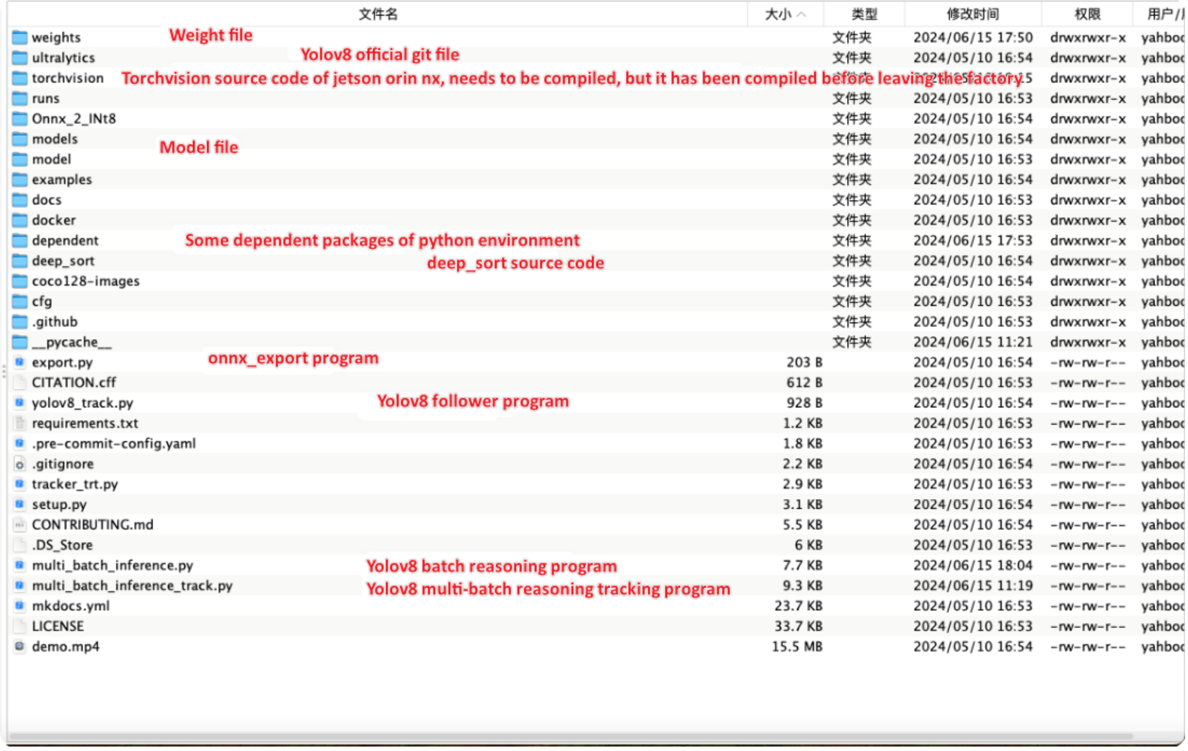
5. deep sort target tracking

Deep sort code path

```
/home/yahboom/YBAMR-COBOT-EDU-00001/soft/yolov8
```

As in the previous section, we can see the file path of our Deep sort from the figure below.



文件名	大小	类型	修改时间	权限	用户/
weights		文件夹	2024/06/15 17:50	drwxrwxr-x	yahbo
ultralytics		文件夹	2024/05/10 16:54	drwxrwxr-x	yahbo
torchvision		文件夹	[illegible]	drwxrwxr-x	yahbo
runs		文件夹	2024/05/10 16:53	drwxrwxr-x	yahbo
Onnx_2_INt8		文件夹	2024/05/10 16:54	drwxrwxr-x	yahbo
models		文件夹	2024/05/10 16:54	drwxrwxr-x	yahbo
model		文件夹	2024/05/10 16:53	drwxrwxr-x	yahbo
examples		文件夹	2024/05/10 16:54	drwxrwxr-x	yahbo
docs		文件夹	2024/05/10 16:53	drwxrwxr-x	yahbo
docker		文件夹	2024/05/10 16:53	drwxrwxr-x	yahbo
dependent		文件夹	2024/06/15 17:53	drwxrwxr-x	yahbo
deep_sort		文件夹	2024/05/10 16:54	drwxrwxr-x	yahbo
coco128-images		文件夹	2024/05/10 16:54	drwxrwxr-x	yahbo
cfg		文件夹	2024/05/10 16:53	drwxrwxr-x	yahbo
.github		文件夹	2024/05/10 16:53	drwxrwxr-x	yahbo
__pycache__		文件夹	2024/06/15 11:21	drwxrwxr-x	yahbo
export.py	203 B		2024/05/10 16:54	-rw-rw-r--	yahbo
CITATION.cff	612 B		2024/05/10 16:53	-rw-rw-r--	yahbo
yolov8_track.py	928 B		2024/05/10 16:54	-rw-rw-r--	yahbo
requirements.txt	1.2 KB		2024/05/10 16:53	-rw-rw-r--	yahbo
.pre-commit-config.yaml	1.8 KB		2024/05/10 16:53	-rw-rw-r--	yahbo
.gitignore	2.2 KB		2024/05/10 16:54	-rw-rw-r--	yahbo
tracker_trt.py	2.9 KB		2024/05/10 16:53	-rw-rw-r--	yahbo
setup.py	3.1 KB		2024/05/10 16:54	-rw-rw-r--	yahbo
CONTRIBUTING.md	5.5 KB		2024/05/10 16:54	-rw-rw-r--	yahbo
.DS_Store	6 KB		2024/05/10 16:53	-rw-rw-r--	yahbo
multi_batch_inference.py	7.7 KB		2024/06/15 18:04	-rw-rw-r--	yahbo
multi_batch_inference_track.py	9.3 KB		2024/06/15 11:19	-rw-rw-r--	yahbo
mkdocs.yml	23.7 KB		2024/05/10 16:53	-rw-rw-r--	yahbo
LICENSE	33.7 KB		2024/05/10 16:53	-rw-rw-r--	yahbo
demo.mp4	15.5 MB		2024/05/10 16:54	-rw-rw-r--	yahbo

Weight file

Yolov8 official git file

Torchvision source code of jetson orin nx, needs to be compiled, but it has been compiled before leaving the factory

Model file

Some dependent packages of python environment

deep_sort source code

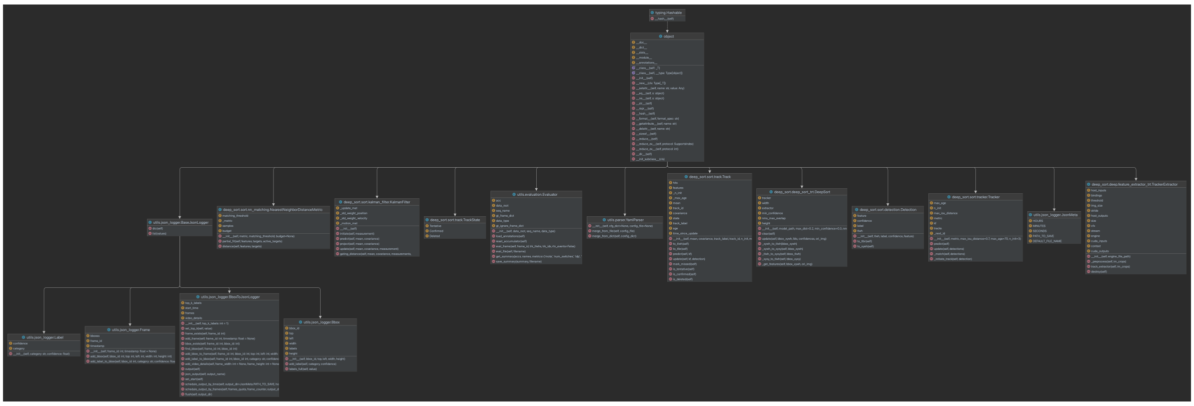
onnx_export program

Yolov8 follower program

Yolov8 batch reasoning program

Yolov8 multi-batch reasoning tracking program

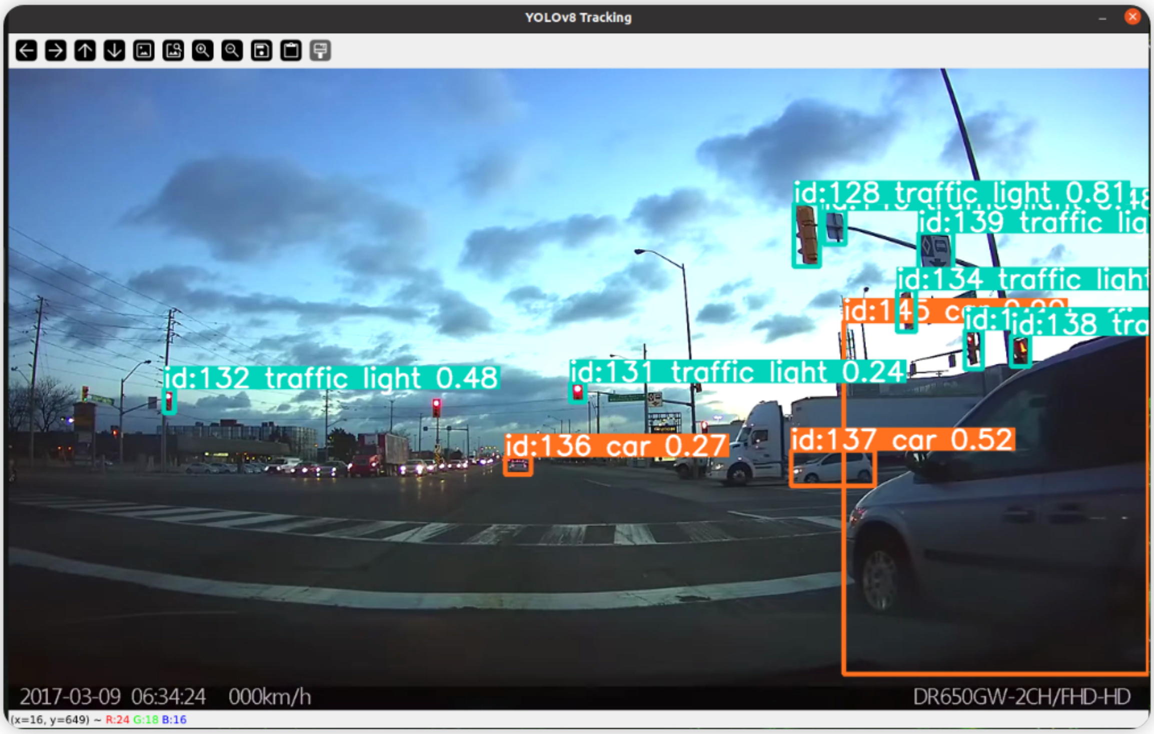
The following project directory code class diagram



We start the demo program. The target tracking startup file is the yolov8_track.py program. We enter the command in the terminal:

```
cd /home/yahboom/YBAMR-COBOT-EDU-00001/soft/yolov8
python yolov8_track.py
```

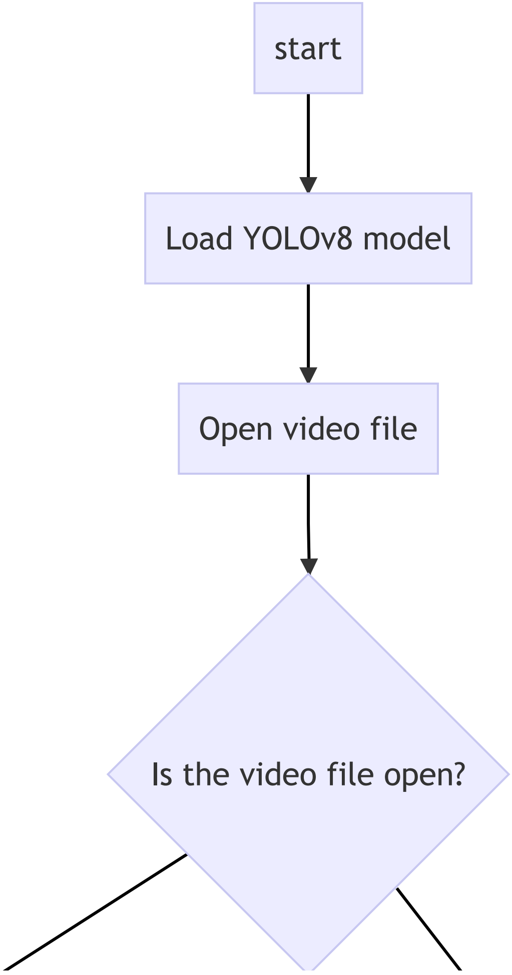
Then the interface pops up as shown below.

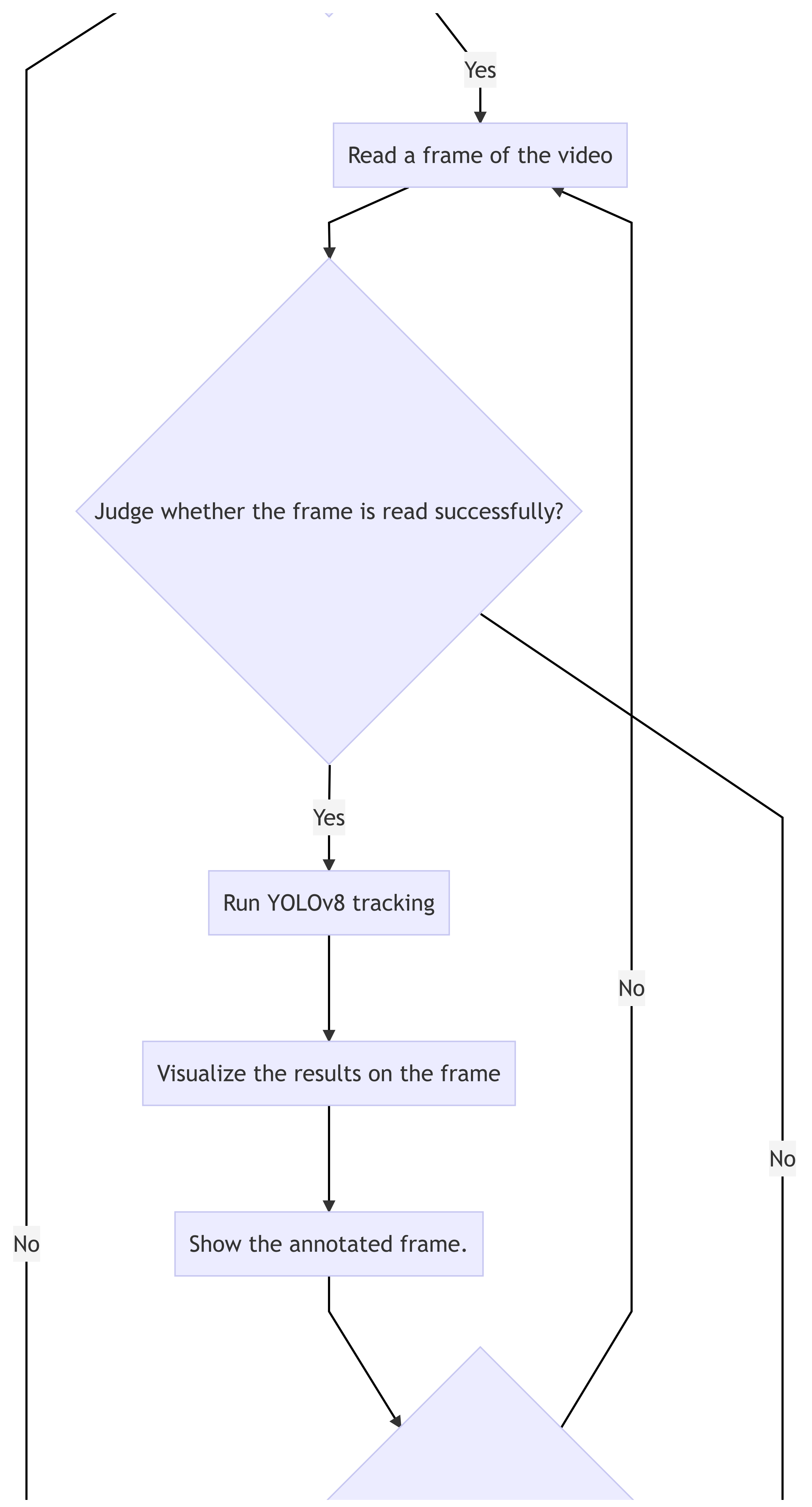

YOLOv8 Tracking
id:128 traffic light 0.81
id:139 traffic lig
id:134 traffic light
id:138 tra
id:132 traffic light 0.48
id:131 traffic light 0.24
id:136 car 0.27
id:137 car 0.52
2017-03-09 06:34:24 000km/h
DR650GW-2CH/FHD-HD
(x=16, y=649) ~ R:24 G:18 B:16

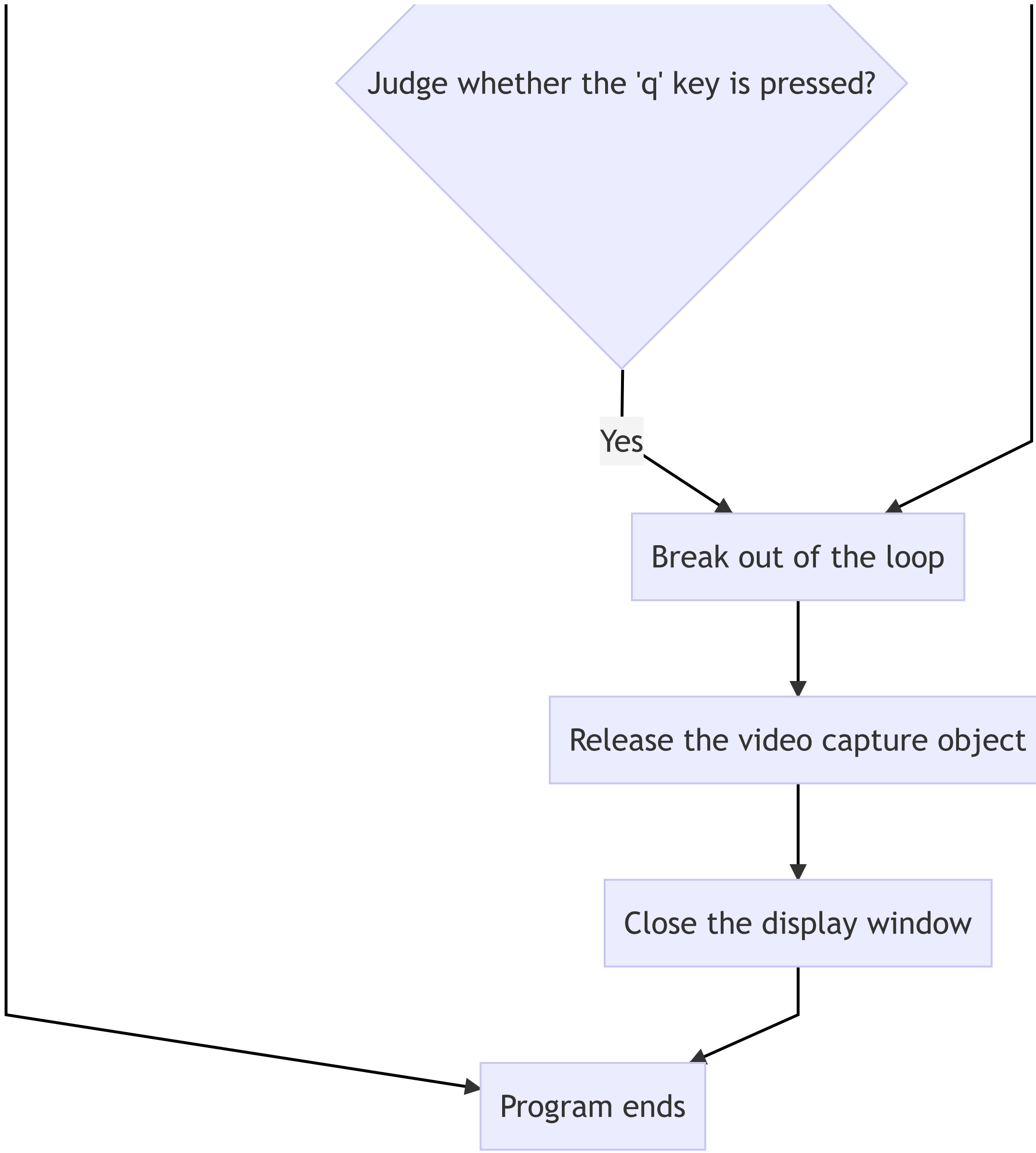
The demo program started above uses yolov8 as the target detection model, and then uses deep sort for target tracking. We can see that each target label will have an ID for the target. If the target is always within the detection range, the ID will remain unchanged.

Code Analysis

Code running flow chart:




Yes
Read a frame of the video
Judge whether the frame is read successfully?
Yes
Run YOLOv8 tracking
Visualize the results on the frame
Show the annotated frame.
No
No
No


Judge whether the 'q' key is pressed?
Yes
Break out of the loop
Release the video capture object
Close the display window
Program ends

Code:

```python
import cv2
from ultralytics import YOLO

# Load the YOLOv8 model
model = YOLO('./weights/yolov8n.pt')

# Open the video file
video_path = "demo.mp4"
cap = cv2.VideoCapture(video_path)

# Loop through the video frames
while cap.isOpened():
    # Read a frame from the video
    success, frame = cap.read()

    if success:
        # Run YOLOv8 tracking on the frame, persisting tracks between frames
        results = model.track(frame, persist=True)
        # Visualize the results on the frame
        annotated_frame = results[0].plot()

        # Display the annotated frame
        cv2.imshow("YOLOv8 Tracking", annotated_frame)

        # Break the loop if 'q' is pressed
        if cv2.waitKey(1) & 0xFF == ord("q"):
            break
    else:
        # Break the loop if the end of the video is reached
        break

# Release the video capture object and close the display window
cap.release()
cv2.destroyAllWindows()
```