## به نام خدا

## بیتا مهدوی – مطهره ظهوری نیا – پونه حیدری

## 

## وارد کردن کتابخانه‌ها

import cv2

from ultralytics import YOLO

from deep\_sort\_realtime.deepsort\_tracker import DeepSort

### توضیح:

* cv2 کتابخانه‌ی OpenCV : برای پردازش تصویر و ویدیو. باهاش می‌تونیم:
  + ویدیو بخونیم
  + تصویر نمایش بدیم
  + روی تصویر اشکال بکشیم
* YOLO از ultralytics : برای بارگذاری و اجرای مدل وشناسایی اشیاء
* DeepSort : الگوریتمی برای **دنبال کردن اشیاء**. مثلاً اگه یه پرنده در چند فریم ظاهر بشه، Deep SORT اون رو با یه شناسه‌ی (ID) ثابت نگه می‌داره.

**۲. بارگذاری مدل و تنظیمات اولیه**

model = YOLO("yolov8s.pt")

video\_path = "D://9.mp4"

cap = cv2.VideoCapture(video\_path)

tracker = DeepSort(max\_age=30)

counted\_ids = set()

**توضیح:**

* YOLO("yolov8s.pt") : ما مدل YOLO نسخه s رو بارگذاری می‌کنیم. این نسخه سبک‌تره و برای اجرای سریع مناسبه.
* cv2.VideoCapture(...) : فایل ویدیو از مسیر مشخص باز میشه.
* DeepSort(max\_age=30):
  + یک شیء از DeepSort ساخته میشه.
  + max\_age=30 یعنی اگه یک پرنده تا ۳۰ فریم دیده نشه، دیگه ردیابی نمی‌شه.
* Counted\_ids : یه مجموعه برای ذخیره کردن ID پرنده‌هایی که قبلاً **شمرده شدن**.

**۳. تعریف اندازه دلخواه تصویر**

target\_width, target\_height = 640, 360

**توضیح:**

* برای سبک کردن عملیات پردازشی، تصاویر ویدیویی رو کوچکتر می‌کنیم.
* این کار باعث افزایش **سرعت شناسایی و ردیابی** میشه.

**۴. شروع حلقه اصلی پردازش فریم‌ها**

while True:

ret, frame = cap.read()

if not ret:

break

**توضیح:**

* با cap.read() هر فریم از ویدیو رو می‌گیریم.
* ret: اگه False باشه یعنی فریم تموم شده (ویدیو تموم شده) → حلقه قطع میشه.

**5. تغییر اندازه فریم**

frame = cv2.resize(frame, (target\_width, target\_height))

**توضیح:**

* فریم فعلی به اندازه مشخص‌شده (640x360) تغییر می‌کنه.
* این کاهش ابعاد باعث میشه YOLO و Deep SORT سریع‌تر کار کنن.

**6. اجرای مدل YOLO روی فریم**

results = model(frame)[0]

**توضیح:**

* YOLO فریم رو تحلیل می‌کنه و اشیاء موجود در فریم رو شناسایی می‌کنه.
* خروجی model(frame) یک لیست از نتایج (برای حالت‌های مختلف) هست که ما فقط اولین مورد رو می‌گیریم ([0]).

**7. فیلتر کردن فقط پرنده‌ها**

detections = []

for box, cls, conf in zip(results.boxes.xyxy, results.boxes.cls, results.boxes.conf):

if int(cls) == 14 and conf > 0.75:

x1, y1, x2, y2 = map(int, box)

w, h = x2 - x1, y2 - y1

detections.append(([x1, y1, w, h], conf.item(), 'bird'))

**توضیح:**

* results.boxes.xyxy: مختصات جعبه (Bounding Box) شناسایی‌شده‌ها (x1, y1, x2, y2).
* cls: کلاس شناسایی شده (مثلاً انسان، سگ، پرنده و غیره).
* conf: میزان اطمینان مدل.

**شرط:**

* cls == 14: فقط اگر کلاس 14 (پرنده) باشه.
* conf > 0.75: فقط اگر اطمینان مدل از 75٪ بیشتر باشه.

**ساختار ورودی برای Deep SORT:**

* [x1, y1, w, h]: مختصات مستطیل
* conf.item(): عدد اطمینان
* 'bird': برچسب شیء (دلخواه)

**8. به‌روزرسانی ردیابی با Deep SORT**

tracks = tracker.update\_tracks(detections, frame=frame)

**توضیح:**

* tracker.update\_tracks موقعیت پرنده‌ها رو به‌روز می‌کنه و ID جدید یا قدیمی بهشون می‌ده.
* هر پرنده در فریم فعلی یک track هست که مشخصه‌هایی مثل ID، موقعیت و وضعیت داره.

**9. رسم Bounding Box و شمارش پرنده‌ها**

bird\_count = 0

for track in tracks:

if not track.is\_confirmed():

continue

* بعضی ردیابی‌ها ممکنه **تأیید نشده** باشن (مثلاً اشتباه یا ناقص) → با is\_confirmed() چک می‌کنیم.

track\_id = track.track\_id

if track\_id not in counted\_ids:

counted\_ids.add(track\_id)

* هر پرنده یک شناسه (track\_id) داره.
* اگه این ID تا حالا دیده نشده بود، به مجموعه counted\_ids اضافه میشه.

**10. کشیدن جعبه و نوشتن ID روی تصویر**

x1, y1, x2, y2 = map(int, track.to\_ltrb())

cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)

cv2.putText(frame, f"Bird {track\_id}", (x1, y1 - 10),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.6, (0, 255, 0), 2)

bird\_count += 1

**توضیح:**

* to\_ltrb(): مختصات جعبه فعلی پرنده رو می‌ده.
* cv2.rectangle: مستطیل سبز دور پرنده می‌کشه.
* cv2.putText: اسم و ID پرنده رو بالای مستطیل می‌نویسه.
* bird\_count: تعداد پرنده‌های توی فریم جاری.

**11. نوشتن تعداد پرنده‌ها روی تصویر**

cv2.putText(frame, f"Birds Now: {bird\_count}", (10, 30),

cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0, 0, 255), 2)

**توضیح:**

* گوشه بالا سمت چپ فریم، تعداد پرنده‌های فعلی رو نشون می‌ده.

**12. نمایش فریم خروجی**

cv2.imshow("Bird Detection (Resized)", frame)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

**توضیح:**

* imshow: نمایش فریم.
* waitKey: هر ۱ میلی‌ثانیه صبر می‌کنه و چک می‌کنه که کاربر کلید q زده یا نه.

**13. پایان کار**

cap.release()

cv2.destroyAllWindows()

**توضیح:**

* release(): فایل ویدیو رو می‌بنده.
* destroyAllWindows(): همه‌ی پنجره‌های نمایشی رو می‌بنده.