In [73]:

```
import cv2
import numpy as np
```

In [74]:

```
from tracker import *
```

In [75]:

```
# Create tracker object
tracker = EuclideanDistTracker()
```

In [76]:

```
cap = cv2.VideoCapture('/Users/myyntiimac/Desktop/object tracking/highway (1).mp

# Object detection from Stable camera
object_detector = cv2.createBackgroundSubtractorMOG2(history=200, varThreshold=5

while cap.isOpened():

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

    roi = frame[340:720, 500:800]

    mask = object_detector.apply(roi)
    _, mask = cv2.threshold(mask, 254, 255, cv2.THRESH_BINARY)

    contours, _ = cv2.findContours(mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

    detections = []
    for cnt in contours:
        area = cv2.contourArea(cnt)
        if area > 100:
            x, y, w, h = cv2.boundingRect(cnt)
            detections.append([x, y, w, h])


    # 2. Object Tracking
    boxes_ids = tracker.update(detections)
    for box_id in boxes_ids:
        x, y, w, h, id = box_id
        cv2.putText(roi, str(id), (x, y - 15), cv2.FONT_HERSHEY_PLAIN, 2, (255,
        cv2.rectangle(roi, (x, y), (x + w, y + h), (0, 255, 0), 3)

    cv2.imshow("roi", roi)
    cv2.imshow("Frame", frame)
    cv2.imshow("Mask", mask)

    key = cv2.waitKey(30)
    if key == 27:
        break

cap.release()
cv2.destroyAllWindows()
```

```
{1: (159, 153)}
{1: (162, 174)}
{1: (158, 150)}
{1: (160, 173)}
{1: (159, 181)}
{1: (158, 190)}
{1: (158, 197)}
{1: (158, 210)}
{1: (157, 220)}
{1: (156, 232)}
{1: (156, 244)}
{1: (156, 256)}
{1: (155, 269)}
{1: (154, 284)}
{1: (153, 298)}
{1: (152, 314)}
{1: (151, 330)}
{1: (150, 342)}
{1: (148, 351)}
{1: (145, 360)}
```

In [ ]:

```
1
```