International Journal of Scientific Research and Engineering Development-– Volume 8 Issue 2, Mar-Apr 2025

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

FARMER's SAFETY FROM REPTILES

Pasupuleti Sobhana Deepthi[1], Dr. N. Sambasiva Rao[2], Devapongu Manoj Kumar[3], Ulasa Naveen[4], Meesala Venkatesh[5], Neelapu Rajesh[6]

Electrical & *Electronic Engineering, NRI Institute of Technology, JNTU – Kakinada,* India[123456].

----------**********----------

Abstract:

This paper presents "Farmer's Safety From Reptiles" that is an IoT-based solution designed to enhance farmer safety by detecting venomous snakes in real-time. It integrates PIR motion sensors, thermal cameras, and acoustic sensors with a Tensor Flow Lite-based AI model for accurate detection. Upon identifying a snake, the system triggers local alerts and sends mobile notifications via a Wi-Fi module. Powered by solar panels and battery backups, it ensures reliable operation in remote fields. Future enhancements include edge AI accelerators and LoRa communication. This system demonstrates the potential of IoT and AI technologies for agricultural safety and broader applications.

Keywords **— Tensor Flow Lite, PIR sensor, thermal imaging, wireless communication, edge computing, solar power, LoRa, real-time alerts.**

----------**********----------

I. INTRODUCTION

The snake bites pose a significant threat to farmers working in agricultural fields, particularly in rural and remote areas where access to immediate medical assistance is limited. According to the World Health Organization (WHO), snake bites cause an estimated 81,000 to 138,000 deaths annually, with many more victims suffering from permanent disabilities. Farmers are especially vulnerable due to frequent encounters with venomous snakes in crops, tall grass, and irrigation systems. Traditional prevention methods, such as manual detection or protective gear, are often ineffective, highlighting the need for an automated, real-time solution to mitigate this critical issue.

The "Farmer's Safety From Reptiles" addresses this challenge by leveraging IoT (Internet of Things) sensors, AI-based image processing, and wireless communication technologies. The system integrates PIR motion sensors, thermal cameras, and acoustic sensors to detect snakes, while a TensorFlow Lite-based object detection model validates their presence.

Upon detection, the system triggers local alerts (buzzers and LEDs) and sends mobile notifications to farmers and a central monitoring unit. Additionally, a human body temperature monitoring module detects sudden changes indicative of a snake bite, enabling rapid emergency response. This multi-layered approach ensures both prevention and post-bite intervention, significantly enhancing farmer safety.

This project represents a significant advancement in the application of IoT and AI technologies for agricultural safety. By combining real-time detection, wireless communication, and edge computing, the system provides a scalable and cost-effective solution for rural communities. Future enhancements, such as edge AI accelerators, LoRa communication, and night vision capabilities, aim to further improve its reliability and effectiveness. The system not only addresses the immediate problem of snake bites but also demonstrates the potential of technology to solve critical challenges in agriculture and public health.

1.**PIR Motion Sensor**

Role: It detects infrared radiation emitted by moving objects in the field.

Purpose: Triggers the camera to capture images upon detecting motion, enabling real-time monitoring and reducing unnecessary image processing.

2. **Raspberry Pi Camera**

Role: It captures high-resolution images of the detected object.

Purpose: Provides visual input for the AI-based snake detection model, ensuring accurate identification and validation of snake presence.

3. **Thermal Camera**

Role: It detects infrared radiation to identify heat signatures of objects.

Purpose: Validates snake presence by analyzing thermal patterns, reducing false positives caused by non-snake movements or environmental factors.

4. **Acoustic Sensor**

Role: It detects frequency patterns of hissing sounds emitted by snakes.

Purpose: Enhances detection accuracy by identifying snake-specific audio signals, complementing visual and thermal data for multi-sensor validation.

5. **Raspberry Pi 4**

Role: It processes captured images using a Tensor Flow Lite-based AI model.

Purpose: Acts as the central processing unit for real-time image analysis, classification, and decision-making, ensuring efficient and accurate snake detection.

6. **Arduino Nano**

Role: It monitors and records human body temperature using a thermal sensor.

Purpose: Detects sudden temperature fluctuations indicative of a snake bite, enabling rapid emergency response and medical intervention.

7. **Wi-Fi Module (ESP8266)**

Role: It facilitates wireless data transmission between devices and mobile apps.

Purpose: Enables real-time communication of alerts and notifications to farmers and monitoring centre's, ensuring timely response to snake detection.

8. **Buzzer & LED**

Role: It generates audible and visual alerts in the field.

Purpose: Provides immediate local notifications to farmers when a snake is detected, ensuring quick action and reducing the risk of snake bites.

9. **Solar Panel**

Role: It Converts solar energy into the electrical power for the system.

Purpose: Ensures continuous and sustainable operation of the system in remote agricultural fields without reliance on grid power.

10. **Battery Backup**

Role: It stores electrical energy for uninterrupted system operation.

Purpose: Maintains system functionality during power outages, low sunlight conditions, or nighttime, ensuring reliable and consistent performance.

II. IMPLEMENTATION

Implementation of the Farmer's Safety From Reptiles involves both hardware setup and software integration to ensure accurate and efficient snake detection in agricultural fields. The system is designed to operate autonomously, integrating various sensors, processing units, a wireless communication module, and an alert mechanism. The primary goal of the implementation is to create a self-sustaining, energy-efficient, and real-time monitoring system that provides instant alerts to farmers and monitoring centre's. The implementation process is divided into two main parts: the hardware setup and the software integration, both of which are essential for the proper functioning of the system.

Software Implementation

Software is developed using Python for the Raspberry Pi, C++ for the Arduino Nano, and Firebase or MQTT for real-time notifications.

Code for Raspberry Pi 4model B

```
Import os
Import xml.etree.ElementTree as ET
Import cv2
Import RPi.GPIO as GPIO
From time import sleep
Import subprocess
# GPIO Pin Setup
```



```
MOTION_SENSOR_PIN = 17  # GPIO pin for motion sensor
BUZZER_PIN = 18  # GPIO pin for buzzer
LED_PIN = 27  # GPIO pin for LED
ARDUINO_TRIGGER_PIN = 22  # GPIO pin to trigger Arduino Nano
ARDUINO_STATUS_PIN = 23  # GPIO pin to receive status from Arduino
# Path to test images
Test_image_dir = r"/media/ajay/bootfs/farmers project/images"  # Update this path
Categories = ["positive", "negative"]
# Initialize GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)  # Disable GPIO warnings
GPIO.setup(MOTION_SENSOR_PIN, GPIO.IN)
GPIO.setup(BUZZER_PIN, GPIO.OUT)
GPIO.setup(LED_PIN, GPIO.OUT)
GPIO.setup(ARDUINO_TRIGGER_PIN, GPIO.OUT)
GPIO.setup(ARDUINO_STATUS_PIN, GPIO.IN)
GPIO.output(ARDUINO_TRIGGER_PIN, GPIO.LOW)  # Ensure trigger is off initially
# Function to parse Pascal VOC XML annotation file
Def parse_annotation(xml_path):
    If not os.path.exists(xml_path):
        Print(f"Error: Annotation file {xml_path} not found.")
        Return []
    Tree = ET.parse(xml_path)
    Root = tree.getroot()
    Objects = root.findall("object")
    Boxes = []
    For obj in objects:
        Name = obj.find("name").text
        If name == "snake":
            Bndbox = obj.find("bndbox")
            Xmin = int(bndbox.find("xmin").text)
            Ymin = int(bndbox.find("ymin").text)
            Xmax = int(bndbox.find("xmax").text)
            Ymax = int(bndbox.find("ymax").text)
            Boxes.append((xmin, ymin, xmax – xmin, ymax – ymin))
    Return boxes

# Function to detect snake in an image (using annotations or model)
Def detect_snake(image_path, annotation_path=None):
    If annotation_path:
        Boxes = parse_annotation(annotation_path)
    Else:
        # For captured images, add detection logic here (e.g., using a trained model)
        Boxes = []  # Placeholder for detection logic
    Return len(boxes) > 0, boxes

# Function to capture image from the camera using libcamera
Def capture_image(output_path):
    Try:
        Subprocess.run(["libcamera-jpeg", "-o", output_path, "—width", "640", "—height", "480"], check=True)
        Print(f"Image captured and saved as {output_path}")
        Return True
    Except subprocess.CalledProcessError as e:
        Print(f"Error capturing image: {e}")
        Return False

# Function to trigger the first alert (LED and Buzzer)
Def trigger_first_alert():
    Print("First Alert: Snake detected! Activating LED and Buzzer...")
    GPIO.output(BUZZER_PIN, GPIO.HIGH)
    GPIO.output(LED_PIN, GPIO.HIGH)
    Sleep(2)  # Keep buzzer and LED on for 2 seconds
    GPIO.output(BUZZER_PIN, GPIO.LOW)
    GPIO.output(LED_PIN, GPIO.LOW)

# Function to trigger the second alert (Temperature Measurement)
Def trigger_second_alert():
    Print("Second Alert: Triggering Arduino Nano for temperature measurement...")
    GPIO.output(ARDUINO_TRIGGER_PIN, GPIO.HIGH)
    Sleep(1)  # Keep the trigger high for 1 second
    GPIO.output(ARDUINO_TRIGGER_PIN, GPIO.LOW)
```



```
    # Wait for Arduino to send status
    Print("Waiting for Arduino status...")
    While True:
        If GPIO.input(ARDUINO_STATUS_PIN) == GPIO.HIGH:
            Print("Status: Danger! Temperature is unsafe.")
            Break
        Elif GPIO.input(ARDUINO_STATUS_PIN) == GPIO.LOW:
            Print("Status: Safe. Temperature is within range.")
            Break
        Sleep(0.1)  # Small delay to avoid excessive CPU usage

# Function to capture image and test for snake
Def capture_and_test():
    Temp_image_path = "temp_image.jpg"
    If not capture_image(temp_image_path):
        Return  # Exit if image capture fails

    # Check if snake is detected (add model inference here)
    Result, boxes = detect_snake(temp_image_path)
    If result:
        Trigger_first_alert()  # First alert
        Trigger_second_alert()  # Second alert
    Else:
        Print("No snake detected.")

# Function to test on the dataset
Def test_dataset():
    For category in categories:
        Print(f"Testing category: {category}")
        Folder = os.path.join(test_image_dir, category)
        For file_name in os.listdir(folder):
            File_path = os.path.join(folder, file_name)
            Annotation_path = file_path.replace(".jpg", ".xml")  # Assuming XML files have the same name as images
            Result, boxes = detect_snake(file_path, annotation_path)
            Print(f"Image: {file_name}, Snake Detected: {result}")
            If result:
                # Draw bounding boxes on the image
                Image = cv2.imread(file_path)
                If image is not None:  # Check if the image is valid
                    For (x, y, w, h) in boxes:
                        Cv2.rectangle(image, (x, y), (x+w, y+h), (0, 255, 0), 2)  # Draw a green rectangle
                    Cv2.imshow("Detection", image)
                    Cv2.waitKey(0)
                    Cv2.destroyAllWindows()

                Trigger_first_alert()  # First alert
                Trigger_second_alert()  # Second alert

                # Optional: Add a delay to avoid rapid triggering
                Sleep(2)
        Print("-" * 50)

# Function to monitor motion and capture images
Def monitor_motion():
    Try:
        Print("Monitoring for motion...")
        While True:
            If GPIO.input(MOTION_SENSOR_PIN):
                Print("Motion detected!")
                Capture_and_test()
                Sleep(5)  # Wait for 5 seconds before checking for motion again
            Sleep(0.1)  # Small delay to avoid excessive CPU usage
    Except KeyboardInterrupt:
        Print("Exiting motion monitoring.")
    Finally:
        GPIO.cleanup()

# Main menu
Def main():
    While True:
        Print("\nChoose an option:")
        Print("1. Test on dataset")
        Print("2. Start motion monitoring")
        Print("3. Exit")
        Choice = input("Enter your choice: ").strip()

        If choice == "1":
```



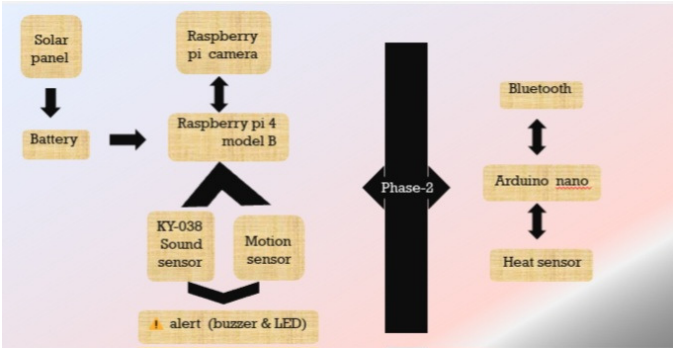
```
        Test_dataset()
    Elif choice == "2":
        Monitor_motion()
    Elif choice == "3":
        Print("Exiting the program.")
        GPIO.cleanup()
        Break
    Else:
        Print("Invalid choice. Please try again.")

If _name_ == "_main_":
    Main()
```

Code for Arduino Nano

```
#include <LiquidCrystal.h>
#include <Adafruit_MLX90614.h>
Adafruit_MLX90614 mlx = Adafruit_MLX90614();
// Pin Definitions
Const int triggerPin = 2;       // Pin to receive trigger signal from Raspberry Pi
Const int statusPin = 3;        // Pin to send status back to Raspberry Pi
Const int buzzerPin = 4;        // Pin for buzzer
Const int ledPin = 5;           // Pin for LED
Const int rs = 7, en = 8, d4 = 9, d5 = 10, d6 = 11, d7 = 12; // LCD pins

// Temperature Thresholds
Const float minTempThreshold = 20.0;    // Minimum safe temperature in Celsius
Const float maxTempThreshold = 40.0;    // Maximum safe temperature in Celsius
// Initialize LCD
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
Void setup() {
  // Initialize pins
  pinMode(triggerPin, INPUT);
  pinMode(statusPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(ledPin, OUTPUT);
  digitalWrite(buzzerPin, LOW);  // Ensure buzzer is off initially
  digitalWrite(ledPin, LOW);     // Ensure LED is off initially
  digitalWrite(statusPin, LOW);  // Ensure status pin is off initially

  // Initialize MLX90614 sensor
  mlx.begin();

  // Initialize LCD
  Lcd.begin(16, 2);  // 16x2 LCD
  Lcd.print("Temp Monitoring");
  Delay(2000);       // Display initial message for 2 seconds
  Lcd.clear();
}

Void loop() {
  If (digitalRead(triggerPin) == HIGH) {
    Float objectTemp = mlx.readObjectTempC();  // Read object temperature in Celsius

    // Display temperature on LCD
    Lcd.setCursor(0, 0);
    Lcd.print("Temp: ");
    Lcd.print(objectTemp);
    Lcd.print(" C");

    // Check temperature range and display status
    If (objectTemp >= 20.0 && objectTemp <= 40.0) {
      Lcd.setCursor(0, 1);
      Lcd.print("Status: Safe");
      digitalWrite(buzzerPin, LOW);    // Turn off buzzer
      digitalWrite(ledPin, LOW);     // Turn off LED
      digitalWrite(statusPin, LOW);  // Send LOW to Raspberry Pi (Safe)
    } else {
      Lcd.setCursor(0, 1);
      Lcd.print("Status: Danger!");
      digitalWrite(buzzerPin, HIGH);    // Turn on buzzer
      digitalWrite(ledPin, HIGH);     // Turn on LED
      digitalWrite(statusPin, HIGH);  // Send HIGH to Raspberry Pi (Danger)
    }

    Delay(5000);  // Wait for 5 seconds before checking again
    Lcd.clear();  // Clear LCD for the next reading
  }
```


}

Hardware Implementation



III. WORKING

The " Farmer's Safety From Reptiles" operates through a seamless integration of IoT sensors, AI-based image processing, and wireless communication technologies to provide real-time detection and alerts. The system begins with the PIR motion sensor, which detects infrared radiation emitted by moving objects in the field. Upon detecting motion, the sensor triggers the Raspberry Pi Camera to capture high-resolution images of the area. Simultaneously, the thermal camera scans the environment for heat signatures, while the acoustic sensor monitors for frequency patterns characteristic of snake hissing sounds. These multi-sensor inputs are transmitted to the Raspberry Pi 4, which serves as the central processing unit. The Raspberry Pi 4 employs a Tensor Flow Lite-based AI model trained on a custom dataset of snake and non-snake images to analyze the captured data. The model processes the visual, thermal, and acoustic inputs to classify the detected object, ensuring high accuracy and minimizing false positives.

If the system confirms the presence of a snake, it triggers local alerts through a buzzer and LED, providing immediate notification to farmers in the vicinity. Concurrently, the Wi-Fi module (ESP8266/ESP32) transmits the detection data, including images and location information, to a mobile application and a central monitoring unit. This enables real-time communication and coordination for rapid response. Additionally, the system incorporates a human body temperature monitoring module powered by an Arduino Nano. The module continuously tracks the farmer's body temperature using a thermal sensor. If a sudden spike and drop in temperature is detected—indicative of a snake bite—the Arduino Nano sends an alert to the Raspberry Pi 4, which triggers an emergency response protocol. This includes activating the buzzer, sending mobile notifications to emergency contacts, and logging the incident in the central monitoring system.

The system is designed for energy efficiency and reliability in remote agricultural settings. It is powered by a solar panel and supported by a lithium-ion battery backup, ensuring uninterrupted operation even in areas without grid power. The integration of multi-sensor fusion and edge AI processing ensures robust performance, while future enhancements such as LoRa communication and night vision capabilities aim to further improve its effectiveness. By combining real-time detection, wireless communication, and emergency response mechanisms, the system provides a comprehensive solution to mitigate the risks posed by venomous snakes, enhancing the safety and productivity of farmers in agricultural environments.

SCOPE OF PAPER

This paper is applicable in agriculture, rural safety, wildlife monitoring, and industrial zones. Farmers and workers in high-risk areas will benefit from automated detection and instant alerts. The system's ability to function in low-network environments using LoRa/Wi-Fi communication ensures wide usability. The addition of a wearable thermal monitoring system enhances emergency response effectiveness.

FUTURE ENHANCEMENTS

1.Integration of AI- powered voice alerts for additional warning.
2.Anhanced data logging to analyze snake movements patterns.
3.Expansion to individual safety zones and forest conservation projects.

CONCLUSION

This paper introduces an efficient, real-time snake detection and prevention system utilizing AI, IoT, and multi-sensor integration. The system provides high detection accuracy, rapid alerts, and energy-efficient operation, making it ideal for agricultural, wildlife, and industrial safety applications. Future improvements will enhance its capabilities, ensuring better protection and real-time monitoring for individuals in high-risk environments.

ACKNOWLEDGMENT

We extend our gratitude to the " Department of Electrical and Electronics Engineering" for providing the necessary resources and infrastructure to develop the "Farmer's Safety From Reptiles". Special thanks to Tensor Flow Lite and Raspberry Pi Foundation for their open-source tools and libraries, which were instrumental in implementing the AI model and hardware integration. We also acknowledge the support of IoT and edge computing communities for their contributions to advancing real-time detection technologies. Lastly, we thank our colleagues and mentors for their technical insights and guidance throughout the project.

REFERENCES

[1] Kumar, R., & Singh, P. (2021). "Real-Time Snake Detection in Agricultural Fields Using Convolutional Neural Networks." "*International Journal of Advanced Computer Science and Applications*", 12(5), 78-85.

[2] Patel, S., & Shah, K. (2019). "IoT-Based Smart Agriculture: A Comprehensive Review." "*IEEE Internet of Things Journal*", 6(2), 1234-1245.

[3] Silva, F., & Silva, A. (2020). "Automated Snake Detection Using Deep Learning: A Review." "*Journal of Herpetology and Wildlife Conservation*", 12(3), 45-60.

[4] Wang, Y., & Chen, Z. (2020). "Wearable Sensors for Real-Time Health Monitoring: A Review." "*IEEE Sensors Journal, 20(10)*", 5432-5445.

[5] Gupta, A., & Sharma, R. (2021). "Thermal Sensors for Early Detection of Health Anomalies." "*Journal of Medical Systems*", 45(4), 1-10.

