

计算机视觉基础与实战

零基础轻松入门（第九讲 yolov8+Deepsort多目标跟踪实战）

路虽远，行则将至！！

目录

1 opencv 读取视频帧

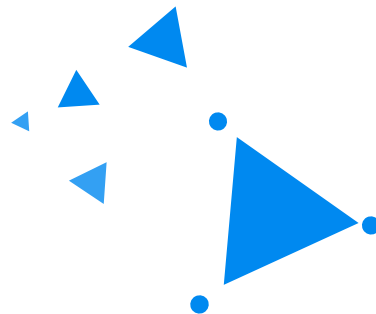
2 opencv画图

3 opencv写入文件

4 yolo v8 实时视频跟踪

5 yolo v8 轨迹绘制

01 opencv 处理输入视频



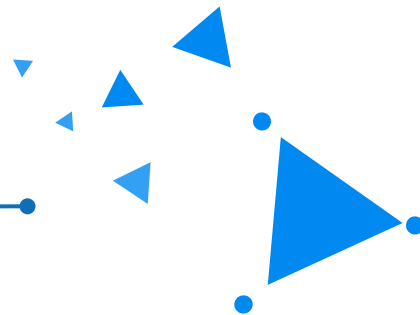
cv2.VideoCapture()

- cv2.VideoCapture 是 OpenCV 库中的一个类，用于捕获视频数据。它可以从视频文件、图像序列或摄像头捕获视频帧。这个模块提供了基本的控制功能，如开始和停止捕获，以及获取和设置视频属性。
 - cv2.VideoCapture()
 - cv2.VideoCapture(device) //device: 摄像头的索引号，例如 0 表示第一个摄像头。
 - cv2.VideoCapture(filename)
- 使用 isOpened() 方法检查 VideoCapture 对象是否成功打开。
- 使用 read() 方法读取视频帧，它返回两个值：布尔值表示是否成功读取帧，以及帧本身。
- 使用 release() 方法释放 VideoCapture 对象，避免资源泄露。
- 使用 cv2.imshow(): 在指定窗口中显示 OpenCV 图像，窗口自适应图像大小
- 使用 cv2.waitKey(): 控制循环的退出条件。
- 使用 cv2.destroyAllWindows(): 关闭所有 OpenCV 创建的窗口。

VideoCapture.get()参数

- `cv2.VideoCapture.get(0)` `CAP_PROP_POS_MSEC` 视频文件的当前位置（播放）以毫秒为单位
- `cv2.VideoCapture.get(1)` `CAP_PROP_POS_FRAMES` 基于以0开始的被捕获或解码的帧索引
- `cv2.VideoCapture.get(2)` `CAP_PROP_POS_AVI_RATIO` 视频文件的相对位置（播放）：0=电影开始，1=影片的结尾。
- `cv2.VideoCapture.get(3)` `CAP_PROP_FRAME_WIDTH` 在视频流的帧的宽度
- `cv2.VideoCapture.get(4)` `CAP_PROP_FRAME_HEIGHT` 在视频流的帧的高度
- `cv2.VideoCapture.get(5)` `CAP_PROP_FPS` 帧速率
- `cv2.VideoCapture.get(6)` `CAP_PROP_FOURCC` 编解码的4字-字符代码
- `cv2.VideoCapture.get(7)` `CAP_PROP_FRAME_COUNT` 视频文件中的帧数
- `cv2.VideoCapture.get(8)` `CAP_PROP_FORMAT` 返回对象的格式
- `cv2.VideoCapture.get(9)` `CAP_PROP_MODE` 返回后端特定的值，该值指示当前捕获模式
- `cv2.VideoCapture.get(10)` `CAP_PROP_BRIGHTNESS` 图像的亮度(仅适用于照相机)
- `cv2.VideoCapture.get(11)` `CAP_PROP_CONTRAST` 图像的对比度(仅适用于照相机)
- `cv2.VideoCapture.get(12)` `CAP_PROP_SATURATION` 图像的饱和度(仅适用于照相机)
- `cv2.VideoCapture.get(13)` `CAP_PROP_HUE` 色调图像(仅适用于照相机)
- `cv2.VideoCapture.get(14)` `CAP_PROP_GAIN` 图像增益(仅适用于照相机)（Gain在摄影中表示白平衡提升）
- `cv2.VideoCapture.get(15)` `CAP_PROP_EXPOSURE` 曝光(仅适用于照相机)
- `cv2.VideoCapture.get(16)` `CAP_PROP_CONVERT_RGB` 指示是否应将图像转换为RGB布尔标志
- `cv2.VideoCapture.get(17)` `CAP_PROP_WHITE_BALANCE` × 暂时不支持
- `cv2.VideoCapture.get(18)` `CAP_PROP_RECTIFICATION` 立体摄像机的矫正标注（目前只有DC1394 v.2.x后端支持这个功能）

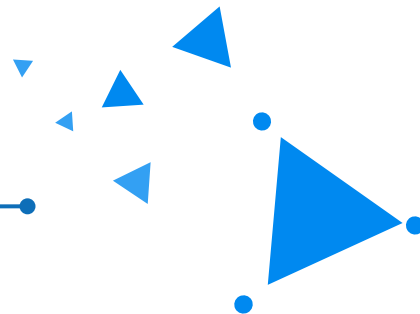
02 opencv画图



opencv 画图

- 在OpenCV中，颜色通常以BGR格式表示，而不是常见的RGB格式。OpenCV中的颜色通道顺序是B（蓝色）、G（绿色）、R（红色）。
- `cv2.line()`: 画线——图像，设置起点和终点，颜色，线条宽度
- `cv2.putText()`: 写文本 ——图像，绘制的文字，位置，字型，字体大小，文字颜色，线型
- `cv2.fillPoly()`:
- `cv2.polylines()`: 绘制多边形—— 图像，点集，是否闭合，颜色，线条粗细
- `cv2.addWeighted()`: 融合两幅图像——src1: 第一幅输入图像，alpha: 第一幅图像的权重，src2: 第二幅输入图像，beta: 第二幅图像的权重，gamma: 加到最终结果上的标量值

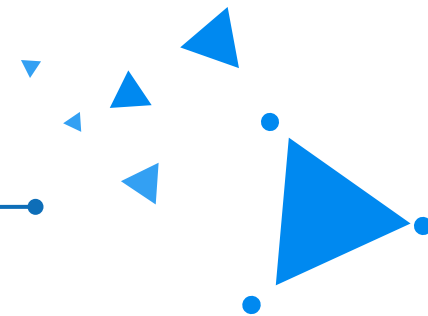
03 opencv写入文件



cv2.VideoWriter

- `cv2.VideoWriter(filename, fourcc, fps, frameSize[, isColor])`
 - 第一个参数是要保存的文件的路径
 - `fourcc` 指定编码器
 - `fps` 要保存的视频的帧率
 - `frameSize` 要保存的文件的画面尺寸
 - `isColor` 指示是黑白画面还是彩色的画面

04 yolo v8 实时视频跟踪



PyCharm断点调试方法

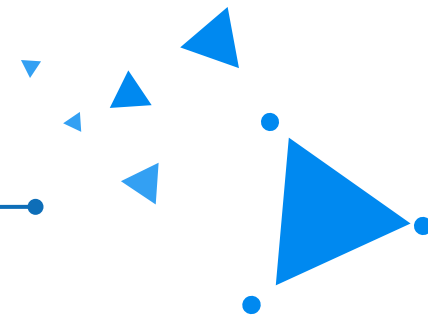
- F8：在断点处执行语句
- F7：进入函数内部执行语句

演示效果

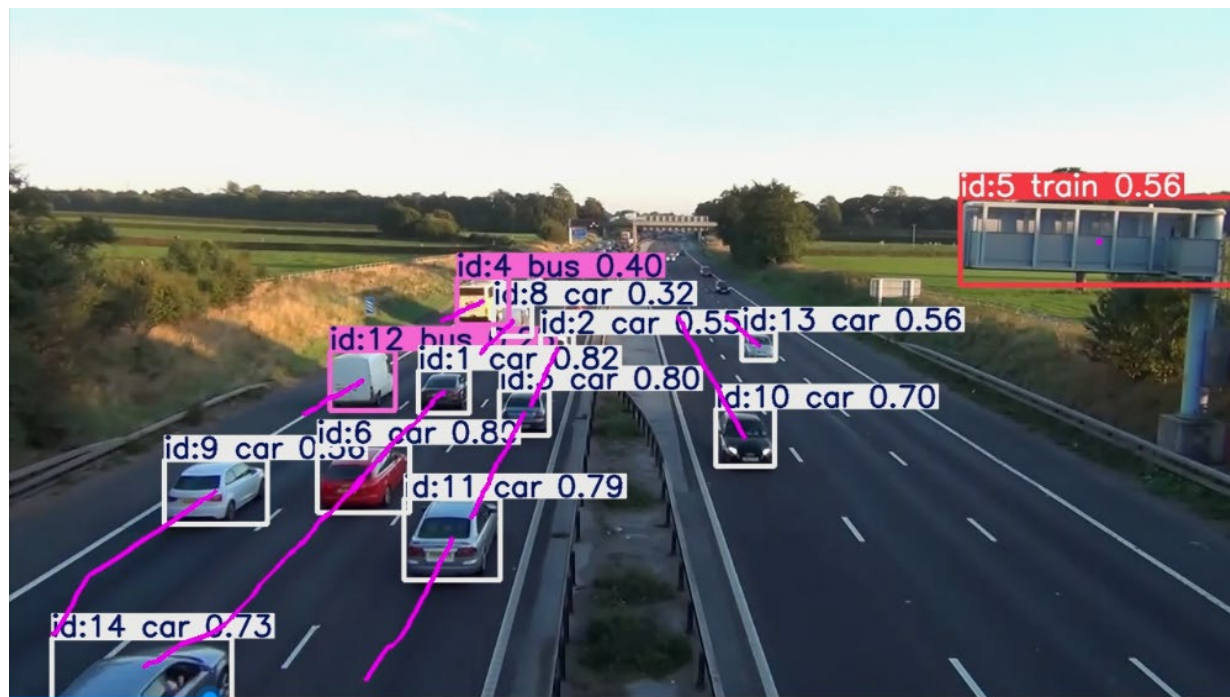


05

yolo v8 轨迹绘制



演示效果





讲解完毕，谢谢！