SOURCE CODE

```
import cv2
import numpy as np
import matplotlib.pyplot as plt import serial
from mail import report_send_mail import time
from mail import *
from pygame import mixer
 net = cv2.dnn.readNetFromDarknet("yolov8_custom.cfg",
"yolov8_custom_last.weights")
class_ = None
classes = ['bear', 'lion', 'peacock', 'Tiger', 'Elephant', 'Chinkara'] def
classifer(label):
print(label)
cap = cv2.VideoCapture(0) while True:
_, img = cap.read()
img = cv2.resize(img, (1280, 720)) height, width, _ = img.shape
 blob = cv2.dnn.blobFromImage(img, 1/255, (416, 416), (0, 0, 0),
swapRB=True, crop=False)
net.setInput(blob)
output_layers_name = net.getUnconnectedOutLayersNames()
layerOutputs = net.forward(output_layers_name) boxes = []
confidences = [] class_ids = []
for output in layerOutputs:
for detection in output:
```

```
score = detection[5:] class_id = np.argmax(score) confidence =score[class_id] if
confidence > 0.7:

center_x = int(detection[0] * width) center_y = int(detection[1] * height) w =
int(detection[2] * width)

h = int(detection[3] * height) x = int(center_x - w / 2)

y = int(center_y - h / 2) boxes.append([x, y, w, h])
confidences.append(float(confidence)) class_ids.append(class_id)

indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4) font =
cv2.FONT_HERSHEY_PLAIN

colors = np.random.uniform(0, 255, size=(len(boxes), 3)) if len(indexes) > 0:

for i in indexes.flatten(): x, y, w, h = boxes[i]

label = str(classes[class_ids[i]])
cv2.imwrite('image.jpg', img) classifer(label)

if label == 'bear': print('bear') time.sleep(2)

report_send_mail(label, 'image.jpg') elif label == 'lion':

print('lion')
report_send_mail(label, 'image.jpg') elif label == 'peacock': print('peacock')

time.sleep(2) report_send_mail(label, 'image.jpg') elif label == 'Tiger':
print('Tiger') time.sleep(2) report_send_mail(label, 'image.jpg') elif label ==
'Elephant': print('Elephant')

time.sleep(2) report_send_mail(label, 'image.jpg') elif label == 'Chinkara':
print('Chinkara') report_send_mail(label, 'image.jpg') try:

mixer.init() mixer.music.load("sound.mp3")

mixer.music.set_volume(0.7) mixer.music.play()
```

```
except:
print('Issues in Speaker')
confidence =str(round(confidences[i], 2)) color = colors[i]
cv2.rectangle(img, (x, y), (x + w, y + h), color, 2)
cv2.putText(img, label + " " + confidence, (x, y + 400), font, 2, color, 2)
cv2.imshow('img', img)
if cv2.waitKey(1) == ord('q'):
break cap.release()
 cv2.destroyAllWindows() ## import packages  import os
import time import smtplib
from email.mime.multipart import MIMEMultipart from email.mime.text import
MIMEText
from email.mime.base import MIMEBase from email.mime.image import
MIMEImage from email import encoders
import imghdr
## define function
def report_send_mail(label, image_path): '''
This function sends mail '''
with open(image_path, rb') as f:
img_data = f.read()
fromaddr = "sangeethasiva2804@gmail.com" toaddr =
"sangeethasiva2804@gmail.com" msg = MIMEMultipart()
msg['From'] = fromaddr msg['To'] = toaddr msg['Subject'] = "Alert" body = label
msg.attach(MIMEText(body, 'plain'))  # attach plain text
```

```
image = MIMEImage(img_data, name=os.path.basename(image_path))
msg.attach(image) # attach image

s = smtplib.SMTP('smtp.gmail.com', 587) s.starttls()

s.login(fromaddr, "iedsixgnppwiucud") text = msg.as_string()
s.sendmail(fromaddr, toaddr, text) s.quit()
```