# Explicação do Código de Identificação com Vídeos

import cv2 as cv

from ultralytics import YOLO

model = YOLO("yolov8n.pt") # Carrega o modelo YOLOv8 a partir do arquivo yolov8n.pt

videoPath = "01.mp4" # Define o caminho para o arquivo de vídeo que será processado

cap = cv.VideoCapture(videoPath) # Cria um objeto VideoCapture para ler o vídeo a partir do caminho especificado

ret, frame = cap.read() # Lê o primeiro frame do vídeo e retorna ret (um valor booleano que indica se a leitura foi bem-sucedida) e frame (o frame lido)

frame\_width = int(cap.get(cv.CAP\_PROP\_FRAME\_WIDTH)) # Obtém a largura dos frames do vídeo

frame\_height = int(cap.get(cv.CAP\_PROP\_FRAME\_HEIGHT)) # Obtém a altura dos frames do vídeo

while ret: # Loop para ler e processar cada frame do vídeo

ret, frame = cap.read() # Lê o próximo frame do vídeo

frame = cv.resize(frame, (frame\_width, frame\_height)) # Redimensiona o frame para garantir que todos os frames tenham o mesmo tamanho

results = model.track(frame, persist=True) # Detecta e rastreia objetos no frame usando o modelo YOLOv8

frameResult = results[0].plot() # Plota os resultados (anotações) no frame

cv.imshow("frame", frameResult) # Mostra o frame processado em uma janela chamada "frame"

if cv.waitKey(1) & 0xFF == ord("q"): # Espera 1 milissegundo para ver se a tecla 'q' foi pressionada para sair do loop

break

cap.release() # Libera o objeto VideoCapture

cv.destroyAllWindows() # Fecha todas as janelas abertas pelo OpenCV

# Explicação do Código de Identificação com Imagens

import cv2 as cv

from ultralytics import YOLO

model = YOLO("yolov8n.pt") # Carrega o modelo YOLOv8 a partir do arquivo yolov8n.pt

imagePath = "cat01.jpg" # Define o caminho para o arquivo de imagem que será processado

frame = cv.imread(imagePath) # Lê a imagem a partir do caminho especificado e armazena o frame na variável frame

results = model.track(frame, persist=True) # Detecta e rastreia objetos no frame da imagem usando o modelo YOLOv8

frameResult = results[0].plot() # Plota os resultados (anotações) no frame

cv.imshow("Image", frameResult) # Mostra o frame processado em uma janela chamada "Image"

cv.waitKey(0) # Espera indefinidamente até que uma tecla seja pressionada para fechar a janela

cv.destroyAllWindows() # Fecha todas as janelas abertas pelo OpenCV