAM 2K24

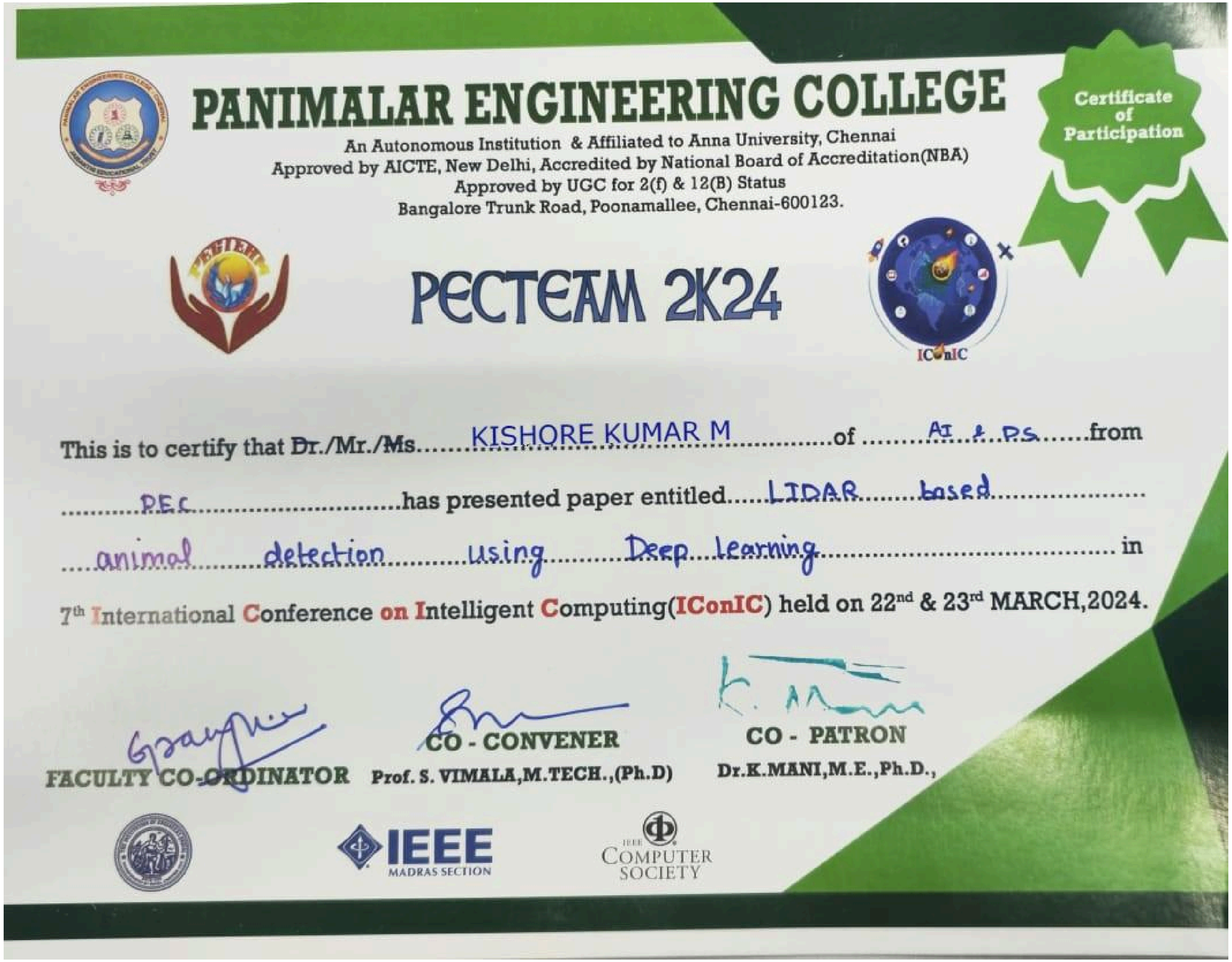
IConIC

This is to certify that ~~Dr.~~/Mr./~~Ms~~..... AAKASH . K. SABAI ....of .....AI & DS........from ....PEC..............................has presented paper entitled..... LIDAR .... based ....animal......... detection .... using .... Deep .... Learning.. in 7th International Conference on Intelligent Computing(IConIC) held on 22nd & 23rd MARCH,2024.

FACULTY CO-ORDINATOR

CO - CONVENER
Prof. S. VIMALA,M.TECH.,(Ph.D)

K. Mani
CO - PATRON
Dr.K.MANI,M.E.,Ph.D.,



PANIMALAR ENGINEERING COLLEGE

An Autonomous Institution & Affiliated to Anna University, Chennai
Approved by AICTE, New Delhi, Accredited by National Board of Accreditation(NBA)
Approved by UGC for 2(f) & 12(B) Status
Bangalore Trunk Road, Poonamallee, Chennai-600123.

Certificate of Participation

PECTEAM 2K24

ICONIC

This is to certify that ~~Dr.~~/Mr./~~Ms.~~ KISHORE KUMAR M of AI & DS from PEC has presented paper entitled LIDAR based animal detection using Deep learning in 7th International Conference on Intelligent Computing(IConIC) held on 22nd & 23rd MARCH,2024.

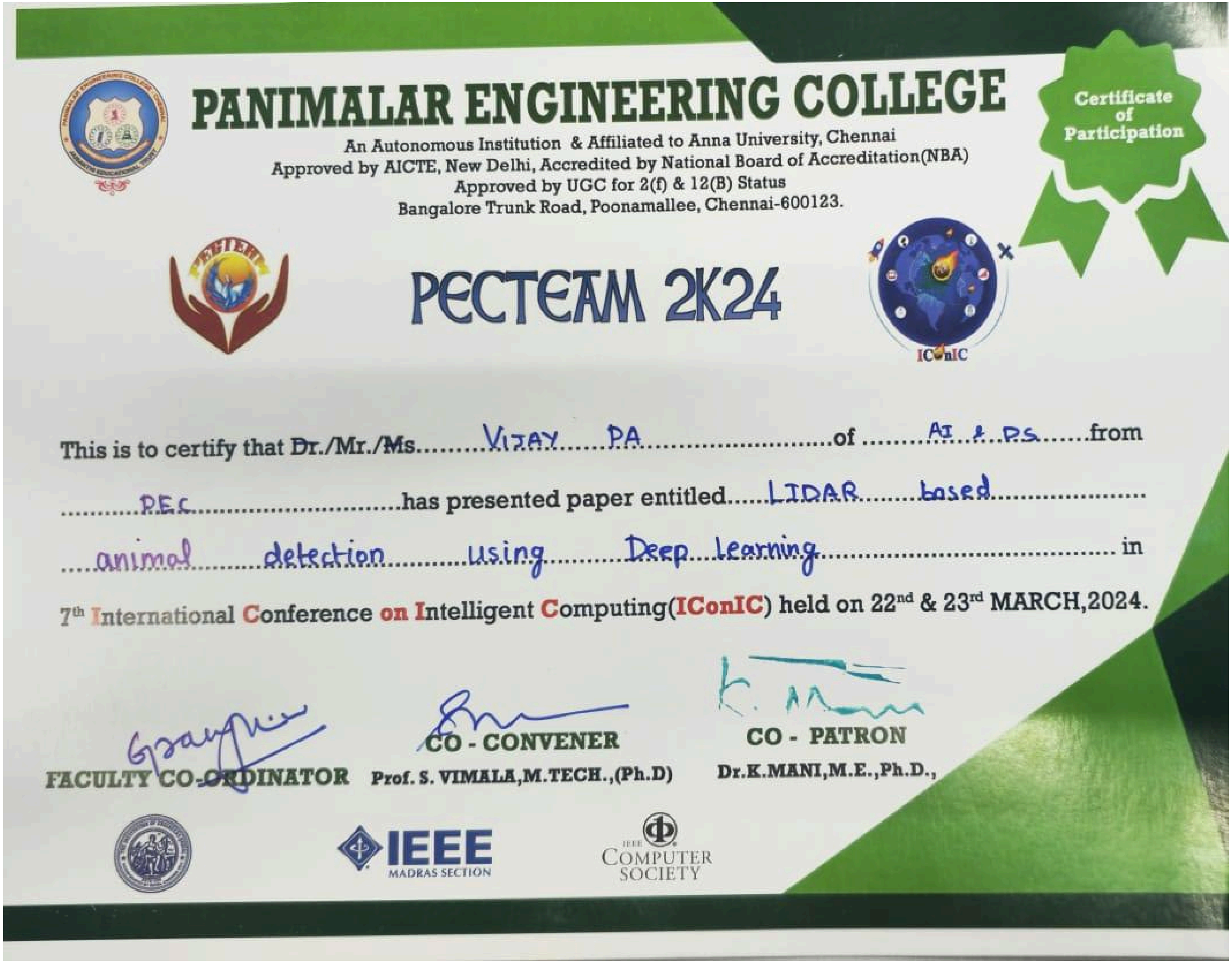
FACULTY CO-ORDINATOR

CO - CONVENER
Prof. S. VIMALA,M.TECH.,(Ph.D)

CO - PATRON
Dr.K.MANI,M.E.,Ph.D.,

IEEE MADRAS SECTION

IEEE COMPUTER SOCIETY



PANIMALAR ENGINEERING COLLEGE

An Autonomous Institution & Affiliated to Anna University, Chennai
Approved by AICTE, New Delhi, Accredited by National Board of Accreditation(NBA)
Approved by UGC for 2(f) & 12(B) Status
Bangalore Trunk Road, Poonamallee, Chennai-600123.

Certificate of Participation

PECTEAM 2K24

ICONIC

This is to certify that Dr./Mr./Ms. VIJAY PA of AI & DS from PEC has presented paper entitled LIDAR based animal detection using Deep learning in 7th International Conference on Intelligent Computing(IConIC) held on 22nd & 23rd MARCH,2024.

FACULTY CO-ORDINATOR

CO - CONVENER
Prof. S. VIMALA,M.TECH.,(Ph.D)

CO - PATRON
Dr.K.MANI,M.E.,Ph.D.,

IEEE MADRAS SECTION

IEEE COMPUTER SOCIETY

Camera based Wild Animals Detection and Alerting system near Wildlife Fences using Deep Learning

Vijay. Pa [1] , Aakash K Sabai [2] , Kishore Kumar M [3], Dr. K. Jayashree [4], Narmadha B [5]

Department of Artificial Intelligence & DataScience, Panimalar Engineering College, Chennai, TamilNadu, India [1,2,3,4,5]

vijaypaofficial@gmail.com [1] , aakashk.offl@gmail.com [2] , kishorem.offl@gmail.com [3] ,
k.jayashri@gmail.com [4] , narmadhab16@gmail.com [5]

Abstract: In the past few years, large scale deforestation has been in the play and as a result, wild animals have lost their living habitat and have started to invade the villages or the people living areas nearby those forest areas. As a result of this there has been a potential threat to the lives of people in those areas . Hence we are in need of an animal detection and alerting system , which could alert as soon as the animals have invaded the living habitat. This paper has proposed such an alert system with a Camera based sensing system along with PointNet deep learning framework to effectively detect animals with IoT devices for data transmission for earliest alerting.

1. INTRODUCTION:

The people living near forest areas have always been facing issues related to Wild animal attacks , which has been increasing for the past few years. Government and various NGOs have been taking several measures to prevent and alert those people through various measures and technological implementations. A few such common implementations are Computer Vision based, PIR sensor based Alert systems which detect based on machine learning. Though there are several such implementations , they are not quite effective when it comes to practical implementation.

Hence we have proposed a much more accurate system that uses Camera data , which after being processed , sends the alert to a remote place wirelessly as a message to the people living nearby.

2. LITERATURE SURVEY:

William D Simonson, et.al,[1] have proposed a Camera based animal species assessment to effectively impose various management actions for animal species protection. Here they have also proposed various remote sensing techniques for Camera based monitoring. Jingrong Chen, et.al, [2] have proposed a method for detection of Deer while crossing the roads using Camera. The Camera data is continuously processed , and when a deer comes within the range of the Camera sensor, it recognizes and alerts the drivers who travel nearby.

Elizaveta Dubrovinskaya, et.al, [3] have proposed a new Camera design with some inexpensive components, that uses low power led lasers to prevent scattering. It is used to analyse marine life without affecting them with traditional Cameras. Cristiano Premebida, et.al, [4] have proposed a Camera and camera based Pedestrian detection system. Here the Camera data is processed , and when a person is detected , it alerts the civilians , who drove through the roads. Chakaravarthi S, et.al,[5] have proposed a detection and tracking technique using Superpixel extraction.

Charles R. Qi, et.al, [6] have proposed a neural network , called PointNet, that can analyse Point Cloud , a data format which is of irregular structure. The Camera that we use in this project also produces similar data as Point Clouds , which can be analysed using this PointNet

3. PROPOSED ARCHITECTURE:

3.1 Proposed System:

Our system consists of both detection as well as transmission sections. The detection section detects the presence of any animal that enters within its range and the transmission section transmits any alert intimated by the detection section, to the people
near that area.

The various requirements needed for this system are :

- Camera
- Raspberry Pi 4 Model B
- PointNet (Python Library)
- Solar panels
- Batteries
- LoRaWAN
- GSM module

VLP-16 Camera :

This is a cost effective Camera sensor with 360 degree field of view which can create a 3D point cloud of about 6,00,000 points/ second with the help of its 16 laser/detector pairs which can rapidly scan the environmental data within a range of up to 328 ft.

Raspberry Pi 4 Model B:

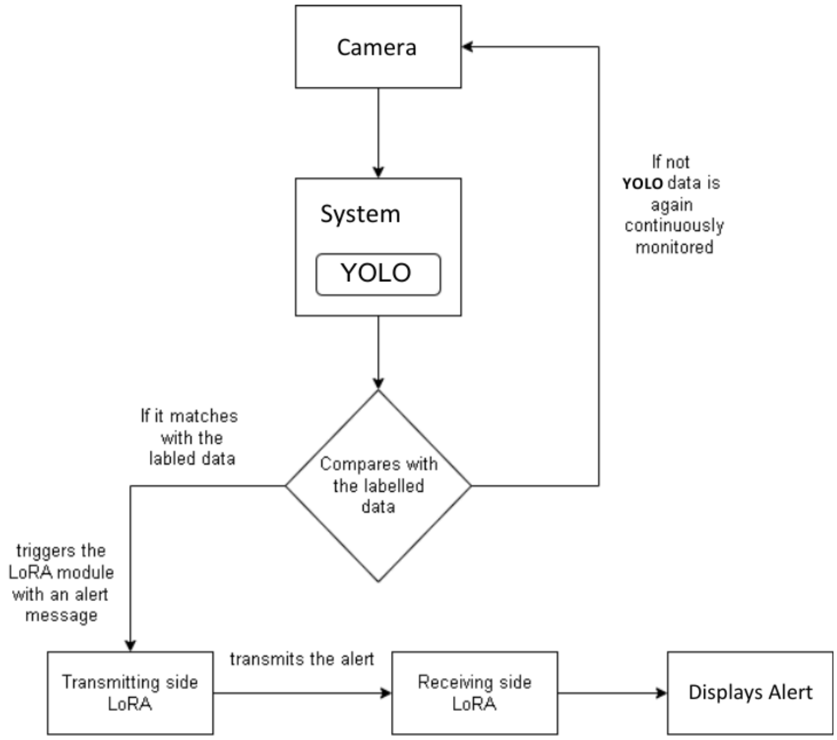
This particular model of Raspberry Pi board has a higher clock speed and suitable for implementing ML models which in our case, is suitable for processing real time point cloud data

PointNet:

It is a python model that can process real time 3D point cloud data, which in our case suitable for realtime animal detection.

The Raspberry Pi board is powered by the batteries that are supported by the solar panels. The various other components are connected to the board.

3.2 Architecture Diagram :



3.3 Working :

I. Detecting Section:

Raspberry Pi 4 Model B board loaded with a pretrained model using PointNet plays a major role in detecting the wildlife in our system. Unlike other traditional systems that use PIR sensors or Cameras , our system uses a VLP-16 Camera sensor that maps the real time data with much more accuracy. If any animal enters within the range of the Camera, and if it matches with the labelled data stored in the trained model , it intimates the LoRA module.

II. Transmission Section:

The LoRA module plays an important role when it comes to message transmission. If the model predicts the presence of any of the labelled data, the the Raspberry Pi sends an alert message to the LoRA module which inturn transmits to the LoRA module present at the receiver side which with the help of GSM module , SMS can be sent to the people nearby to alert them regarding the approach of those animals.

4. IMPLEMENTATION AND RESULT:

In this proposed system , we have used VLP-16 Camera to gather the environment data which has many advantages over the traditional methods of acquiring it. Some of the major advantages include accurate 3D spatial data, Low Light operation, all weather support, minimal false positives, low power consumption, etc.

The PointNet deep learning framework that we have used here, can handle the data from the Camera, which is also in Point Cloud format. It doesn't require as much as labelled data required by other deep learning models like Convolutional Neural Networks (CNN), which makes it easier to train the model using Python.

As from the paper [6] , we can infer that this PointNet model is said to have an accuracy of about 89.2% when it comes to classification based on the ModelNet40 benchmark. This justifies the use of this model for our implementation.

	input	#views	accuracy avg. class	accuracy overall
SPH [11]	mesh	-	68.2	-
3DShapeNets [25]	volume	1	77.3	84.7
VoxNet [15]	volume	12	83.0	85.9
Subvolume [16]	volume	20	86.0	**89.2**
LFD [25]	image	10	75.5	-
MVCNN [20]	image	80	**90.1**	-
Ours baseline	point	-	72.6	77.4
Ours PointNet	point	1	86.2	**89.2**

Fig 4.1 ModelNet40 benchmark result for classification of data

5. CONCLUSION AND FUTUREWORKS :

Thus we have provided an innovative method for animal detection and alerting that leverages the use of 3D spatial data with the help of Camera. The integration of Camera and the PointNet deep learning model, a practical, effective and a more accurate method for precisely identifying the animals has been made possible.

Additionally, the incorporation of LoRA has ,made it possible to transmit the message to a much longer distance, as the use of network is quite impossible in the forest side village areas. Along with the GSM module, it is made easier to send messages to the villagers as soon as the animal is locates metres away from them.

Our current work is majorly limited to the detection of elephants, as they are the major threats to life and property. In future, with upon more data and resources, we would develop our system to detect a wide variety of animals including tigers, bears, etc, which could also potentially cause a threat.

REFERENCES:

1. Will D SImonson, Harriet D.Allen , et.al, Applications of airborne Camera for the assessment of animal species diversity, Methods in Ecology and Evolution 2014.

2. Jingrong Chen, Hoa Xu, Jianqing Wc, Rui Yue, Changwei Yuan and LuWang, “ Deer Crossing Road Detection With Roadside Camera Sensor “ ,IEE Access,2019.

3. Elizaveta Dubrovinskaya , Fraser Dalgleish ,Bing Ouyang, Paolo Casari, “ Underwater Camera Signal Processing for Enhanced Detection and Localization of Marine Life “,OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO) ,2018.

4. Cristiano Premebida, Oswaldo Ludwig,and Urbano Nunes, “ Camera and Vision-BasedPedestrian Detection System “, Journal of Field Robotics, 2009.

5. Chakaravarthi S, Visu P, Ganesan R, “An Object Detection and Tracking Techniques Using Superpixel Extraction in Compressed Video Sequences”, International Journal of Pure and Applied Mathematics , Volume 119, 2018.

6. Charles R. Qi, Hao Su, Kaichun Mo, Leonidas J. Guibas,“ PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation “,IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.