```
from picamera2 import Picamera2
import cv2
import numpy as np
import tflite_runtime.interpreter as tflite
from gpiozero import LED
from gpiozero import Button
import serial
import RPi.GPIO as GPIO
from time import sleep

GPIO.setmode(GPIO.BCM)
GPIO.setup(21,GPIO.IN)

buz=LED(20)

# Load model
interpreter = tflite.Interpreter(model_path="model.tflite")
interpreter.allocate_tensors()

input_details = interpreter.get_input_details()
output_details = interpreter.get_output_details()
h, w = input_details[0]['shape'][1:3]

labels = open("labels.txt").read().splitlines()

# Camera
picam2 = Picamera2()
picam2.configure(picam2.create_preview_configuration(
    main={"size": (640, 480)}
))
picam2.start()

recipient="8248229967"
ser = serial.Serial('/dev/ttyS0', baudrate=9600,
                    parity=serial.PARITY_NONE,
                    stopbits=serial.STOPBITS_ONE,
                    bytesize=serial.EIGHTBITS
                    )

ser.write('AT\r\n'.encode())
sleep(1)
ser.write('AT+CMGF=1\r\n'.encode())
sleep(1)

while True:
    rfid_state=GPIO.input(21)
    print(rfid_state)

    frame = picam2.capture_array()

    rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    resized = cv2.resize(rgb, (w, h))
    input_data = (resized.astype(np.float32) / 127.5) - 1
    input_data = np.expand_dims(input_data, axis=0)

    interpreter.set_tensor(input_details[0]['index'], input_data)
    interpreter.invoke()
    pred = interpreter.get_tensor(output_details[0]['index'])

    idx = np.argmax(pred)
    label = f"{labels[idx]} {pred[0][idx]:.2f}"
    print("------")
    print(label)
    print(labels[idx])
    print(rfid_state)
    print("------")

    if rfid_state== 0 and labels[idx]=="1 Elephant":
        buz.on()
        cv2.putText(frame, label, (20, 40),cv2.FONT_HERSHEY_SIMPLEX, 1, (0,255,0), 2)
        ser.write('''AT+CMGS="'''.encode() + recipient.encode() + '''"\r'''.encode())
        sleep(1)
        ser.write("Elephant Detected".encode())
        sleep(1)
        ser.write(chr(26).encode())
        buz.off()

    cv2.imshow("Elephant Detection", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cv2.destroyAllWindows()
picam2.stop()
```