Smart Wildlife Surveillance System using C and PYTHON

1.C Code for Sensor module

```
#include <stdio.h>
#include <stdlib.h>
#include <wiringPi.h>
#include <softPwm.h>

// Define sensor pins and other parameters
#define CAMERA_PIN 2
#define MOTION_PIN 3
#define TEMPERATURE_PIN 4

void setup() {
    wiringPiSetup();
    pinMode(CAMERA_PIN, INPUT);
    pinMode(MOTION_PIN, INPUT);
    pinMode(TEMPERATURE_PIN, INPUT);
}

void loop() {
    // Read sensor data
    int camera_data = digitalRead(CAMERA_PIN);
    int motion_data = digitalRead(MOTION_PIN);
    int temperature_data = analogRead(TEMPERATURE_PIN);

    // Process data (e.g., filter noise, extract features)
    // ...

    // Transmit data to server (e.g., using MQTT, HTTP)
    // ...

    delay(1000); // Adjust delay as needed
}
```

2.Python Code for Central Server

```
import paho.mqtt.client as mqtt
import time
import cv2
import tensorflow as tf
```

```
# MQTT client setup
def on_connect(client, userdata, flags, rc):
    print("Connected with result code " + str(rc))
    client.subscribe("sensor_data")

def on_message(client, userdata, msg):
    data = msg.payload.decode()
    # Process received data (e.g., parse JSON, extract features)
    # ...

    # Use machine learning models for analysis
    model = tf.keras.models.load_model('wildlife_detection_model.h5')
    predictions = model.predict(preprocessed_data)
    # ...

    # Generate alerts and notifications
    # ...

# MQTT client initialization
client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message
client.connect("broker.hivemq.com", 1883, 60)

# Start the MQTT client
client.loop_start()

# Main loop
while True:
    # Other tasks (e.g., user interface, data visualization)
    time.sleep(1)
```