```
import cv2
import numpy as np
import time
import os
import threading
from datetime import datetime
import RPi.GPIO as GPIO
import requests
from ultralytics import YOLO


# CONFIG ===
TRIG_PIN = 16  # Board pin 16
ECHO_PIN = 18  # Board pin 18
RELAY_PIN = 22  # Light relay for elephant
SERVO_PIN = 32  # Board pin 32 (PWM)
BUZZER_PIN = 36  # Board pin 36 (beep when cow/elephant detected)


AUDIO_FILE = "alert.mp3"
ALERT_URL = "http://192.168.198.118:5050/alert" # Replace with your IP


# SETUP GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(TRIG_PIN, GPIO.OUT)
GPIO.setup(ECHO_PIN, GPIO.IN)
GPIO.setup(RELAY_PIN, GPIO.OUT)
GPIO.setup(SERVO_PIN, GPIO.OUT)
GPIO.setup(BUZZER_PIN, GPIO.OUT)


GPIO.output(RELAY_PIN, GPIO.LOW)
GPIO.output(BUZZER_PIN, GPIO.LOW)


servo = GPIO.PWM(SERVO_PIN, 50) # 50Hz PWM
servo.start(0)


# YOLOv8 MODEL
model = YOLO("yolov8n.pt")


# MODEL YOLOv8n.pt
cap = cv2.VideoCapture(0)
cap.set(3, 640)
cap.set(4, 480)


# FUNCTIONS


def set_servo_angle(angle):
  duty = 2 + (angle / 18)
  servo.ChangeDutyCycle(duty)
  time.sleep(0.03)
  servo.ChangeDutyCycle(0)
```

```
def measure_distance():
  GPIO.output(TRIG_PIN, False)
  time.sleep(0.05)
  GPIO.output(TRIG_PIN, True)
  time.sleep(0.00001)
  GPIO.output(TRIG_PIN, False)


  pulse_start, pulse_end = time.time(), time.time()
  while GPIO.input(ECHO_PIN) == 0:
      pulse_start = time.time()
  while GPIO.input(ECHO_PIN) == 1:
      pulse_end = time.time()
  duration = pulse_end - pulse_start
  distance = round(duration * 17150, 2)
  return distance


def play_audio():
  threading.Thread(target=lambda: os.system(f"mpg321 {AUDIO_FILE} > /dev/null 2>&1"), daemon=True).start()


def buzz_beep(repeat=5, on_time=0.2, off_time=0.2):
  def _buzz():
      for _ in range(repeat):
          GPIO.output(BUZZER_PIN, GPIO.HIGH)
          time.sleep(on_time)
          GPIO.output(BUZZER_PIN, GPIO.LOW)
```

```
        time.sleep(off_time)
  threading.Thread(target=_buzz, daemon=True).start()


def save_image(frame, tag):
  timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
  filename = f"{tag}_{timestamp}.jpg"
  cv2.imwrite(filename, frame)
  return filename


def send_alert(image_path, message):
  try:
      with open(image_path, 'rb') as img:
          files = {'image': img}
          data = {'message': message}
          requests.post(ALERT_URL, files=files, data=data, timeout=5)
  except Exception as e:
      print(f"[x] Alert failed: {e}")


# MAIN LOOP


try:
  print("YOLOv8n + Ultrasonic + Smooth Servo Sweep Started...")
  angle = 0
  step = 2
  direction = 1
```

```
while True:
    set_servo_angle(angle)
    print(f"Servo at {angle}")


    ret, frame = cap.read()
    if not ret:
        continue


    distance = measure_distance()
    print(f"Measured Distance: {distance:.1f} cm")
    cv2.putText(frame, f"Distance: {distance:.1f} cm", (10, 38),
                cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 255, 255), 2)


    temp_img = "current.jpg"
    cv2.imwrite(temp_img, frame)
    results = model(temp_img)
    labels = [model.names[int(cls)] for cls in results[0].boxes.cls]
    cow_detected = "cow" in labels
    elephant_detected = "elephant" in labels


    for box in results[0].boxes.xyxy:
        x1, y1, x2, y2 = map(int, box)
        cv2.rectangle(frame, (x1, y1), (x2, y2), (6, 255, 0), 2)
```

```
# ALERT LOGIC
if 60 < distance <= 70 and (cow_detected or elephant_detected):
    print("[A] Alert Zone: 70 cm")
    play_audio()
    buzz_beep(3)
    img = save_image(frame, "zone70")
    time.sleep(5)
elif 50 < distance <= 60 and (cow_detected or elephant_detected):
    print("[A] Alert Zone: 60 cm")
    play_audio()
    buzz_beep(3)
    img = save_image(frame, "zone60")
    time.sleep(5)
elif distance <= 50 and (cow_detected or elephant_detected):
    print("[0] Critical Zone: 50 cm")
    img = save_image(frame, "critical")
    buzz_beep(5)
    if cow_detected:
        print("Cow detected! Sending alert.")
        play_audio()
        send_alert(img, "Cow is entering the field!")
    if elephant_detected:
        print("Elephant detected! Turning on Light.")
        GPIO.output(RELAY_PIN, GPIO.HIGH)
        play_audio()
        send_alert(img, "Elephant detected! Light triggered.")
        time.sleep(10)
```

```
            GPIO.output(RELAY_PIN, GPIO.LOW)


        cv2.imshow("YOLOv8n Animal Surveillance", frame)
        if cv2.waitKey(1) & 0xFF == 27:
            raise KeyboardInterrupt


        # Continuous sweep logic
        angle += step * direction
        if angle >= 180:
            angle = 180
            direction = -1
        elif angle <= 0:
            angle = 0
            direction = 1


        time.sleep(0.05) # Smooth servo + time for detection


except KeyboardInterrupt:
    print("Stopped by user.")


finally:
    cap.release()
    cv2.destroyAllWindows()
    servo.stop()
```

```
GPIO.cleanup()
```