1. YOLOv8 Model Training (on PC / Google Colab / PowerShell)

bash

```
# Install YOLOv8 (Ultralytics)

pip install ultralytics


# Train the model

yolo task=detect mode=train model=yolov8n.pt data=data.yaml epochs=100 imgsz=640


# Check model results (optional)

yolo task=detect mode=val model=best.pt data=data.yaml


# Inference on test image

yolo task=detect mode=predict model=best.pt source=path/to/image.jpg


# Export trained model (if needed)

yolo export model=best.pt format=onnx
```

🔷 2. Dataset Management using Roboflow (PC)

```
# Roboflow auto-install script (in Colab/PC)

pip install roboflow


# Download dataset

from roboflow import Roboflow
```

```
rf = Roboflow(api_key="your_api_key")

project = rf.workspace().project("project-name")

dataset = project.version(1).download("yolov8")
```

◆ 3. Installing Required Python Libraries (on Raspberry Pi / PC)

```
pip install opencv-python

pip install easyocr

pip install pytesseract

pip install ultralytics

pip install numpy

import smtplib

pip install imutils
```

Explain

```
pip install opencv-python
```

◆ OpenCV is used for capturing and processing real-time video frames from the camera (e.g., drawing boxes, cropping, etc.).

```
pip install easyocr
```

◆ EasyOCR is used to extract text (vehicle number) from the number plate image using deep learning-based Optical Character Recognition.

```
pip install pytesseract
```

◆ Pytesseract is an alternative OCR library (wrapper for Tesseract) to read number plates from images.

```
pip install ultralytics
```

◆ Ultralytics provides access to YOLOv8 models, which are used to detect mobile phone usage, two-wheelers, riders, and number plates.

```
pip install numpy
```

🔹 NumPy is used for efficient numerical operations and handling image arrays and detection outputs.

```
pip install smtplib**
```

❌ This command is incorrect — smtplib is part of Python's standard library and does not need installation via pip. You can directly import smtplib.

```
pip install imutils
```

🔹 Imutils simplifies image processing tasks like resizing, rotating, and displaying video frames with OpenCV.

✅ Correct pip install command (excluding smtplib):

```
pip install opencv-python easyocr pytesseract ultralytics numpy imutils
```

🔷 4. Run Real-Time Detection (on Raspberry Pi / PC)

```
# Python script to run YOLOv8 detection and OCR

python detect_and_notify.py
```

Inside your script, you would have included:

python

```
import cv2

from ultralytics import YOLO

import easyocr

import smtplib
```

Detection logic :

This code segment is part of the object detection and classification logic in a real-time YOLOv8-based violation detection system. It processes each detected object in the current video frame, extracting the bounding box coordinates, class ID, and confidence score. If the confidence is above 0.3, it identifies the object's type—such as mobile phone, number plate, rider, two-wheeler, or person—based on the class ID. It then assigns a specific color to each class for visual annotation and stores relevant detections in separate lists (riders_with_mobile, number_plates, two_wheelers, persons). These lists are later used for logic such as detecting if a rider is using a mobile phone by checking spatial overlap, extracting the number plate for OCR, and logging or reporting violations accordingly.

🔷 5. Tesseract OCR (if used instead of EasyOCR)

```
# Install Tesseract OCR (Linux/Raspberry Pi)

sudo apt-get install tesseract-ocr



# Test Tesseract

tesseract image.png output.txt



# In Python

import pytesseract

text = pytesseract.image_to_string(cropped_image)
```

O. Transfer Files to Raspberry Pi

Use **SCP** from your PC to Pi:

```
scp best.pt pi@<RaspberryPi_IP>:/home/pi/Desktop/

scp detect_and_notify.py pi@<RaspberryPi_IP>:/home/pi/Desktop/
```

🔹 6. Email Automation (SMTP in Python)

python

```
import smtplib
from email.message import EmailMessage

msg = EmailMessage()
msg['Subject'] = 'Traffic Violation Detected'
msg['From'] = 'sender@example.com'
msg['To'] = 'receiver@example.com'
msg.set_content('Violation detected. Vehicle number: XYZ1234')

with open("violation.jpg", 'rb') as f:
    img_data = f.read()
    msg.add_attachment(img_data, maintype='image', subtype='jpeg', filename='violation.jpg')

server = smtplib.SMTP_SSL('smtp.gmail.com', 465)
server.login("sender@example.com", "your_password")
server.send_message(msg)
server.quit()
```

🔹 7. Camera Initialization and Display (OpenCV)

```
cap = cv2.VideoCapture(0)  # For USB camera
```

```
while True:
    ret, frame = cap.read()
    if not ret:
        break
    cv2.imshow("Frame", frame)
    if cv2.waitKey(1) == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

🔹 8. Raspberry Pi GPIO (If LCD or LED used)

```
# Install RPi.GPIO if not already
sudo apt-get install python3-rpi.gpio
```

```
python
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(18, GPIO.OUT)
GPIO.output(18, GPIO.HIGH)  # Turn ON
```

LCD Pin	Function	Raspberry Pi GPIO (BCM)	Physical Pin
RS	Register Select	GPIO 25	Pin 22
E	Enable	GPIO 24	Pin 18
D4	Data 4	GPIO 23	Pin 16

LCD Pin	**Function**	**Raspberry Pi GPIO (BCM)**	**Physical Pin**
D5	Data 5	GPIO 17	Pin 11
D6	Data 6	GPIO 18	Pin 12
D7	Data 7	GPIO 22	Pin 15