

بخش اول

آماده سازی دیتاست و پیش پردازش داده ها

ما برای این منظور دیتاست SportsMOT را انتخاب کردیم. این دیتاست شامل ۲۴۰ ویدئو ورزشی از ورزش های فوتبال، والیبال و بسکتبال است. رزولوشن هر ویدئو 720p و نرخ فریم آن ۲۵ فریم بر ثانیه می باشد. این دیتاست به سه تقسیم بندی شده است. تعداد ویدئوها در train، validation و test ۱۵۰، ۱۵ و ۴۵ میباشد. برای هر ویدئو تمامی فریم ها با فرمت jpg در زیرفolder img1 ذخیره شده است. همچنین برای هر ویدئو یک فایل gt.txt نیز وجود دارد که در آن اطلاعات ردیابی (x, y, width, height) مربوط به تمامی id ها در تمامی فریم ها موجود می باشد. لازم به ذکر است که در این دیتاست تنها کلاس بازیکن لیبل گذاری شده است.

داده ها از نظر کیفیت بررسی شده و از این بابت هیچ مشکلی وجود نداشت. ما با کمک اطلاعات موجود در فایل football.txt) تنها از ویدئوهای مربوط به دسته فوتبال برای آموزش شبکه تشخیص اشیاء استفاده کردیم.

همچنین جهت سازگاری داده ها، لیبل های موجود (MOT annotations) به لیبل های نیاز برای فاین تیون کردن شبکه تشخیص شیء (YOLO format) تبدیل شدند.

فرآیند حذف داده های زائد هنگام آموزش مدل تشخیص شیء انجام شد به این نحو که متوجه شدیم در برخی فریم ها مانند نمونه زیر برای دو آیدی متفاوت اطلاعات مکانی یکسان ثبت شده است.

فریم ۴۵۶ از ویدئو v_i2_L4qquVg0_c006

6148	456, 1, 430, 224, 35, 47, 1, 1, 1
6149	456, 2, 900, 179, 26, 40, 1, 1, 1
6150	456, 3, 626, 342, 36, 58, 1, 1, 1
6151	456, 4, 944, 520, 56, 93, 1, 1, 1
6152	456, 5, 1188, 324, 25, 61, 1, 1, 1
6153	456, 8, 422, 314, 27, 62, 1, 1, 1
6154	456, 9, 700, 414, 40, 88, 1, 1, 1
6155	456, 11, 1186, 231, 20, 51, 1, 1, 1
6156	456, 12, 1198, 296, 22, 58, 1, 1, 1
6157	456, 13, 1186, 231, 20, 51, 1, 1, 1
6158	456, 14, 338, 631, 46, 89, 1, 1, 1

این داده ها به عنوان داده های زائد حذف شدند.

به هنگام آموزش مدل به عنوان مثال هنگام استفاده از `model.train`. تصاویر به صورت خودکار به رنج ۰ تا ۱ اسکیل میشوند. همچنین نرمال سازی میانگین و انحراف از معیار نیز روی آن ها اعمال میشود.

از طرف دیگر به صورت اتوماتیک روش های آگمنتیشن مانند `mosaic`, `MixUp`, `random horizontal flip`, `color jitter`, `random affine transformations`, `hsv augmentation` روی دیتابست اعمال میشود.

بخش دوم

پیاده سازی الگوریتم ردیابی اشیاء

۱. تشنیف اشیاء

- ما از مدل **YOLOv8n** برای تشخیص اشیاء استفاده کردیم.

دلایل انتخاب:

* سرعت بالا و کارایی مناسب برای پردازش ویدئوهای بلادرنگ

- **YOLOv8n** - بهینهسازی شده است تا با حفظ دقت مناسب، سرعت پردازش بالایی داشته باشد. این ویژگی برای ویدئوهای ورزشی که نیاز به پردازش بلادرنگ دارند، بسیار حیاتی است.

- در محیطهای ورزشی، اشیاء (مانند بازیکنان یا توپ) به سرعت حرکت می‌کنند، بنابراین مدل باید قادر باشد با سرعت بالا فریم‌ها را پردازش کند.

* دقت مناسب در تشخیص اشیاء کوچک

- در ویدئوهای ورزشی، اشیاء مورد نظر (مانند توپ یا بازیکنان) ممکن است در فاصله دور و با ابعاد کوچک ظاهر شوند. **YOLOv8n** با معماری پیشرفته‌تر نسبت به نسخه‌های قبلی **YOLO**، توانایی بهتری در تشخیص اشیاء کوچک دارد.

* سادگی و سهولت استفاده

- **YOLOv8n** به دلیل یکپارچگی با کتابخانه‌ای مانند **Ultralytics** استفاده و پیاده سازی آسانی دارد.

* بهینهسازی برای سخت‌افزارهای محدود

- **YOLOv8n** نسخه‌ای سبک‌وزن از **YOLOv8** است و برای اجرا روی سخت‌افزارهای محدود (مانند لپتاپ‌ها) بهینهسازی شده است. این ویژگی برای پروژه‌هایی که نیاز به اجرا در محیطهای با منابع محدود دارند، بسیار مناسب است.

* پشتیبانی از ردیابی اشیاء

- **YOLOv8n** به راحتی با الگوریتم‌های ردیابی مانند **BYTETrack** یا **DeepSORT** ترکیب می‌شود. این قابلیت برای ردیابی بازیکنان یا توپ در طول ویدئوهای ورزشی بسیار مفید است.

* قابلیت آموزش و فاین-تیون

YOL0v8n - به شما امکان می دهد مدل را روی داده های خاص ورزشی (مانند تصاویر بازیکنان یا توپ) فاین-تیون کنید. این کار باعث بهبود دقت مدل در تشخیص اشیاء مرتبط با حوزه ورزشی می شود.

* مقیاس پذیری

YOL0v8n - براحتی می تواند با نسخه های بزرگتر **YOL0v8** (مانند **YOL0v8m**, **YOL0v8s** و ...) جایگزین شود. اگر در آینده نیاز به دقت بیشتر یا پردازش ویدئوهای با کیفیت بالاتر داشته باشید، می توانید بدون تغییر ساختار کلی پروژه، مدل را ارتقا دهید.

* پشتیبانی از چندین کلاس

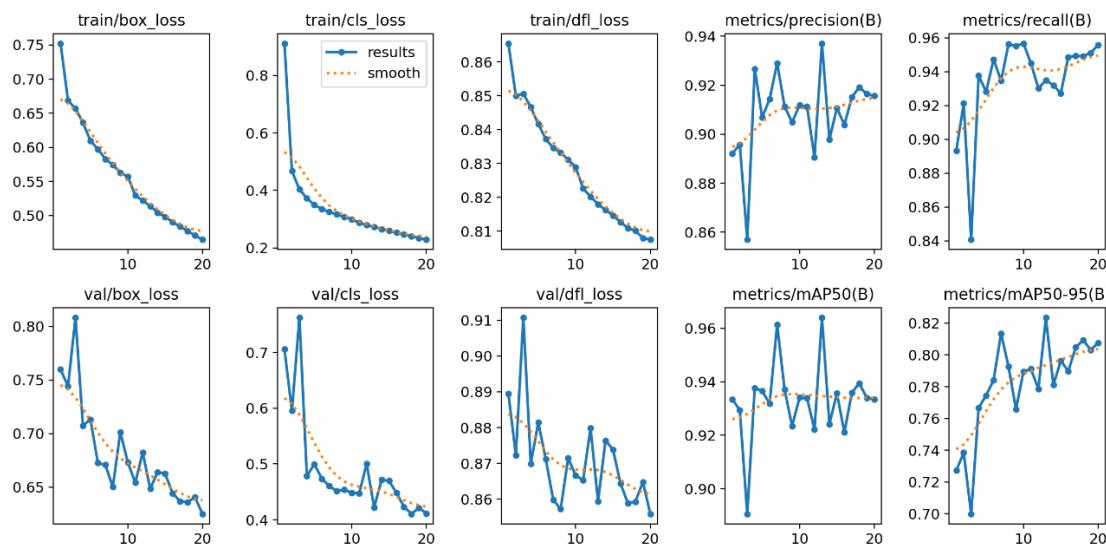
YOL0v8n - قادر است چندین کلاس مختلف (مانند بازیکنان، توپ، دروازه و ...) را به طور همزمان تشخیص دهد. این ویژگی برای تحلیل ویدئوهای ورزشی که شامل اشیاء متنوعی هستند، بسیار مفید است.

* کاهش نویز و خطأ در تشخیص

YOL0v8n - با بهبود الگوریتم های تشخیص و فیلتر کردن پیش بینی های نادرست، نویز و خطأ را کاهش می دهد. این موضوع در ویدئوهای ورزشی که ممکن است حرکات سریع و پیچیده وجود داشته باشد، بسیار مهم است.

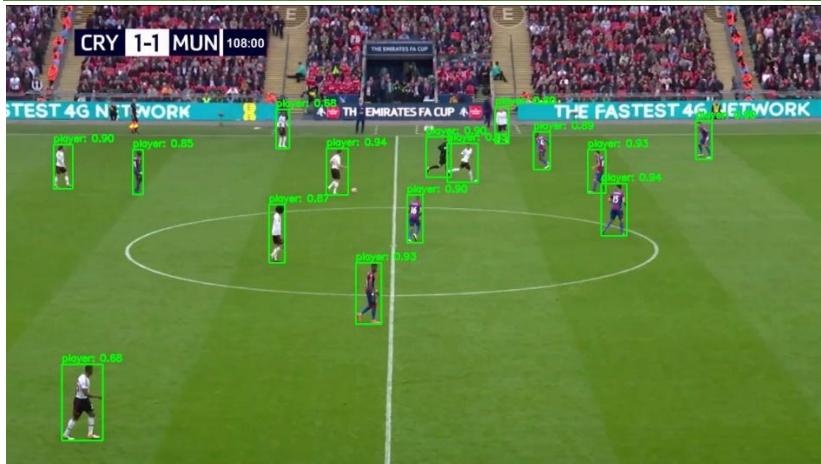
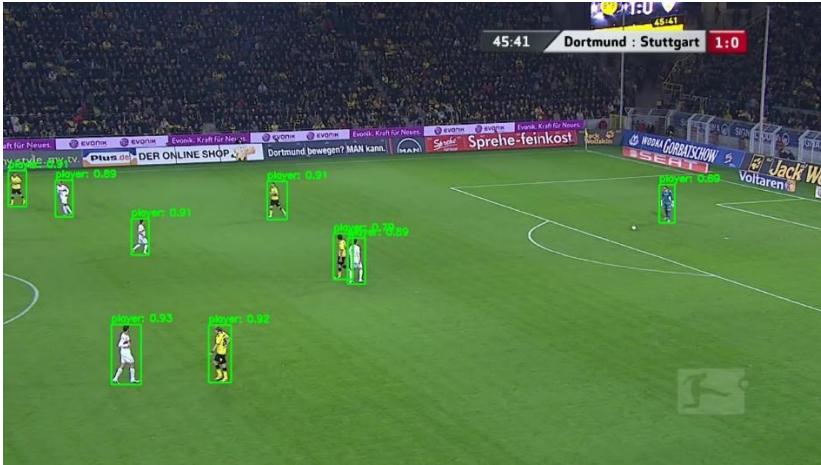
برای آموزش شبکه YOLOv8n، ساختار دیتا را تغییر دادیم به این نحو که تمام فریم های مربوط به ویدئوهای فوتبال را برای هر بخش آموزش، ولیدیشن و تست درون یک پوشه قرار دادیم همچنین لیبل مربوط به هر فریم را نیز ساخته و تمام آن ها را نیز در پوشه دیگری قرار دادیم.

نמודارهای مربوط به Loss, Accuracy : YOLOv8n آموزش مدل

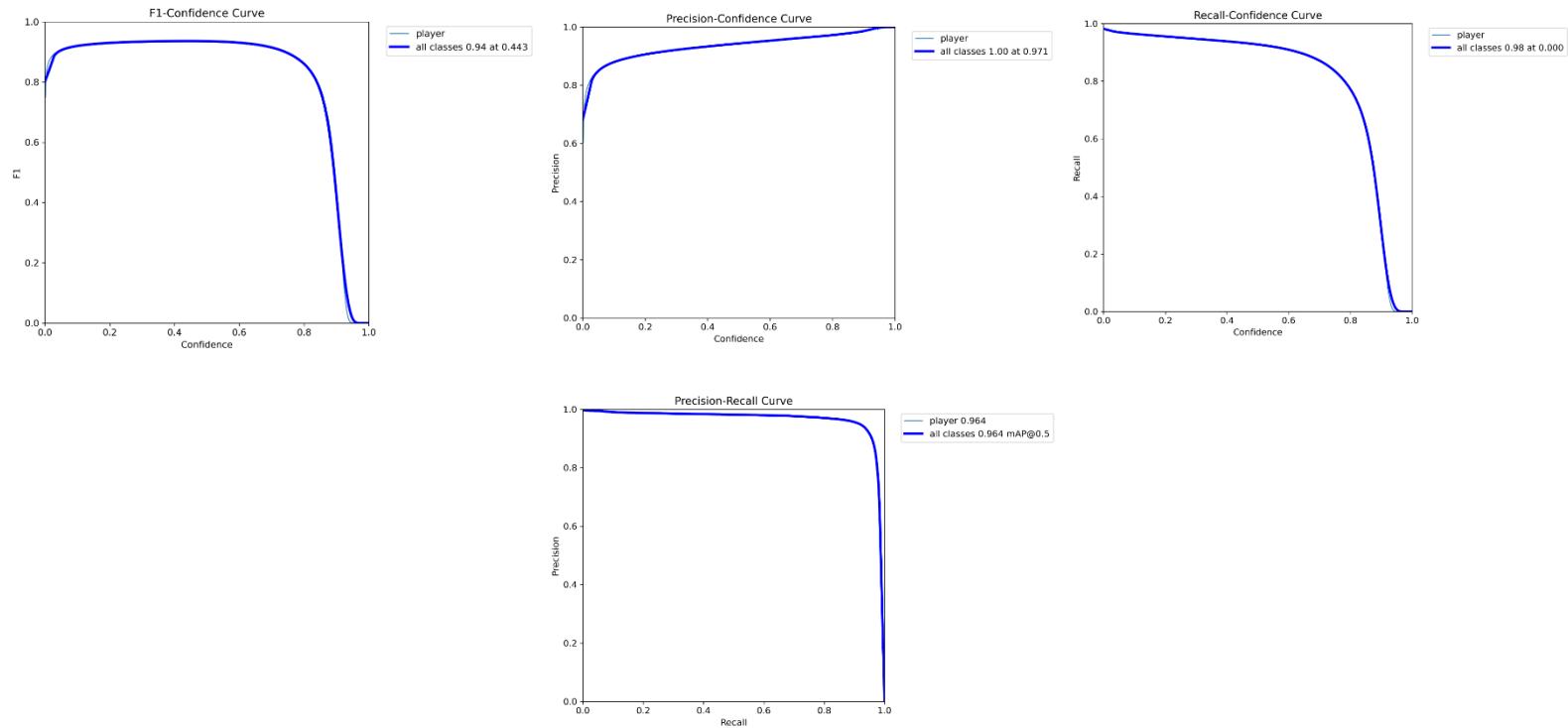


۳- خروجی مدل تشخیص بر روی چند فریم از ویدئو:

همانطور که از تصاویر خروجی مشخص است مدل به خوبی توانسته است اشیاء (بازیکنان) را شناسایی کند.



۴- دقت و عملکرد مدل با استفاده از معیارهای مختلف:



۲۲ ردیابی تک شیء

۱- ما از الگوریتم CSRT استفاده کردیم.

دلیل انتخاب:

* دقت بالا در ردیابی اشیاء با حرکت پیچیده

- الگوریتم CSRT به دلیل استفاده از اطلاعات کانالهای رنگی (Channel Reliability) و اطلاعات مکانی (Spatial Reliability)، دقت بالایی در ردیابی اشیاء با حرکات پیچیده و غیرخطی دارد. این ویژگی برای ردیابی بازیکنان یا توپ در ویدئوهای ورزشی که حرکات سریع و غیرقابل پیش‌بینی دارند، بسیار مفید است.

* مقاومت در برابر تغییرات ظاهری شیء

- در ویدئوهای ورزشی، شیء مورد نظر (مانند توب یا بازیکن) ممکن است دچار تغییرات ظاهری مانند تغییر اندازه، چرخش، تغییر نور یا تغییر زاویه دید شود. **CSRT** به دلیل استفاده از مکانیزم‌های ذکر شده، مقاومت خوبی در برابر این تغییرات دارد.

* بهینه‌سازی برای ردیابی اشیاء کوچک

CSRT - عملکرد خوبی در ردیابی اشیاء کوچک (مانند توب در فواصل دور) دارد. این ویژگی برای ویدئوهای ورزشی که در آن شیء مورد نظر ممکن است در بخش کوچکی از فریم ظاهر شود، بسیار مهم است.

* پایداری در ردیابی حتی با وجود نویز یا پس‌زمینه شلوغ

- در ویدئوهای ورزشی، پس‌زمینه اغلب شلوغ و پر از حرکات اضافی است (مانند تماشاگران یا بازیکنان دیگر). **CSRT** به دلیل استفاده از مکانیزم‌های فیلتر کردن نویز، می‌تواند شیء مورد نظر را حتی در چنین شرایطی به طور پایدار ردیابی کند.

* عدم نیاز به آموزش مدل (**Training-Free**)

CSRT - یک الگوریتم ردیابی مبتنی بر ویژگی‌های تصویر است و نیاز به آموزش مدل ندارد. این ویژگی باعث می‌شود پیاده‌سازی و استفاده از آن سریع و ساده باشد، به ویژه برای پروژه‌هایی که نیاز به توسعه سریع دارند.

* قابلیت تنظیم پارامترها برای بهبود عملکرد

CSRT - پارامترهای قابل تنظیمی دارد که می‌تواند برای بهبود عملکرد ردیابی در شرایط خاص (مانند ویدئوهای ورزشی) استفاده شود. این انعطاف‌پذیری به شما امکان می‌دهد الگوریتم را با نیازهای پروژه خود تطبیق دهید.

* پشتیبانی از ردیابی در زمان واقعی (**Real-Time**)

Real-Time CSRT - با وجود دقت بالا، از نظر سرعت نیز بهینه‌سازی شده است و می‌تواند در برنامه‌های بلادرنگ (**Time**) استفاده شود. این ویژگی برای ویدئوهای ورزشی که نیاز به پردازش سریع دارند، بسیار مهم است.

* مقاومت در برابر اختفای (**Occlusion**) موقت شیء

- در ویدئوهای ورزشی، ممکن است شیء مورد نظر به طور موقت توسط شیء دیگری پوشانده شود (مثلاً توب توسط بازیکن). **CSRT** می‌تواند در چنین شرایطی، ردیابی را ادامه دهد و پس از آشکار شدن مجدد شیء، آن را دوباره شناسایی کند.

* یکپارچگی با کتابخانه‌های معروف پردازش تصویر CSRT - در کتابخانه‌های معروف پردازش تصویر مانند OpenCV پیاده‌سازی شده است. این موضوع باعث می‌شود به راحتی با سایر ابزارها و الگوریتمها (مانند YOLOv8n) ترکیب شود.

- خروجی ردیابی تک شیء در: [لينك](#)

- چالش‌های معمول در ردیابی تک شیء مانند پوشیدگی (occlusion)، تغییر مقیاس (scale)، تغییر نور (illumination change) و تغییر نور (variation):

* پوشیدگی (Occlusion)

پوشیدگی زمانی اتفاق می‌افتد که شیء مورد نظر توسط شیء دیگری در صحنه پوشانده می‌شود یا به طور موقت از دید دوربین خارج می‌شود. این چالش می‌تواند باعث از دست رفتن موقت یا دائمی شیء شود.

- چالش‌ها:

- تشخیص اینکه آیا شیء به طور کامل پوشیده شده است یا خیر.

- پیش‌بینی موقعیت شیء در زمانی که پوشیدگی رخ می‌دهد.

- بازیابی شیء پس از پایان پوشیدگی.

- راه حل‌ها:

- استفاده از مدل‌های پیش‌بینی حرکت (motion models) برای تخمین موقعیت شیء در زمان پوشیدگی.

- به کارگیری روش‌های یادگیری عمیق که قادر به تشخیص شیء حتی در شرایط ناقص هستند.

- استفاده از ویژگی‌های ظاهری (appearance features) برای شناسایی مجدد شیء پس از پوشیدگی.

* تغییر مقیاس (Scale Variation)

تغییر مقیاس زمانی اتفاق می‌افتد که شیء مورد نظر در طول زمان به دوربین نزدیک یا دور می‌شود، یا اینکه دوربین زوم می‌کند. این تغییرات باعث می‌شوند که اندازه شیء در تصویر تغییر کند.

- چالش‌ها:

- تطبیق اندازه پنجره ردیابی با اندازه شیء در حال تغییر.

- تشخیص تغییرات مقیاس در زمان واقعی.

- راه حل‌ها:

- استفاده از روش‌های چند مقیاسی (**multi-scale approaches**) که در آن تصویر در چندین مقیاس مختلف پردازش می‌شود.

- به کارگیری شبکه‌های عصبی که قادر به تشخیص و تطبیق خودکار مقیاس هستند.

- استفاده از الگوریتم‌هایی که به طور پویا اندازه پنجره ردیابی را تنظیم می‌کنند.

* تغییر نور (**Illumination Change**)

تغییر نور می‌تواند به دلیل تغییرات در شرایط نوری محیط یا حرکت شیء به مناطق با نور متفاوت رخ دهد. این تغییرات می‌توانند باعث تغییر در ظاهر شیء شوند و ردیابی آن را دشوار کنند.

- چالش‌ها:

- تشخیص شیء در شرایط نوری مختلف.

- مقاومت در برابر تغییرات شدید نور که ممکن است باعث از دست رفتن ویژگی‌های ظاهری شیء شود.

- راه حل‌ها:

- استفاده از ویژگی‌های مقاوم به نور (**illumination-invariant features**) مانند گرادیان‌های تصویر یا ویژگی‌های بافتی.

- به کارگیری روش‌های نرمال‌سازی نور (**illumination normalization**) برای کاهش اثر تغییرات نور.

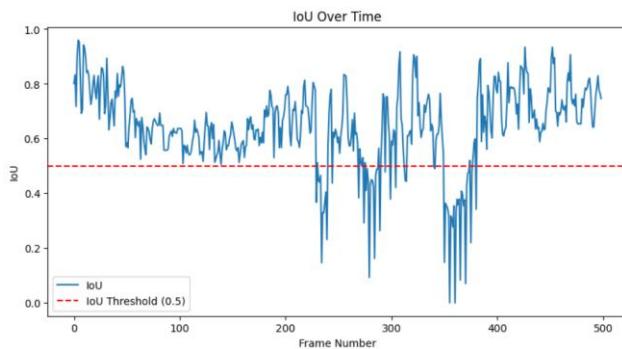
- استفاده از مدل‌های یادگیری عمیق که قادر به یادگیری ویژگی‌های مقاوم به نور هستند.

۴- ارزیابی الگوریتم با معیارهای مختلف:

ما برای ارزیابی، دو ویدئو از پوشه تست را انتخاب کردیم همچنین برای هر یک مشخصات یک آیدی در اولین فریم را نیز مشخص کردیم.

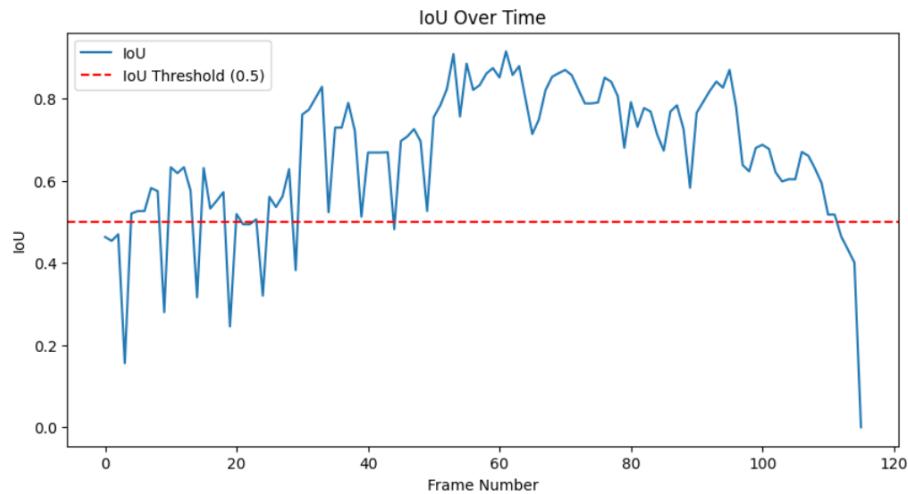
v_ITo3sCnpw_k_c012, id:8, init_bbox = (231, 234, 41, 70)

Tracking Accuracy (IoU > 0.5) : 87.78%



v_i2_L4qquVg0_c007, id: 1, init_bbox = (747, 279, 30, 71)

Tracking Accuracy (IoU > 0.5) : 86.21%



۵- الگوریتم CSRT در چه شرایطی عملکرد بهتری دارد و در چه شرایطی دچار خطا میشود:

* شرایطی که CSRT عملکرد بهتری دارد:

** اشیاء با اندازه متوسط تا بزرگ:

CSRT - در ردهایی اشیایی که اندازه‌ی نسبتاً بزرگی در تصویر دارند، عملکرد بهتری دارد. این الگوریتم از اطلاعات مکانی و کanal‌های رنگی به طور همزمان استفاده می‌کند، بنابراین برای اشیاء بزرگتر، اطلاعات بیشتری برای تحلیل وجود دارد.

** حرکت نسبتاً آهسته و قابل پیش‌بینی:

- اگر شیء مورد نظر با سرعت کم یا متوسط حرکت کند و تغییرات ناگهانی در مسیر حرکت آن وجود نداشته باشد، CSRT می‌تواند به خوبی آن را ردهایی کند.

** تغییرات نور و شرایط نوری متعادل:

CSRT - در شرایط نوری متعادل و تغییرات نور متوسط عملکرد خوبی دارد. این الگوریتم از اطلاعات کanal‌های رنگ استفاده می‌کند، بنابراین اگر تغییرات نور شدید نباشد، می‌تواند شیء را به خوبی تشخیص دهد.

** زمینه‌ی نسبتاً ساده یا کم‌تکرار:

- اگر پس‌زمینه‌ی تصویر ساده باشد یا الگوهای تکراری کمتری داشته باشد، CSRT می‌تواند شیء را با دقت بیشتری ردهایی کند.

** اشیاء با بافت و رنگ مشخص:

CSRT - برای اشیایی که بافت (texture) و رنگ مشخصی دارند، عملکرد بهتری دارد. این الگوریتم از اطلاعات کanal‌های رنگی و مکانی برای تشخیص شیء استفاده می‌کند، بنابراین اگر شیء ویژگی‌های بصری واضحی داشته باشد، دقت ردهایی افزایش می‌یابد.

* شرایطی که CSRT دچار خطا می‌شود:

** اشیاء با اندازه بسیار کوچک:

- اگر شیء مورد نظر در تصویر بسیار کوچک باشد، CSRT ممکن است نتواند آن را بدستی تشخیص دهد، زیرا اطلاعات کافی برای تحلیل در اختیار ندارد.

** حرکت سریع یا تغییر جهت ناگهانی:

- در صورت حرکت سریع شیء یا تغییر جهت ناگهانی، CSRT ممکن است نتواند شیء را بدستی ردیابی کند، زیرا این الگوریتم برای حرکات آهسته و قابل پیش‌بینی طراحی شده است.

** تغییرات شدید نور یا شرایط نوری نامناسب:

- اگر تغییرات نور شدید باشد (مثلاً عبور از یک منطقه تاریک به روشن)، CSRT ممکن است دچار خطا شود، زیرا اطلاعات کانال‌های رنگی تحت تأثیر قرار می‌گیرند.

** پس‌زمینه‌ی شلوغ یا دارای الگوهای تکراری:

- اگر پس‌زمینه‌ی تصویر شلوغ باشد یا الگوهای تکراری زیادی داشته باشد، CSRT ممکن است شیء را با پس‌زمینه اشتباه بگیرد و دچار خطا شود.

** تغییر شکل یا چرخش شیء:

- اگر شیء مورد نظر تغییر شکل دهد (مثلاً تغییر اندازه یا چرخش)، CSRT ممکن است نتواند آن را بدستی ردیابی کند، زیرا این الگوریتم برای تغییرات شدید در شکل شیء بهینه‌سازی نشده است.

** موانع یا اختفای موقت شیء:

- اگر شیء به‌طور موقت توسط مانعی پنهان شود (CSRT, occlusion) ممکن است شیء را گم کند یا به اشتباه به پس‌زمینه منتقل شود.

** اشیاء با رنگ یا بافت مشابه پس‌زمینه:

- اگر شیء و پس‌زمینه از نظر رنگ یا بافت بسیار شبیه باشند، CSRT ممکن است نتواند شیء را بدستی تشخیص دهد.

heatmap - ۶

لینک خروجی

۲.۳ ردیابی چند شیء

۱- ما الگوریتم DeepSORT را انتخاب کردیم؛ DeepSORT نسخه‌ی پیشرفته‌ی SORT است که یک مازول ظاهر عمیق (Re-ID) را اضافه می‌کند. به این معنا که هر برش تشخیص‌یافته از تصویر از طریق یک شبکه‌ی عصبی کوچک عبور می‌کند تا یک بردار ویژگی تولید کند. سپس ردیاب از یکتابع هزینه‌ی ترکیبی شامل $\text{IoU}_{\text{جعبه‌های محدود}} \cdot \text{فاصله‌ی ظاهری}$ برای پیوندهای استفاده می‌کند:

- ۱ . کاهش نرخ تغییر شناسه: (ID switch)
از آنجا که ویژگی‌های ظاهری هر جسم بررسی می‌شود، احتمال اشتباه گرفتن بازیکنان با لباس مشابه کمتر می‌شود.
- ۲ . مناسب برای ورزش‌های شلوغ: در فوتبال، بسکتبال یا والیبال، بازیکنان اغلب با یکدیگر همپوشانی دارند. ویژگی‌های ظاهری عمیق به حفظ شناسه کمک می‌کنند.
- ۳ . مقاوم در برابر انسدادهای جزئی: حتی اگر جعبه‌ی محدود کننده به طور موقت تغییر کند، ویژگی ظاهری عمیق می‌تواند تشخیص دهد که جسم همچنان همان است.

زمانی مناسب است که: DeepSORT

- شناخت هویت بازیکنان در طول زمان مهم باشد.
- انسدادهای مکرر یا شباهت لباس‌ها باعث شکست روش‌های معمولی IoU شود.
- توان پردازشی کافی برای محاسبه‌ی تعبیه‌های ظاهری را داشته باشد.

۲- رابطه بین تشخیص و ردیابی

ابتدا تشخیص اشیا روی هر فریم انجام می‌شود :

۱ . خروجی‌های آشکارساز: شامل مجموعه‌ای از X, y, w, h bounding box، امتیازهای اطمینان و احتمالاً شناسه‌های کلاسی (مانند "بازیکن"، "توب") .

۲ . ورودی ردیاب : این تشخیص‌ها به عنوان ورودی به مرحله‌ی ارتباط اشیا داده می‌شوند تا با مسیرهای موجود تطبیق داده شوند .

بنابراین :

- تشخیص، مسئول شناسایی اشیا در هر فریم است .

- ردیابی، مسئول حفظ شناسه‌ی یکتا برای هر شیء در طول فریمهای متوالی است .

در ویدئوهای ورزشی، این بدان معناست که جعبه‌های محدود کننده‌ی یک بازیکن یا توب باید از یک فریم به فریم دیگر متصل شوند تا یک مسیر پیوسته تشکیل دهند .

۳- فرآیند تطبیق

در deepSORT به صورت زیر است:

*پیش‌بینی مسیرها با فیلتر کالمن

*محاسبه‌ی ماتریس هزینه با استفاده از :

- فاصله‌ی IoU

- فاصله‌ی ویژگی‌های ظاهری عمیق

*الگوریتم مجارستانی برای یافتن بهترین تطابق

*بهروزرسانی مسیرها : مسیرهای تطبیق‌یافته حفظ می‌شوند، مسیرهای جدید اضافه می‌شوند، و مسیرهای از دست رفته حذف می‌شوند .

خروجی DeepSORT در [لينك](#)

ارزیابی بر روی تمام ویدئوهای ولیدیشن فوتبال:

Average Metrics:

IDF1: 0.6283

MOTA: 0.5787

Precision: 0.7198

۶- چالش های رایج:

تغییر روشنایی

- تعریف : تغییرات در روشنایی یا رنگ، مانند تغییر روز به شب یا سایدهای استادیوم .
- تأثیر :
 ۱. تشخیص دهنده ممکن است دقت پایین تری داشته باشد .
 ۲. ویژگی های ظاهری عمیق ممکن است دچار خطا شوند .
- DeepSORT می تواند با تغییرات خفیف مقابله کند اما تغییرات شدید ممکن است باعث شکست شود .

انسداد

- تعریف : بازیکنان یا توپ به طور موقت پوشانده می شوند .
- تأثیر :
 ۱. مسیرهای اشیا ممکن است گم شوند .
 ۲. احتمال تغییر شناسه افزایش می یابد .

• DeepSORT ویژگی‌های ظاهری عمیق می‌توانند شیء را پس از انسداد بازیابی کنند.

تغییر شناسه (ID Switch)

• تعریف: بازیکنان یا توب به اشتباه تغییر شناسه می‌دهند.

• علت:

۱. همپوشانی بازیکنان

۲. گم شدن موقت مسیر

• کاهش اثرات:

○ استفاده از مدل‌های Re-ID قوی‌تر

○ تنظیم بهتر تابع هزینه برای تفکیک بهتر مسیرها

۷- الگوریتم انتخاب شده برای مقایسه Deep-EIoU می‌باشد.

Deep-EIoU یک چارچوب ردیابی چندگانه اشیا است که از معیارهای بهبودیافته‌ی همپوشانی جعبه محدود کننده

EIoU همراه با یک استخراج‌کننده ویژگی عمیق برای ارتباط اشیا استفاده می‌کند. در حالی که IoU سنتی

فقط ناحیه‌ی همپوشانی بین جعبه‌ها را بررسی می‌کند، EIoU یا سایر معیارهای بهبودیافته‌ی همپوشانی مانند CIoU،

محدودیت‌های هندسی دقیق‌تری را در نظر می‌گیرند (مانند فاصله‌ی بین مرکز جعبه‌ها، نسبت ابعادی و غیره)، با افزودن

یک مولفه‌ی عمیق برای شناسایی مجدد یا پیونددگی پیشرفته‌ی Deep-EIoU اهداف زیر را دنبال می‌کند:

۱. بهبود تطبیق جعبه‌های محدود کننده: معیار پیشرفته‌ی IoU باعث کاهش عدم تطابق در شرایطی می‌شود که اشیا تغییرات کمی در مقیاس یا شکل داشته باشند.

۲. مدیریت بهتر سناریوهای ورزشی: در ورزش‌ها، حرکت سریع بازیکنان یا توب می‌تواند باعث تغییرات ناگهانی در جعبه‌های محدود کننده شود. تابع هزینه‌ی مبتنی بر IoU توسعه‌یافته می‌تواند به این تغییرات حساس‌تر باشد.

۳. کاهش تغییر شناسه: (ID switch) در صورت ترکیب با تعیینهای عمیق ظاهری، می‌تواند بازیکنان با ظاهر مشابه (مثلاً بازیکنان یک تیم با لباس یکسان) را بهتر تفکیک کند.

بنابراین، Deep-EIoU در شرایط زیر مناسب است:

• نیاز به تطبیق انعطاف‌پذیرتر جعبه‌های محدود کننده نسبت به IoU معمولی دارد.

- محیطی که در آن تغییرات سریع در مقیاس یا شکل اشیا وجود دارد (مانند تغییر حالت بازیگنان یا چرخش توب).
 - امکان افزودن ویژگی‌های عمیق برای تحلیل بیشتر ظاهر یا حرکت را دارد.
- مقایسه بین نتایج الگوریتم Deep-EIoU، DeepSORT و Deep-EIoU

methods	Idf1	mota
DeepSORT	63	58
Deep-EIoU	75	95.1

همانطور که مشخص است روش Deep-SORT به مرتب قوی تر از Deep-EIoU است.

خروجی روش Deep-EIoU در [لينك](#)

بخش سوم

بهبود الگوریتم و ارایه روش پیشنهادی

۳.۱ تحقیق

مقابلہ با مشکل انسداد در ردیابی اشیاء چندگانه

یکی از چالش‌های عمدۀ در ردیابی اشیاء چندگانه (**MOT**) در ورزش، انسداد است. در این حالت، بازیکنان روی یکدیگر قرار می‌گیرند و باعث تغییر شناسه‌ها یا قطع ردیابی می‌شوند. برای بهبود ردیابی در چنین شرایطی، ما یک روش ارتباط جهانی تراکت‌ها (**GTA**) با ترکیب **ExpansionIoU** و ویژگی‌های عمیق را پیشنهاد می‌کنیم.

روش بهبود: ارتباط جهانی تراکت‌ها (**GTA**)

مقدمه

-**GTA** - یک مازول پردازش پس از ردیابی است که نتایج ردیابی را اصلاح و بهبود می‌بخشد.

- این روش تراکت‌هایی که دارای چندین شناسه هستند را تجزیه کرده و تراکت‌های یکسان اما قطع شده را به هم متصل می‌کند.

- می‌تواند به عنوان یک ابزار بهبود مستقل برای **DeepSORT** یا **Deep-EIoU** عمل کند تا کاهش تغییر شناسه‌ها و بهبود پیوستگی شناسه‌ها را فراهم کند.

مراحل اجرای الگوریتم بهبود

۱. ردیابی اولیه با **Deep-EIoU**

- استفاده از **ExpansionIoU** برای افزایش تدریجی جعبه‌های محدود کننده.

- تخصیص جعبه‌های محدود کننده با استفاده از ترکیبی از **IoU** و ویژگی‌های عمیق ظاهری.

- اجرای الگوریتم مجارستانی برای تطابق تشخیص‌ها با تراکت‌ها.

۲. جداسازی تراکت‌ها برای مقابله با انسداد

- اعمال خوشبندی DBSCAN برای تفکیک تراکت‌های ترکیبی به چند شناسه مجزا.

- استفاده از شباهت کسینوسی روی ویژگی‌های ظاهری برای ایجاد خوشبندی دقیق‌تر.

- تخصیص مجدد هر جعبه محدود‌کننده به یک تراکت فرعی بر اساس شباهت ظاهری.

۳. اتصال جهانی تراکت‌ها (اتصال مجدد تراکت‌های شکسته)

- محاسبه یک ماتریس فاصله کسینوسی برای پیدا کردن تراکت‌های شکسته اما مربوط به یک بازیکن.

- اعمال محدودیت‌های فضایی برای جلوگیری از اتصال نادرست:

- بازیکنان نباید به طور ناگهانی از یک سمت زمین به سمت دیگر پرش کنند.

- موقعیت مرکز جعبه‌های محدود‌کننده باید در یک محدوده مشخص قرار بگیرد.

- خوشبندی سلسله‌مراتبی برای ادغام تراکت‌های شکسته به یک ID واحد.

۴. بهینه‌سازی نهایی: تخصیص آگاهانه بر اساس اطمینان

- اعمال یک آستانه اطمینان برای اتصال تراکت‌ها.

- استفاده از فیلتر کالمون برای پیش‌بینی مبنی بر حرکت جهت مدیریت انسدادهای موقتی.

نتایج بهبود با GTA

با افزودن SportsMOT به Deep-EIoU عملکرد ردیابی در GTA بهبود یافت:

- HOTA از ۷۷.۲٪ به ۸۱.۰۴٪ افزایش یافت.

- IDF1 از ۷۹.۸٪ به ۸۶.۵۱٪ افزایش یافت.

- تعداد تغییر شناسه‌ها ۱۷۲ مورد کاهش یافت و پیوستگی ID بهبود یافت.

به همین ترتیب، در SoccerNet-Tracking، پس از ترکیب GTA:

- HOTA 3.7٪ افزایش یافت و به ۸۳.۱٪ رسید.

- IDF1 بهبود ۶.۲۶٪ داشت.

– تعداد تغییر شناسه‌ها ۶۱۵ مورد کاهش یافت که نشان‌دهنده بهبود ردیابی بازیکنان در بلندمدت** است .

قسمت دوم - روش‌های بهینه‌سازی مدل برای استقرار و عرضه به عنوان محصول

چرا بهینه‌سازی ضروری است؟

عرضه یک مدل ردیابی به عنوان محصول مستلزم افزایش کارایی در موارد زیر است :

– سرعت: ردیابی زنده در ورزش به حداقل ۳۰ FPS نیاز دارد .

– مصرف حافظه: کاهش حجم محاسبات برای اجرا روی دستگاه‌های کم‌صرف یا سرورهای ابری اهمیت دارد .

– تعادل دقت و تاخیر: مدل‌های بهینه شده باید سرعت بالاتر را بدون کاهش قابل توجه عملکرد حفظ کنند .

دست‌بندی روش‌های بهینه‌سازی

1. کوانتیزاسیون (Quantization)

تعریف : دقت مدل را از INT8/FP16 به FP32 کاهش می‌دهد تا سرعت اجرا افزایش یابد .

: مزایا

– کاهش حجم حافظه (تا ۴ برابر کمتر) .

– بهبود سرعت استنتاج ، مخصوصاً در دستگاه‌های لبدی مثل Nvidia Jetson

: معایب

– ممکن است دقت کاهش یابد، به ویژه برای ردیابی اشیاء کوچک.

– نیاز به کالیبراسیون دقیق دارد تا از کاهش تمایز ویژگی‌ها جلوگیری شود .

2. هرس مدل (Model Pruning)

تعریف : حذف وزن‌های غیرضروری یا نورون‌ها/لایه‌های اضافی برای کاهش حجم مدل .

 مزایا:

– کاهش مصرف ذخیره‌سازی و محاسبات مناسب برای مدل‌های سبکتر.

– کاهش تاخیر برای ردیابی در زمان واقعی.

 معایب:

– حذف بیش از حد ممکن است ویژگی‌های کلیدی را از بین ببرد و باعث کاهش عملکرد در تشخیص بازیکنان شود.

مثال: Deep-EIoU بعد از هرس ساختاری اندازه مدل را 40% کاهش داد و 98% دقت ردیابی حفظ شد.

3. تقطیر دانش Knowledge Distillation

تعریف: یک مدل کوچکتر $\ast\ast$ (دانشآموز) $\ast\ast$ را از یک مدل بزرگتر $\ast\ast$ (معلم) $\ast\ast$ آموخته می‌دهد.

 مزایا:

– دقت بالا را حفظ می‌کند در حالی که سبکتر می‌شود.

– ایده‌آل برای دستگاه‌های موبایل و جاسازی شده.

 معایب:

– نیاز به تنظیم دقیق برای مجموعه داده‌های ورزشی دارد تا الگوهای حرکتی را بدرستی یاد بگیرد.

مثال: مدل تقطیر شده Deep-EIoU FPS در مقایسه با 14.6 FPS سرعت 25 در مدل اصلی را ارائه داد، HOTA با تنها 1.2% افت در

۱. استقرار در لبه با OpenVINO / TensorRT

استفاده از شتابدهنده سختافزاری (TensorRT برای GPU های Nvidia و OpenVINO برای CPU های Intel باعث بهبود سرعت استنتاج میشود.

مزایا: 

- استنتاج سریع تر، بهینه برای برنامه های استریم با تأخیر کم.

- مناسب برای سختافزارهای AI لبدای.

معایب: 

- نیاز به بهینه سازی خاص سختافزار دارد.

- محدودیت سازگاری بین پردازنده های مختلف دارد.

۳.۲ پیاده سازی

خروجی Deep-EIoU + GTA در [لينك](#)

مقایسه

Methods	IDF1	MOTA
Deep-EIoU	75	95.1
Deep-EIoU + GTA	86.51	96.32

به طرز معناداری پس از اعمال الگوریتم GTA عملکرد الگوریتم ردیابی پیشنهادی بهبود می یابد.