

Seguiment de jugadors en el futbol  
mitjançant tècniques de visió per  
computador basades en xarxes neuronals

Carme Corbi Sulé

---



Universitat  
Pompeu Fabra  
*Barcelona*

# Seguiment de jugadors en el futbol mitjançant tècniques de visió per computador basades en xarxes neuronals

TREBALL FI DE GRAU DE

Carme Corbi Sulé

Director: Alejandro Cartas, Coloma Ballester  
i Gloria Haro

Grau en Enginyeria en Sistemes Audiovisuals

Curs 2023-2024



Universitat  
Pompeu Fabra  
*Barcelona*

Escola  
d'Enginyeria







## **Agraïments**

En primer lloc, vull donar les gràcies als meus tutors de treball de fi de grau, Alejandro, Coloma i Gloria, per la seva guia, suport i paciència al llarg de tot aquest procés. La seva experiència i els seus consells han estat fonamentals per a la realització d'aquest treball.

També vull agrair als meus companys de la universitat i amics, que han estat una font constant de suport i motivació. Sense ells, no hauria pogut aconseguir tots els meus objectius al llarg de la carrera. Gràcies per les llargues hores d'estudi compartides i els moments de desconnexió que tant necessitavem. He tingut la sort de conèixer persones increïbles que sempre portaré amb mi.

Finalment, vull agrair especialment a la meva família, que sempre ha estat al meu costat recolzant-me en totes les decisions que he pres.



## **Resum**

El treball de final de grau se centra a investigar el seguiment de jugadors en el futbol a través de tècniques de visió per ordinador. S'explora l'eficàcia de diversos mètodes de seguiment d'objectes utilitzant el conjunt de dades SoccerNet, que ofereix una àmplia gamma d'imatges i vídeos de partits de futbol de diferents escenaris amb anotacions detallades. A través d'una anàlisi exhaustiva, es revisa l'estat actual dels mètodes de seguiment d'objectes, des de mètodes tradicionals fins a les últimes estratègies de xarxes neuronals profundes. S'avaluen els resultats amb mètriques específiques i s'identifiquen els errors i desafiaments trobats. A més, s'obté una millora en els resultats mitjançant l'ús del mètode StrongSORT, centrat a optimitzar la detecció i l'associació d'objectes. Finalment, s'obté un nou model entrenant un detector utilitzant deteccions automàtiques obtingudes a partir de tècniques com Background Subtraction i Human Pose Estimation, amb l'objectiu de millorar la precisió i fiabilitat de les deteccions de jugadors i, per tant, la qualitat global del seguiment de jugadors.

## **Resumen**

El trabajo de fin de grado se centra en investigar el seguimiento de jugadores en el fútbol a través de técnicas de visión por computadora. Se explora la eficacia de diversos métodos de seguimiento de objetos utilizando el conjunto de datos SoccerNet, que ofrece una amplia gama de imágenes y videos de partidos de fútbol de diferentes escenarios con anotaciones detalladas. A través de un análisis exhaustivo, se revisa el estado actual de los métodos de seguimiento de objetos, desde métodos tradicionales hasta las últimas estrategias de redes neuronales profundas. Se evalúan los resultados con métricas específicas y se identifican los errores y desafíos encontrados. Además, se obtiene una mejora en los resultados mediante el uso del método StrongSORT, centrado en optimizar la detección y la asociación de objetos. Finalmente, se obtiene un nuevo modelo entrenando un detector utilizando detecciones automáticas obtenidas a partir de técnicas como Background Subtraction y Human Pose Estimation, con el objetivo de mejorar la precisión y fiabilidad de las detecciones de jugadores y, por tanto, la calidad global del seguimiento de jugadores.

## **Abstract**

This work focuses on researching player tracking in soccer through computer vision techniques. The effectiveness of various object tracking methods is explored using the SoccerNet dataset, which provides a wide range of images and videos of soccer matches from different scenarios with detailed annotations. Through comprehensive analysis, the current state of object tracking methods is reviewed, from traditional methods to the latest deep neural network strategies. Results are evaluated using specific metrics, and errors and challenges are identified. Furthermore, an improvement in results is achieved through the use of the StrongSORT method, which focuses on optimizing object detection and association. Finally, a new model is obtained by training a detector using automatic detections obtained from techniques such as Background Subtraction and Human Pose Estimation, aiming to improve the accuracy and reliability of player detections and thus the overall quality of player tracking.



# Índex

<b>Introducció</b>	<b>1</b>
1.1 Motivació	1
1.2 Objectius	1
1.3 Assoliments	2
1.4 Organització de la memòria	2
<b>Estat de l'art</b>	<b>3</b>
2.1 Multiple Object Tracking (MOT)	3
2.2 Classificació dels mètodes	5
2.3 JDE	5
2.4 FairMOT	6
2.5 CenterTrack	7
2.6 ByteTrack	8
2.7 DeepSORT	8
2.8 BoT-SORT	9
2.9 OC-SORT	10
2.10 Configuració dels mètodes de seguiment	11
<b>Anàlisi, evaluació i resultats</b>	<b>12</b>
3.1 Conjunt de dades	12
3.2 Mètriques pel seguiment d'objectes	13
3.2 Resultats	14
3.3 Errors i problemes en els resultats	16
<b>Milfores, implementacions i resultats</b>	<b>19</b>
4.1 StrongSORT	19
4.2 Entrenament d'un detector	22
<b>Conclusions i treball futur</b>	<b>30</b>
<b>Apèndix</b>	<b>35</b>

## Llista de figures

Figura 2.1: Comparació entre detectors d'objectes d'una sola etapa (b) i de dues etapes (a). Figura de [15]	4
Figura 2.2: Il·lustració de (a) l'arquitectura de la xarxa i (b) el cap de predicció. Figura de [39]	6
Figura 2.3: Visió general de FairMOT. Figura de [47]	7
Figura 2.4: Il·lustració del mètode CenterTrack. Figura de [49]	7
Figura 2.5: Exemple d'associació de cada caixa de detecció utilitzant ByteTrack. Figura de [46]	8
Figura 2.6: Esquema del mètode OC-SORT. Figura de [7]	11
Figura 3.1: Distribució de les classes d'acció. Figura de [11]	12
Figura 3.2: Comparació de SoccerNet-Tracking amb altres conjunts de dades de seguiment. Figura de [11]	13
Figura 3.3: Representació TPA, FNA i FPA. Figura de [25]	14
Figura 3.4: Detecció dels jugadors en escenaris borrosos.	16
Figura 3.5: Oclusions quan s'acumulen jugadors.	17
Figura 3.6: Detecció del públic.	17
Figura 3.7: Detecció de la pilota en diferents fotogrames.	17
Figura 3.8: Escenari on els jugadors abandonen l'escena durant un període de temps.	18
Figura 3.9: Escenari on els jugadors s'acumulen i es produeixen oclusions.	18
Figura 3.10: Diferents identificadors per a la pilota.	18
Figura 4.1: Comparació de marc i rendiment entre DeepSORT i StrongSORT. Figura de [17]	19
Figura 4.2: Estructura del model AFLink de dues branques. Figura de [17]	20
Figura 4.3: Il·lustració de la diferència entre la interpolació lineal i GSI. Figura de [17]	20
Figura 4.4: Comparació visual entre DeepSORT i StrongSORT.	22
Figura 4.5: Comparació de l'associació entre DeepSORT i StrongSORT.	22
Figura 4.6: Resultats del model pix2pix.	23
Figura 4.7: Comparació de la màscara de camp abans i després de l'actualització.	24
Figura 4.8: Imatges binàries després d'aplicar Background Subtraction.	24
Figura 4.9: Deteccions dels jugadors aplicant Background Subtraction.	25
Figura 4.10: Estructura mètode Realtme Multi-person 2D Pose Estimation. Figura de [8]	25
Figura 4.11: Deteccions dels jugadors aplicant Human Pose Estimator.	26
Figura 4.12: Deteccions de jugadors combinant els dos mètodes.	26
Figura 4.13: Escenaris de millora del mètode proposat.	28
Figura 4.14: Escenaris on el mètode proposat no millora els resultats.	29

## Llista de taules

Taula 2: Configuració dels mètodes.	11
Taula 3: Resultats HOTA, DetA i AssA en el conjunt de test.	15
Taula 4.1: Comparativa entre els resultats de DeepSORT, StrongSORT i StrongSORT++.	21
Taula 4.2: Comparativa II dels resultats entre DeepSORT, StrongSORT i StrongSORT++.	21
Taula 4.3: Taula comparativa dels mètodes BoT-SORT i ByteTrack.	27
Taula 4.4: Taula II comparativa dels mètodes BoT-SORT i ByteTrack.	27

## Nomenclatures

AFLink Appearance-Free Link

AssA Association Accuracy

CMC Camera Motion Compensation

CNN Convolution Neural Network

CVAT Computer Vision Annotation Tool

DetA Detection Accuracy

DFL Distribution Focal Loss

DLA Deep Layer Aggregation

ECC Envelope Correlation Coefficient

EMA Exponential Moving Average

FCOS Fully Convolutional One-Stage

FN False Negative

FNA False Negative Accuracy

FP False Positive

FPA False Positive Accuracy

FPN Feature Pyramid Network

GMC Global Motion Compensation

GSI Gaussian-Smoothed Interpolation

IoU Intersection over Union

JDE Joint Detection and Embedding

HOTA Higher Order Tracking Accuracy

KF Kalman Filter

LocA Location Accuracy

ML Mostly Lost

MLP MultiLayer Perceptron

MOT Multiple Object Tracking

MT Mostly Tracked  
NMS Non-Maximum Suppression  
NSA Noise Scale Adaptive  
OCM Observation Centric Momentum  
OCR Observation Centric Recovery  
ORU Observation Centric Re-Update  
PAN Path Aggregation Network  
PT Partially Tracked  
RANSAC RANdom SAmple Consensus  
R-CNN Region-based Convolutional Neural Network  
Re-ID Re-identification  
RPN Region Proposal Network  
ROI Region of Interest  
SDE Separate Detection and Embedding  
SORT Simple Online and Realtime Tracking  
SOT Simple Object Tracking  
TAL Task Alignment Learning  
TN True Negative  
TP True Positive  
TPA True Positive Accuracy  
VFL Varifocal Loss  
YOLO You Only Look Once

# Capítol 1

## Introducció

L'esport s'està convertint cada cop més en un camp impulsat per les dades, ja que actualment hi ha una gran quantitat d'informació disponible sobre la condició física dels atletes, les seves actuacions tècniques en els partits, entre altres aspectes, sovint complementada amb vídeos dels esdeveniments esportius o partits complets. El volum, la complexitat i la riquesa d'aquestes fonts de dades han convertit l'aprenentatge automàtic en una eina d'anàlisi cada vegada més important. Com a resultat, l'aprenentatge automàtic s'utilitza per prendre decisions en l'àmbit de l'esport professional. D'una banda, s'utilitza per extreure coneixements raonables de les grans quantitats de dades relacionades amb el rendiment dels jugadors, els enfocaments tàctics i l'estat físic dels esportistes. D'altra banda, s'utilitza per automatitzar algunes tasques orientades a l'anàlisi de vídeos de partits. Fins ara, l'anàlisi de vídeos esportius es feia manualment, la qual cosa suposava un gran cost i temps. Per tant, la seva automatització aporta avantatges considerables [13].

### 1.1 Motivació

El seguiment d'objectes, conegut com a *video tracking*, és una tasca fonamental en la visió per computador, que té com a objectiu estimar les caixes delimitadores i les identitats d'un o més objectes en seqüències de vídeo.

Els avenços en qualitat i resolució d'imatges han fet possible l'ús del *video tracking* en nombroses aplicacions, inclos l'àmbit esportiu. No obstant això, el fet que estigui en constant investigació no vol dir que sigui una tasca senzilla de dur a terme. Seguir un objecte al llarg d'una seqüència pot ser molt complicat a causa de factors externs com canvis d'il·luminació, la presència d'altres objectes amb característiques semblants o fins i tot oclusions que fan que l'objecte quedi totalment o parcialment ocluit.

Amb l'objectiu de fer front als desafiaments que es presenten, s'han desenvolupat nous algoritmes que permeten seguir els objectes de manera eficient, independentment dels factors externs que puguin aparèixer. Aquest treball està motivat per la necessitat de trobar solucions a aquests problemes en l'àmbit del futbol mitjançant l'ús d'algoritmes i implementacions pròpies.

### 1.2 Objectius

Els objectius d'aquest treball són els següents:

- Investigar i analitzar diversos mètodes de visió per ordinador centrats en el seguiment de múltiples objectes per comprendre les seves aplicacions i funcionalitats.
- Avaluar aquests mètodes en un conjunt de dades específic relacionat amb el futbol, que inclogui diferents escenes que reflecteixin les condicions reals del joc i permeti identificar objectes com jugadors, àrbitres i la pilota.
- Avaluar l'eficàcia i precisió d'aquests mètodes per determinar la qualitat del seguiment d'objectes en el context del futbol.
- Analitzar els resultats obtinguts per extreure conclusions que ajudin a comprendre l'eficàcia dels mètodes en l'àmbit del futbol, identificant els seus punts forts i punts febles.
- Intentar millorar els resultats i abordar els punts febles dels mètodes creant un mètode personalitzat o millorant un existent. A més, posar a prova aquesta implementació i comparar-la amb els resultats dels mètodes anteriors.

## 1.3 Assoliments

En aquest treball, s'han aconseguit els següents objectius proposats:

- **Avaluació i anàlisi d'algoritmes:** S'han avaluat un total de 8 algoritmes de seguiment de múltiples objectes, identificant els seus punts forts i febles. Aquesta avaluació s'ha dut a terme utilitzant un conjunt de dades relacionat amb el futbol, i ha permès determinar quin algoritme és el més eficaç. A més, s'ha realitzat una anàlisi tant quantitativa com qualitativa per identificar possibles errors i millorar l'eficàcia d'aquests.
- **Investigació de mètodes externs:** A més dels algoritmes inclosos en la plataforma PaddleDetection, s'han explorat mètodes externs per millorar els resultats de DeepSORT, que inicialment era el més eficaç.
- **Desenvolupament de deteccions pròpies:** Per abordar les limitacions identificades, com ara la detecció del públic i la falta de deteccions en escenaris borrosos, s'han creat deteccions pròpies utilitzant la tècnica de subtracció de fons i l'estimació de la postura humana. Tot i que l'entrenament no ha estat tan efectiu com s'esperava, s'han solucionat amb èxit les dues limitacions.

## 1.4 Organització de la memòria

El contingut de la memòria es divideix en els següents capítols:

- **Capítol 2:** s'introduceix el concepte de seguiment de múltiples objectes (MOT) i s'explora l'estat de l'art amb diferents mètodes i algoritmes, destacant les tècniques més avançades i eficients.
- **Capítol 3:** s'explica el conjunt de dades utilitzat per a l'anàlisi dels mètodes, s'avaluen aquests amb diverses mètriques i es presenten els resultats i els errors detectats.
- **Capítol 4:** es milloren els resultats anteriors mitjançant l'ús de nous mètodes de seguiment i l'entrenament d'un detector.
- **Capítol 5:** es presenten les conclusions i les propostes de futur.

## Capítol 2

# Estat de l'art

En aquest capítol, es defineix el concepte de Multiple Object Tracking i es realitza una classificació dels diferents mètodes utilitzats. A més, s'expliquen cadascun dels mètodes des de la secció 2.3 fins a la secció 2.9, i finalment, es presenta la configuració detallada de cadascun d'ells.

### 2.1 Multiple Object Tracking (MOT)

MOT juga un paper important en la visió per computador. Les tasques de MOT es divideixen principalment en localitzar múltiples objectes, assignar diferents identitats i generar les seves trajectòries individuals a partir d'una seqüència d'imatges o un vídeo d'entrada. Els objectes a seguir poden ser vianants al carrer, vehicles a la carretera, esportistes a la pista, etc.

En comparació amb Simple Object Tracking (SOT), que se centra principalment a dissenyar models d'aparença o models de moviment per fer front a problemes com canvis d'escala, rotacions fora del pla i variacions d'il·luminació, MOT requereix addicionalment resoldre dues tasques: determinar el nombre d'objectes, que varia típicament amb el temps, i mantenir les seves identitats. En MOT apareixen problemes com les oclusions freqüents, la similitud entre objectes i les interaccions entre múltiples objectes. Per abordar aquests problemes, s'han plantejat diverses solucions en els últims anys en forma d'algoritmes des de mètodes clàssics fins a les innovadores aplicacions de xarxes neuronals profundes [15].

L'enfocament general per realitzar el seguiment múltiple d'objectes consta de dues etapes: la detecció d'objectes, que implica detectar tots els objectes mitjançant caixes delimitadores, i l'associació d'instàncies, que consisteix a associar els objectes detectats en els diferents fotogrames del vídeo. Alguns algoritmes també poden incloure una etapa de re-identificació d'objectes.

#### Detecció d'objectes

La detecció d'objectes és un aspecte molt important en el seguiment múltiple d'objectes, ja que té un impacte significatiu en el rendiment del seguiment. Aquest procés implica identificar la ubicació i la categoria de la classe de l'objecte en cada fotograma del vídeo.

A mesura que avança la tecnologia de l'aprenentatge profund, els detectors basats en Convolution Neural Network (CNN) s'han convertit en la principal opció per a la detecció d'objectes en tasques de MOT. Les CNN són un tipus especial de xarxes neuronals, formades per neurones interconnectades per capes. Cada neurona té pesos i biaixos d'aprenentatge, rep múltiples entrades, realitza una suma ponderada, aplica una funció d'activació, i produeix una resposta de sortida. Aquestes xarxes són efectives quan reconeixen patrons com vores, colors, formes i textures [3].

Hi ha dos tipus diferents de detectors d'objectes [2]:

1. Detectors d'objectes d'una sola etapa: classifiquen i fan la regressió directament de les caixes ancorades candidates sense identificar les regions d'interès (RoI). Alguns exemples són la família YOLO [37] i CenterNet [50].
2. Detectors d'objectes de dues etapes: divideixen la tasca de detecció en dues etapes. Primerament, utilitzen una Region Proposal Network (RPN) per determinar les RoI. Aquestes RoI són caixes delimitadores que potencialment contenen un objecte. Després, classifiquen i

fan la regressió únicament de les RoI per generar els resultats finals de detecció. Alguns exemples són R-CNN [21] i Faster-RCNN [34].

Com es mostra en la Figura 2.1, es detallen més els dos tipus de detectors i les seves diferències. En general, els detectors d'objectes d'una sola etapa tenen una velocitat d'inferència més alta, mentre que els detectors d'objectes de dues etapes tenen una millor precisió.

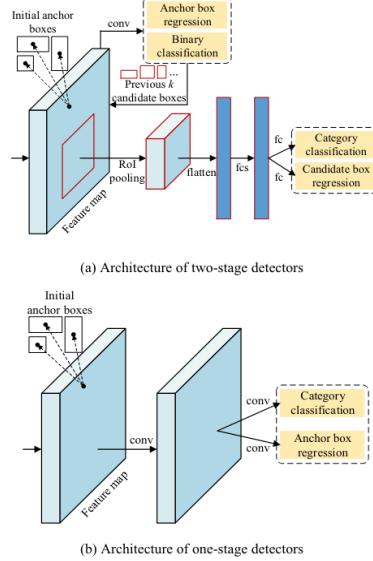


Figura 2.1: Comparació entre detectors d'objectes d'una sola etapa (b) i de dues etapes (a). Figura de [15]

## Associació d'instàncies

L'associació d'instàncies implica assignar de manera coherent una identificació als objectes de manera que els mateixos objectes tinguin les mateixes identificacions durant tota la seqüència del vídeo. Aquest procés utilitza indicis de moviment, aparença i temporals, o una combinació d'aquests, per generar trajectòries individuals per a cada objecte.

Hi ha dos enfocaments principals [4]:

1. Utilitzar indicis de moviment per assignar les caixes de detecció als *tracklets*. Normalment, això implica aplicar un Kalman Filter (KF) [40] per predir les caixes delimitadores dels *tracklets* en el fotograma actual i comparar les caixes delimitadores detectades amb les caixes delimitadores predites basant-se en la mètrica de similitud Intersection over Union (IoU). KF és un mètode probabilístic bayesià que proporciona una solució computacional eficient per estimar estats passats, presents i fins i tot futurs, estima la ubicació esperada de l'objecte en el següent fotograma. La similitud de moviment i ubicació és precisa en l'associació a curt termini, en casos on hi ha oclusions breus. Algunes implementacions són SORT i ByteTrack.
2. Utilitzar informació de característiques d'aparença dins de les caixes delimitadores per coincidir les instàncies a través dels fotogrames. Aquest procés sovint requereix l'ús d'una xarxa neuronal addicional per extreure característiques, les quals s'utilitzaran per comparar el contingut de les caixes detectades amb el conjunt de *tracklets* utilitzant una mètrica de distància (per exemple, la similitud del cosinus de les característiques de re-identificació). La similitud d'aparença és precisa en l'associació a llarg termini, en casos on un objecte pot ser reidentificat després d'estar ocult durant un període llarg de temps o quan hi ha canvis significatius entre fotogrames, ja que l'algoritme de coincidència no depèn de la ubicació exacta de les caixes delimitadores. Alguns exemples d'algoritmes són DeepSORT, JDE i FairMOT.

## Re-identificació d'objectes

La re-identificació d'objectes consisteix a predir la identitat d'un objecte específic al llarg d'una seqüència de vídeo. En aquest treball, un exemple seria la re-identificació d'un jugador que surt del camp de visió de la càmera durant un cert temps o en situacions d'oclosió causades per una acumulació de jugadors.

Aquesta re-identificació es basa en l'extracció de característiques rellevants. Aquesta tasca ha suposat un repte constant en el camp de la recerca. En els primers estudis, es donava èmfasi a les representacions de característiques dissenyades manualment, però actualment, amb els avenços de l'aprenentatge automàtic, s'utilitzen CNN per extreure característiques de nivell superior [15].

## 2.2 Classificació dels mètodes

En primer lloc, els mètodes que s'introduiran es poden agrupar en dues categories: Joint Detection and Embedding (JDE) [39] i Separate Detection and Embedding (SDE).

### JDE

En els models JDE, unifiquen les tasques de detecció d'objectes i generació d'*embeddings* d'aparença corresponents de les caixes detectades en un sol model. Aquests mètodes busquen aprendre de forma conjunta aquestes dues tasques mitjançant una xarxa neuronal que opera en una sola passada endavant.

Tot i que aquests són eficients gràcies a la seva naturalesa col·laborativa, es veuen afectats per dues limitacions. En primer lloc, les diferents característiques necessàries per a les tasques de detecció i *embedding* poden causar conflictes, dificultant l'optimització del model. En segon lloc, un altre obstacle és la desalineació espacial per als *trackers* [15]. Els mètodes JDE inclouen: JDE, FairMOT [47] i CenterTrack [49].

### SDE

En els models SDE, les tasques de detecció i associació d'objectes es realitzen en dues etapes separades. En la primera etapa, s'identifiquen els objectes en cada fotograma mitjançant un detector de localització inicial. En la segona etapa, les deteccions es connecten i s'assignen a trajectòries existents mitjançant un model d'*embedding* (a vegades s'inclou un mòdul de re-identificació) per a l'associació a llarg termini.

Malgrat la seva precisió de seguiment, aquests sovint requereixen una gran quantitat de recursos informàtics. Això és a causa de la necessitat d'extreure informació d'*embedding* en cada caixa candidata mitjançant una xarxa Re-ID independent, la qual cosa resulta en una velocitat de seguiment lenta. Els mètodes SDE inclouen ByteTrack [46], DeepSORT [42], BoT-SORT [1] i OC-SORT [7].

## 2.3 JDE

El mètode JDE [39] té com a objectiu generar simultàniament la localització i els *embeddings* d'aparença dels objectes en un únic model d'una sola passada endavant.

Per millorar la detecció, JDE utilitza l'arquitectura Feature Pyramid Network (FPN) [29], que fa prediccions a múltiples escales. La xarxa passa per diverses capes, generant mapes de característiques en tres escales. Després, el mapa de característiques més petit, que conté les característiques semànticament més fortes, és augmentat i fusionat amb el mapa de la segona escala més petita mitjançant una connexió de salt. Aquest procés es repeteix per a totes les altres escales. Finalment, s'afegeixen els caps de predicció als mapes de característiques fusionats a les tres escales, com es pot veure en la imatge (a) de la Figura 2.2. Un cap de predicció consta de diverses capes convolucionals

apilades que emeten un mapa dens de prediccions. Aquest mapa es divideix en tres parts: classificació de caixes, regressió de caixes i mapa d'*embedding* dens, com es pot observar a la imatge (b) de la Figura 2.2.

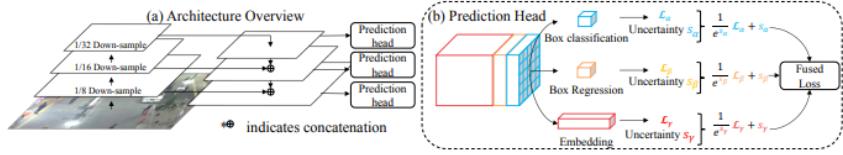


Figura 2.2: Il·lustració de (a) l'arquitectura de la xarxa i (b) el cap de predicció. Figura de [39]

El procés d'entrenament es modela com un problema d'aprenentatge de múltiples tasques que inclou classificació d'àncores, regressió de caixes i aprenentatge d'*embeddings*. Per tal d'equilibrar la importància de cada tasca, s'utilitza la incertesa dependent de la tasca [26] per ponderar dinàmicament les pèrdues heterogènies.

JDE utilitza un mètode simple i ràpid. Cada *tracklet* es descriu amb un estat d'aparença  $e_i$  i un estat de moviment  $m_i$ . A partir d'un fotograma d'entrada, es calculen les matrius d'afinitat de moviment  $A_m$  i d'aparença  $A_e$  entre totes les observacions i els *tracklets*. L'afinitat d'aparença es calcula utilitzant la similitud del cosinus, mentre que l'afinitat de moviment es calcula utilitzant la distància de Mahalanobis. L'assignació lineal es resol mitjançant l'algoritme Hongarès [27] amb la matriu de costos  $C$  (Eq. 2.1).

$$C = \lambda A_e + (1 - \lambda) A_m \quad (2.1)$$

L'estat de moviment de tots els *tracklets* associats s'actualitza amb el KF, i l'estat d'aparença s'actualitza mitjançant l'Eq. 2.2, on  $f_i^t$  és l'*embedding* d'aparença de l'observació coincident actual i  $\alpha = 0,9$  és un terme de moment.

$$e_i^t = \alpha e_i^{t-1} + (1 - \alpha) f_i^t \quad (2.2)$$

Finalment, les observacions no assignades a cap *tracklet* s'inicialitzen com a nous *tracklets* si apareixen en dos fotogrames consecutius. Un *tracklet* es considera finalitzat si no s'actualitza en els 30 fotogrames següents.

## 2.4 FairMOT

FairMOT [47] utilitza un enfocament equitatiu per a les tasques de detecció i extracció de característiques de Re-ID d'objectes. Per a alinear les tasques, s'utilitza com a *backbone* ResNet-34 i s'aplica una versió millorada de DLA [44] anomenada DLA-34, que té més connexions de salt entre característiques de baix nivell i alt nivell semblants a FPN.

La branca de detecció està construïda sobre CenterNet, afegint tres caps paral·lels a DLA-34 per estimar mapes de calor, desplaçaments del centre de l'objecte i mides de les caixes delimitadores, respectivament. La branca de Re-ID genera característiques per distingir objectes utilitzant una capa de convolució [28].

Les dues branques s'entrenen conjuntament, com es mostra la Figura 2.3, amb pèrdues específiques per a cada tasca. S'utilitza una pèrdua d'incertesa per equilibrar-les automàticament. La xarxa rep una imatge com a entrada i utilitza el mètode Non-Maximum Suppression (NMS) basat en els pics del mapa de calor per extreure els punts més importants. Les caixes delimitadores es calculen a partir de les desviacions i mides de les caixes estimades i els *embeddings* d'identitat s'estreuen en els centres dels objectes estimats.

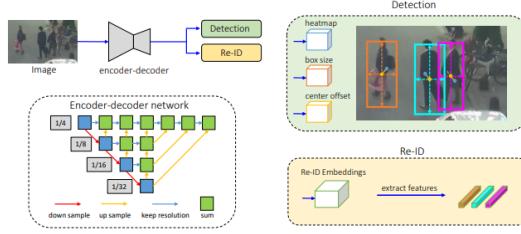


Figura 2.3: Visió general de FairMOT. Figura de [47]

Per a l'associació, es basa en MOTDT [10] i s'utilitza un mètode d'associació de dades en línia jeràrquic. Primerament, s'inicialitzen múltiples *tracklets* basats en les caixes detectades al primer fotograma. En els fotogrames següents, es vinculen les deteccions als *tracklets* existents utilitzant KF i les característiques de Re-ID. S'utilitza el KF per predir les ubicacions dels *tracklets* al fotograma següent i es calcula la distància de Mahalanobis  $D_m$  entre les caixes predites i detectades seguint DeepSORT (Eq. 2.4). Seguidament, es fusiona  $D_m$  amb la distància de cosinus  $D_r$  a partir de les característiques de Re-ID (Eq. 2.3). En aquesta equació,  $\lambda = 0,98$  és un paràmetre de ponderació per determinar quina de les dues mètriques té més pes.

$$D = \lambda D_r + (1 - \lambda) D_m \quad (2.3)$$

En la primera etapa, s'utilitza l'algoritme Hongarès per resoldre l'associació, mentre que en la segona etapa s'intenta associar les deteccions sense coincidència amb els *tracklets* mitjançant la superposició de les caixes delimitadores amb un llindar  $\tau=0,5$ . Les deteccions sense coincidència es consideren noves trajectòries i els *tracklets* sense coincidència es conserven durant 30 fotogrames en cas que reapareguin en el futur.

## 2.5 CenterTrack

CenterTrack [49] és un *tracker* senzill i ràpid que es basa en el detector CenterNet. A diferència d'altres detectors, CenterNet prediu el centre de les caixes delimitadores en lloc de la seva posició. Cada objecte es caracteritza per un únic punt al centre de la seva caixa delimitadora, el qual se segueix al llarg del temps. CenterNet ofereix informació clau com ara ubicacions, mides i confiança. Per millorar la coherència temporal, CenterTrack incorpora dos fotogrames addicionals com a entrada a la xarxa de detecció, a més de deteccions prèvies addicionals representades amb un mapa de calor d'un sol canal basat en punts. Només es representen els objectes amb una puntuació superior a un llindar per reduir falsos positius. L'arquitectura de CenterTrack és similar a la de CenterNet, amb quatre canals d'entrada addicionals, però no manté una connexió entre les deteccions al llarg del temps.

La Figura 2.4 il·lustra la part de detecció de CenterTrack. La xarxa pren el fotograma actual, el fotograma anterior i un mapa de calor renderitzat a partir dels centres dels objectes rastrejats com a entrades. Després, produeix un mapa de calor de detecció  $\hat{Y}$  al centre per al fotograma actual, el mapa de mida de la caixa delimitadora  $\hat{S}$  i un mapa de desplaçament  $\hat{O}$ . Finalment, s'estreuen les mides dels objectes i els desplaçaments dels pics del mapa de calor.



Figura 2.4: Il·lustració del mètode CenterTrack. Figura de [49]

Pel que fa a l'associació, utilitza un algoritme senzill per associar les deteccions al llarg del temps, si un objecte surt del fotograma o queda ocult i torna a aparèixer, se li assigna una nova identitat.

CenterTrack prediu un desplaçament 2D com a dos canals de sortida addicionals. Aquest desplaçament es calcula com la diferència entre la ubicació de l'objecte en el fotograma actual i en el fotograma anterior. Per a cada objecte detecció en el punt  $p$ , s'associa amb la detecció prèvia no emparellada més propera a la posició  $p - D$ , on  $D$  és el desplaçament, en ordre descendent de confiança  $w$ . Si no hi ha cap detecció prèvia sense emparellar dins d'un radi  $K$ , es genera una nova trajectòria. El valor de  $K$  és la mitjana geomètrica de l'amplada i l'altura de la caixa delimitadora predita per a cada trajectòria. Per a més informació sobre el mètode CenterTrack veure Apèndix-2.

## 2.6 ByteTrack

ByteTrack [46] és un mètode d'associació que rastreja tots els objectes associant totes les caixes de detecció en lloc de només les que tenen una puntuació IoU alta. Utilitza el detector YOLOX [20] amb YOLOX-X com a *backbone* per a la detecció.

El mètode implementa el mètode d'associació BYTE, que consta de dues etapes. En la primera etapa, les caixes de detecció amb puntuació alta s'associen a totes les trajectòries utilitzant la distància IoU o Re-ID entre les caixes de detecció amb puntuació alta i les caixes predites de totes les trajectòries. Després, s'aplica l'algoritme Hongarès [27] per resoldre l'associació, guardant les deteccions i les trajectòries no associades. En la segona etapa, les caixes de detecció amb puntuació baixa s'associen a les trajectòries no associades utilitzant només la distància IoU per recuperar algunes de les caixes de detecció que podrien perdre's en altres mètodes de seguiment. La sortida de cada fotograma són les caixes delimitadores i les identitats de les pistes en el fotograma actual. Per a més informació sobre el *tracker* BYTE veure Apèndix-2.

La Figura 2.5 il·lustra com s'associen les caixes de detecció. En les imatges (a) es mostren totes les caixes de detecció amb les seves puntuacions. En les imatges (b) es mostren els tres *tracklets* obtinguts a partir de l'associació entre les caixes de detecció amb puntuacions superiors a 0,5, com a resultat, s'observa com les caixes de detecció vermelles desapareixen en els fotogrames 2 i 3. Finalment, en les imatges (c) es mostren els *tracklets* obtinguts mitjançant una segona associació, on es pot veure com dues caixes de detecció de puntuació baixa es corresponen amb els *tracklets* (caixes de color vermell amb línies puntejades), recuperant així els objectes, mentre que les caixes que detectaven fons s'eliminen, ja que no s'han associat a cap *tracklet*.

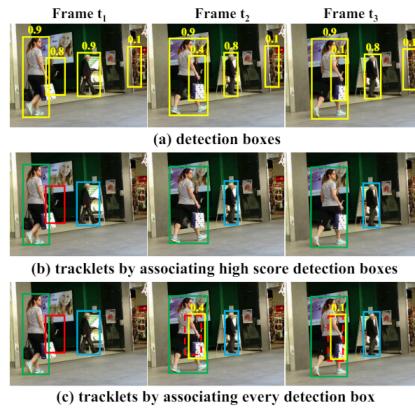


Figura 2.5: Exemple d'associació de cada caixa de detecció utilitzant ByteTrack. Figura de [46]

## 2.7 DeepSORT

DeepSORT [42] és una versió millorada del mètode SORT [5] aprofitant un model d'aparença mitjançant una xarxa CNN entrenada en un conjunt de dades de re-identificació de persones [48].

Aquest model pren una imatge d'una persona detectada com a entrada i la converteix en un vector d'*embedding* que representa característiques distintives per identificar persones.

El mètode utilitza una metodologia de seguiment amb una única hipòtesi, utilitzant el KF i l'associació de dades de fotograma a fotograma. Les trajectòries es gestionen incrementant un comptador des de l'última associació, i aquelles que superen un determinat límit es consideren que han abandonat l'escena i s'eliminen del conjunt de trajectòries. Les noves trajectòries s'inicien per a cada detecció sense associar, però s'eliminen si no s'associen amb èxit dins dels seus primers tres fotogrames.

Per resoldre l'associació entre els estats predictius del KF i les mesures que arriben, es construeix un problema d'assignació que es resol utilitzant l'algoritme hongarès. S'utilitzen dues mètriques: la distància (al quadrat) de Mahalanobis (Eq. 2.4), que té en compte la informació de moviment, i la distància de cosinus (Eq. 2.5), que considera la informació d'aparença obtinguda del model d'aparença.

$$d^{(1)}(i, j) = (d_j - y_i)^T S_i^{-1} (d_j - y_i) \quad (2.4)$$

En l'Eq. 2.5, es mesura la menor distància de cosinus entre la trajectòria  $i$  i la detecció  $j$  a l'espai d'aparença on  $r_j$  és el descriptor d'aparença extret d'una CNN preentrenada.

$$d^{(2)}(i, j) = \min\{1 - r_j^T r_k^{(i)} \mid r_k^{(i)} \in R_i\} \quad (2.5)$$

Les dues mètriques es combinen amb una suma ponderada per formar un cost d'associació controlat per l'hiperparàmetre  $\lambda$  (Eq. 2.6). En DeepSORT,  $\lambda=0$ , considerant només la informació d'aparença.

$$c_{i,j} = \lambda d^{(1)}(i, j) + (1 - \lambda)d^{(2)}(i, j) \quad (2.6)$$

En lloc de resoldre associacions de manera global, s'aplica una cascada d'emparellaments que resol una sèrie de subproblemes. L'algoritme està estructurat en una cascada que dona prioritat als objectes amb una edat menor, és a dir, els que s'han vist més recentment. En l'última etapa, s'aplica una associació basada en IoU per augmentar la robustesa del sistema davant de canvis d'aparença bruscos. Per a més informació sobre l'algoritme de cascada d'emparellament veure Apèndix-2.

## 2.8 BoT-SORT

BoT-SORT [1] és un mètode que introduceix millors específiques per abordar els reptes associats al moviment de la càmera i als canvis en la superposició entre les caixes delimitadores predites i detectades. A més, inclou BoT-SORT-ReID, una extensió de BoT-SORT que té un mòdul de re-identificació. Per a més informació sobre aquesta extensió veure Apèndix-2.

Per modelar el moviment de l'objecte, utilitza el KF adaptant el vector d'estat per estimar directament l'amplada i l'alçada de la caixa delimitadora per millorar l'ajust de l'amplada de la caixa delimitadora.

El seguiment basat en la detecció depèn en gran manera de la superposició entre les caixes delimitadores dels *tracklets* predictius i les detectades. En escenaris dinàmics o de càmera en moviment, la ubicació de les caixes pot canviar dràsticament, cosa que podria resultar en un augment dels canvis d'ID o falsos negatius. BoT-SORT utilitza la tècnica Global Motion Compensation (GMC) per obtenir el moviment de fons.

Primer s'extreuen els punts clau i el flux òptic de la imatge [6], i després es realitza la transformació de caixes delimitadores entre fotogrames consecutius utilitzant una matriu afí resolta mitjançant RANSAC [19], corregint els canvis significatius causats pel moviment de la càmera.

Pel que fa a l'associació, s'integren característiques d'aparença utilitzant ResNeSt50 [45] com a *backbone*. S'utilitza l'Exponential Moving Average (EMA) per actualitzar l'estat d'aparença del seguiment  $e_i^k$  (Eq. 2.7) on  $f_i^k$  és l'*embedding* d'aparença de la detecció coincident actual i  $\alpha = 0,9$  és un terme de moment.

$$e_i^k = \alpha e_i^{k-1} + (1 - \alpha) f_i^k \quad (2.7)$$

D'altra banda, per a la coincidència entre  $e_i^k$  i el nou vector d'*embedding* de detecció  $f_i^k$ , s'utilitza la mètrica de similitud de cosinus. En aquest cas, no s'utilitza la suma ponderada comuna entre el cost d'aparença i el cost de moviment per calcular la matriu de costos. S'ha desenvolupat un nou mètode per combinar la informació de moviment i d'aparença, utilitzant la matriu de distància IoU i la matriu de distància del cosinus. Primer, es rebutgen els candidats amb baixa similitud cosinus o llunyans en termes de la puntuació IoU calculant el cost d'aparença  $\hat{d}_{i,j}^{\cos}$  (Eq. 2.8). Després, s'utilitza el mínim en cada element entre les dues matrius com el valor final de la matriu de costos C (Eq. 2.9).

A l'Eq. 2.8,  $d_{i,j}^{\text{iou}}$  és la distància IoU entre la caixa delimitadora predita del *tracklet*  $i$  i la caixa delimitadora de la detecció  $j$ ,  $d_{i,j}^{\cos}$  és la distància cosinus entre el descriptor d'aparença del *tracklet*  $i$  i el nou descriptor de detecció  $j$ ,  $\theta_{\text{iou}} = 0,5$  és un llindar de proximitat per rebutjar parelles poc probables de *tracklets* i deteccions, i  $\theta_{\text{emb}} = 0,25$  és el llindar d'aparença, que s'utilitza per separar l'associació positiva dels estats d'aparença dels *tracklets* i els vectors d'*embeddings* de les deteccions.

$$\hat{d}_{i,j}^{\cos} \begin{cases} 0,5 \cdot d_{i,j}^{\cos}, (d_{i,j}^{\cos} < \theta_{\text{emb}}) \wedge (d_{i,j}^{\text{iou}} < \theta_{\text{iou}}) \\ 1, \text{altrament} \end{cases} \quad (2.8)$$

$$C_{i,j} = \min\{d_{i,j}^{\text{iou}}, \hat{d}_{i,j}^{\cos}\} \quad (2.9)$$

Finalment, es resol el problema d'assignació lineal de les deteccions d'alta confiança utilitzant l'algoritme Hongarès basant-se en la matriu de costos C.

## 2.9 OC-SORT

OC-SORT [7] és una extensió de l'algoritme SORT [5] que ha estat dissenyada per abordar les seves limitacions quan es perden objectes o quan hi ha moviments no lineals.

SORT té algunes limitacions, com ara la sensibilitat al soroll d'estat i l'ampliació de l'error temporal, que es produeix quan no hi ha observacions disponibles per actualitzar els paràmetres del KF. A més, segueix una perspectiva centrada en les estimacions en lloc de fer ús de les observacions.

L'objectiu d'OC-SORT és dissenyar un *tracker* centrat en les observacions en lloc de centrar en l'estimació, utilitzant tres solucions diferents.

La primera solució és ORU (Observation-Centric re-Update), que actualitza els paràmetres del KF basant-se en observacions d'una trajectòria virtual quan una pista es reactiva després d'un període sense observacions. Aquest mètode ajuda a evitar l'acumulació d'errors temporals durant el període sense rastrejar.

La segona solució OCM (Observation-Centric Momentum), que millora l'associació d'objectes en un interval de temps curt utilitzant les observacions per calcular la direcció de moviment i reduir el soroll en les estimacions d'estat. Ajusta el cost d'associació entre trajectòries existents i noves deteccions tenint en compte la distància IoU i la consistència entre les direccions de i) vincular dues observacions en una trajectòria existent i ii) vincular una observació històrica d'una trajectòria i una nova observació.

Finalment, la tercera solució ORU (Observation-Centric Recovery) realitza un segon intent d'associació entre les últimes observacions de les trajectòries no associades amb les observacions sense associar després de l'etapa d'associació habitual. Aquest mètode ajuda a recuperar objectes que podrien haver estat temporalment ocults.

La Figura 2.6 mostra les deteccions com a caixes vermelles, les trajectòries actives com a caixes taronges, les trajectòries sense associar com a caixes blaves i les estimacions del KF com a caixes discontinues. Durant el procés d'associació, s'utilitza OCM per afegir el cost de consistència de la velocitat. S'il·lustra un exemple on l'objecte es perd i es recupera gràcies a OCR, amb una re-actualització activa mitjançant ORU dels paràmetres del KF. Per a més informació sobre el pseudocodi d'OC-SORT veure Apèndix-2.

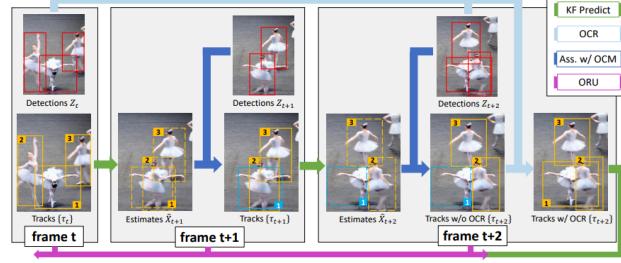


Figura 2.6: Esquema del mètode OC-SORT. Figura de [7]

## 2.10 Configuració dels mètodes de seguiment

Una vegada explicats els diferents mètodes de seguiment d'objectes utilitzats al llarg d'aquest treball, es detalla més explícitament en la Taula 2 quin detector, *tracker* o addicionalment model de Re-ID utilitza cada mètode. Aquests mètodes provenen de la plataforma PaddlePaddle [33]. És important destacar que cada mètode pot canviar de detector, model de Re-ID o fins i tot el *tracker* respecte als presentats en els diferents papers anteriorment. L'objectiu d'aquest treball és comprendre com aquests canvis afecten el rendiment general del mètode. Per a més informació sobre els detectors emprats veure Apèndix-1.

Taula 2: Configuració dels mètodes.

Mètode	Detector	Re-ID	Tracker
DeepSORT 1	YOLOv3	PCB+Pyramid	DeepSort
DeepSORT 2	YOLOv3	PPLCNet	DeepSort
DeepSORT 3	PP-YOLOE	ResNet101	DeepSort
FairMOT 1	CenterNet	FairMOT	JDE
FairMOT 2	CenterNet	FairMOT	BYTE
ByteTrack 1	YOLOX	Cap	BYTE
ByteTrack 2	YOLOv3	Cap	BYTE
ByteTrack 3	PP-YOLOE	Cap	BYTE
JDE	YOLOv3	JDE	JDE
BoT-SORT	PP-YOLOE	Cap	BoT-SORT
OC-SORT 1	YOLOX	Cap	OCSORT
OC-SORT 2	PP-YOLOE	Cap	OCSORT
CenterTrack	CenterNet	Cap	CenterTracker

## Capítol 3

# Anàlisi, avaluació i resultats

En aquest capítol, es descriu el conjunt de dades utilitzat per avaluar els diferents mètodes, les mètriques emprades, els resultats tant quantitatius com qualitatius, i finalment, s'identifiquen alguns errors i limitacions.

### 3.1 Conjunt de dades

En aquest treball, l'anàlisi i la implementació es basen únicament en el conjunt de dades proporcionat per SoccerNet [35]. SoccerNet no és només un conjunt de dades, sinó que també una comunitat activa que organitza reptes anuals, permetent als equips de recerca competir internacionalment.

S'ha utilitzat el conjunt de dades del repte de *Tracking*, que consisteix en 200 seqüències de vídeo de 30 segons de la Lliga Suïssa del 2019 dividides en quatre conjunts: prova, entrenament i dos de *challenges*. Aquestes seqüències són capturades des d'una única càmera, la qual cosa les fa adequades per a la tasca de seguiment, ja que no experimenten canvis de càmera ni repeticions. Han estat seleccionades específicament per representar un desafiament en el seguiment, on hi ha, per exemple, moviments ràpids de la càmera, grups de jugadors i la pilota de futbol movent-se a alta velocitat. Com es pot veure en la Figura 3.1, les diferents seqüències es divideixen en escenaris de futbol que es classifiquen en 17 classes.

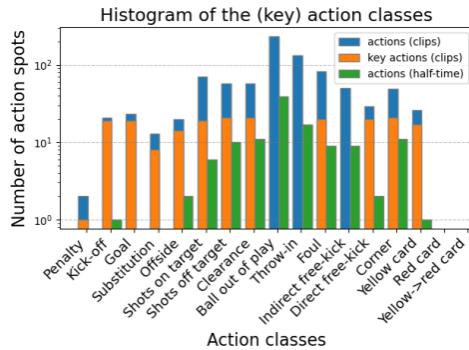


Figura 3.1: Distribució de les classes d'acció. Figura de [11]

A més de les seqüències de vídeo, el conjunt de dades conté anotacions de seguiment realitzades amb la plataforma SuperAnnotate [36]. Aquestes anotacions contenen cinc classes d'objectes d'interès: jugador, porter, àrbitre, pilota i altres. S'han anotat al voltant de 3,6 milions de caixes delimitadores que corresponen a 5.009 objectes únics, amb un 96% dels jugadors i porters identificats amb un número de la samarreta.

A diferència d'altres conjunts de dades, aquest conserva ID únics per als objectes que surten i tornen del camp de visió de la càmera. Aquest conjunt de dades destaca com un dels més grans disponibles públicament per al seguiment en el futbol i és el primer que inclou el seguiment múltiple d'objectes, com es pot veure en la Figura 3.2. La seva estructura és similar al format del conjunt de dades MOT20, simplificant així la comparació i l'avaluació de nous mètodes.

Dataset	Sequences	Frames	Tracklets	Bounding boxes	Domain	Task
MOT16 [52]	14	11,235	1,276	292,733	Pedestrians	MOT
MOT20 [19]	8	13,410	3,833	2,102,385	Urban (crowded)	MOT
KITTI-T [30]	50	10,870	977	65,213	Autonomous Driving	MOT
Head [68]	5	5,723	2,965	1,086,790	Pedestrian (heads)	MOT
TAO [16]	3	4,447,038	16,104	332,401	Generic	MOT
3DZef-T [55]	8	14,398	32	86,452	Fish	3D MOT
CTMC [2]	86	152,498	2,900	2,045,834	Cells	MOT
SSET [25]	80	12,000	80	12,000	Soccer	SOT
SN-Tracking (ours)	201	225,375	5,009	3,645,661	Soccer	MOT

Figura 3.2: Comparació de SoccerNet-Tracking amb altres conjunts de dades de seguiment. Figura de [11]

## 3.2 Mètriques pel seguiment d'objectes

Per avaluar els diferents mètodes, es fa servir la plataforma TrackEval [24] i s'utilitza la mètrica Higher Order Tracking Accuracy (HOTA) per avaluar el rendiment de MOT entre d'altres [25].

Aquesta mètrica és una combinació de tres puntuacions de IoU. Divideix l'avaluació del seguiment en tres subtasques (detecció, associació i localització) i calcula una puntuació per a cadascuna utilitzant l'Índex de Jaccard. Després, combina aquestes tres puntuacions per a cada subtasca per obtenir la puntuació final de HOTA.

### Localització

Mesura l'alignació espacial entre una detecció predita i una detecció real. Loc-IoU es calcula com la relació d'intersecció entre les dues deteccions i l'àrea total coberta per ambdues (unió). La mètrica Location Accuracy (LocA) (Eq. 3.1) es calcula prenent la mitjana de Loc-IoU per a totes les parelles de deteccions predites i reals que coincideixen a tot el conjunt de dades.

$$LocA = \frac{1}{|TP|} \sum_{c \in TP} Loc - IoU(c) \quad (3.1)$$

### Detecció

Mesura l'alignació entre el conjunt de totes les deteccions predites i el conjunt de totes les deteccions reals. Cal definir un llindar de localització per sobre del qual es considera que dues deteccions es creuen. S'aplica l'algoritme Hongarès per determinar una correspondència biunívoca entre les deteccions predites i reals.

True Positive (TP) representa les parelles coincidents de deteccions, False Positive (FP) són les deteccions predites que no coincideixen, i False Negative (FN) són les deteccions reals que no coincideixen.

La mètrica Detection Accuracy (DetA) (Eq. 3.2) es calcula utilitzant el Det-IoU, que és el nombre de TP dividit per la suma del nombre de TP, FN i FP sobre tot el conjunt de dades.

$$DetA = Det - IoU = \frac{|TP|}{|TP| + |FN| + |FP|} \quad (3.2)$$

### Associació

Mesura la capacitat d'un *tracker* per vincular les deteccions al llarg del temps en les mateixes identitats, donada la sèrie de vincles d'identitat reals en les seqüències reals. Es calcula comparant una detecció predita amb una detecció real que coincideixen, i mesurant l'alignació entre la seqüència completa de la detecció predita i la seqüència completa de la detecció real.

True Positive Accuracy (TPA) representa el nombre de coincidències de TP entre les dues seqüències, mentre que False Positive Accuracy (FPA) és qualsevol detecció restant a la seqüència predita i False Negative Accuracy (FNA) és qualsevol detecció restant a la seqüència real.

$$Ass - IoU = \frac{|TPA|}{|TPA| + |FNA| + |FPA|} \quad (3.3)$$

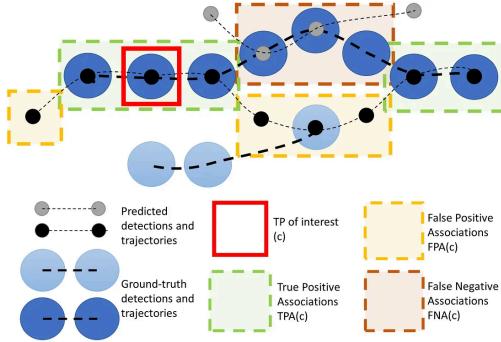


Figura 3.3: Representació TPA, FNA i FPA. Figura de [25]

En la Figura 3.3, el quadrat vermell indica la parella TP coincident entre una detecció predicta i una real, per la qual es vol trobar una puntuació d'associació. Per mesurar com s'alinea temporalment aquesta associació entre les deteccions, es busquen totes les deteccions en aquestes dues trames que coincideixen entre elles (TPA en verd) i totes les deteccions on no coincideixen (FPA en groc i FNA en marró).

La mètrica Association Accuracy (AssA) (Eq. 3.4) es calcula prenent la mitjana de l'Ass-IoU (Eq. 3.3) per a totes les parelles de deteccions coincidents predictes i reals a tot el conjunt de dades.

$$AssA = \frac{1}{|TP|} \sum_{c \in TP} Ass - IoU(c) = \frac{1}{|TP|} \sum_{c \in TP} \frac{|TPA(c)|}{|TPA(c)| + |FNA(c)| + |FPA(c)|} \quad (3.4)$$

Per últim, la mètrica HOTA (Eq. 3.5, 3.6) es calcula utilitzant les tres puntuacions de IoU definides anteriorment on  $\alpha$  és el llindar IoU.

$$HOTA_\alpha = \sqrt{DetA_\alpha \cdot AssA_\alpha} = \sqrt{\frac{\sum_{c \in TP} Ass - IoU(c)}{|TP_\alpha| + |FN_\alpha| + |FP_\alpha|}} \quad (3.5)$$

$$HOTA = \int_{0 < \alpha \leq 1} HOTA_\alpha \quad (3.6)$$

Es realitza una mitjana geomètrica en combinar la detecció i l'associació per garantir que ambdues tinguin el mateix pes en el resultat final i funcionin correctament.

## 3.2 Resultats

Per a avaluar els diversos algoritmes i obtenir resultats, s'ha utilitzat PaddleDetection [33], que és un kit de desenvolupament per a la detecció d'objectes d'extrem a extrem basat en PaddlePaddle.

Avaluant els resultats obtinguts pels diferents mètodes en el conjunt de dades de test, s'obtenen les puntuacions que es mostren a la Taula 3. Aquesta taula presenta una comparació quantitativa de les mètriques de rendiment per a cada mètode avaluat.

**DeepSORT** destaca especialment per la seva capacitat de detecció, ja que el nombre de deteccions s'aproxima el nombre real d'objectes presents, amb un 97% de deteccions correctes i obtenint la millor puntuació DetA. A més, és el mètode amb menor assignació d'IDs, gràcies a l'eficàcia de la xarxa CNN utilitzada per extreure les característiques d'aparença. No obstant això, cal tenir en compte que pot presentar dificultats en situacions amb moviments de càmera a causa dels canvis de forma dels jugadors.

Taula 3: Resultats HOTA, DetA i AssA en el conjunt de test.

Mètode	HOTA (↑)	DetA (↑)	AssA (↑)
DeepSORT 1	<b>69,552</b>	<b>82,628</b>	<b>58,668</b>
DeepSORT 3	69,414	82,6	58,455
DeepSORT 2	67,867	82,376	56,033
OC-SORT 2	45,529	50,614	41,085
FairMOT 1	43,909	46,318	41,773
FairMOT 2	43,746	45,776	41,949
OC-SORT 1	42,479	37,686	48,095
ByteTrack 1	39,780	33,271	47,677
BoT-SORT	39,178	37,996	40,511
JDE	36,379	40,594	32,739
ByteTrack 2	34,826	41,701	29,245
CenterTrack	31,75	36,032	28,33
ByteTrack 3	30,710	24,143	39,141

**OC-SORT** és molt influenciat pel detector utilitzat. Per exemple, OC-SORT 1, que utilitza el detector YOLOX, té una puntuació DetA significativament baixa (-12,9) en comparació amb OC-SORT 2, que utilitza el detector PP-YOLOE. Això es deu a la sobre detecció generada pel detector YOLOX, que resulta en un 50% de deteccions falses. En contrast, utilitzant OC-SORT 2, el nombre de deteccions és més baix que el nombre real de deteccions, assolint un 77% de deteccions correctes i només un 23% de deteccions falses. La puntuació AssA és inferior a causa de l'assignació d'un nombre d'ID massa elevat, probablement perquè aquest model no utilitza informació d'aparença i només es basa en informació de moviment i en l'alt rendiment dels detectors.

La principal diferència entre FairMOT 1 i FairMOT 2 és que el primer utilitza el *tracker* original de **FairMOT** mentre que el segon aplica el *tracker* BYTE. Amb el *tracker* BYTE s'obtenen més deteccions, ja que té en compte totes les caixes de detecció. Això provoca un lleuger augment de deteccions falses, la qual cosa fa que la puntuació DetA del segon sigui més baixa (-0,5). No obstant això, la disminució del nombre d'IDs fa que la puntuació AssA augmenti (+0,17). En aquest mètode, les tasques de detecció i Re-ID s'entrenen conjuntament per assegurar la seva equitat, la qual cosa es reflecteix en els resultats, ja que les puntuacions DetA i AssA són similars.

**ByteTrack** és un mètode del tipus SDE que no utilitza informació d'aparença, el que significa que el detector utilitzat té un gran impacte en els resultats. ByteTrack 2 és el més eficient en la detecció, ja que l'ús del detector YOLOv3 produeix menys deteccions, amb un 68% d'elles correctes. D'altra banda, ByteTrack 1 i ByteTrack 3, que utilitzen YOLOX i PP-YOLOE com a detectors respectivament, generen més deteccions, però amb un 56% i 68% de deteccions falses, afectant negativament la puntuació DetA (-8,4 i -17,5). Quant a la puntuació AssA, ByteTrack 2 obté la puntuació més baixa, probablement perquè aquest mètode només es basa en la puntuació IoU i no té en compte els canvis d'aparença dels jugadors quan hi ha oclusions. Per tant, el model interpretarà l'objecte com una nova instància i iniciarà un nou seguiment.

**BoT-SORT** utilitza el detector PP-YOLOE, que com s'ha vist en altres mètodes, genera una gran quantitat de deteccions. Com a resultat, hi ha un gran nombre de deteccions falses (48%), la qual cosa afecta negativament la puntuació DetA. A més a més, no s'utilitza el model Re-ID, la qual cosa provoca un nombre elevat d'IDs i empitjora l'associació.

**JDE** utilitza el detector YOLOv3, que produeix menys deteccions que altres mètodes, però ofereix una major precisió, reduint els falsos positius al 27%. Específicament, el 73% de les deteccions són correctes. Té una puntuació AssA inferior (-8,3) en comparació amb FairMOT. Aquest marc de detecció utilitza ancoratges, el que provoca alguns problemes en l'associació (múltiples caixes delimitadores per un sol jugador). En canvi, FairMOT utilitza CenterNet, un marc de detecció sense ancoratges basat en els punts dels objectes.

**CenterTrack** produeix una puntuació AssA més baixa, ja que només considera la situació en què els jugadors no es perdren al llarg del vídeo. Quan un jugador desapareix i torna a aparèixer, el mètode crea una nova trajectòria i li assigna un nou ID. Això provoca un augment significatiu en el nombre d'IDs assignats (+15,8%), el qual afecta negativament l'associació entre les deteccions i les trajectòries existents.

Una anàlisi més detallada dels resultats presentats en la Taula 3 es pot trobar en l'Apèndix-1 i l'Apèndix-2.

### 3.3 Errors i problemes en els resultats

Tot i els nombrosos avenços aconseguits en els darrers anys i la incorporació de nous algoritmes implementats, s'han detectat alguns errors que afecten els resultats obtinguts a l'apartat 3.2.

#### Detecció d'objectes

Durant el procés de detecció d'objectes, hi ha diversos motius pels quals un objecte pot ser no detectat o ser detectat de manera incorrecta.

En analitzar els resultats, es pot observar que el cas més freqüent és quan els jugadors apareixen borrosos a causa del moviment de la càmera, com es mostra a la Figura 3.4. En aquesta figura, es presenten tres imatges obtingudes amb diferents mètodes, i es pot veure que només uns pocs jugadors han estat identificats. Fins i tot en la imatge (c) del mètode BoT-SORT, que considera aquests escenaris utilitzant GMC, no s'aconsegueixen detectar tots els jugadors.

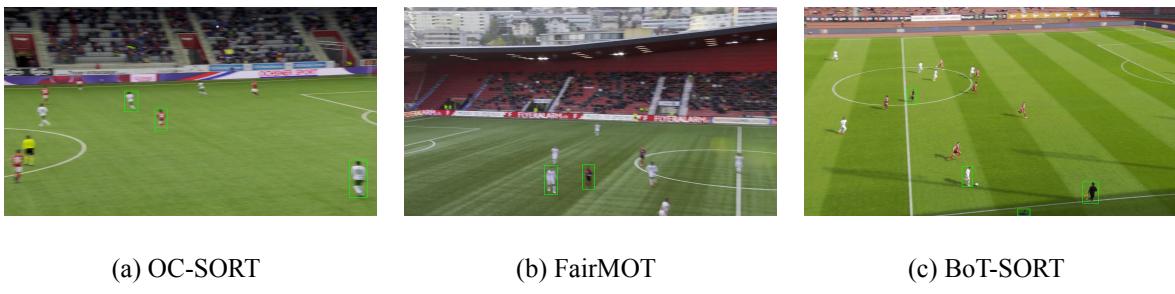


Figura 3.4: Detecció dels jugadors en escenaris borrosos.

Un altre problema comú és la detecció dels jugadors que es troben més lluny de la càmera en el camp i que, per tant, apareixen més petits. Això fa que sigui més difícil la seva detecció.

A més, les oclusions són un altre desafiant que es produeix quan els jugadors s'acumulen, especialment en situacions com els còrners, tal com es mostra a la Figura 3.5. En aquestes circumstàncies, les deteccions no s'ajusten bé, i de vegades, una única caixa delimitadora detecta més d'un jugador, mentre que en altres casos es generen més caixes delimitadores que jugadors.



Figura 3.5: Oclusions quan s'acumulen jugadors.

Alguns detectors, com YOLOX i PP-YOLOE [43], generen moltes deteccions gràcies a la seva capacitat d'extreure característiques molt detallades. Com a resultat, també es detecta el públic, com es mostra a la Figura 3.6. En aquell treball, no interessa detectar el públic, ja que només es busca detectar jugadors, àrbitres i la pilota que es troben dins del camp. Aquestes deteccions del públic es consideren falses i afecten negativament la puntuació DetA.

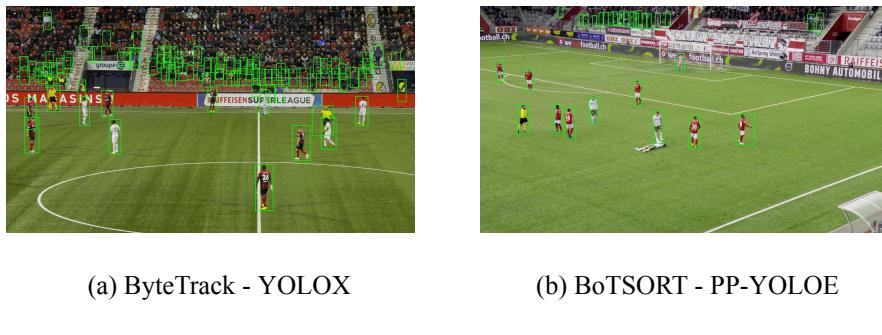


Figura 3.6: Detecció del públic.

Finalment, la detecció de la pilota és un dels elements que genera més problemes, ja que és un objecte molt petit i en constant moviment. En algunes seqüències on els jugadors ja són molt petits, la pilota encara ho és més i resulta impossible detectar-la. Com es mostra a la Figura 3.7, hi ha dues situacions: en la primera, la pilota no es detecta en el fotograma (a), mentre que en la segona sí que s'ha detectat adequadament (b).

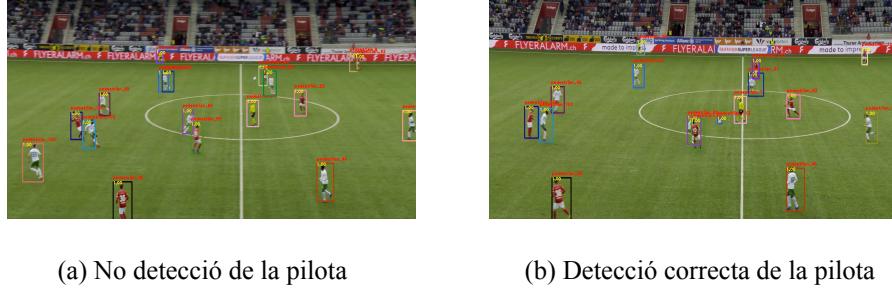
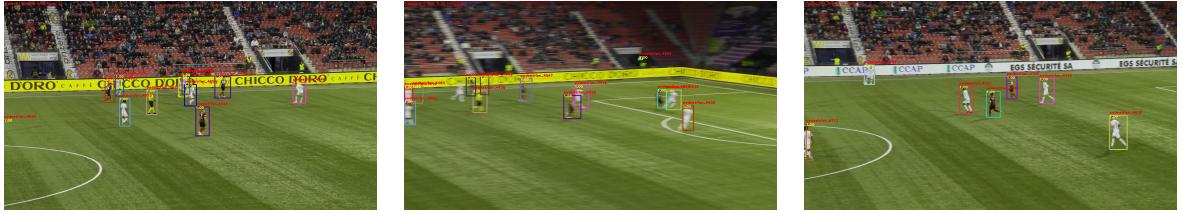


Figura 3.7: Detecció de la pilota en diferents fotogrames.

## Seguiment d'objectes

En el procés de seguiment d'objectes, hi ha diversos factors que poden provocar que un objecte no segueixi correctament, afectant així el comportament del model i la puntuació HOTA final.

Un dels casos més freqüents és quan un jugador surt del fotograma per un període de temps llarg, com s'observa en la Figura 3.8. A la imatge (a), el jugador amb el dorsal 4 té l'ID 4553. A la imatge (b), es mostra l'últim fotograma on apareix aquest jugador abans de sortir de l'escena. Després de 336 fotogrames, el jugador amb el dorsal 4 torna a aparèixer amb un nou ID, el 4712 (c). Aquest canvi afecta la puntuació AssA, ja que el mètode crea una nova trajectòria per a aquest jugador.



(a) Fotograma 130

(b) Fotograma 210

(c) Fotograma 546

Figura 3.8: Escenari on els jugadors abandonen l'escena durant un període de temps.

Com s'ha observat en la detecció, les oclusions segueixen sent un repte en el seguiment. Quan hi ha molts jugadors junts, els IDs sovint s'intercanvien i, a vegades, es perd l'ID original i se n'assigna un de nou, tal com es mostra a la Figura 3.9.

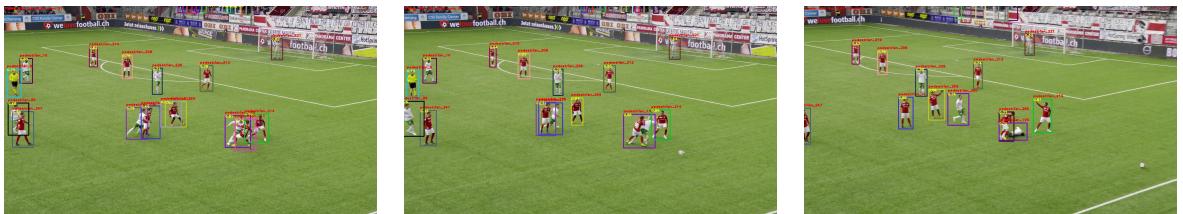


Figura 3.9: Escenari on els jugadors s'acumulen i es produueixen oclusions.

Un altre problema identificat en la detecció va ser la pilota. Aquest continua sent un obstacle en el seguiment d'objectes. A la Figura 3.10 es pot veure que cada vegada que un jugador toca la pilota o aquesta rebota, se li assigna un nou identificador.



(a) Pilota amb ID 37

(b) Pilota amb ID 63

(c) Pilota amb ID 112

Figura 3.10: Diferents identificadors per a la pilota.

## Fitxers ground truth

Cal destacar que els fitxers *ground truth* proporcionats per SoccerNet i utilitzats per calcular la mètrica HOTA no estan perfectament anotats. L'eina CVAT [12] ha estat essencial per analitzar els fotogrames de seqüències d'imatges i identificar oclusions.

Alguns problemes identificats inclouen l'anotació incorrecta dels jugadors en cas d'occlusions, ja que no sempre s'assigna el mateix ID al mateix jugador, i algunes caixes delimitadores que no s'ajusten perfectament al cos dels jugadors, arribant a tallar part seu cos.

## Capítol 4

# Millores, implementacions i resultats

En aquest capítol, s'han millorat els resultats del Capítol 3 mitjançant l'exploració de nous mètodes com StrongSORT, a més de provar l'entrenament d'un detector per abordar errors com la detecció del públic i la detecció en escenaris borrosos.

### 4.1 StrongSORT

StrongSORT [17] és una millora de DeepSORT en diversos aspectes, incloent-hi la detecció, l'*embedding* i l'associació.

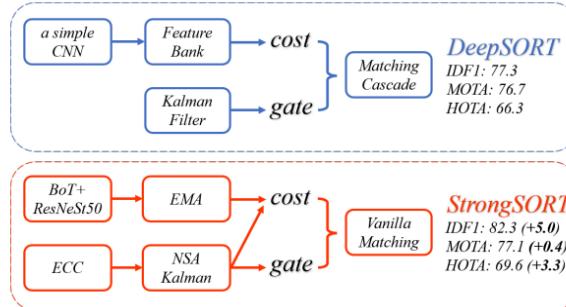


Figura 4.1: Comparació de marc i rendiment entre DeepSORT i StrongSORT. Figura de [17]

La comparació entre els algoritmes DeepSORT i StrongSORT es mostra a la Figura 4.1. DeepSORT es resumeix com un marc de dues branques: una per l'aparença i una altra pel moviment. D'altra banda, StrongSORT incorpora mòduls avançats i alguns trucs d'inferència.

Els mòduls avançats de StrongSORT inclouen l'ús de YOLOX-X en lloc del detector utilitzat per DeepSORT. A més, substitueix la simple CNN per un extractor de característiques més potent, BoT [30], que utilitza ResNeSt50 com *backbone*.

StrongSORT reemplaça el mecanisme del banc de característiques de DeepSORT amb una estratègia d'actualització d'aparença basada en EMA. Aquesta estratègia actualitza l'estat d'aparença  $e_i^t$  del *tracklet* i en el fotograma  $t$  tal com es feia en el *tracker* JDE (Eq.2.2), reduint així el soroll de detecció i el temps de processament.

Per tal de compensar els moviments de la càmera, StrongSORT utilitza el model Envelope Correlation Coefficient (ECC) [18] per minimitzar el soroll de moviment causat pel moviment de la càmera. Aquest model és una tècnica d'alignació paramètrica d'imatges que estima la rotació global i la translació entre fotogrames adjacents.

Per millorar la robustesa del KF utilitzat a DeepSORT davant de deteccions de baixa qualitat, StrongSORT implementa l'algorisme Noise Scale Adaptive (NSA) Kalman de GIAOTracker [16]. Aquest algorisme ajusta dinàmicament la covariància del soroll basant-se en la confiança de les deteccions, millorant l'exactitud dels estats actualitzats. Un valor de covariància baix significa que la detecció tindrà un pes més gran en el pas d'actualització de l'estat, i viceversa.

DeepSORT utilitza únicament la distància de característiques d'aparença com a cost d'associació durant la primera etapa d'associació, mentre que la distància de moviment actua com a barrera. D'altra banda, StrongSORT resol el problema d'assignació incorporant informació tant d'aparença com de

moviment, calculant una matriu de costos  $C$  com a suma ponderada del cost d'aparença  $A_a$  i el cost de moviment  $A_m$  (Eq.17). En aquesta equació,  $\lambda = 0.98$  representa el factor de ponderació.

$$C = \lambda A_a + (1 - \lambda) A_m \quad (17)$$

Finalment, StrongSORT opta per substituir l'algorisme de coincidència en cascada de DeepSORT amb una assignació global lineal bàsica per millorar la precisió a mesura que el *tracker* es fa més potent.

StrongSORT millora les seves capacitats incorporant dos algoritmes externs per abordar millor els problemes d'associació i de deteccions perdudes: el model Apperance-Free Link (AFLink) per millorar l'associació, i Gaussian-Smoothed Interpolation (GSI) per millorar la localització. Aquests algoritmes s'afegeixen sense generar costos addicionals, i aquesta nova versió es coneix com a StrongSORT++.

AFLink és un model sense aparença dissenyat per predir la connectivitat entre dos *tracklets* basant-se únicament en informació espaciotemporal. La Figura 4.2 il·lustra l'estructura de les dues branques del model. Com a entrada, s'utilitzen dos *tracklets* que contenen la identificació del fotograma en les posicions x i y dels últims 30 fotogrames, amb l'aplicació de *zero-padding* si és necessari. S'aplica un mòdul temporal per extreure les característiques, mentre que el mòdul de fusió integra la informació de diferents dimensions de característiques. Els dos mapes de característiques resultants es fusionen i es redueixen a vectors de característiques que finalment es concatèn. Finalment, s'utilitza una xarxa neuronal MultiLayer Perceptron (MLP) per predir una puntuació de confiança per a l'associació.

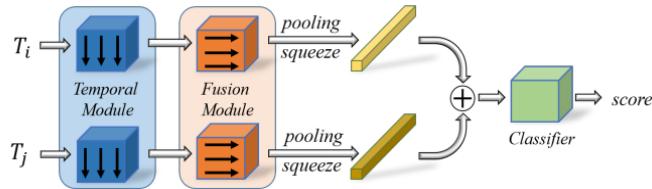


Figura 4.2: Estructura del model AFLink de dues branques. Figura de [17]

El procediment d'associació es formula com una tasca de classificació binària i s'optimitza amb la pèrdua de l'entropia creuada binària. Les parelles de *tracklets* poc raonables es filtrejant amb restriccions espaciotemporals. A continuació, el vincle global es resol com una tasca d'assignació lineal [27] amb la puntuació de connectivitat predicta.

GSI és un algoritme d'interpolació lleuger per omplir els buits en els *tracklets* causats per deteccions perdudes. Encara que s'han proposat altres estratègies com el filtre de Kalman i ECC, GSI utilitza la regressió de processos gaussians [41] per modelar el moviment no lineal en comptes d'ignorar la informació de moviment com fa la interpolació lineal.

La Figura 4.3 il·lustra un exemple de la diferència entre GSI i la interpolació lineal. Els resultats rastrejats en brut (en taronja) inclouen tremolars sorollosos, mentre que la interpolació lineal (en blau) ignora la informació del moviment. GSI (en vermell) resol ambdós problemes simultàniament en suavitzar tota la trajectòria amb un factor de suavitat adaptatiu.

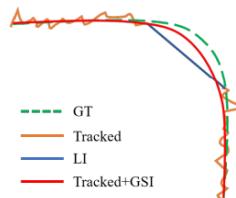


Figura 4.3: Il·lustració de la diferència entre la interpolació lineal i GSI. Figura de [17]

## Resultats

La Taula 4.1 presenta una comparativa dels resultats obtinguts pels mètodes DeepSORT, StrongSORT i StrongSORT++ amb diverses mètriques d'avaluació com HOTA, DetA i AssA, en el conjunt de test. S'observa que StrongSORT té un rendiment significativament superior a DeepSORT, tant en detecció com en associació. Pel que fa a StrongSORT++, s'ha provat l'aplicació dels dos mòduls simultàniament i cadascun per separat. Només s'observa una millora quan s'aplica el mòdul de AFLink, ja que es registra un lleuger augment en la puntuació AssA. Aquest resultat concorda amb l'affirmació del mètode , que destaca per associar dos *tracklets* en un. En canvi, l'aplicació del mòdul GSI no millora les deteccions, sinó que les empitjora, i tampoc millora l'associació. Això pot ser degut a la complexitat de les trajectòries, ja que en aplicar el mòdul, les trajectòries es suavitzen i no acaben d'associar-se bé.

Taula 4.1: Comparativa entre els resultats de DeepSORT, StrongSORT i StrongSORT++.

Mètode	HOTA (↑)	DetA (↑)	AssA (↑)
DeepSORT 1	69,552	82,628	58,668
StrongSORT	81,972	<b>92,22</b>	72,872
StrongSORT++ (AFLink + GSI)	78,278	85,619	71,695
StrongSORT++ (AFLink)	<b>82,265</b>	92,219	<b>73,385</b>
StrongSORT ++ (GSI)	78,002	85,621	71,193

La Taula 4.2 ofereix una visió més detallada del nombre total de deteccions i IDs respecte als reals dels mètodes DeepSORT, StrongSORT i StrongSORT++. En comparar DeepSORT i StrongSORT, s'observa com StrongSORT registra menys deteccions i assigna menys IDs que DeepSORT. Això suggereix que les deteccions de StrongSORT són més fiables, ja que amb un nombre menor de deteccions és capaç d'assignar menys IDs. A més, en afegir el mòdul AFLink a StrongSORT, es manté el mateix nombre de deteccions StrongSORT, però amb una assignació menor de IDs, donant lloc a una associació més fiable.

Taula 4.2: Comparativa II dels resultats entre DeepSORT, StrongSORT i StrongSORT++.

Mètode	Detections totals/Detections reals	ID totals/ ID reals (↓)
DeepSORT 1	555.793/564.547	3768/1935
StrongSORT	547.497/564.547	2836/1935
StrongSORT++ (AFLink + GSI)	554.261/564.547	<b>2766/1935</b>
StrongSORT++ (AFLink)	547.497/564.547	<b>2766/1935</b>
StrongSORT ++ (GSI)	553.993/564.547	2836/1935

Visualment, també es pot apreciar la millora entre el StrongSORT i el DeepSORT. Com es mostra a la Figura 4.4, amb l'ús del StrongSORT, es pot observar una detecció més precisa, identificant un major nombre de jugadors i amb caixes delimitadores que s'ajusten de manera més precisa als seus contorns.

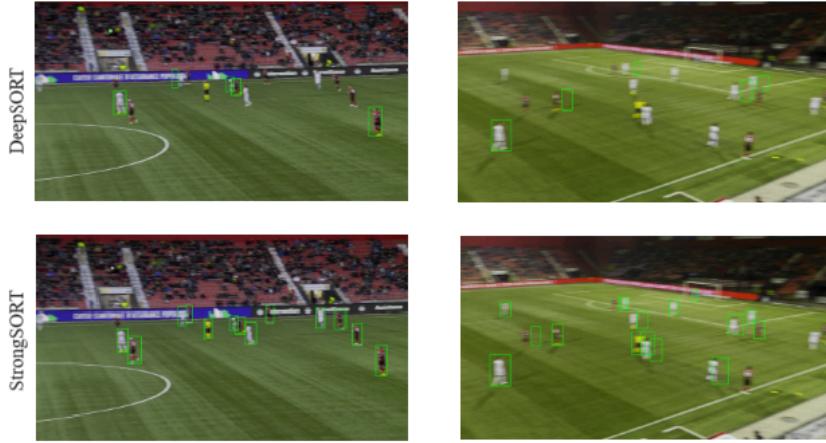


Figura 4.4: Comparació visual entre DeepSORT i StrongSORT.

En la part d’associació, es pot observar a la Figura 4.5 com en cada parella de fotogrames, es destaca un jugador específic que apareix i desapareix de la pantalla entre aquests fotogrames. Amb StrongSORT, es pot veure com el jugador manté la mateixa identificació en ambdós fotogrames. En canvi, amb DeepSORT, el jugador és assignat amb dues IDs diferents.

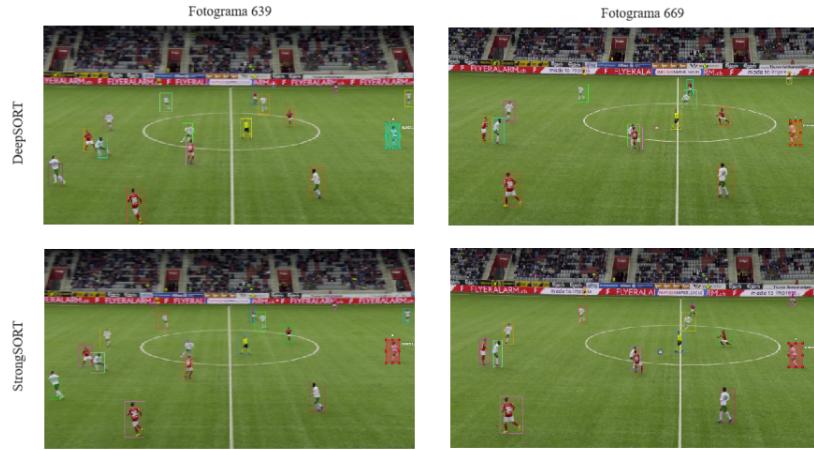


Figura 4.5: Comparació de l’associació entre DeepSORT i StrongSORT.

StrongSORT millora els resultats de DeepSORT en diversos aspectes. Aquesta millora es pot observar en l’ajust més precís de les caixes delimitadores als objectes i en la reassignació dels jugadors que desapareixen durant un cert nombre de fotogrames. Tot i això, encara es presenten algunes limitacions, especialment en escenes borroses on no es detecten tots els jugadors i en altres aspectes com la detecció de la pilota, que segueix sent un repte. Les associacions també presenten imperfeccions, amb casos on es perden jugadors i s’assignen amb un altre ID. Com es mostra a la Taula 4.2, encara hi ha un 40% més d’IDs assignats dels que hi hauria realment. Per a una ànalisi més concreta veure Apèndix-3.

## 4.2 Entrenament d’un detector

Després de revisar els resultats de StrongSORT i identificar una limitació en la detecció dels jugadors en situacions amb moviment de càmera i la detecció del públic, en aquest apartat es busca millorar les deteccions en aquestes circumstàncies. Amb aquest objectiu, s’utilitzaran dos mètodes principals: *Background Subtraction* i *Human Pose Estimation*. L’objectiu és entrenar un detector basat en les deteccions obtingudes amb aquests dos mètodes, esperant obtenir deteccions més precises i fiables dels jugadors.

El detector seleccionat per realitzar l'entrenament és la versió més avançada de la família de detectors YOLO, coneguda com a YOLOv8. Dins de YOLOv8, hi ha diversos models disponibles, i s'ha optat pel model YOLOv8x, reconegut per la seva alta precisió. Aquest detector està implementat per la plataforma d'aprenentatge profund Ultralytics [38]. Per obtenir informació sobre aquest detector, veure Apèndix-1.

## Background Subtraction

La tècnica de *Background Subtraction* és àmpliament utilitzada en visió per ordinador i processament d'imatges. El seu objectiu principal és detectar objectes en moviment en una seqüència de fotogrames capturats per una càmera estàtica. Permet extreure el que està en primer pla i el fons, facilitant així un procediment posterior [31].

Aquesta tècnica parteix del supòsit que el fons és sempre estàtic. Mitjançant la subtracció de fons del primer pla, es facilita la identificació d'objectes en moviment. Normalment, aquesta tècnica genera una màscara que pot ser considerada com el primer pla en la seqüència d'imatges. Abans d'aplicar aquesta tècnica, s'ha realitzat una detecció de camp que ajuda a eliminar les deteccions de persones de fora del camp de futbol, com podrien ser els aficionats.

### Detecció de camp

Per realitzar la detecció del camp, s'ha utilitzat el mètode [22], que combina la detecció de camp i línies. Primerament, es busca una màscara que contingui els píxels potencials del camp. La màscara del camp es calcula seguint aquests passos de processament successius:

- Aplicació d'un filtre verd per crear una primera imatge binària, utilitzant com a límits de color els valors (15,50,50) i (70, 255, 255).
- Selecció del contorn amb el component connectat més gran.
- Aproximació del contorn amb un polígon.
- Detecció de la línia inferior del camp i eliminació dels píxels de la màscara per sota d'aquesta línia.

Per detectar la línia inferior, s'utilitza el model pix2pix [9] per obtenir una imatge binària que s'aproxima a les línies del camp. A continuació, s'aplica una transformada de línia de Hough en aquesta imatge binària i s'utilitzen valors de longitud, angle i posició com a llindars per a les línies detectades.

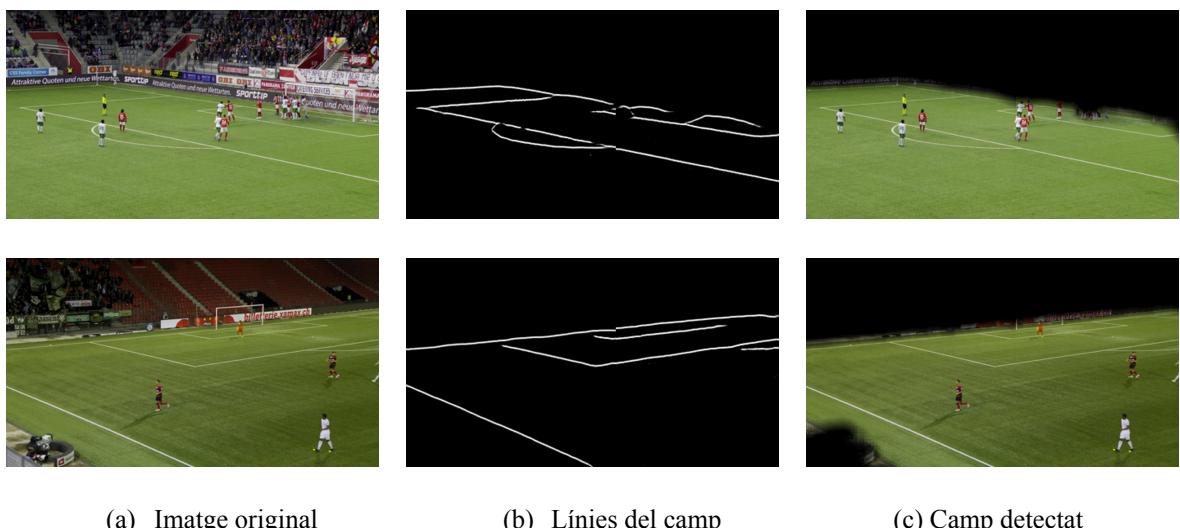


Figura 4.6: Resultats del model pix2pix.

La Figura 4.6 presenta els resultats obtinguts després d'aplicar el model pix2pix. A les imatges (b), es poden veure imatges binàries que detecten les línies del camp, mentre que a les imatges (c) s'identifica el camp detectat. Les zones de color negre mostren tot allò que no forma part del camp.

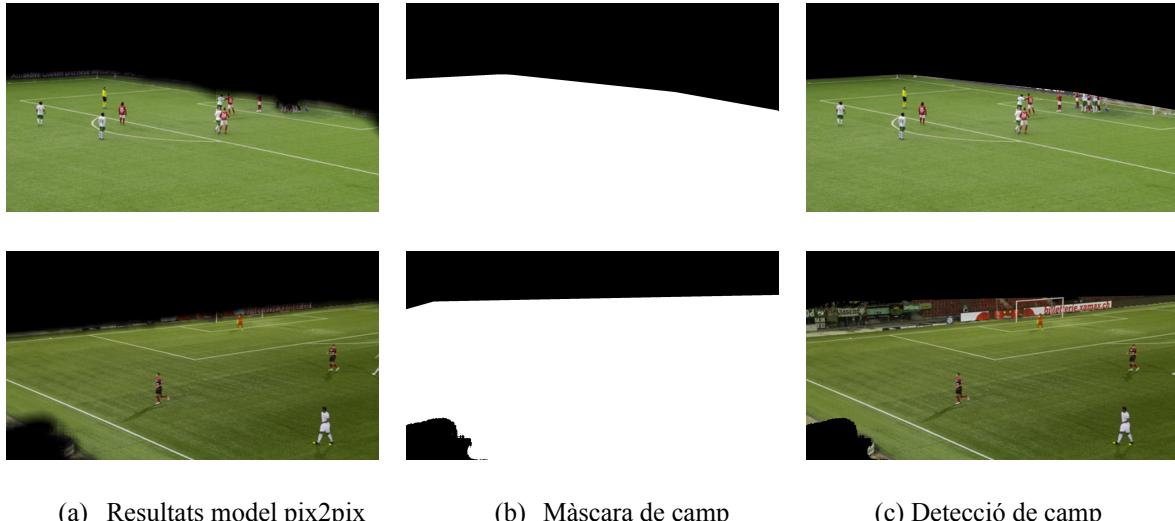


Figura 4.7: Comparació de la màscara de camp abans i després de l'actualització.

A la Figura 4.7 es pot observar com, amb els passos que s'ha mencionat anteriorment, la màscara del camp s'ajusta de manera més precisa al camp i produeix resultats millors. Si comparem les imatges (a), que són els resultats després d'aplicar el model pix2pix, amb les imatges (c), que són els resultats després d'aplicar l'actualització de la màscara de camp, es pot observar com les segones mostren resultats més realistes i aconsegueixen excloure més àrea que no pertany al camp.

Després de detectar el camp, s'ha utilitzat la tècnica de *Background Subtraction* per detectar els jugadors i separar el fons. S'ha desenvolupat un codi personalitzat on aquesta separació es realitza mitjançant una màscara verda per identificar els píxels que estan dins del rang de color verd i que es consideren part del fons. Com a resultat, es genera una imatge binària com es mostra a la Figura 4.8 on els jugadors estan representats amb píxels blancs i el fons amb píxels negres.



Figura 4.8: Imatges binàries després d'aplicar Background Subtraction.

Per generar les deteccions dels jugadors, s'ha aplicat diversos paràmetres per garantir que les caixes delimitadores coincideixin amb els jugadors:

- Operació de tancament: emprada per connectar parts properes d'un mateix jugador per evitar la generació de més d'una caixa delimitadora per jugador.
- Llindar d'àrea: emprat per eliminar possibles *blobs* o regions petites que no són jugadors, considerant només regions amb una àrea superior a aquest llindar com a possibles jugadors.
- Llindar d'*aspect ratio*: emprat per filtrar les regions que no s'ajustin a un aspecte semblant d'un jugador, descartant, per exemple, les balles publicitàries amb una amplada molt major que l'altura.

- Llindar per les línies del camp: emprat per comparar la quantitat de píxels blancs entre la imatge de les línies del camp i la regió detectada per la caixa delimitadora. Si les quantitats són semblants, es descarta la detecció.

La Figura 4.9 mostra algunes de les imatges resultants després de la detecció amb els paràmetres descrits anteriorment. Cada imatge correspon a una seqüència diferent. En analitzar les deteccions, es pot veure que la majoria dels jugadors són detectats, però no tots. En el cas dels jugadors amb una samarreta verda, es pot observar que la detecció no és precisa a causa de la confusió amb el color del camp. A més, en algunes ocasions es pot veure més d'un jugador dins de la mateixa caixa delimitadora.



Figura 4.9: Detections dels jugadors aplicant Background Subtraction.

### Human pose estimation

Per millorar la precisió de les deteccions, s'ha utilitzat un estimador de *human pose* basat en el mètode Realtime Multi-person 2D Pose Estimation [8], utilitzant una implementació disponible a GitHub [14].

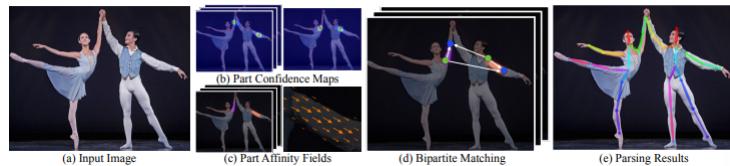


Figura 4.10: Estructura mètode Realtime Multi-person 2D Pose Estimation. Figura de [8]

La Figura 4.10 mostra l'estructura d'aquest mètode. El mètode rep una imatge (a) com a entrada i produeix les localitzacions 2D dels punts per a cada persona com a sortida (e). Utilitza una xarxa CNN de dues branques per predir simultàniament mapes de confiança 2D (b) per a la detecció de parts del cos i camps d'afinitat de parts (c) per a l'associació de parts. Els mapes de confiança indiquen la probabilitat de què una part del cos aparegui en una ubicació de píxel determinada, mentre que els camps d'afinitat codifiquen l'associació entre les parts del cos. Finalment, aquests són analitzats per la inferència *greedy* (d) per produir els punts clau 2D per a totes les persones de la imatge.

La Figura 4.11 mostra els resultats de les deteccions després d'aplicar Human Pose Estimation. Cada persona detectada es representa amb línies de colors que connecten totes les parts detectades d'una mateixa persona. A més, les caixes delimitadores s'han calculat a partir els punts superiors i inferiors de les parts detectades. En algunes seqüències, dividir els fotogrames en més parts pot permetre detectar més jugadors, especialment si són petits.



Figura 4.11: Deteccions dels jugadors aplicant Human Pose Estimator.

La Figura 4.12 mostra els resultats de combinar les deteccions dels dos mètodes per verificar si les caixes pertanyen al mateix jugador utilitzant diferents llindars. En cas que hi hagi més d'una caixa que pertany a un mateix jugador, se selecciona la caixa que tingui una alçada superior, ja que defineix millor el jugador.



(a) Background Subtraction      (b) Human Pose Estimation      (c) Combinació dels dos mètodes

Figura 4.12: Deteccions de jugadors combinant els dos mètodes.

Un cop s'ha obtingut totes les deteccions de les 57 seqüències del conjunt d'entrenament que corresponen a un total de 42.750 imatges, s'ha seleccionat el 10% d'aquestes imatges de cada seqüència, aproximadament 75 imatges corresponents al principi i final de la seqüència, per al conjunt de validació. Les imatges restants, que representen el 90%, s'han utilitzat per al conjunt d'entrenament.

Per dur a terme l'entrenament, s'ha utilitzat el model YOLOv8x amb pesos preentrenats en el conjunt de dades COCO. Aquest procés implica realitzar un *fine-tuning* per adaptar la xarxa preentrenada a les característiques del conjunt de dades d'aquest treball. L'entrenament es realitza durant 100 èpoques. Per a més informació sobre l'entrenament realitzat veure Apèndix-4.

La Taula 4.3 presenta una comparació dels resultats obtinguts amb dos mètodes de seguiment disponibles a la plataforma Ultralytics: BoT-SORT i ByteTrack. Es comparen en tres escenaris diferents: (1) utilitzant els pesos resultants de l'entrenament del detector YOLOv8x (color blau), (2)

utilitzant els pesos preentrenats del model YOLOv8x, i (3) utilitzant la configuració de seguiment específica emprada al Capítol 2.

BoT-SORT és el *tracker* que proporciona millors resultats, amb una puntuació HOTA més alta que ByteTrack (+15,97). Aquesta diferència ja s'havia identificat en el Capítol 3.

Pel que fa als tres escenaris, és evident que el mètode proposat, a priori, aconsegueix una puntuació HOTA més baixa en comparació amb els altres dos. L'ús del detector YOLOv8 en lloc dels detectors emprats al Capítol 3, millora significativament la puntuació DetA tant per a BoT-SORT (+23) com per a ByteTrack (+4).

Taula 4.3: Taula comparativa dels mètodes BoT-SORT i ByteTrack.

Mètode	HOTA (↑)	DetA (↑)	AssA (↑)
BoT-SORT + Inferència YOLOv8x (prop)	26,394	29,628	23,997
BoT-SORT + YOLOv8x	54,761	61,404	48,992
BoT-SORT	39,178	37,996	40,511
ByteTrack + Inferència YOLOv8x (prop)	25,217	29,089	22,344
ByteTrack + YOLOv8x	38,786	37,653	40,067
ByteTrack 1	39,780	33,271	47,677
ByteTrack 2	34,826	41,701	29,245
ByteTrack 3	30,710	21,143	39,141

La Taula 4.4 presenta una comparació més detallada entre el mètode proposat i el mètode que utilitza els pesos preentrenats del model YOLOv8x. Les mètriques avaluen el comportament de les trajectòries reals en cada algoritme de seguiment, classificant-les com majoritàriament rastrejada (MT), parcialment rastrejada (PT) o majoritàriament perduda (ML). Un objecte es considera MT si es rastreja amb èxit durant almenys el 80% de la seva vida útil, ML si es rastreja amb èxit com a màxim el 20%, mentre que tots els altres objectes es consideren PT.

Taula 4.4: Taula II comparativa dels mètodes BoT-SORT i ByteTrack.

Mètode	MT	PT	ML	Deteccions totals/ Deteccions reals	ID totals/ ID reals (↓)
BoT-SORT + Inferència YOLOv8x (prop)	116	732	1087	389.363/564.547	3652/1935
BoT-SORT + YOLOv8x	946	183	806	543.139/564.547	3006/1935
ByteTrack + Inferència YOLOv8x (prop)	120	703	1112	361.042/564.547	3549/1935
ByteTrack + YOLOv8x	898	221	816	875.732/564.547	17812/1935

Es pot observar una diferència significativa en el nombre de deteccions generades. Tot i que el mètode proposat produeix considerablement menys deteccions, l'altre mètode supera el nombre de deteccions reals, indicant un excés de deteccions amb un 13% de deteccions falses. Els resultats del mètode proposat mostren un augment del nombre d'IDs i una tendència a perdre trajectòries (augment de ML) a causa d'obtenir menys deteccions.

La manca de deteccions es pot atribuir a diversos factors. En primer lloc, el model preentrenat YOLOv8x ha estat entrenat amb conjunts de dades grans i diversos, com ara COCO, permetent-li generalitzar millor i detectar més objectes. En canvi, el model proposat ha estat entrenat amb un conjunt de dades molt més petit. A més, el model preentrenat ha passat per un entrenament extensiu amb hiperparàmetres optimitzats, mentre que el model proposat ha estat entrenat amb alguns hiperparàmetres diferents com el *learning rate*, el nombre d'èpoques, la qual cosa pot afectar el seu rendiment.

La Figura 4.13 il·lustra els principals escenaris en els quals el mètode proposat millora als altres dos mètodes.



Figura 4.13: Escenaris de millora del mètode proposat.

En el primer escenari, quan els jugadors apareixen borrosos en els fotogrames, el mètode proposat aconsegueix més deteccions. En el segon escenari, quan la càmera està molt allunyada i els jugadors es veuen molt petits, una vegada més, el mètode proposat obté més deteccions. Pel tercer escenari, quan els jugadors estan acumulats, per exemple, prop de la porteria, i el públic és més identifiable, el mètode proposat aconsegueix no detectar-lo, abordant així un dels problemes identificats en el Capítol 3. Aquestes millors són el resultat de l'entrenament realitzat, ja que les deteccions extrems es centren a eliminar el públic i millorar els casos borrosos. Per tant, es pot afirmar que s'ha assolit l'objectiu de l'entrenament.

En alguns casos, el mètode proposat no millora els resultats, com es mostra la Figura 4.14. A vegades, es detecten menys jugadors que amb els altres dos mètodes, i també es produeixen deteccions falses quan es detecten múltiples caixes per a un únic jugador. Aquesta situació pot ser causada pel fet que les deteccions tant de *Background Subtraction* com *Human Pose Estimation* no van ser filtrades prou quan es van combinar.



(a) BoT-SORT/ByteTrack



(b) BoT-SORT/ByteTrack +  
YOLOv8x



(c) BoT-SORT/ByteTrack +  
Inferència YOLOv8x (propri)

Figura 4.14: Escenaris on el mètode proposat no millora els resultats.

## Capítol 5

# Conclusions i treball futur

En aquest treball, s'ha aconseguit realitzar la detecció i el seguiment dels elements típics d'un camp de futbol en més d'un partit, incloent-hi jugadors, pilota i àrbitres mitjançant l'ús d'algoritmes avançats. Per aconseguir-ho, s'ha dut a terme una investigació profunda prèvia sobre els diferents algoritmes.

També s'ha seleccionat un dels millors conjunts de dades per a l'anàlisi del futbol, el qual ha requerit un estudi previ detallat per adaptar-lo als diferents algoritmes. S'ha aconseguit identificar el model de seguiment més eficient per al conjunt de dades seleccionat mitjançant una comparació exhaustiva dels diferents mètodes emprant diverses mètriques. Aquesta comparació i avaluació han permès entendre quins aspectes milloraven i quins aspectes deterioraven els diferents models.

A més, s'ha realitzat una recerca de mètodes fora de la plataforma PaddleDetection per millorar els resultats del mètode DeepSORT, que inicialment tenia els millors resultats ( $HOTA = 69,552$ ). S'ha trobat el mètode StrongSORT, enfocat a millorar DeepSORT, amb resultats millorats tant en detecció com en seguiment ( $HOTA = 82,265$ ).

No obstant, s'han identificat altres àrees de millora, com la detecció del públic i la falta de deteccions en escenaris borrosos. Això ha portat a desenvolupar deteccions pròpies utilitzant la subtracció de fons i l'estimació de la postura humana, amb l'objectiu de realitzar un entrenament amb un detector d'última generació. Tot i haver abordat els problemes identificats, l'entrenament no ha estat tan efectiu com s'esperava, principalment a causa de la manca de deteccions en comparació amb altres models. Per millorar l'entrenament, seria necessari utilitzar una quantitat més gran de dades i incorporar múltiples conjunts de dades específicament enfocats en el futbol.

D'altra banda, un aspecte que no ha rebut tant d'èmfasi en aquest treball és la detecció de la pilota, ja que és l'objecte més petit i difícil de seguir, considerant que la pilota pot sortir de l'àrea del camp de futbol durant el joc, tot i que és un objecte essencial, ja que el joc se centra en ella. Algunes implementacions que es podrien considerar per detectar la pilota serien l'aplicació de segmentació basada en el color blanc característic de la pilota per identificar regions que coincideixin amb aquest color i utilitzar un filtre per reconèixer la forma circular de la pilota per acotar aquestes regions. Pel seguiment, es podria implementar un algoritme que intenti predir la trajectòria de la pilota basant-se en la seva posició anterior i la direcció del moviment dels jugadors.

Per aconseguir un entrenament eficient i millorar la detecció i el seguiment en futurs treballs, es podrien realitzar les següents accions:

1. Explotar més informació per a la detecció: es podria calcular l'homografia del camp de futbol juntament amb les línies del camp per obtenir les coordenades reals en metres dels jugadors. Amb aquesta informació, es podrien eliminar persones que no estan jugant, com ara el públic. A més, també es podria determinar el perfil del moviment de les persones dins del camp de futbol, com per exemple, els àrbitres de banda que només es mouen per una meitat de la banda o els porters que no s'allunyen gaire de l'àrea de la porteria.
2. Buscar altres conjunts de dades: recopilar més conjunts de dades que continguin partits de futbol des de diferents perspectives, amb diverses condicions de llum i en diferents escenes de joc ajudarà a millorar la generalització del model.
3. Explorar mètodes de re-identificació: per abordar el problema del seguiment i reduir el nombre d'identificadors únics necessaris, es podrien explorar algoritmes de re-identificació.

## Bibliografia

- [1] Aharon, N., Orfaig, R., & Bobrovsky, B. Z. (2022). BoT-SORT: Robust associations multi-pedestrian tracking. Recuperat de <https://arxiv.org/pdf/2206.14651v2.pdf>
- [2] Alberto Rizzoli (2021). *Object Detection Models, Architectures & Tutorial [2023]*. Recuperat 5 de gener del 2024, des de <https://www.v7labs.com/blog/object-detection-guide>
- [3] Bathija, A., & Sharma, G. (2019). Visual object detection and tracking using yolo and sort. *International Journal of Engineering Research Technology*, 8(11), 345-355. Recuperat de <https://www.ijert.org/research/visual-object-detection-and-tracking-using-yolo-and-sort-IJERTV8IS110343.pdf>
- [4] Ben Le (2023). *Introduction to Multiple Object Tracking and Recent Developments*. Recuperat 10 de gener del 2024, des de <https://www.datature.io/blog/introduction-to-multiple-object-tracking-and-recent-developments>
- [5] Bewley, A., Ge, Z., Ott, L., Ramon, F., & Upcroft, B. (2016). Simple online and realtime tracking. *2016 IEEE international conference on image processing (ICIP)*. (p. 3464-3468). IEEE
- [6] Bouquet, J. Y. (2001). Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel corporation*, 5(1-10), 4. Recuperat de [http://robots.stanford.edu/cs223b04/algo\\_affine\\_tracking.pdf](http://robots.stanford.edu/cs223b04/algo_affine_tracking.pdf)
- [7] Cao, J., Pang, J., Weng, X., Khirodkar, R., & Kitani, K. (2023). Observation-centric sort: Rethinking sort for robust multi-object tracking. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (p. 9686-9696).
- [8] Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. *Proceedings of the IEEE conference on computer vision and pattern recognition* (p. 7291-7299).
- [9] Chen, J., & Little, J. J. (2019). Sports camera calibration via synthetic data. *Proceedings of the IEE/CVF conference on computer vision and pattern recognition workshops* (p. 0-0).
- [10] Chen, L., Ai, H., Zhuang, Z., & Shang, C. (2018). Real-time multiple people tracking with deeply learned candidate selection and person re-identification. *2018 IEEE international conference on multimedia and expo (ICME)* (p.1-6). IEEE
- [11] Cioppa, A., Giancola, S., Deliege, A., Kang, L., Zhou, X., Cheng, Z., Ghanem, B., & Van Droogenbroeck, M. (2022). Soccernet-tracking: Multiple object tracking dataset and benchmark in soccer videos. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (p. 3491-3502).
- [12] CVAT (s.d.) *Open Data Annotation Platform*. Recuperat 30 d'octubre del 2023, des de <https://www.cvcat.ai/>
- [13] Davis, J., Bransen, L., Devos, L., Meert, W., Robberechts, P., Van Haaren, J., & Van Roy, M. (2022). Evaluating sports analytics models: Challenges, approaches, and lessons learned. *AI Evaluation Beyond Metrics Workshop at IJCAI 2022* (Vol. 3169, p. 1-11). CEUR Workshop Proceedings.
- [14] DeNA (2022). *Chainer\_Realtime\_Multi-Person\_Pose\_Estimation: Chainer version of Realtime Multi-Person Pose Estimation*. Recuperat 10 de febrer del 2024, des de [https://github.com/DeNA/Chainer\\_Realtime\\_Multi-Person\\_Pose\\_Estimation](https://github.com/DeNA/Chainer_Realtime_Multi-Person_Pose_Estimation)

- [15] Du, C., Lin, C., Jin, R., Chai, B., Yao, Y., & Su, S. (2024). Exploring the State-of-the-Art in Multi-Object Tracking: A Comprehensive Survey, Evaluation, Challenges, and Future Directions. *Multimedia Tools and Applications*, 1-39. Recuperat de <https://doi.org/10.1007/s11042-023-17983-2>
- [16] Du, Y., Wan, J., Zhao, Y., Zhang, B., Tong, Z., & Dong, J. (2021). Giotracker: A comprehensive framework for memot with global information and optimizing strategies in visdrone 2021. *Proceedings of the IEEE/CVF International conference on computer vision* (p. 2809-2819).
- [17] Du, Y., Zhao, Z., Song, Y., Zhao, Y., Su, F., Gong, T., & Meng, H. (2023). Strongsort: Make deepsort great again. *IEEE Transactions on Multimedia*. Recuperat de <https://arxiv.org/pdf/2202.13514.pdf>
- [18] Evangelidis, G. D., & Psarakis, E. Z. (2008). Parametric image alignment using enhanced correlation coefficient maximization. *IEEE transaction on pattern analysis and machine intelligence*, 30(10), 1858-1865. Recuperat de <https://inria.hal.science/hal-00864385/document>
- [19] Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381-395. Recuperat de <https://dl.acm.org/doi/abs/10.1145/358669.358692>
- [20] Ge, Z., Liu, S., Wang, F., Li, Z., & Sun, J. (2021). Yolox: Exceeding yolo series in 2021. Recuperat de <https://arxiv.org/pdf/2107.08430>
- [21] Girshick, R., Donahue, J., Darell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition* (p. 580-587).
- [22] Hurault, S., Ballester, C., & Haro, G. (2020). Self-supervised small soccer player detection and tracking. *Proceedings of the 3rd international workshop on multimedia content analysis in sports* (p. 9-18).
- [23] jm12138 (2022). *SoccerNet\_Tracking\_PaddleDetection: A Baseline for Multi Objective Tracking (MOT) of Soccer and Soccer Players Based on SoccerNet Tracking Dataset and PaddleDetection*. Recuperat 6 d'octubre del 2023, des de [https://github.com/jm12138/SoccerNet\\_Tracking\\_PaddleDetection](https://github.com/jm12138/SoccerNet_Tracking_PaddleDetection)
- [24] Jonathon Luiten (2020). *TrackEval: HOTA (and other) evaluation metrics for Multi-Object Tracking (MOT)*. Recuperat 12 d'octubre del 2023, des de <https://github.com/JonathonLuiten/TrackEval>
- [25] Jonathon Luiten (2021). *How to evaluate tracking with the HOTA metrics*. Recuperat 12 d'octubre del 2023, des de <https://jonathonluiten.medium.com/how-to-evaluate-tracking-with-the-hota-metrics-754036d183e1>
- [26] Kendall, A., Gal, Y., & Cipolla, R. (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *Proceedings of the IEEE conference on computer vision and pattern recognition* (p. 7482-7491).
- [27] Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2), 83-97. Recuperat de <https://web.eecs.umich.edu/~pettie/matching/Kuhn-hungarian-assignment.pdf>
- [28] Labhesh Valechha (2022). *Object Tracking and Reidentification with FairMOT*. Recuperat 23 de setembre del 2023, des de <https://learnopencv.com/object-tracking-and-reidentification-with-fairmot/>

- [29] Lin, T. Y., Dollar, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. Proceedings of the IEEE conferences on computer vision and pattern recognition (p. 2117-2125).
- [30] Luo, H., Jiang, W., Gu, Y., Liu, F., Liao, X., Lai, S., & Gu, J. (2019). A strong baseline and batch normalization neck for deep person re-identification. *IEEE Transactions on Multimedia*, 22(10), 2597-2609. Recuperat de <https://arxiv.org/abs/1906.08332>
- [31] Muhammad Sabih (2022). *Background subtraction in computer vision*. Recuperat 1 de febrer del 2024, des de <https://medium.com/@muhammadsabih56/background-subtraction-in-computer-vision-402ddc79cb1b>
- [32] OpenAI. (2022). *ChatGPT*. <https://chat.openai.com/>
- [33] PaddlePaddle (2019). *PaddleDetection: Object Detection toolkit based on PaddlePaddle. It supports object detection, instance segmentation, multiple object tracking and real-time multi-person keypoint detection*. Recuperat 6 d'octubre del 2023, des de <https://github.com/PaddlePaddle/PaddleDetection>
- [34] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28. Recuperat de <https://arxiv.org/abs/1506.01497>
- [35] SoccerNet (2018). *SoccerNet*. Recuperat 21 de juliol del 2023, des de <https://www.soccer-net.org/>
- [36] SuperAnnotate AI Inc (s.d.). *SuperAnnotate | Empowering Enterprises with Custom LLM/GenAI/CV Models*. Recuperat 21 de juliol del 2023, des de <https://www.superannotate.com/>
- [37] Terven, J., Córdova-Esparza, D. M., & Romero-González, J. A. (2023). A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction*, 5(4), 1680-1716. Recuperat de <https://arxiv.org/pdf/2304.00501>
- [38] Ultralytics (s.d.). *Ultralytics | Revolutionizing the world of Vision AI*. Recuperat 27 d'abril del 2024, des de <https://www.ultralytics.com/>
- [39] Wang, Z., Zheng, L., Liu, Y., & Wang, S. (2020). Towards real-time multi-object tracking. *European conference on computer vision* (p. 107-122). Cham: Springer International Publishing.
- [40] Welch, G., & Bishop, G. (1995). An introduction to the Kalman filter. Recuperat de <https://perso.crans.org/club-krobot/doc/kalman.pdf>
- [41] Williams, C. K. I., & Rasmussen, C. E. (1996). Gaussian processes for regression, advances in neural information processing systems. *MIT Press*, 514-520. Recuperat de [https://proceedings.neurips.cc/paper\\_files/paper/1995/file/7cce53cf90577442771720a370c3c723-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/1995/file/7cce53cf90577442771720a370c3c723-Paper.pdf)
- [42] Wojke, N., Bewley, A., & Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. *2017 IEEE international conference on image processing (ICIP)* (p. 3645-3649). IEEE
- [43] Xu, S., Wang, X., Lv, W., Chang, Q., Cui, C., Deng, K., Wang, G., Dang, Q., Wei, S., Du, Y., & Lai, B. (2022). PP-YOLOE: An evolved version of YOLO. Recuperat de <https://arxiv.org/pdf/2203.16250>

- [44] Yu, F., Wang, D., Shelhamer, E., & Darrell, T. (2018). Deep layer aggregation. *Proceedings of the IEEE conference on computer vision and pattern recognition* (p. 2403-2412).
- [45] Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Lin, H., Zhang, Z., ... & Smola, A. (2022). Resnest: Split-attention networks. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (p. 2736-2746).
- [46] Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., & Wang, X. (2022). Bytetrack: Multi-object tracking by associating every detection box. *European conference on computer vision* (p. 1-21). Cham: Springer Nature Switzerland.
- [47] Zhang, Y., Wang, C., Wang, X., Zeng, W., & Liu, W. (2021). Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129, 3069-3087. Recuperat de <https://arxiv.org/pdf/2004.01888v6.pdf>
- [48] Zheng, L., Bie, Z., Sun, Y., Wang, J., Su, C., Wang, S., & Tian, Q. (2016). Mars: A video benchmark for large-scale person re-identification. *Computer Vision-ECCV 2016: 14th European Conference* (p. 868-884). Springer International Publishing.
- [49] Zhou, X., Koltun, V., & Krähenbühl, P. (2020). Tracking objects as points. *European conference on computer vision* (p. 474-490). Cham: Springer International Publishing.
- [50] Zhou, X., Wang, D., & Krähenbühl, P. (2019). Objects as points. Recuperat de <https://arxiv.org/pdf/1904.07850>

# Apèndix

## Llista de continguts

<b>1. Detectors d'objectes</b>	<b>36</b>
Família YOLO	36
CenterNet	39
Anàlisi comparativa dels resultats dels diferents detectors	39
<b>2. Seguiment d'objectes</b>	<b>43</b>
DeepSORT	43
ByteTrack	43
BoT-SORT	44
OC-SORT	45
CenterTrack	46
Anàlisi comparativa dels resultats dels mètodes de seguiment d'objectes	47
<b>3. StrongSORT</b>	<b>49</b>
ECC	49
Modificació de paràmetres	50
<b>4. Entrenament d'un detector</b>	<b>51</b>
<b>Referències</b>	<b>54</b>

## 1. Detectors d'objectes

En aquest treball s'utilitzen diversos detectors d'objectes. En aquesta secció es proporciona una anàlisi detallada de cadascun d'aquests detectors, oferint una comprensió més completa de les seves respectives capacitats.

### Família YOLO

La família YOLO s'ha destacat gràcies a la seva combinació única de velocitat i precisió, que permet una identificació ràpida i fiable d'objectes en imatges. Ha experimentat diverses iteracions, des del model inicial YOLOv1 fins a l'última versió, YOLOv8, tal com s'observa en la Figura 1.1. Cada versió posterior ha estat dissenyada per abordar les mancances i el rendiment de les versions anteriors [20].

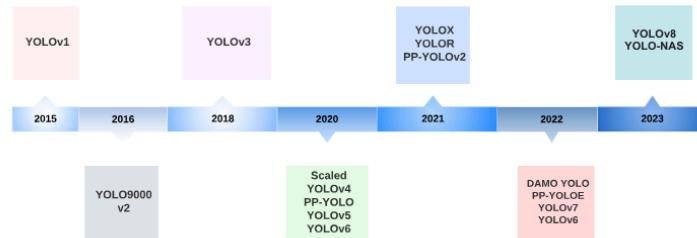


Figura 1.1: Evolució de la família YOLO. Figura de [20]

Joseph Redmon va presentar per primera vegada l'any 2015 una aproximació d'extrem a extrem en temps real per a la detecció d'objectes. L'acrònim *YOLO* fa referència al fet que l'algoritme és capaç de realitzar la tasca de detecció amb una sola passada de la xarxa. La motivació de Redmon va ser millorar mètodes que s'utilitzaven en aquell moment, com detectors de dues etapes que trigaven molt de temps a entrenar i eren difícils d'optimitzar.

L'arquitectura dels detectors d'objectes YOLO es divideix en tres parts principals:

- Backbone: s'encarrega d'extreure característiques útils de la imatge d'entrada. Normalment, es tracta d'una CNN entrenada en una tasca de classificació d'imatges a gran escala. Extreu característiques a diferents escales, des de baix nivell (vores i textures) extretes a les primeres capes fins a alt nivell (parts d'objectes i informació semàntica) extretes a capes més profundes.
- Neck: component intermedi que connecta el *backbone* amb el cap. Agrega i refina les característiques extretes pel *backbone*, sovint centrant-se a millorar la informació espacial i semàntica a diferents escales. També pot incloure capes convolucionals addicionals, FPN o altres mecanismes per millorar la representació de les característiques.
- Cap: component final, responsable de fer prediccions basades en les característiques proporcionades pel *backbone* i el *neck*. Normalment, inclou una o més subxarxes específiques de la tasca que realitzen la classificació, la localització, la segmentació d'instàncies i l'estimació de la posició. Processa les característiques que ha rebut del neck, generant prediccions per a cada objecte candidat. Finalment, es pot aplicar un pas de postprocessament, com ara NMS, per filtrar les deteccions superposades i conservar només les més confiades.

### YOLOv3

El detector YOLOv3 [17] presenta canvis significatius i una arquitectura més complexa en comparació amb els detectors anteriors, com YOLOv1 i YOLOv2.

Utilitza clústers de dimensions com a caixes d'ancoratge [16] per predir les caixes delimitadores, proporcionant les quatre coordenades per a caixa cada delimitadora. També calcula una puntuació d'objectivitat per a cada caixa delimitadora utilitzant la regressió logística, la qual és 1 si la caixa delimitadora anterior superposa un objecte  $gt$  més que qualsevol altra caixa delimitadora anterior. A diferència de Faster R-CNN [18], YOLOv3 només assigna una caixa delimitadora anterior per a cada objecte  $gt$ . A més, cada caixa prediu les classes que pot contenir mitjançant la classificació *multilabel*.

YOLOv3 fa prediccions de caixes en múltiples escales, utilitzant tres escales diferents. Extreu característiques d'aquestes escales utilitzant un concepte similar a les FPN [13]. Això permet generar caixes delimitadores amb més detall i millorar significativament la predicció d'objectes petits, que abans era un repte pels detectors anteriors.

Una de les principals novetats és l'ús d'una nova xarxa que combina la xarxa utilitzada a YOLOv2, DarkNet-19, i aquesta nova xarxa residual. DarkNet-53 té una arquitectura amb 53 capes convolucionals i connexions residuals. Aquesta xarxa neuronal és més potent i robusta que la seva versió anterior, DarkNet-19. A més, també és més ràpida i eficient, ja que l'estruatura de la xarxa utilitza millor la GPU.

La sortida de la xarxa és un tensor 3D que codifica les caixes delimitadores, la probabilitat de l'objecte i les prediccions de la classe.

## YOLOX

YOLOX [10] presenta canvis significatius respecte a YOLOv3, tot i compartir el mateix *backbone* DarkNet-53. En lloc d'un cap acoblat, YOLOX utilitza un cap desacoblat com es mostra a la Figura 1.2. Per a cada nivell de la característica FPN, s'afegeix una capa de convolució 1x1 per reduir la dimensió del canal, seguida de dues branques paral·leles amb dues capes convolucionals 3x3 cadascuna per a la classificació i regressió. La branca IoU s'afegeix a la branca de regressió. Aquest canvi millora significativament la velocitat de convergència.

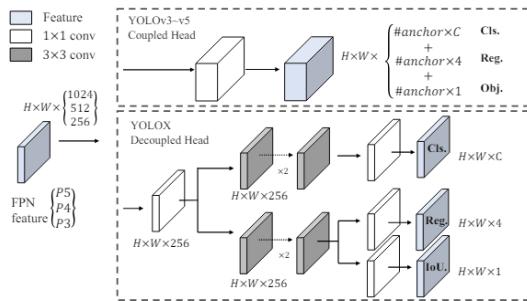


Figura 1.2: Il·lustració de la diferència entre el cap de YOLOv3 i el cap desacoblat proposat.  
Figura de [10].

S'utilitzen tècniques d'augment de dades com Mosaic i MixUp [25] durant l'entrenament per millorar el seu rendiment.

Una altra característica destacable és la seva arquitectura sense ancoratge. El mecanisme d'ancoratge presenta diversos problemes, com la necessitat de realitzar una anàlisi de *clustering* per determinar un conjunt d'ancoratges òptims abans de l'entrenament, així com la complexitat dels caps de detecció i el nombre de prediccions per a cada imatge. Per canviar a un model sense ancoratges, YOLOX redueix les prediccions per a cada ubicació de 3 a 1 i prediu directament quatre valors (dos desplaçaments i l'alçada i l'amplada de la caixa predita) [8].

L'assignació avançada d'etiquetes [9] és un altre progrés important en la detecció d'objectes. S'utilitza SimOTA per identificar prediccions positives i negatives.

Tot i que les prediccions positives s'etiqueten com a coincidents amb els objectes *ground truth* i les negatives com a no coincidents, aquestes últimes no es descarten, ja que es fan servir en una de les funcions de pèrdua.

La sortida consta de tres tensors que contenen informació diferent, en lloc d'un únic tensor amb tota la informació. Cada tensor inclou Cls (classe de cada caixa delimitadora), Reg (coordenades de la caixa delimitadora) i IoU (confiança de l'objecte en la caixa delimitadora).

### YOLOv8

Com es detalla a l'Apartat 4.2, s'ha utilitzat el detector YOLOv8 [20] per a l'entrenament. Aquest detector pertany a la família YOLO i és la versió més moderna desenvolupada per Ultralytics [22]. Hi ha cinc models diferents, que van des de YOLOv8 Nano, el més petit i ràpid, fins a YOLOv8 Extra Large (YOLOv8x), que és més lent, però més precís, i és el que s'ha utilitzat en l'entrenament.

Incorpora diverses millores respecte a les versions anteriors de YOLO. Aquestes millores inclouen una nova arquitectura de xarxa neuronal que integra els mòduls Feature Pyramid Network (FPN) [13] i Path Aggregation Network (PAN) [14]. El mòdul FPN s'utilitza per generar mapes de característiques a múltiples escales i resolucions, mentre que el mòdul PAN agrega característiques de diferents nivells de la xarxa, millorant així la precisió de la detecció.

Una altra novetat destacada és la incorporació de RoboFlow Annotate [19], una nova eina d'etiquetatge que facilita el procés d'anotació d'imatges per a l'entrenament del model.

Incorpora tècniques de postprocessament avançades que consisteixen en un conjunt d'algoritmes aplicats a les caixes delimitadores predites i les puntuacions d'objectivitat generades per la xarxa neuronal. Aquestes tècniques ajuden a millorar els resultats de detecció, eliminar les deteccions redundants i augmentar la precisió global de les prediccions. Un exemple és Soft-NMS [2], una variant de la tècnica NMS que aplica un llindar suau a les caixes delimitadores superposades en lloc de descartar-les directament. En canvi, NMS elimina les caixes delimitadores que se superposen i només conserva les que tenen la puntuació d'objectivitat més alta.

Una característica clau és el seu mecanisme de detecció sense ancoratge, que prediu el centre d'un objecte directament en lloc del desplaçament d'una caixa d'ancoratge coneiguda. Això redueix el nombre de prediccions de caixes i accelera el procés de postprocessament.

Finalment, YOLOv8 utilitza la tècnica d'augment de mosaics en el conjunt d'entrenament. Aquesta tècnica consisteix a seleccionar quatre imatges aleatòries del conjunt d'entrenament i combinar-les en una única imatge de mosaic.

### PPYOLOE

PPYOLOE [24] és un dels models implementats per la plataforma de *Deep Learning* PaddlePaddle [15], d'aquí el nom PP. Aquests models s'han desenvolupat paral·lelament als models YOLO, exercint una influència considerable en l'evolució de la família YOLO.

PPYOLOE es basa en el model PP-YOLOv2, que utilitza ResNet50-vd [11] com a *backbone* amb convolució deformable. En essència, PP-YOLOv2 s'inspira en YOLOv3 perquè assigna una caixa d'ancoratge per a cada objecte. A més de les pèrdues de classificació, regressió i objectivitat, PP-YOLOv2 incorpora dues noves pèrdues: la pèrdua IoU i la pèrdua de consciència de IoU, que milloren el rendiment del model.

A diferència del model PP-YOLOv2, utilitza un mètode d'assignació d'etiquetes sense ancoratge, ja que la introducció d'un mecanisme d'ancoratge requereix la incorporació de nombrosos hiperparàmetres, el qual no pot generalitzar-se bé en altres conjunts de dades. Aquesta tècnica, basada

en Fully Convolution One-Stage (FCOS) [21], utilitza límits superiors i inferiors per a cada cap de detecció i prediu un vector 4D per a la regressió.

Proposa un nou bloc RepRes que combina les connexions residuals i les connexions denses per millorar l'eficiència i el rendiment del model en el *backbone* i *neck*.

Per millorar encara més la precisió, l'assignació d'etiquetes és un altre aspecte a tenir en compte. A diferència de YOLOX, que utilitza SimOTA com a estratègia d'assignació d'etiquetes, PP-YOLOE utilitza Task Alignment Learning (TAL) de TOOD [7]. Aquesta tècnica consisteix en una assignació dinàmica d'etiquetes i una pèrdua alineada amb les tasques. En alinear explícitament les tasques de classificació i localització, TAL aconsegueix simultàniament la puntuació de classificació més alta i la caixa delimitadora més precisa.

El cap desacoblat proposat per YOLOX pot fer que les tasques de classificació i localització siguin separades i independents, i manquin d'aprenentatge específic de la tasca. PP-YOLOE es basa en TOOD i proposa el cap Efficient Task-aligned (ET) per millorar la precisió de la detecció d'objectes. Aquest cap aprèn les tasques de classificació i localització utilitzant pèrdues com VariFocal Loss (VFL) [26] i Distribution Focal Loss (DFL) [12] respectivament.

## CenterNet

CenterNet [30] és un detector que utilitza un enfocament basat en punts clau per predir els centres d'objectes en una imatge. En primer lloc, una xarxa totalment convolucional processa la imatge i genera un mapa de característiques final amb mapes de calor per a diferents punts clau. Els pics d'aquests mapes de característiques de sortida es consideren els centres predictius, es mantenen els 100 pics principals. A més, la xarxa prediu les dimensions de les caixes a partir de l'estimació de punts clau sense la necessitat d'aplicar NMS o altres postprocessaments. Pel que fa a la classificació, els pics de mapes de calor també s'associen a una classe concreta.

CenterNet considera que el centre d'una caixa és alhora un objecte i un punt clau, i després utilitza aquest centre predictiu per trobar les coordenades i el desplaçament de la caixa delimitadora. El model consta de tres caps de sortida: el cap del mapa de calor, que estima els punts clau donada una imatge d'entrada; el cap de dimensions, que prediu les dimensions de la caixa (amplada i alçada); i el cap d'offset que estima el desplaçament del centre de l'objecte, mitigant així l'impacte del canvi de mida.

## Anàlisi comparativa dels resultats dels diferents detectors

La Taula 1.1 presenta els resultats dels diferents detectors utilitzats pels mètodes. Per a cada detector, s'enumeren els mètodes que en fan ús, les deteccions realitzades en comparació amb les deteccions reals i la puntuació DetA associada. La selecció del detector té un impacte significatiu en el rendiment de cada mètode, especialment en els models SDE que tracten les tasques de detecció i associació per separat.

En el context específic del conjunt de dades SoccerNet, es pot observar que el YOLOv3 té el millor rendiment. També és evident com el nombre total de deteccions té un impacte directe en la puntuació DetA. Per exemple, malgrat que el YOLOX genera un gran nombre de deteccions, obté una puntuació DetA relativament baixa. Això es deu al fet que, amb un nombre tan elevat de deteccions, hi ha una alta probabilitat d'incloure deteccions falses, superant el nombre de deteccions reals. El mateix passa amb el detector PP-YOLOE, que utilitzant el mètode ByteTrack 3 aconsegueix el major nombre de deteccions, però al mateix temps la puntuació DetA més baixa.

En canvi, detectors com YOLOv3 o CenterNet presenten un equilibri entre el nombre total de deteccions i les deteccions reals. Amb un nombre total de deteccions proper al nombre real d'objectes del conjunt de dades, aquests detectors aconsegueixen una puntuació DetA més alta.

Taula 1.1: Resultats comparatius dels diferents detectors en el conjunt de test.

Detector	Mètode	Deteccions totals / Deteccions reals	DetA ( $\uparrow$ )
YOLOv3	DeepSORT 1	555.793/564.547	<b>82,628</b>
	DeepSORT 2	555.706/564.547	82,376
	ByteTrack 2	586.721/564.547	41,701
	JDE	502.750/564.547	40,594
YOLOX	ByteTrack 1	1.109.590/564.547	33,271
	OC-SORT 1	942.438/564.547	37,686
PP-YOLOE	ByteTrack 3	1.473.017/564.547	24,143
	BoT-SORT	887.365/564.547	37,996
	OC-SORT 2	549.683/564.547	50,614
	DeepSORT 3	555.284/564.547	82,6
CenterNet	FairMOT 1	567.471/564.547	46,318
	FairMOT 2	592.361/564.547	45,776
	CenterTrack	804.580/564.547	36,032

Malgrat la importància del nombre de deteccions per avaluar l'eficàcia d'un detector, també és crucial tenir en compte la precisió de les deteccions, siguin correctes (TP) o falses (FP). Un detector amb moltes deteccions, però amb un alt nombre de FP no serà útil. A la Taula 1.2 es detallen els percentatges de TP i FP de cada mètode.

En analitzar els resultats, es pot observar que el detector YOLOv3 sol produir un percentatge més alt de deteccions positives, mentre que els detectors com YOLOX i PP-YOLOE tendeixen a generar un percentatge més alt de deteccions falses. No obstant això, aquesta diferència no només depèn del detector utilitzat, sinó també del mètode de seguiment emprat. Per exemple, el mètode DeepSORT genera un alt percentatge de deteccions positives, independentment de si utilitza el detector YOLOv3 o PP-YOLOE. En canvi, els mètodes ByteTrack i OC-SORT depenen més del tipus de detector utilitzat.

Taula 1.2: Resultats comparatius II dels diferents detectors en el conjunt de test.

Detector	Mètode	TP (↑)	FP (↓)
YOLOv3	DeepSORT 1	97,05 %	2,95 %
	DeepSORT 2	97,04 %	2,96 %
	ByteTrack 2	68,06 %	31,94 %
	JDE	73,22 %	26,78 %
YOLOX	ByteTrack 1	43,66 %	56,34 %
	OC-SORT 1	50,28 %	49,72 %
PP-YOLOE	ByteTrack 3	31,46 %	68,54 %
	BoT-SORT	52,01 %	47,99 %
	OC-SORT 2	77,02%	22,98 %
	DeepSORT 3	<b>97,09 %</b>	<b>2,91 %</b>
CenterNet	FairMOT 1	74,31 %	25,69 %
	FairMOT 2	73,46 %	26,54 %
	CenterTrack	53,28 %	46,72 %

La Figura 1.3 mostra un exemple visual de com l'elecció del detector influeix en el mètode ByteTrack. Les imatges (a) i (c) mostren les deteccions dels detectors YOLOX i PP-YOLOE respectivament, i es pot observar que hi ha una gran quantitat d'aficionats a les grades que són detectats, generant deteccions no desitjades. L'objectiu d'aquest treball és detectar només jugadors, àrbitres, i la pilota, entre d'altres, excloent tots els altres objectes que estiguin fora del camp de futbol. En canvi, a les imatges (b), que corresponen a les deteccions generades pel detector YOLOv3, només es detecta una petita quantitat de persones del públic.



Figura 1.3: Resultats de diferents detectors utilitzant ByteTrack.

Des d'una perspectiva analítica i visual, es pot concloure que el detector YOLOv3 és el més adequat per al conjunt de dades utilitzat en aquest treball. Tot i això, detectors com YOLOX i PP-YOLOE, que poden extreure característiques detallades, poden proporcionar una detecció més precisa d'objectes. Tanmateix, aquest avantatge també pot comportar la sobre detecció d'objectes no rellevants, com els espectadors a les grades, que generen falses deteccions i una puntuació DetA baixa.

En altres conjunts de dades on la distinció entre objectes rellevants i no rellevants sigui més difusa, detectors com YOLOX i PP-YOLOE poden resultar més útils que YOLOv3. Ara bé, cal tenir en compte que això pot implicar una etapa de postprocessament per filtrar les deteccions i eliminar les no desitjades.

## 2. Seguiment d'objectes

Al llarg d'aquest treball, s'ha explorat diversos mètodes de seguiment d'objectes. En aquest apartat es presentarà una anàlisi més exhaustiva dels mètodes esmentats anteriorment, juntament amb el seu pseudocodi si està disponible, per facilitar una comprensió més completa del seu funcionament.

### DeepSORT

#### Matching Cascade

La Figura 2.1 descriu l'algoritme d'emparellament. Com a entrada, es proporciona el conjunt d'índexs de pistes  $T$  i deteccions  $D$ , així com l'edat màxima  $A_{max}$ . A les línies 1 i 2, es calcula la matriu de cost d'associació  $C$  i la matriu d'associacions admissibles  $B$ . Posteriorment, l'algoritme itera sobre l'edat de la pista  $n$  per resoldre el problema d'assignació lineal per a pistes d'edat creixent. A la línia 6, es selecciona el subconjunt de pistes  $T_n$  que no s'han associat amb una detecció en els darrers  $n$  fotogrames. A la línia 7, es resol l'assignació lineal entre les pistes en  $T_n$  i les deteccions no associades  $U$ . A les línies 8 i 9, s'actualitza el conjunt d'emparellaments i deteccions no associades, que es retorna després de completar la línia 11.

**Listing 1** Matching Cascade

---

**Input:** Track indices  $\mathcal{T} = \{1, \dots, N\}$ , Detection indices  $\mathcal{D} = \{1, \dots, M\}$ , Maximum age  $A_{max}$

- 1: Compute cost matrix  $C = [c_{i,j}]$  using Eq. 5
- 2: Compute gate matrix  $B = [b_{i,j}]$  using Eq. 6
- 3: Initialize set of matches  $\mathcal{M} \leftarrow \emptyset$
- 4: Initialize set of unmatched detections  $\mathcal{U} \leftarrow \mathcal{D}$
- 5: **for**  $n \in \{1, \dots, A_{max}\}$  **do**
- 6:     Select tracks by age  $\mathcal{T}_n \leftarrow \{i \in \mathcal{T} \mid a_i = n\}$
- 7:      $[x_{i,j}] \leftarrow \text{min\_cost\_matching}(C, \mathcal{T}_n, \mathcal{U})$
- 8:      $\mathcal{M} \leftarrow \mathcal{M} \cup \{(i, j) \mid b_{i,j} \cdot x_{i,j} > 0\}$
- 9:      $\mathcal{U} \leftarrow \mathcal{U} \setminus \{j \mid \sum_i b_{i,j} \cdot x_{i,j} > 0\}$
- 10: **end for**
- 11: **return**  $\mathcal{M}, \mathcal{U}$

---

Figura 2.1: Algoritme Matching Cascade de DeepSORT. Figura de [23]

### ByteTrack

#### BYTE

La Figura 2.2 mostra el pseudocodi del *tracker* BYTE. L'entrada de BYTE és una seqüència de vídeo  $V$ , juntament amb un detector d'objectes  $Det$  i s'estableix un llindar de puntuació de detecció  $\tau$ . La sortida de BYTE és un conjunt de pistes  $T$  del vídeo, cadascuna de les quals conté una caixa delimitadora i la identitat de l'objecte a cada caixa.

Per a cada fotograma del vídeo, es prediu les caixes delimitadores i les puntuacions utilitzant el detector  $Det$ . Les caixes es divideixen en dues categories,  $D_{high}$  i  $D_{low}$ , segons  $\tau$ . Per les caixes amb puntuacions més altes que  $\tau$ , s'afegeixen a  $D_{high}$  i per a les caixes més baixes que  $\tau$ , s'afegeixen a  $D_{low}$  (línies 3-13). Després de separar les caixes segons la puntuació, s'aplica el KF per predir les noves ubicacions en el fotograma actual de cada pista a  $T$  (línies 14-16).

La primera associació es realitza entre  $D_{high}$  i totes les pistes  $T$ , incloses les pistes perdudes  $T_{lost}$ . La *Similarity#1* es pot calcular per l'IoU o per les distàncies de característiques Re-ID entre  $D_{high}$  i la caixa predita de les pistes  $T$ . Posteriorment, s'utilitza l'algoritme Hongarès per acabar l'emparellament basat en la similitud. Es conserven les deteccions no emparellades a  $D_{remain}$  i les pistes no emparellades a  $T_{remain}$  (línies 17-19).

La segona associació es realitza entre  $D_{low}$  i  $T_{remain}$  després de la primera associació. Es mantenen les pistes no emparellades a  $T_{re-remain}$  i només s'eliminen les caixes de baixa puntuació no emparellades, ja es consideren fons (línies 20-21). En aquest cas, *Similarity2#* es basa únicament en la mètrica IoU, atès que les caixes de detecció de baixa puntuació sovint presenten oclusions o desenfocament de moviment, i les característiques d'aparença no són fiables.

Després de l'associació, les pistes no emparellades restants s'eliminen de les seqüències de pistes. Per a  $T_{re-remain}$  després de la segona associació, es posen a  $T_{lost}$ . Cada pista de  $T_{lost}$ , només s'elimina si existeix durant més de 30 fotogrames. En cas contrari, es manté  $T_{lost}$  a T (línia 22). Finalment, s'inicialitza noves pistes amb  $D_{remain}$  després de l'associació (línies 23-27). La sortida de cada fotograma individual són les caixes delimitadores i identitats de les pistes  $T$  al fotograma actual.

**Algorithm 1:** Pseudo-code of BYTE.

```

Input: A video sequence V; object detector Det; detection score
       threshold  $\tau$ 
Output: Tracks  $\mathcal{T}$  of the video
1 Initialization:  $\mathcal{T} \leftarrow \emptyset$ 
2 for frame  $f_k$  in  $V$  do
3   /* Figure 2(a) */
4   /* predict detection boxes & scores */
5    $\mathcal{D}_k \leftarrow \text{Det}(f_k)$ 
6    $\mathcal{D}_{high} \leftarrow \emptyset$ 
7    $\mathcal{D}_{low} \leftarrow \emptyset$ 
8   for  $d$  in  $\mathcal{D}_k$  do
9     if  $d.score > \tau$  then
10      |  $\mathcal{D}_{high} \leftarrow \mathcal{D}_{high} \cup \{d\}$ 
11    else
12      |  $\mathcal{D}_{low} \leftarrow \mathcal{D}_{low} \cup \{d\}$ 
13    end
14   /* predict new locations of tracks */
15   for  $t$  in  $\mathcal{T}$  do
16     |  $t \leftarrow \text{KalmanFilter}(t)$ 
17   end

```

```

/* Figure 2(b) */
/* first association */
Associate  $\mathcal{T}$  and  $\mathcal{D}_{high}$  using Similarity#1
 $\mathcal{D}_{remain} \leftarrow$  remaining object boxes from  $\mathcal{D}_{high}$ 
 $\mathcal{T}_{remain} \leftarrow$  remaining tracks from  $\mathcal{T}$ 

/* Figure 2(c) */
/* second association */
Associate  $\mathcal{T}_{remain}$  and  $\mathcal{D}_{low}$  using similarity#2
 $\mathcal{T}_{re-remain} \leftarrow$  remaining tracks from  $\mathcal{T}_{remain}$ 

/* delete unmatched tracks */
 $\mathcal{T} \leftarrow \mathcal{T} \setminus \mathcal{T}_{re-remain}$ 

/* initialize new tracks */
23 for  $d$  in  $\mathcal{D}_{remain}$  do
24   |  $\mathcal{T} \leftarrow \mathcal{T} \cup \{d\}$ 
25 end
26 end
27 Return:  $\mathcal{T}$ 

```

Figura 2.2: Pseudocodi del tracker BYTE. Figura de [27]

BYTE també pot ser aplicat en altres mètodes d'associació. Un exemple és quan BYTE es combina amb el mètode FairMOT, les característiques Re-ID s'afegeixen a la primera associació.

## BoT-SORT

La Figura 2.3 mostra el pseudocodi de l'algoritme BoT-SORT amb la incorporació del mòdul Re-ID. L'entrada és una seqüència de vídeo  $V$ , juntament amb un detector d'objectes  $Det$ , un extractor de característiques d'aparença  $Enc$ , un llindar de puntuació de detecció  $\tau$  i un llindar de seguiment  $\eta$ . La sortida és un conjunt de pistes  $T$  del vídeo, cadascuna de les quals conté una caixa delimitadora i la identitat de l'objecte a cada caixa.

Per a cada fotograma del vídeo, es preduïen les caixes delimitadores i les puntuacions utilitzant el detector  $Det$ . Les caixes es divideixen en dues categories,  $D_{high}$  i  $D_{low}$ , segons  $\tau$ . Per les caixes amb puntuacions més altes que  $\tau$ , s'afegeixen a  $D_{low}$  i per a les caixes més baixes que  $\tau$ , s'afegeixen a  $D_{low}$ . Les característiques d'aparença només s'estreuen de  $D_{high}$  amb  $Enc$  passant-li l'*embedding* d'aparença  $f_k$  de la detecció juntament amb les caixes delimitadores detectades  $d.box$  i s'afegeixen a  $F_{high}$  (línies 3-12).

Després de separar les caixes segons la puntuació i calcular les característiques d'aparença de les caixes  $D_{high}$ , es calcula la matriu afí  $A_{k-1}^k$  utilitzant RANSAC per estimar el moviment de la càmera entre  $f_{k-1}$  i  $f_k$  (línia 13). Seguidament, s'utilitza el KF per predir les noves ubicacions en el fotograma actual de cada pista a  $T$  i la compensació del moviment de la càmera GMC utilitzant la matriu afí  $A_{k-1}^k$  (línies 14-16).

La primera associació es realitza entre  $D_{high}$  i totes les pistes  $T$ , incloses les pistes perdudes  $T_{lost}$ . Es calculen les matrius de costos  $C_{iou}$  (cost de moviment) i  $C_{emb}$  (cost d'aparença) (línies 17-18).

Seguidament, s'utilitza el valor mínim de cada element de les matrius com a valor final de  $C_{high}$  i es resol l'assignació lineal de les deteccions de confiança  $D_{high}$  utilitzant l'algorisme Hongarès i basat en  $C_{high}$  (línia 20). Es conserven les deteccions no emparellades a  $D_{remain}$  i les pistes no emparellades a  $T_{remain}$  (línes 21-22).

La segona associació es realitza entre  $D_{high}$  i  $T_{remain}$ . Només es calcula la matriu de cost  $C_{low}$  mitjançant les caixes delimitadores de les pistes  $T_{remain}$  i les caixes de detecció  $D_{low}$ . A més, l'assignació lineal de les deteccions de confiança  $D_{low}$  es resol mitjançant l'algorisme Hongarès i basat en  $C_{low}$ . Es conserven les pistes no emparellades a  $T_{re-remain}$  i només s'eliminen les caixes de baixa puntuació no emparellades, ja es consideren fons (línes 23-25). Després de l'associació, s'actualitza el KF de les pistes coincidents, s'actualitzen les característiques d'aparença de les pistes i s'eliminen les pistes no emparellades de les seqüències de pistes. Per a  $T_{re-remain}$  després de la segona associació, es posen a  $T_{lost}$ . Per a cada pista a  $T_{lost}$ , només s'elimina si existeix durant més de 30 fotogrames. En cas contrari, es manté  $T_{lost}$  a T (línes 26-28).

Finalment, s'inicialitza noves pistes amb  $D_{remain}$  després de l'associació si la puntuació de detecció és superior al llindar de seguiment  $\eta$ . (línes 29-31). La sortida de cada fotograma individual són les caixes delimitadores i les identitats de les pistes T al fotograma actual.

Opcionalment, es realitza una interpolació lineal de les pistes T, amb un interval màxim de 20, com a fase de postprocessament per compensar les imperfeccions en el *ground truth* (línes 32-33).

<b>Algorithm 1:</b> Pseudo-code of BoT-SORT-ReID.	<pre> <math>\text{Input:}</math> A video sequence V; object detector Det; appearance           (features) extractor Enc; high detection score threshold           <math>\tau</math>; new track score threshold <math>\eta</math> <math>\text{Output:}</math> Tracks <math>\mathcal{T}</math> of the video 1 Initialization: <math>\mathcal{T} \leftarrow \emptyset</math> 2 <b>for</b> frame <math>f_k</math> in <math>V</math> <b>do</b> 3   /* Handle new detections */ 4   <math>\mathcal{D}_k \leftarrow \text{Det}(f_k)</math> 5   <math>\mathcal{D}_{high} \leftarrow \emptyset</math> 6   <math>\mathcal{D}_{low} \leftarrow \emptyset</math> 7   <math>\mathcal{F}_{high} \leftarrow \emptyset</math> 8   <b>for</b> <math>d</math> in <math>\mathcal{D}_k</math> <b>do</b> 9     <b>if</b> <math>d.score &gt; \tau</math> <b>then</b> 10       /* Store high scores detections */ 11       <math>\mathcal{D}_{high} \leftarrow \mathcal{D}_{high} \cup \{d\}</math> 12       /* Extract appearance features */ 13       <math>\mathcal{F}_{high} \leftarrow \mathcal{F}_{high} \cup \text{Enc}(f_k, d.box)</math> 14     <b>else</b> 15       /* Store low scores detections */ 16       <math>\mathcal{D}_{low} \leftarrow \mathcal{D}_{low} \cup \{d\}</math> 17 18   /* Find warp matrix from k-1 to k */ 19   <math>\mathcal{A}_{k-1}^k = \text{findMotion}(f_{k-1}, f_k)</math> 20 21   /* Predict new locations of tracks */ 22   <b>for</b> <math>t</math> in <math>\mathcal{T}</math> <b>do</b> 23     <math>t \leftarrow \text{KalmanFilter}(t)</math> 24     <math>t \leftarrow \text{MotionCompensation}(t, \mathcal{A}_{k-1}^k)</math> 25 26   /* First association */ 27   <math>C_{iou} \leftarrow \text{IOUDist}(\mathcal{T}.boxes, \mathcal{D}_{high})</math> 28   <math>C_{emb} \leftarrow \text{FusionDist}(\mathcal{T}.features, \mathcal{F}_{high}, C_{iou})</math> 29   // Eq. 12 30   <math>C_{high} \leftarrow \min(C_{iou}, C_{emb})</math> // Eq. 13 31   Linear assignment by Hungarian's alg. with <math>C_{high}</math> 32   <math>\mathcal{D}_{remain} \leftarrow \text{remaining object boxes from } \mathcal{D}_{high}</math> 33   <math>\mathcal{T}_{remain} \leftarrow \text{remaining tracks from } \mathcal{T}</math> 34 35   /* Second association */ 36   <math>C_{low} \leftarrow \text{IOUDist}(\mathcal{T}_{remain}.boxes, \mathcal{D}_{low})</math> 37   Linear assignment by Hungarian's alg. with <math>C_{low}</math> 38   <math>\mathcal{T}_{re-remain} \leftarrow \text{remaining tracks from } \mathcal{T}_{remain}</math> 39 40   /* Update matched tracklets */ 41   <math>\mathcal{T} \leftarrow \text{UpdateMatchedTracklets}(\mathcal{T})</math> 42   Update matched tracklets Kalman filter. 43   Update tracklets appearance features. 44 45   /* Delete unmatched tracks */ 46   <math>\mathcal{T} \leftarrow \mathcal{T} \setminus \mathcal{T}_{re-remain}</math> 47 48   /* Initialize new tracks */ 49   <b>for</b> <math>d</math> in <math>\mathcal{D}_{remain}</math> <b>do</b> 50     <b>if</b> <math>d.score &gt; \eta</math> <b>then</b> 51       <math>\mathcal{T} \leftarrow \mathcal{T} \cup \{d\}</math> 52 53   /* (Optional) Offline post-processing */ 54   <math>\mathcal{T} \leftarrow \text{LinearInterpolation}(\mathcal{T})</math> 55 56   <b>Return:</b> <math>\mathcal{T}</math> </pre>
---	---

Figura 2.3: Pseudocodi de l'algoritme BoT-SORT + ReID. Figura de [1]

## OC-SORT

La Figura 2.4 mostra el pseudocodi del *tracker* OC-SORT. L'entrada són les deteccions Z, el KF i un llindar per eliminar pistes no associades  $t_{expire}$ . La sortida són les pistes T, i cada pista conté una caixa delimitadora juntament amb la identitat de l'objecte a cada caixa.

Per a cada pas de temps  $t$ , es realitzen diferents passos. En el pas 1, l'objectiu és que les prediccions de les pistes coincideixin amb les observacions. Primer, s'inicialitzen les matrius  $Z_t$  que representen les observacions,  $X_t$  que representen els estats estimats després d'aplicar el KF, i  $Z$  que conté la trajectòria d'observacions de totes les pistes existents. (línes 3-5). Després, es calcula la matriu de costos  $C_t$  fent servir el terme del mètode OCM i es resol l'assignació lineal mitjançant l'algoritme Hongarès basat en  $C_t$  (línes 6-7).

Finalment, es guarden les pistes associades a una observació a  $T_t^{matched}$ , les pistes que no s'han pogut associar a cap observació a  $T_t^{remain}$  i les observacions que no s'han pogut associar a cap pista a  $Z_t^{remain}$  (línies 8-10).

Al pas 2, l'objectiu és identificar les pistes perdudes mitjançant OCR. S'inicia un segon intent d'associació entre l'última observació de les pistes no associades  $Z^{Tremain}$  i les observacions no associades  $Z_t^{remain}$  utilitzant només  $C_{IoU}$  (línies 11-13). Finalment, es guarden les pistes  $T_t^{remain}$  que s'han associat amb observacions  $Z^{Tremain}$  a  $T_t^{recovery}$ , les observacions  $Z^{Tremain}$  que encara no s'han associat a cap pista a  $Z_t^{unmatched}$ , i les pistes  $T_t^{remain}$  que encara no s'han associat a les observacions; a més, les pistes  $T_t^{matched}$  s'afegeixen  $T_t^{recovery}$  (línies 14-17).

En el pas 3, l'objectiu és actualitzar l'estat de les pistes. Un cop una pista està associada de nou amb una observació després d'un període de no ser rastrejada, es retrocedeix al període en què estava perduda i s'actualitzen els paràmetres del KF. L'actualització es basa en observacions d'una trajectòria virtual  $Z_t^T$  (línies 19-24).

En l'últim pas, s'inicialitzen noves pistes i s'eliminen pistes caducades. Les noves pistes generades a partir de  $Z_t^{unmatched}$  es guarden a  $T_t^{new}$  (línia 30). Per a cada pista  $\tau$  en  $T_t^{unmatched}$ , el comptador de temps sense rastreig  $\tau.untracked$  s'incrementa en 1 (línies 31-34). Finalment, es crea  $T_{reserved}$  que conté les pistes de  $T_t^{unmatched}$  que encara no han superat  $t_{expire}$  i les pistes  $T$  finals es componen de  $T_t^{new}$ ,  $T_t^{matched}$  i  $T_t^{reserved}$  (línies 35-39).

```

n 1: Pseudo-code of OCSORT.
Detections  $Z = \{z_k^t | 1 \leq k \leq T, 1 \leq t \leq N_k\}$ ; Kalman Filter KF; threshold to remove untracked tracks  $t_{expire}$ 
: The set of tracks  $\mathcal{T} = \{\tau_i\}$ 
: union:  $\mathcal{T} \leftarrow \emptyset$  and KF;
step  $t \leftarrow 1 : T$  do
Step 1: match track prediction with observations */
 $- [z_1^t, \dots, z_{N_t}^t]^T$  /* Observations */
 $- [x_1^t, \dots, x_{N_t}^t]^T$  from  $\mathcal{T}$  /* Estimations by KF.predict */
-Historical observations of the existing tracks
 $- C_{tot}(\hat{X}_t, Z_t) + \lambda C_r(Z, Z_t)$  /* Cost Matrix with OCM term */
ar assignment by Hungarians with cost  $C_t$ 
chd  $\leftarrow$  tracks matched to an observation
unm  $\leftarrow$  tracks not matched to any observation
amn  $\leftarrow$  observations not matched to any track

Step 2: perform OCR to find lost tracks back */
 $\tau_{un} \leftarrow$  last matched observations of tracks in  $T_t^{remain}$ 
uin  $\leftarrow C_{HU}(Z_t^{remain}, Z_t^{remain})$ 
ar assignment by Hungarians with cost  $C_t^{remain}$ 
very  $\leftarrow$  tracks from  $T_t^{remain}$  and matched to observations in  $Z_t^{remain}$ 
watched  $\leftarrow$  observations from  $Z_t^{remain}$  that are still unmatched to tracks
satched  $\leftarrow$  tracks from  $T_t^{remain}$  that are still unmatched to observations
chd  $\leftarrow \{T_t^{matched}, T_t^{recovery}\}$ 

for  $\tau$  in  $T_t^{matched}$  do
    if  $\tau.tracked = False$  then
        /* Perform ORU for track from untracked to tracked */
         $z_{\tau,t}, t' \leftarrow$  The last observation matched to  $\tau$  and the time step
        Rollback KF parameters to  $t'$ 
        /* Generate virtual observation trajectory */
         $Z_t^\tau \leftarrow [z_{\tau,t+1}^t, \dots, z_{\tau,T}^t]$ 
        Online smooth KF parameters along  $Z_t^\tau$ 
    end
     $\tau.tracked = True$ 
     $\tau.untracked = 0$ 
    Append the new matched associated observation  $z_\tau^t$  to  $\tau$ 's observation history
    Update KF parameters for  $\tau$  by  $z_\tau^t$ 
end

/* Step 3: update status of matched tracks */
for  $\tau$  in  $T_t^{matched}$  do
    if  $\tau.untracked = False$  then
        /* Perform ORU for track from untracked to tracked */
         $z_{\tau,t}, t' \leftarrow$  The last observation matched to  $\tau$  and the time step
        Rollback KF parameters to  $t'$ 
        /* Generate virtual observation trajectory */
         $Z_t^\tau \leftarrow [z_{\tau,t+1}^t, \dots, z_{\tau,T}^t]$ 
        Online smooth KF parameters along  $Z_t^\tau$ 
    end
     $\tau.tracked = True$ 
     $\tau.untracked = 0$ 
    Append the new matched associated observation  $z_\tau^t$  to  $\tau$ 's observation history
    Update KF parameters for  $\tau$  by  $z_\tau^t$ 
end

/* Step 4: initialize new tracks and remove expired tracks */
 $T_t^{new} \leftarrow$  new tracks generated from  $Z_t^{unmatched}$ 
for  $\tau$  in  $T_t^{unmatched}$  do
    if  $\tau.tracked = False$ 
        if  $\tau.untracked = True$ 
             $\tau.untracked = \tau.untracked + 1$ 
        end
    end
     $T_t^{reserved} \leftarrow \{\tau | \tau \in T_t^{unmatched} \text{ and } \tau.untracked < t_{expire}\}$  /* remove expired unmatched tracks */
     $\mathcal{T} \leftarrow \{T_t^{new}, T_t^{matched}, T_t^{reserved}\}$  /* Conclude */
end
 $\mathcal{T} \leftarrow$  Postprocess( $\mathcal{T}$ ) /* [Optional] offline post-processing */
Return:  $\mathcal{T}$ 

```

Figura 2.4: Pseudocodi del mètode OCSORT. Figura de [3]

## CenterTrack

La Figura 2.5 presenta el pseudocodi del *tracker* CenterTrack. L'entrada es compon pels objectes rastrejats en el fotograma anterior,  $T^{(t-1)}$ , amb centre  $p$ , els pics del mapa de calor  $B^{(t)}$  amb desplaçament  $d$  al fotograma actual, ordenats per confiança descendent, i les deteccions  $D^{(t)}$ . La sortida és el conjunt d'objectes rastrejats en el fotograma actual,  $T^{(t)}$ .

S'inicialitzen tant  $T^{(t)}$  com el conjunt de pistes associades  $S$  (línies 1-3). Posteriorment, es calcula la matriu de costos  $W$  entre  $B^{(t)}$  i  $T^{(t-1)}$  (línies 4-5).

S'utilitza un algoritme de coincidència *greedy* per associar els objectes al llarg del temps. Per a cada detecció a la posició  $p$ , s'associa amb la detecció no associada més propera a la posició  $p-D_p$ , en ordre de confiança  $w$  descendent. El valor de  $k$  es defineix com una mitjana geomètrica de l'amplada i de l'alçada de la caixa delimitadora predicta per a cada pista. Si no hi ha cap detecció anterior sense associar dins d'un radi  $k$ , es crea una nova pista. (línies 6-25).

---

**Algorithm 1:** Private Detection

---

**Input :**  $T^{(t-1)} = \{(\mathbf{p}, \mathbf{s}, id)_j^{(t-1)}\}_{j=1}^M$ : Tracked objects in the previous frame, with center  $\mathbf{p}$ , size  $\mathbf{s} = (w, h)$ .  
 $B^{(t)} = \{(\hat{\mathbf{p}}, \hat{\mathbf{d}})_i^{(t)}\}_{i=1}^N$ : Heatmap peaks with offset  $\hat{\mathbf{d}}$  in the current frame, sorted in descending confidence.

**Output:**  $T^{(t)} = \{(\mathbf{p}, \mathbf{s}, id)_i^{(t)}\}_{i=1}^N$ : Tracked objects in the current frame.

1 // Initialization:  $T^{(t)}$  and  $S$  are initialized as empty lists.  
2  $T^{(t)} \leftarrow \emptyset$   
3  $S \leftarrow \emptyset$  // Set of matched tracks  
4  $W \leftarrow Cost(B^{(t)}, T^{(t-1)})$ //  
 $W_{ij} = ||\hat{\mathbf{p}}_i^{(t)} - \hat{\mathbf{d}}_i^{(t)}, \mathbf{p}_j^{(t-1)}||_2$

5

---

6 **for**  $i \leftarrow 1$  to  $N$  **do**  
7      $j \leftarrow \arg \min_{j \notin S} W_{ij}$   
8     // calculate the distance threshold  $\kappa$   
9      $\kappa \leftarrow \min(\sqrt{\hat{w}_i \cdot h_i}, \sqrt{w_j \cdot h_j})$   
10     // if the cost is smaller the threshold.  
11     **if**  $w_{ij} < \kappa$  **then**  
12         // Propagate matched id  
13          $T^{(t)} \leftarrow T^{(t)} \cup (\hat{\mathbf{p}}_i^{(t)}, \hat{\mathbf{s}}_i^{(t)}, id_j^{(t-1)})$   
14          $S \leftarrow S \cup \{j\}$  // Mark track j as matched  
15     **end**  
16     **else**  
17     |  
18     |  
19     |     // Create a new track.  
20     |      $T^{(t)} \leftarrow T^{(t)} \cup (\hat{\mathbf{p}}_i^{(t)}, \hat{\mathbf{s}}_i^{(t)}, NewId)$   
21     |  
22     **end**  
23     **end**  
24 **end**  
25 **Return:**  $T^{(t)}$

---

Figura 2.5: Pseudocodi del mètode CenterTrack. Figura de [29]

## Anàlisi comparativa dels resultats dels mètodes de seguiment d'objectes

La Taula 2 mostra els resultats de diversos *trackers* utilitzats pels diferents mètodes. Per a cada *tracker*, s'indica si s'utilitza o no un model de Re-ID, quins mètodes utilitzen el *tracker*, el nombre total d'ID obtinguts per cada mètode en comparació amb els ID reals i mètriques com MT, PT, ML.

Aquestes mètriques avaluen el comportament de les trajectòries reals en cada algoritme de seguiment, classificant-les com majoritàriament rastrejada (MT), parcialment rastrejada (PT) o majoritàriament perduda (ML). Un objecte es considera MT si es rastreja amb èxit durant almenys el 80% de la seva vida útil, ML si es rastreja amb èxit com a màxim el 20%, mentre que tots els altres objectes es consideren PT.

Els *trackers* que incorporen un model Re-ID assignen un nombre més baix d'ID, indicant que aquests models poden ajudar a reduir la quantitat d'ID assignats i permetre seguir els objectes durant més temps, millorant així el rendiment del sistema de seguiment. No obstant això, l'eficàcia del Re-ID únicament és notable en el *tracker* de DeepSORT, ja que aconsegueix el major nombre de trajectòries MT i el menor nombre de trajectòries ML. ració amb altres mètodes que també utilitzen Re-ID.

En canvi, altres models Re-ID utilitzats en mètodes com FairMOT i JDE, que formen part dels mètodes de seguiment de tipus JDE, redueixen el nombre d'ID en comparació amb aquells que no incorporen models Re-ID. No obstant això, aquests models són menys efectius, ja que augmenten el nombre de trajectòries ML i redueixen el nombre de trajectòries MT.

Això suggereix que els models JDE, que combinen les tasques de detecció i l'associació dels objectes mitjançant un únic model d'incrustació, són menys eficients que els models SDE com BoT-SORT, OC-SORT i ByteTrack, els quals realitzen aquestes tasques de manera separada. Tot i no incorporar un model Re-ID, aquests últims obtenen resultats més satisfactoris.

A més, les deteccions també tenen un impacte en l'eficiència del mètode de seguiment. En *trackers* com BYTE o OC-SORT, que no utilitzen un model de Re-ID, la selecció del detector té una gran influència, com es pot veure en la Taula 2 amb els diferents mètodes. En alguns casos, si es detecten molts jugadors, els mètodes de seguiment poden tenir dificultats per assignar el mateix ID a un jugador en diferents instàncies, com es veu en els mètodes ByteTrack 3 i OC-SORT 1. D'altra banda, si es detecten pocs jugadors, com en el cas dels mètodes ByteTrack 2 i OC-SORT 2, és probable que en alguns fotogrames no es detecti un jugador determinat i, per tant, el *tracker* assignarà un nou ID quan reaparegui en fotogrames posteriors.

Taula 2: Resultats dels *trackers* utilitzats pels diferents mètodes en el conjunt de test.

Tracker	Re-ID	Mètode	ID totals/ ID reals (↓)	MT	PT	ML
DeepSORT	PCB Pyramid Embedding [28]	DeepSORT 1	<b>3768/1935</b>	1130	50	755
	PPLCNet Embedding [4]	DeepSORT 2	3780/1935	1129	52	754
	ResNet Embedding	DeepSORT 3	3822/1935	1129	51	755
JDE	FairMOT EmbeddingHead	FairMOT1	5960/1935	663	472	827
	JDE EmbeddingHead	JDE	6379/1935	441	616	878
BYTE	No	ByteTrack 1	17896/1935	987	130	818
	No	ByteTrack 2	15258/1935	586	512	837
	No	ByteTrack 3	38074/1935	896	223	816
	FairMOT EmbeddingHead	FairMOT 2	4364/1935	703	405	827
BoT-SORT	No	BoT-SORT	18893/1935	890	228	817
OC-SORT	No	OC-SORT 1	13209/1935	931	182	822
	No	OC-SORT 2	7312/1935	662	448	825
CenterTracker	No	CenterTrack	30645/1935	708	383	844

### 3. StrongSORT

Tal com s'ha demostrat en aquest treball, StrongSORT [5] és el mètode amb els millors resultats. Aquest model està compost per diversos mòduls i paràmetres, i en aquesta secció s'analitzaran alguns d'aquests mòduls i la possible modificació dels paràmetres per investigar si es poden millorar encara més els resultats.

#### ECC

StrongSORT utilitza el mòdul ECC [6] per abordar el soroll de moviment causat pel moviment de la càmera, aquesta secció comprovarà si aquest mòdul realment millora els resultats o, per contra, introduceix distorsions a les deteccions.

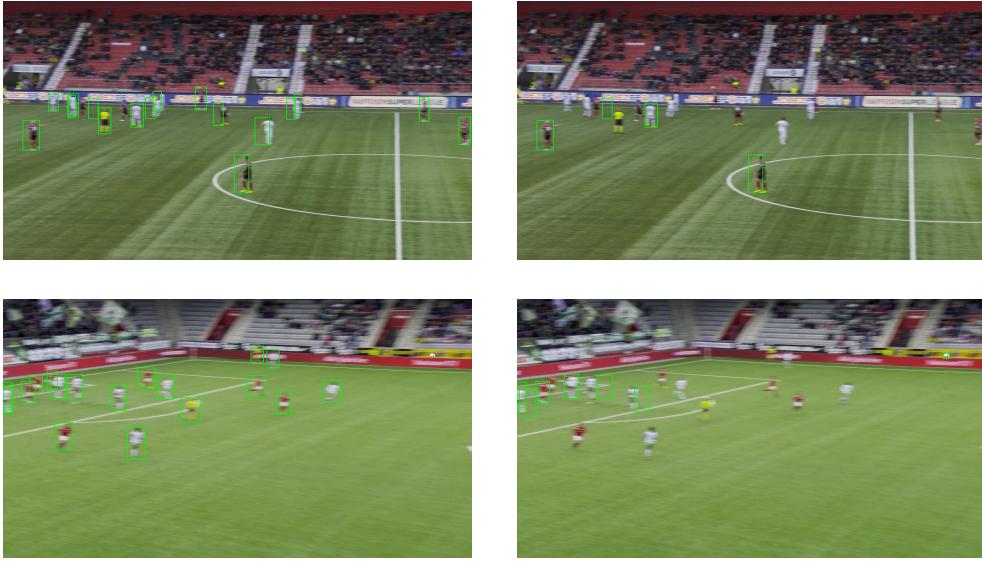
La Taula 3.1 presenta els resultats de la implementació de StrongSORT i StrongSORT++ sense utilitzar el mòdul ECC.

Taula 3.1: Resultats de StrongSORT i StrongSORT++ aplicant i sense aplicar ECC.

Mètode	HOTA ( $\uparrow$ )	DetA ( $\uparrow$ )	AssA ( $\uparrow$ )	Detections totals/ Detections reals	ID totals/ ID reals( $\downarrow$ )
StrongSORT (sense ECC)	79,445	91,854	68,712	544.860/564.547	3328/1935
StrongSORT	81,972	<b>92,22</b>	72,872	547.497/564.547	2836/1935
StrongSORT++ (GSI i sense ECC)	75,48	85,017	67,169	554.018/564.547	3328/1935
StrongSORT++ (GSI)	78,002	85,621	71,193	553.993/564.547	2836/1935
StrongSORT++ (AFLink i sense ECC)	79,808	91,852	69,344	544.859/564.547	3216/1935
StrongSORT++ (AFLink)	<b>82,265</b>	92,219	<b>73,385</b>	547.497/564.547	<b>2766/1935</b>

Es pot observar que la puntuació HOTA disminueix significativament en absència d'aquest mòdul. En general, el nombre de deteccions és menor i el nombre d'ID augmenta, i, per tant, l'associació es veu afectada negativament.

La Figura 3.1 mostra una comparació visual de les deteccions en diversos fotogrames amb i sense l'ús del mòdul ECC amb StrongSORT + AFLink. En les imatges (a), es pot veure que la majoria dels jugadors es detecten en fotogrames on estan borrosos. D'altra banda, en les imatges (b), no es detecten tots els jugadors i les caixes delimitadores no s'ajusten correctament als jugadors. Per tant, es pot concloure que l'ús del model ECC permet extreure informació de moviment més precisa i millora els resultats.



(a) Deteccions amb el mòdul ECC

(b) Deteccions sense el mòdul ECC

Figura 3.1: Comparació visual de les deteccions amb i sense l'aplicació del mòdul ECC.

## Modificació de paràmetres

S'han modificat dos paràmetres per millorar els resultats d'aquest mètode. En primer lloc, s'ha augmentat el factor de pes  $\lambda$  del cost de moviment de 0,98 a 0,995. A mesura que  $\lambda$  s'apropa a 1, es dóna més pes a la informació d'aparença que a la de moviment. Tal com es pot veure a la Taula 3.2, aquesta modificació ha comportat una millora en l'associació global, ja que s'observa un augment en la puntuació HOTA (+ 0,1). Això es deu a un lleuger augment en la puntuació AssA (+ 0,18), que s'ha traduït en una reducció del nombre de deteccions i del nombre total d'ID. Per contra, utilitzar un valor més petit per a  $\lambda$ , com ara 0,2, dóna més pes a la informació de moviment, la qual cosa redueix la puntuació AssA (-3,46) a causa de l'assignació del doble d'ID.

Per altra banda, s'ha ajustat el paràmetre *max\_distance*, augmentant el valor inicial de 0,7 a 0,9. Aquest paràmetre determina el llindar de filtratge, on les associacions amb un cost superior són ignorades. El nou valor de 0,9 ha comportat una millora notable dels resultats tant en la puntuació de detecció com en l'associació, provocant un augment significatiu en la puntuació HOTA (+1,32), com es pot observar a la Taula 3.2.

Taula 3.2: Resultats amb diferents valors de paràmetres.

Mètode	HOTA ( $\uparrow$ )	DetA ( $\uparrow$ )	AssA ( $\uparrow$ )	Deteccions totals/ Deteccions reals	ID totals/ ID reals ( $\downarrow$ )
StrongSORT ++ (AFLink)	82,265	92,219	73,385	547.497/564.547	2766/1935
StrongSORT ++ (AFLink) $\lambda = 0,995$	82,367	92,217	73,569	547.457/564.547	2763/1935
StrongSORT ++ (AFLink) $\lambda = 0,2$	80,499	92,68	69,92	550.992/564.547	4984/1935
StrongSORT ++ (AFLink) $\lambda = 0,995$ $\text{max\_distance} = 0,9$	<b>83,586</b>	<b>93,786</b>	<b>74,497</b>	552.907/564.547	<b>2577/1935</b>

## 4. Entrenament d'un detector

Primerament, les deteccions generades pels dos mètodes (Background Subtraction i Human Pose Estimation) s'han convertit en format YOLO, cada imatge de cada seqüència ha de tenir un fitxer separat que contingui totes les deteccions per aquell fotograma amb les coordenades de les caixes delimitadores normalitzades. Després, s'ha entrenat el detector YOLOv8x durant 100 èpoques utilitzant la plataforma d'Ultralytics [22]. Totes les deteccions han estat assignades a una única classe anomenada *person*.

La Figura 4.1 mostra diverses mètriques resultants a partir de l'entrenament del detector. La precisió mesura el nombre d'objectes detectats correctament, indicant quantes deteccions van ser correctes. Es calcula dividint els TP per la suma dels TP i FP. El recall és la capacitat del model per identificar totes les instàncies dels objectes a les imatges. Es calcula dividint els TP per la suma dels TP i FN [24]. Es pot observar com els valors de recall i precisió augmenten al llarg de les diferents èpoques fins a assolir uns valors màxims de 0,489 i 0,625 respectivament.

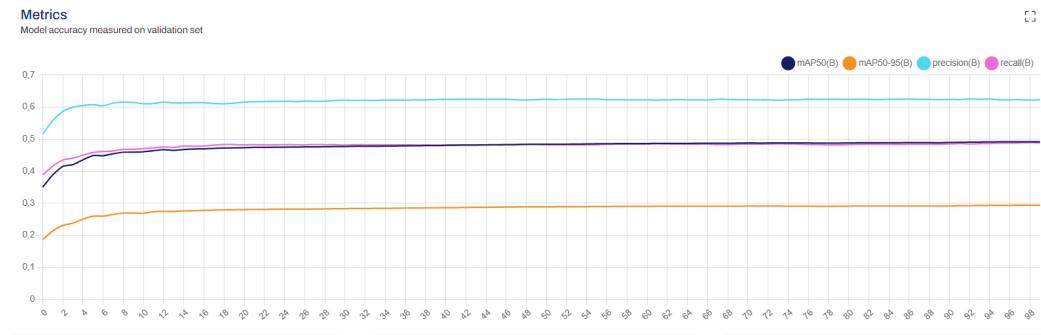


Figura 4.1: Progressió de mètriques durant l'entrenament en 100 èpoques.

Després de completar l'entrenament, s'ha passat a la fase de seguiment d'objectes. Ultralytics ofereix dos models de seguiment: Bot-SORT i ByteTrack, els quals ja s'ha discutit en aquest treball. Per avaluar l'entrenament, s'han realitzat diverses proves utilitzant els pesos de diferents èpoques per observar la progressió, com es mostra a la Taula 4.1. En aquesta taula es mostren els resultats en el conjunt de test utilitzant el *tracker* de BoT-SORT.

En la Taula 4.1 es pot observar que a mesura que les èpoques augmenten, també ho fa el nombre de deteccions, indicant que el detector està aprenent correctament. No obstant això, un gran nombre de deteccions resulta en un augment del nombre d'IDs, fet que fa que la puntuació HOTA disminueixi, ja que les trajectòries que estaven majoritàriament identificades ara estan parcialment identificades. En aquest cas, l'augment d'IDs es deu al *tracker* de BoT-SORT. Si s'utilitzés el *tracker* de DeepSORT o StrongSORT, és probable que el nombre d'IDs disminuís.

Taula 4.1: Resultats BoT-SORT en diferents èpoques d'entrenament.

Època	HOTA ( $\uparrow$ )	MT	PT	ML	ID totals/ ID reals ( $\downarrow$ )	Deteccions totals/ Deteccions reals
8	27.369	134	705	1096	<b>3044/1935</b>	357.670/564.547
12	<b>27.525</b>	137	704	1094	3115/1935	361.765/564.547
19	27.207	125	720	1090	3209/1935	373.541/564.547
23	27.277	124	722	1089	3233/1935	377.177/564.547
25	27.196	126	720	1089	3253/1935	377.198/564.547
28	27.088	128	720	1087	3281/1935	377.232/564.547
37	27.344	128	715	1092	3320/1935	379.789/564.547
42	27.249	131	717	1087	3344/1935	382.036/564.547
64	26.703	119	722	1094	3478/1935	384.353/564.547
94	26.394	116	732	1087	3652/1935	389.363/564.547

Les Figures 4.2 i 4.3 il·lustren l'increment de les deteccions al llarg de les diferents èpoques en escenaris borrosos i escenaris normals.

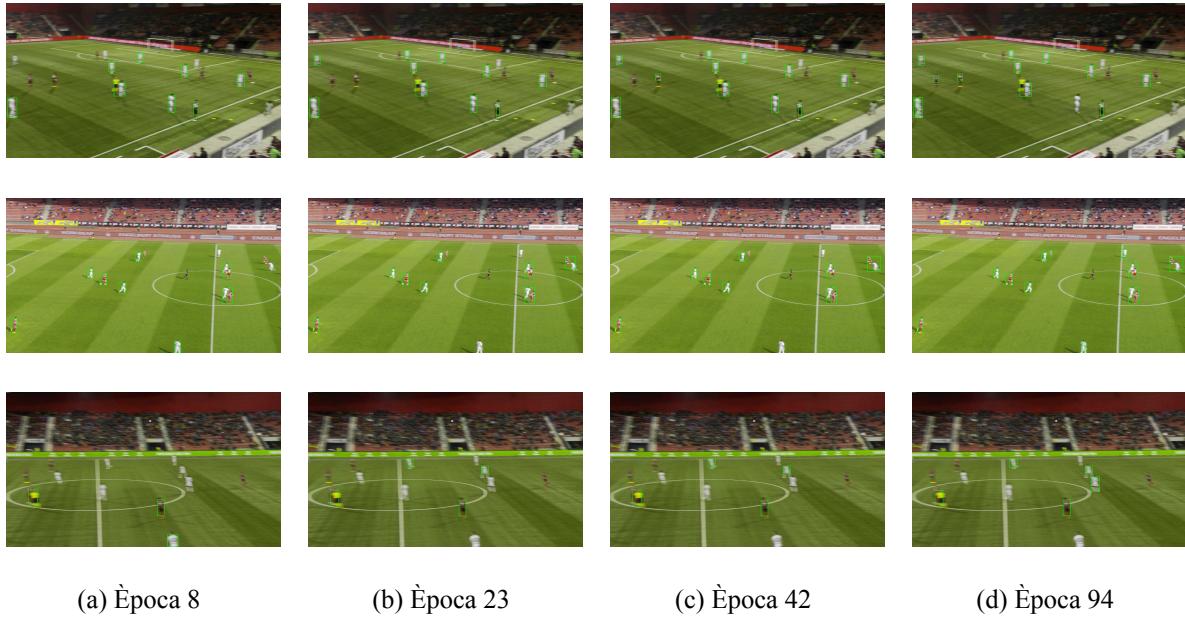


Figura 4.2: Deteccions resultants per a diferents èpoques en escenes borroses.



(a) Època 8

(b) Època 23

(c) Època 42

(d) Època 94

Figura 4.3: Deteccions resultants per a diferents èpoques en diferents escenes

Després d'analitzar l'entrenament, es pot afirmar que ha estat adequat, però encara hi ha marge per millorar els resultats de manera significativa. Una manera d'aconseguir aquesta millora seria ampliant el conjunt de dades d'entrenament. En lloc de limitar-se al conjunt d'entrenament de SoccerNet, es podrien incloure els dos conjunts de *challenges* o altres conjunts de dades on hi hagi seqüències de futbol per obtenir una major diversitat de partits. Una altra millora possible seria classificar els objectes en classes separades, com ara separar la pilota en una classe i els jugadors en una altra. Per últim, per aconseguir un seguiment més precís, seria beneficiós ajustar els pesos d'entrenament utilitzant mètodes com DeepSORT i StrongSORT, ja que s'ha demostrat que s'adapten millor al conjunt de dades seleccionat en aquest treball.

## Referències

- [1] Aharon, N., Orfaig, R., & Bobrovsky, B. Z. (2022). BoT-SORT: Robust associations multi-pedestrian tracking. Recuperat de <https://arxiv.org/pdf/2206.14651v2.pdf>
- [2] Bodla, N., Singh, B., Chellappa, R., & Davis, L. S. (2017). Soft-NMS—improving object detection with one line of code. *Proceedings of the IEEE international conference on computer vision* (p. 5561-5569).
- [3] Cao, J., Pang, J., Weng, X., Khirodkar, R., & Kitani, K. (2023). Observation-centric sort: Rethinking sort for robust multi-object tracking. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (p. 9686-9696).
- [4] Cui, C., Gao, T., Wei, S., Du, Y., Guo, R., Dong, S., ... & Ma, Y. (2021). PP-LCNet. A lightweight CPU convolutional network. Recuperat de <https://arxiv.org/pdf/2109.15099>
- [5] Du, Y., Zhao, Z., Song, Y., Su, F., Gong, T., & Meng, H. (2023). Strongsort: Make deepsort great again. *IEEE Transactions on Multimedia*. Recuperat de <https://arxiv.org/pdf/2202.13514>
- [6] Evangelidis, G. D., & Psarakis, E. Z. (2008). Parametric image alignment using enhanced correlation coefficient maximization. *IEEE transactions on pattern analysis and machine intelligence*, 30(10), 1858-1865.
- [7] Feng, C., Zhong, Y., Gao, Y., Scott, M. R., & Huang, W. (2021). Tood: Task-aligned one-stage object detection. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (p. 3490-3499). IEEE Computer Society.
- [8] Gabriel Mongaras (2022). *YOLOX explanation — How does YOLOX work?*. Recuperat 30 d'agost del 2023 , des de <https://gmongaras.medium.com/yolox-explanation-how-does-yolox-work-3e5c89f2bf78>
- [9] Ge, Z., Liu, S., Z., Yoshie, O., & Sun, J. (2021). Ota: Optimal transport assignment for object detection. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (p. 303-312).
- [10] Ge, Z., Liu, S., Wang, F., Li, Z., & Sun, J. (2021). Yolox: Exceeding yolo series in 2021. Recuperat de <https://arxiv.org/pdf/2107.08430>
- [11] He, T., Zhang, Z., Zhang, H., Xie, J., & Li, M. (2019). Bag of tricks for image classification with convolutional neural networks. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (p. 558-567).
- [12] Li, X., Eng, W., Wu, L., Chen, S., Hu, X., Li, J., ... & Yang, J. (2020). Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Advances in Neural Information Processing Systems*, 33, 21002-21012.
- [13] Lin, T. Y., Dollar, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. *Proceedings of the IEEE conferences on computer vision and pattern recognition* (p. 2117-2125).
- [14] Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path aggregation network for instance segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition* (p. 8759-8768).

- [15] PaddlePaddle (2019). PaddleDetection: Object Detection toolkit based on PaddlePaddle. It supports object detection, instance segmentation, multiple object tracking and real-time multi-person keypoint detection. Recuperat 6 d'octubre del 2023, des de <https://github.com/PaddlePaddle/PaddleDetection>
- [16] Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. *Proceedings of the IEEE conference on computer vision and pattern recognition* (p. 7263-7271).
- [17] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. Recuperat de <https://arxiv.org/pdf/1804.02767v1>
- [18] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28. Recuperat de <https://arxiv.org/pdf/1506.01497>
- [19] Roboflow. (s.d.). *Roboflow Annotate: Label Images Faster Than Ever*. Recuperat 23 de maig del 2024, des de <https://roboflow.com/annotate>
- [20] Terven, J., Córdova-Esparza, D. M., & Romero-González, J. A. (2023). A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction*, 5(4), 1680-1716. Recuperat de <https://arxiv.org/pdf/2304.00501>
- [21] Tian, Z., Shen, Chunhua, S., Chen, H., He, T. (2019). Fcos: Fully convolutional one-stage object detection. *Proceedings of the IEEE/CVF international conference on computer vision* (p. 9627-9636).
- [22] Ultralytics. (2023). *YOLO Performance Metrics - Ultralytics YOLO Docs*. Recuperat 24 de maig del 2024, des de <https://docs.ultralytics.com/guides/yolo-performance-metrics/>
- [23] Wojke, N., Bewley, A., & Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. *2017 IEEE international conference on image processing (ICIP)* (p.3645-3649). IEEE
- [24] Xu, S., Wang, X., Lv, W., Chang, Q., Cui, C., Deng, K., Wang, G., Dang, Q., Wei, S., Du, Y., & Lai, B. (2022). PP-YOLOE: An evolved version of YOLO. Recuperat de <https://arxiv.org/pdf/2203.16250>
- [25] Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. Recuperat de <https://arxiv.org/pdf/1710.09412>
- [26] Zhang, H., Wang, Y., Dayoub, F., & Sunderhauf, N. (2021). Varifocalnet: An iou-aware dense object detector. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (p. 8514-8523).
- [27] Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., & Wang, X. (2022). ByteTrack: Multi-object tracking by associating every detection box. *European conference on computer vision* (p. 1-21). Cham: Springer Nature Switzerland.
- [28] Zheng, F., Deng, C., Sun, X., Jiang, X., Guo, X., Yu, Z., ... & Ji, R. (2019). Pyramidal re-identification via multi-loss dynamic training. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (p. 8514-8522).
- [29] Zhou, X., Koltun, V., & Krähenbühl, P. (2020). Tracking objects as points. *European conference on computer vision* (p. 474-490). Cham: Springer International Publishing.

- [30] Zhou, X., Wang, D., & Krähenbühl, P. (2019). Objects as points. Recuperat de <https://arxiv.org/pdf/1904.07850>