
This is the **published version** of the bachelor thesis:

Britos Silveiro, Pablo Ezequiel; Ponsa Mussarra, Daniel, dir. Seguiment d'animals salvatges en seqüències d'imatges infraroges. 2024. (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/290102>

under the terms of the  license

Seguiment d'animals salvatges en seqüències d'imatges infraroges

Pablo Ezequiel Britos Silveiro

Resum—El cos d'agents rurals de la Generalitat realitza censos de fauna, en una tasca que els requereix una anàlisi visual d'imatges tèrmiques llarg i tediós. Per facilitar aquesta feina, en aquest projecte s'ha desenvolupat un sistema de seguiment d'animals en seqüències d'imatges tèrmiques. S'ha seleccionat el conjunt de dades BIRDSAI, que inclou imatges infraroges amb animals de diverses mides, alguns d'ells molt petits. S'ha utilitzat l'estratègia "track by detection", la qual consisteix a fer servir les deteccions realitzades per un detector per fer el seguiment dels objectes mitjançant un algoritme de seguiment. S'han utilitzat algorismes que estan dins de l'estat de l'art: per la detecció de la fauna s'ha utilitzat el detector YOLOv8, mentre que per al seguiment s'han seleccionat dos algorismes: ByteTrack i BoT-Sort. S'han realitzat diferents experiments per tal d'obtenir el millor rendiment del sistema proposat. Els resultats indiquen que s'ha obtingut un rendiment similar en la part de detecció i una millora en la part de seguiment, en comparació amb els resultats obtinguts pels autors de BIRDSAI [26]. BoT-Sort ha demostrat un rendiment superior a ByteTrack en el seguiment d'animals de diferents mides en imatges infraroges.

Paraules clau—Detecció, Seguiment, BIRDSAI, YOLOv8, Bytetrack, BoT-Sort, Animals Salvatges, Molt Petits, Imatges Tèrmiques.

Abstract—The rural agents of the Generalitat conduct wildlife censuses, engaging in a task that involves a lengthy and tedious visual analysis of thermal images. To streamline this process, a system for tracking animals in thermal image sequences has been developed in this project. The BIRDSAI dataset has been selected for this purpose, it comprises infrared images featuring animals of varying sizes, including some very small. The "track by detection" strategy has been employed, entailing the use of detections performed by a detector to facilitate object tracking through a tracking algorithm. State-of-the-art algorithms have been employed: YOLOv8 has been utilized for fauna detection, while two algorithms, ByteTrack and BoT-Sort, have been chosen for tracking. Several experiments have been conducted to optimize the performance of the proposed system. The results indicate a comparable performance in the detection aspect and an enhancement in the tracking component, in contrast to the outcomes reported by the authors of BIRDSAI [26]. BoT-Sort has demonstrated superior performance to ByteTrack in tracking animals of diverse sizes in infrared images.

Index Terms—Detection, Tracking, BIRDSAI, YOLOv8, ByteTrack, BoT-Sort, Wildlife, Very Small, Thermal Images.



1 INTRODUCCIÓ - CONTEXT DEL TREBALL

EL cos d'agents rurals de la Generalitat vetlla per la conservació de la natura i la protecció del medi ambient, així com la defensa del patrimoni natural. Una de les seves tasques és el monitoratge i el recompte de la població de diverses espècies de fauna salvatge. Aquesta tasca es realitza mitjançant la inspecció visual d'imatges tèrmiques captades per drons. Els animals es veuen des de força distància com petites taques de major intensitat que el seu entorn amb característiques poc definides (dins de zones de 10x10 píxels). Sovint hi ha altres elements a l'escena d'aparença similar, que generen falses alarmes. El seguiment d'objectes petits en imatges tèrmiques pot convertir-se en una tasca complicada, ja que aquests ob-

jectes solen presentar una aparença feble i característiques poc definides. A més, tenen més tendència a perdre's en comparació amb els objectes de mida normal.

En treballs previs [1], [2], [3] s'ha explorat l'ús d'un seguidor que utilitza un detector basat en visió per computador per identificar regions a les imatges on podria trobar-se un animal, permetent així el seu seguiment. El seguiment dels animals detectats al llarg del temps pot, d'una banda, ajudar a descartar les falses alarmes generades pel detector basat en visió i, d'altra banda, prevenir que un mateix animal sigui detectat repetidament, ja que es pot moure significativament durant el temps que s'inspecciona una àrea determinada, que pot estendre's fins a un llarg període de temps. En aquest treball, s'ha desenvolupat un sistema de seguiments d'animals, per tal d'ajudar a millorar l'eficiència de la feina dels agents rurals.

- E-mail de contacte: pabloezequiel.britos@gmail.com
- Menció realitzada: Computació
- Treball tutoritzat per: Daniel Ponsa Mussarra (Computació)
- Curs 2023/24

2 OBJECTIUS

L'objectiu d'aquest projecte és desenvolupar un sistema de seguiment d'animals en seqüències d'imatges tèrmiques, mitjançant tècniques d'aprenentatge computacional, que faciliti el cens d'animals salvatges per als agents rurals. Per assolir-ho, s'han seleccionat algorismes de seguiment basats en la detecció d'objectes i s'ha quantificat el rendiment, amb l'objectiu de validar si poden contribuir de manera efectiva al sistema desitjat.

3 METODOLOGIA

La metodologia seguida en aquest treball és l'estàndard en l'àmbit de la recerca científica: identificar i revisar els treballs científics publicats en l'àmbit d'interès, recopilar els conjunts de dades que s'utilitzen en aquests treballs, i finalment implementar i quantificar objectivament el rendiment de les diferents propostes. Específicament, les tasques principals que s'han realitzat són les següents:

- Revisió de l'estat de l'art d'algoritmes de detecció i seguiment d'objectes petits en imatges tèrmiques i selecció dels que millor s'ajustin al treball.
- Recopilació de dades: Recopilació d'un conjunt divers i extens de dades d'objectes petits en imatges tèrmiques.
- Aplicació d'algoritmes de detecció i seguiment d'objectes: Implementació dels diferents algorismes de detecció i seguiment d'objectes seleccionats i entrenament del sistema amb les dades recopilades.
- Validació i avaluació: Avaluació del sistema utilitzant conjunts de dades de prova que no s'han utilitzat prèviament en l'entrenament.
- Optimització i ajustament: Ajustament dels hiperparàmetres del model i realització d'optimitzacions per millorar el rendiment i la robustesa del sistema.
- Anàlisi i conclusions: Avaluació dels algorismes utilitzats i comparació dels resultats amb treballs previs per obtenir conclusions.

4 PLANIFICACIÓ DEL TREBALL

En la secció A de l'Apèndix es pot trobar un diagrama de Gantt que reflecteix la planificació del treball realitzat. En aquest diagrama, les files representen les tasques amb les seves hores corresponents i les columnes representen els dies, agrupats per setmanes. La suma total de les hores és de 300 hores.

5 ESTAT DE L'ART

El seguiment de múltiples objectes, conegut per les seves sigles en anglès com a MOT, és un desafiament important en el camp de la visió per computador. Consisteix a identificar i rastrejar la posició de diversos objectes al llarg d'una seqüència de vídeo. L'objectiu no és només detectar els objectes d'interès en cada fotograma, sinó també mantenir un seguiment del seu moviment al llarg del temps. Els treballs [4], [5], [6], [7], [8], [9] realitzen una revisió dels mètodes aplicats al MOT, on s'identifica la tècnica *track by detection*, que consta d'una seqüència de dues activitats principals: detecció i seguiment.

5.1 DETECCIÓ

En aquesta primera fase es detecten els objectes presents en una imatge o fotograma. Actualment, les Xarxes Neuronals Convolucionals (CNN) són excel·lents per captar característiques i superen els mètodes tradicionals en termes de retenció d'informació característica. Els detectors d'objectes es poden dividir en dos tipus [10]: *Two-Shot Detection* (TSD) i *Single-Shot Detection* (SSD). TSD utilitza dues etapes separades per detectar objectes, mentre que SSD utilitza una xarxa neuronal per a aquest propòsit en una única etapa. TSD produeix propostes de regions en una primera etapa i les classifica i ajusta en una segona per millorar la precisió. TSD és més precís que SSD, mentre que SSD és més ràpid que TSD. L'article [4] menciona que entre els detectors d'objectes més destacats es troben R-FCN [11], SSD [12], Faster R-CNN [13] i la sèrie YOLO [14], que tenen un rendiment sòlid en la detecció d'objectes.

Els detectors Faster R-CNN i R-FCN són del tipus TSD. El detector Faster R-CNN utilitza una CNN per detectar objectes en una imatge utilitzant una tècnica anomenada *Region Proposal Network* (RPN). La RPN genera regions candidates a la imatge que poden contenir objectes i després aquestes propostes de regions s'utilitzen per extreure característiques de la imatge i classificar els objectes. R-FCN també utilitza una xarxa CNN per detectar objectes en una imatge. La xarxa CNN s'entrena per generar una sortida de mapa de característiques que s'utilitza per identificar les regions d'interès (ROI) a la imatge. Posteriorment, s'aplica una capa de ROI *pooling* per extreure característiques de cada ROI i classificar l'objecte de la ROI.

Respecte als detectors del tipus SSD, el detector YOLO (You Only Look Once) destaca com un dels més populars, gràcies a la seva facilitat tant per a l'entrenament com per a l'aplicació en temps real. YOLO per fer les deteccions, divideix la imatge en una quadrícula de mida $S \times S$. En cada cel·la d'aquesta quadrícula, YOLO realitza la predicció de N possibles caixes delimitadores i calcula la probabilitat associada a cadascuna. Això implica la generació de $S \times S \times N$ caixes delimitadores diferents. Posteriorment, s'eliminen aquelles caixes delimitadores que presenten

una probabilitat inferior a un determinat llindar. Finalment, s'elimina les deteccions duplicades i es conserva només la detecció més precisa de cada objecte detectat a la imatge. Actualment, la darrera versió és YOLOv8, desenvolupada per Ultralytics [15]. YOLOv8, es basa en els avenços de les versions anteriors de YOLO, introduint noves funcionalitats i optimitzacions.

5.2 SEGUIMENT

En el seguiment es relaciona la informació de detecció amb la trajectòria dels objectes. El seu objectiu és trobar la millor manera de seguir objectes en un sistema de seguiment, tenint en compte aspectes com objectes que es resguarden l'un a l'altre i canvis en la seva forma, mitjançant la correlació i l'associació de dades. L'article [5] parla d'algorismes de seguiment, mencionant els que estan dins de l'estat de l'art: com ara BoT-SORT [16], ByteTrack [17], Deep OC-SORT [18], OC-SORT [19] i StrongSORT [20].

Ultralytics, a part d'oferir diverses versions de YOLO, incorpora els algorismes de seguiment ByteTrack i BoT-SORT, convertint-lo en un bon marc de treball per al seguiment d'objectes. ByteTrack es basa en una metodologia d'associació senzilla, efectiva i genèrica, que associa pràcticament totes les caixes de detecció, incloent-hi les de baixa puntuació. Aquest enfocament té com a objectiu recuperar els objectes i eliminar les deteccions de fons. ByteTrack utilitza el filtre de Kalman [21] per predir la posició de l'objecte en el següent fotograma i corregir la predicció utilitzant la detecció actual. L'algoritme de seguiment BoT-SORT, integra dades de moviment i aparença, juntament amb compensació de moviment de càmera i un vector d'estat de filtre de Kalman, per desenvolupar un seguiment robust.

5.3 Seguiment d'objectes petits

El seguiment d'objectes petits presenta reptes únics a causa de la seva petita mida i sovint d'una aparença limitada. Yabin Zhu et al. [22] proposen un nou algoritme anomenat MKDNet. Aquest algoritme millora la representació de característiques, la discriminació i les habilitats de localització en el seguiment d'objectes petits. Els experiments demostren que MKDNet és superior i més efectiu en comparació amb els mètodes existents. En l'article de Ryota Yoshihashi et al. [23] es va abordar el repte de la detecció d'objectes petits mitjançant una nova estratègia anomenada *Red Correlacional Recurrent*. Aquesta tècnica permet la detecció i el seguiment simultani de manera més efectiva i mostra millores significatives en comparació amb els mètodes existents, especialment en imatges d'ocells i drons. En canvi, Camilo Aguilar et al. [24] presenten un mètode per detectar i seguir objectes petits en vídeos satel·litaris utilitzant una xarxa neuronal convolucional i un filtre bayesià. El mètode supera a altres en proves amb vídeos satel·litaris que són desafiants i de baixa resolució amb objectes petits en moviment. Jun

Fan et al. [25] proposen un marc per la detecció i seguiment de petits objectius de vehicles infrarojos. Aquest marc està format per tres components: detecció d'objectes en tota la imatge, detecció i seguiment d'objectes en imatges retallades, i predicció de la trajectòria de l'objecte. Els resultats mostren que el marc proposat pot seguir constantment els objectius dels vehicles i adaptar-se bé en situacions com la desaparició temporal d'objectius i les interferències d'altres vehicles.

6 CONJUNT DE DADES

En el desenvolupament actual de sistemes basats en *deep learning*, les dades per entrenar el sistema són tant o més importants que l'arquitectura de la xarxa del model. Per tant, en aquest treball s'han revisat diversos conjunts de dades existents, seleccionat el conjunt de dades BIRDSAI [26] per entrenar i avaluar el nostre model per les similituds que guarda amb el projecte.

El conjunt de dades BIRDSAI consisteix en imatges infraroges que capturen escenes nocturnes d'animals i éssers humans al sud d'Àfrica, captades amb drons on alguns objectes presents en l'escena es veuen a una escala petita. Aquest conjunt de dades inclou 48 vídeos reals d'infraroig tèrmic aeris i 124 vídeos sintètics d'infraroig tèrmic aeris (generats amb AirSim [27]), amb un total de 62.000 i 100.000 imatges, respectivament. És important assenyalar que aquests fotogrames d'imatge es presenten en escala de grisos, amb una baixa resolució sobre els objectes d'interès. A més, molts altres objectes als fotogrames de vídeo poden semblar similars als objectes d'interès. S'observen també variacions d'escala, confusió de fons a causa de reflexos tèrmics, grans rotacions de càmera i desenfocament del moviment. En la Fig. 1 podem observar exemples del conjunt de dades.

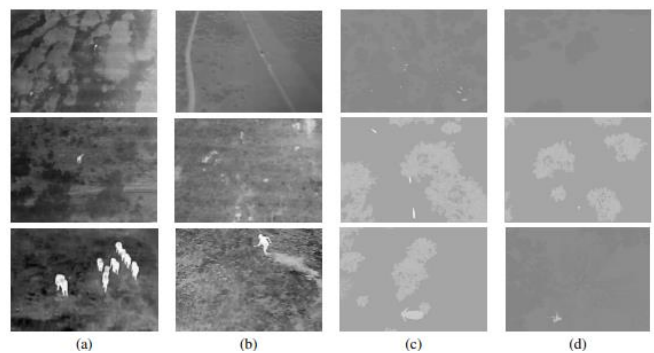


Fig. 1. Exemples d'imatges del conjunt de dades BIRDSAI. Les columnes a) i b) són fotogrames de seqüències de vídeos reals. Les columnes c) i d) són fotogrames generats sintèticament.

7 MÈTRIQUES

El repte de MOT ofereix mètriques reconegudes per avaluar el seguiment d'objectes en moviment. En aquest treball, per avaluar les deteccions s'ha utilitzat la mètrica mAP, que és la més comuna per mesurar la precisió de les deteccions [28]. Pel seguiment, s'ha fet servir les mètriques MOTA i MOTP, les més utilitzades segons els treballs revisats [4], [5], [6], [7], [8], [9].

• mAP (mean Average Precision)

El mAP és una mètrica que avalua la precisió dels detectors d'objectes en totes les classes, calculant la mitjana de la precisió promig (AP) en cada classe. L'AP representa l'àrea sota la corba de la *Precision-Recall*, on una àrea major indica un millor rendiment del model de detecció.

Per determinar si una detecció és correcta o no, el mAP utilitza l'*Intersection Over Union* (IOU), que mesura l'àrea de solapament entre les caixes delimitadores predites i reals. Un valor comú de l'IOU en la mètrica mAP és del 50%, el que significa que les deteccions amb més del 50% de solapament es consideren correctes i es representa com mAP@50. La fórmula és la següent:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i$$

On AP_i és l'AP en la classe i -èssima i N és el nombre total de classes que s'estan avaluant.

• MOTA (Multi-Object-Tracking Accuracy)

Aquesta mètrica mesura com de bé el sistema de seguiment detecta objectes i prediu les trajectòries sense tenir en compte la precisió. És a dir, no avalua com de precisos es localitzen les trajectòries. Aquesta mètrica considera tres tipus d'errors: canvis d'identitat, objectius perduts i falsos positius. La MOTA es troba en el rang de $[-\infty, 100\%]$, on valors negatius impliquen que els errors són més nombrosos que els objectes a seguir. La fórmula és la següent:

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDWS_t)}{\sum_t GT_t}$$

On t és l'índex del fotograma de la seqüència. FN (Fals Negatiu) és el nombre de fotogrames de detecció que no coincideixen amb el fotograma de predicció. FP (Fals positiu) és el nombre de fotogrames de predicció que no coincideixen amb el fotograma de detecció. $IDSW$ (Canvi d'ID) és el nombre de vegades que canvia la identificació de l'objecte. GT (*Ground truth*) és el nombre d'objectes de seguiment.

• MOTP (Multiple Object Tracking Precision)

Mesura quan precís és el seguidor a l'hora de determinar la posició de l'objecte, independentment de la capacitat del seguidor per reconèixer la configuració de l'objecte i mantenir trajectòries coherents. La fórmula és la següent:

$$MOTP = 1 - \frac{\sum Bt}{\sum Ct}$$

On Bt és la distància entre la posició corresponent de l'objecte en el fotograma t -èssim i la posició predita i Ct és el nombre de coincidències entre l'objecte i l'objecte predit en el fotograma t -èssim.

8 PIPELINE (DETECCIÓ +SEGUIMENT)

En aquest treball s'ha implementat un sistema de seguiment d'animals en imatges infraroges. Per tal de fer-ho, s'ha utilitzat la tècnica *track by detection*, on s'utilitza un algorisme de detecció i un de seguiment per poder traçar les trajectòries dels animals. El sistema s'ha desenvolupat en Python 3.10.11 i s'ha escollit la llibreria que ofereix Ultralytics per a Python [29]. Ultralytics és popular per a la detecció i seguiment d'objectes gràcies a la seva eficàcia, flexibilitat, facilitat d'ús i personalització.

Aquesta llibreria incorpora algorismes que estan a l'estat de l'art en detecció i seguiment. Ultralytics ofereix totes les versions del detector YOLO. En aquest treball s'ha escollit l'última versió, YOLOv8, que proporciona millores en la velocitat i precisió de la detecció d'objectes respecte a les versions anteriors. Pel que fa al seguiment, inclou els algorismes de seguiment ByteTrack i BoT-Sort.

S'ha implementat una seqüència de dues accions on l'entrada són les imatges d'una seqüència de vídeo. En la primera s'utilitza el detector YOLOv8 per fer les deteccions i en la segona part s'ha utilitzat els algorismes ByteTrack i BoT-Sort (es pot escollir qualsevol dels dos) per fer les associacions de les deteccions i poder fer el seguiment dels animals.

9 EXPERIMENTS I RESULTATS

En aquest apartat, es revisaran els diferents experiments realitzats amb el conjunt de dades BIRDSAI, utilitzant només les seqüències reals del conjunt de dades.

L'apartat es divideix en dues seccions: la primera presenta els diversos entrenaments del model YOLOv8 avaluat amb el conjunt de test, mentre que la segona mostra els resultats del seguiment dels animals amb els algorismes de seguiment ByteTrack i BoT-SORT, avaluat sobre el conjunt de test, utilitzant el model, entrenat amb YOLOv8, amb millor resultats.

9.1 ENTRENAMENTS DEL MODEL AMB YOLOV8

Tots els entrenaments s'han realitzat en una màquina amb dues GPU (NVIDIA GeForce RTX 2080 Ti, 11012 MiB). Per dur a terme aquests entrenaments, s'han utilitzat models preentrenats de YOLOv8 amb el conjunt de dades COCO [30], un ampli conjunt de dades dissenyat, entre altres funcions, per a la detecció d'objectes, i que compta amb més de 200.000 imatges RGB etiquetades. També s'han seguit algunes recomanacions d'Ultralytics per obtenir els millors resultats a l'hora d'entrenar un model amb un conjunt de dades propi [31].

El conjunt de dades BIRDSAI proporciona un conjunt d'entrenament i un altre de test. En cada experiment, s'ha treballat amb tres conjunts: el d'entrenament, el de validació i el conjunt de test per tal de validar els entrenaments amb el conjunt de validació i realitzar diferents experiments amb l'objectiu de millorar els resultats sense condicionar el conjunt de test. S'ha adaptat la distribució del conjunt de dades a l'estructura de directoris requerida per entrenar el model amb YOLOv8. També s'han normalitzat les coordenades i s'ha calculat el centre de les caixes delimitadores com a punt de referència per indicar l'amplada i l'altura de les caixes. A continuació, en el treball es farà referència als tres conjunts per les seves traduccions a l'anglès: train, validation i test.

Tots els entrenaments s'han realitzat amb 50 èpoques. En la Fig. 2 podem veure que de l'experiment 2, les mètriques de pèrdua pel conjunt validation en cert punt comencen a augmentar mentre que les del conjunt train decreixen en tot moment, indicant clarament la presència d'*overfitting*. De cada entrenament, s'ha seleccionat el millor model, aquell que presenta el rendiment òptim just abans de l'aparició d'*overfitting*. En l'Apèndix B, es pot observar totes les gràfiques resultants dels entrenaments i com es produeix *overfitting* en totes elles.

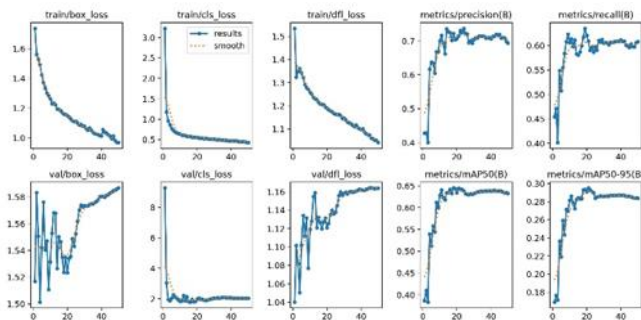


Fig. 2. Mètriques de l'entrenament de l'experiment 2.

Per avaluar els models, solament es tindrà en compte la mètrica mAP, explicada anteriorment.

• Experiment 1: Mida de la imatge (640 px vs. 1280 px)

El primer experiment tracta de fer una comparativa de l'hiperparàmetre *imgsz* a l'hora d'entrenar el model amb

els valors 640 i 1280. Aquest paràmetre indica la mida a la qual es redimensionarà les imatges d'entrada per a entrenar el model.

Segons Ultralytics, entrenar el model amb una resolució de 1280 píxels pot proporcionar resultats millors, especialment si hi ha objectes petits, com és el cas del conjunt de dades seleccionat en aquest treball. Augmentar la resolució en les imatges de l'entrenament del model pot aportar diversos avantatges, com una major precisió i una millor capacitat per capturar detalls més subtils presents a les imatges.

El model preentrenat utilitzat en aquest experiment és el model nano YOLOv8n, el més petit de YOLOv8. Té 3,2 milions de paràmetres i s'ha entrenat amb el conjunt de dades COCO en una resolució nativa de 640 píxels. Les seqüències reals del conjunt de dades BIRDSAI tenen una resolució propera a 640 píxels. Per al conjunt d'entrenament s'ha agafat el 80% dels vídeos reals del conjunt d'entrenament que proporciona BIRDSAI i el 20% restant per al conjunt validation, on s'ha considerat una única classe *animal*.

A la Taula 1 podem trobar els resultats dels entrenaments sobre el conjunt de test i, tal com s'esperava, una resolució de 1280 px millora lleugerament el mAP@50 i el mAP@50-95.

Taula 1. Mètriques de Models Entrenats amb Resolucions 640 px i 1280 px.

imgsz	P	R	mAP@50	mAP@50-95
640 px	0.622	0.366	0.398	0.152
1280 px	0.671	0.38	0.427	0.153

La Fig. C.2 de l'Apèndix ofereix un exemple visual més clar de com una resolució de 1280 px pot millorar la capacitat del model per identificar característiques més precises dels objectes petits. En comparació amb la Fig. C.1 de l'Apèndix, es pot apreciar com el model ha estat capaç de predir un objecte petit amb aquesta resolució major. En canvi, el model amb una resolució de 640 px no ha aconseguit predir cap objecte petit que aparegui a la Fig. C.1 de l'Apèndix.

En els següents experiments s'utilitzarà un *imgsz* de 1280 px.

• Experiment 2: Aplicació de Mostreig Estratificat en el conjunt de dades

En l'experiment anterior, s'ha avaluat els models amb una sola classe: *animal*. No obstant això, per a una avaluació més completa, en aquest experiment considerarem totes les classes disponibles, ja que hi ha objectes de diferents mides en aquestes classes com s'observa en la Fig. 6. Això ens permetrà analitzar com el model detecta tant animals petits com grans, obtenint resultats més realistes.

Si observem la Fig. 3, es pot veure que, en la distribució de dades de l'experiment anterior, els conjunts no estan balancejats a nivell de classes: les classes *elephant* i *lion* no tenen dades al conjunt validation, i la classe *dog* no té dades ni en el conjunt de train ni en el de validation. També es pot visualitzar que la classe *elephant* té la major quantitat d'instàncies del conjunt de dades reals de BIRDSAI, seguides per les classes *human* i *unknown*.

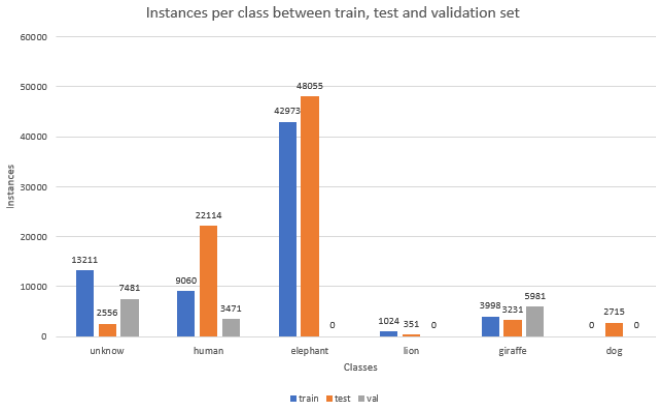


Fig. 3. Instàncies per classe per cada conjunt de dades.

Per tal d'equilibrar els conjunts de dades a nivell de classes, s'han dividit les seqüències de vídeos en dues seqüències úniques per obtenir una quantitat més àmplia de seqüències i poder fer una distribució més homogènia de les classes. Aquesta divisió ha elevat el nombre de seqüències de 48 a 96. S'han agrupat els 96 vídeos segons la classe predominant en cada vídeo i s'han dividit de forma estratificada, assignant, per classe, un 60% dels vídeos al conjunt de train i un 20% a cada un dels conjunts de validació i prova. A part d'assegurar una distribució equilibrada de les classes, assegurem també que hi hagi una millor distribució d'escenaris diferents i objectes de diferents mides en els tres conjunts.

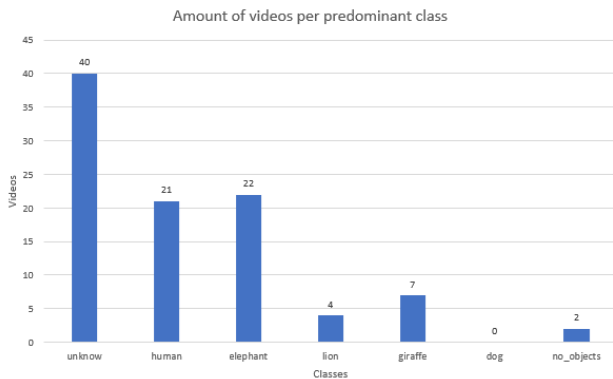


Fig. 4. Quantitat de vídeos per classe predominant.

En la Fig. 4 es pot observar que no hi ha seqüències de vídeos on la classe predominant sigui *dog* i que hi ha dues seqüències on no hi ha objectes a detectar. Ultralytics aconsella que dins del conjunt de dades, per cada classe hi hagi 10.000 instàncies o més. Per tant, les classes *dog* i

lion, com es pot apreciar a la Fig. 3, no arriben a aquesta xifra límit i no seran considerades en l'avaluació del model. Les seqüències on no hi ha cap objecte, s'han inclòs en el conjunt de train. En la Fig. 5 podem veure que les dades, a nivell de classe, estan més balancejades un cop s'ha fet una distribució estratificada.

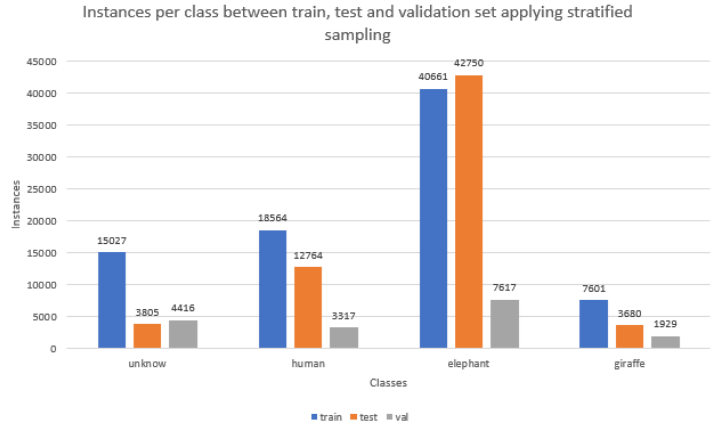


Fig. 5. Instàncies per classe per cada conjunt de dades aplicant un mostreig estratificat.

El model s'ha entrenat amb la configuració de l'experiment anterior. Únicament ha canviat la distribució dels conjunts de dades i que s'ha considerat les classes *unknown*, *human*, *elephant* i *giraffe*.

Taula 2. Mètriques del Model Entrenat Dividint els Conjunts de forma Estratificada per Classe.

class	P	R	mAP@50	mAP@50-95
all	0.612	0.516	0.552	0.231
unknown	0.68	0.458	0.49	0.173
human	0.522	0.407	0.423	0.193
elephant	0.894	0.238	0.635	0.273
giraffe	0.353	0.96	0.656	0.283

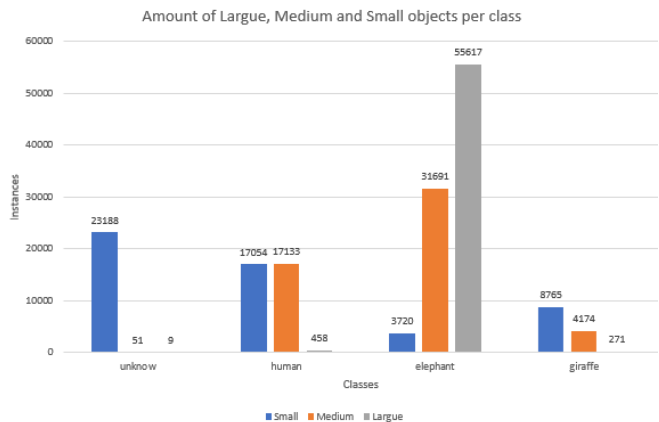


Fig. 6. Quantitat d'objectes petits, mitjans i grans per classe. Agrupats segons els percentils 33 i 66 de les àrees de les caixes delimitadores.

En la Taula 2, amb la nova distribució de les dades es pot veure que s'ha millorat les mètriques respecte a l'experiment anterior, el mAP@50 i el mAP@50-95 han augmentat un 0.13 i un 0.08 respectivament. La classe *giraffe* obté el millor mAP, tot i que s'esperava que fos la classe *elephant*, degut a la seva quantitat d'instàncies i també al fet que té objectes de mida més gran que la resta, com es pot veure a la Fig. 6, que són més fàcils de detectar. Si ens fixem en la Taula 2, en el *recall* (R) de la classe *elephant*, podem veure que té un valor baix indicant que el model té dificultats per identificar correctament tots els casos positius reals d'aquesta classe. Aquests resultats poden ser atribuïts al fons de les imatges, on les girafes presenten un contrast més evident, facilitant-ne la detecció en comparació amb les imatges on hi ha elefants, que es poden confondre amb elements de l'escena. En l'Apèndix, a la Fig. D.1, es pot veure un exemple del fons on hi ha girafes, mentre que en la Fig. D.2 es pot veure un exemple del fons on hi ha elefants.

En els següents experiments s'utilitzarà la distribució de dades utilitzada en aquest experiment.

• Experiment 3: Batch Size

Aquest experiment tracta de fer una comparativa de l'hiperparàmetre *batch*, que defineix la quantitat d'exemples de dades que es processen simultàniament durant cada iteració d'entrenament. Aquest hiperparàmetre està relacionat amb la utilització dels recursos; un valor més gran ocupa més memòria, però l'entrenament és més ràpid, mentre que un valor més petit ocupa menys memòria, però triga més en el procés d'entrenament. Encara que a vegades pot afectar el rendiment, l'article [32] parla de com valors grans poden afectar a una pitjor generalització del model. En aquest experiment, s'observa com afecten diferents valors de l'hiperparàmetre *batch* als resultats.

Taula 3. Mètriques del Model Entrenat amb Batch Size 4.

class	P	R	mAP@50	mAP@50-95
all	0.593	0.532	0.568	0.247
unknow	0.578	0.38	0.384	0.14
human	0.471	0.389	0.408	0.201
elephant	0.839	0.421	0.682	0.283
giraffe	0.483	0.936	0.799	0.365

Taula 4. Mètriques del Model Entrenat amb Batch Size 8.

class	P	R	mAP@50	mAP@50-95
all	0.565	0.523	0.568	0.248
unknow	0.625	0.409	0.465	0.175
human	0.361	0.4	0.379	0.174
elephant	0.85	0.313	0.621	0.268
giraffe	0.422	0.969	0.807	0.373

En els experiments anteriors s'ha utilitzat un *batch* de 16. En aquest experiment s'ha fet la comparativa amb valors de 4 i 8, en les Taules 3 i 4 es poden observar, respectivament, els resultats dels entrenaments. Per una banda, les variacions en el mAP entre les configuracions de mida de lot de 4, 8 i 16 no mostren diferències excessivament notables. Els models amb un *batch* de 4 i 8 obtenen un mAP pràcticament igual i mostren un augment poc significatiu respecte al *batch* de 16 (experiment anterior). A nivell de classe, la mida del *batch* pot presentar variacions en el rendiment. Per exemple, per a la classe *unknown*, es pot observar un empitjorament de més de 0,1 en el mAP@50 amb una mida de *batch* de 4 en comparació amb un *batch* de 16. També, amb un *batch* de 8, es millora un 0.15 el mAP@50 de la classe *giraffe* respecte a un *batch* de 16.

Tot i que els models amb una *batch* de 4 i 8 exhibeixen el millor rendiment, en els següents experiments s'utilitzarà el model amb un *batch* de 16. Aquest model ofereix un rendiment pràcticament similar al de 4 i 8 i, al mateix temps, accelera el temps d'entrenament.

• Experiment 4: Diferents models preentrenats

En aquest experiment, s'han utilitzat altres models preentrenats proporcionats per Ultralytics com a base per entrenar el model, fent servir el conjunt de dades BIRDSAI. En els experiments anteriors, s'ha utilitzat el model més petit, el *Nano*, amb 3,2 milions de paràmetres. A la Taula 5 es mostren els diferents models preentrenats disponibles d'Ultralytics, amb els resultats obtinguts en la mètrica mAP@50-95 sobre el conjunt de dades COCO. Ultralytics mostra que un model amb més paràmetres millora el rendiment en la mètrica mAP@50-95 sobre el conjunt de dades COCO, però això implica un augment en el temps d'entrenament.

Taula 5. Models Preentrenats que proporciona Ultralytics.

Model	mAP@50-95 (%)	Speed A100 TensorRT (ms)	Paràmetres (M)
YOLOv8n	37.3	0.99	3.2
YOLOv8s	44.9	1.2	11.2
YOLOv8m	50.2	1.83	25.9
YOLOv8l	52.9	2.39	43.7
YOLOv8x	53.9	3.53	68.2

En aquest experiment s'han utilitzat el model *Small* amb 11,2 milions de paràmetres i el model *Medium* amb 25,9 milions de paràmetres. A les Taules 6 i 7 es poden observar els resultats d'ambdós entrenaments, respectivament. El model *Medium* s'ha entrenat amb un *batch* de 8, per manca de recursos. Si tenim en compte el rendiment de l'experiment 2, el model *Small* obté un valor pràcticament igual en el mAP@50, mentre que el model *Medium* ho millora en 0.05.

Taula 6. Mètriques del Model Entrenat amb el model preentrenat Small.

class	P	R	mAP@50	mAP@50-95
all	0.666	0.496	0.55	0.236
unknow	0.722	0.419	0.503	0.17
human	0.631	0.372	0.433	0.21
elephant	0.903	0.247	0.634	0.275
giraffe	0.407	0.947	0.629	0.289

Taula 7. Mètriques del Model Entrenat amb el model preentrenat Medium.

class	P	R	mAP@50	mAP@50-95
all	0.691	0.522	0.602	0.257
unknow	0.693	0.453	0.53	0.185
human	0.681	0.381	0.474	0.227
elephant	0.893	0.31	0.63	0.278
giraffe	0.497	0.945	0.773	0.339

Com era d'esperar, un model amb més paràmetres ha millorat el rendiment, encara que sigui lleugerament. Un model amb més paràmetres d'entrenament pot millorar la capacitat de detectar més característiques o aspectes, ja que disposa d'una capacitat més gran per aprendre i representar detalls subtils. En els següents experiments, s'utilitzarà el model preentrenat Medium.

• Experiment 5: Eliminació d'imatges sense objectes.

En aquest experiment, seguirem una recomanació d'Ultralytics [31]. Ultralytics suggereix que en el conjunt de dades hi hagi entre un 0% i un 10% d'imatges sense objectes per tal de reduir els falsos positius. Si hi ha massa exemples d'imatges sense objectes, el model podria aprendre que l'estratègia òptima per optimitzar el rendiment és no predir res.

Si fem un recompte de les imatges en el conjunt de dades BIRDSAI, veiem que hi ha un 40% d'imatges sense objectes de les 61.994 imatges disponibles. En aquest experiment, s'han eliminat 15.189 imatges sense objectes del conjunt de train, deixant un 5% d'aquest conjunt amb imatges sense objectes.

En la Taula 8 es pot observar els resultats de l'entrenament. S'ha reduït el mAP@50 mínimament respecte a l'entrenament de l'experiment anterior. La classe *unknow*, que és la classe amb més percentatge d'objectes petits, ha disminuït un 0.056. Això pot indicar que el model necessita imatges sense objectes per tal de diferenciar aquells petits objectes del fons. Donat que es tracta d'imatges infraroges, hi ha diversos elements a les imatges, com taques blanques petites, que es poden confondre amb animals.

Taula 8. Mètriques del Model Entrenat amb Conjunt Train amb un 5% solament d'Imatges sense Objectes.

class	P	R	mAP@50	mAP@50-95
all	0.697	0.492	0.581	0.252
unknow	0.633	0.412	0.474	0.166
human	0.707	0.368	0.462	0.22
elephant	0.928	0.258	0.622	0.273
giraffe	0.521	0.93	0.766	0.348

• Experiment 6: Més instàncies al conjunt train de la classe *elephant*

En l'últim experiment s'ha mogut una seqüència del conjunt de test al conjunt de train. Això s'ha fet amb l'objectiu d'intentar millorar el mAP per a la classe *elephant*, ja que com es pot observar a la Fig. 5, hi ha més instàncies de la classe *elephant* en el conjunt de test que en el de train. S'ha traslladat un vídeo que conté 23.123 instàncies d'elefants.

A la Taula 9 es pot observar que un augment en el nombre d'instàncies d'elefants millora el mAP@50 de la classe *elephant* en 0.26 respecte al millor model de l'experiment 4. A més, en aquest experiment també s'han registrat millores en el mAP@50 de les classes *human* i *giraffe*, amb increments de 0.075 i 0.1, respecte a l'experiment 4.

Taula 9. Mètriques del Model Entrenat amb més Instàncies de la Classe Elephant al Conjunt de Train.

class	P	R	mAP@50	mAP@50-95
all	0.745	0.651	0.71	0.338
unknow	0.653	0.422	0.532	0.191
human	0.811	0.401	0.549	0.258
elephant	0.838	0.832	0.888	0.504
giraffe	0.677	0.951	0.872	0.398

• Model amb millor rendiment.

Els millors resultats s'han obtingut en l'experiment 6 amb un mAP@50 de 0.71. Les classes *elephant* i *giraffe* són les que millor detecta el model, mentre que la classe *unknow* és la que pitjor detecta.

9.2 AVALUACIÓ DELS ALGORISMES DE SEGUIMENT

En aquest apartat es documenta la comparativa dels algorismes de seguiment ByteTrack i BoT-Sort amb el millor model entrenat de detecció obtingut en els experiments sobre el conjunt de test. També es compararà amb els resultats obtinguts en el treball on es presenta el conjunt de dades BIRDSAI [26].

Hi ha vídeos on els algorismes de seguiment no han pogut seguir cap objecte, en aquests vídeos se les ha assignat un 0 en les mètriques MOTA i MOTP. En la Fig. E.1 de l'Apèndix podem observar com en general, en tots els vídeos BoT-Sort és superior en la mètrica MOTA, és a dir, és capaç de seguir els objectes en una seqüència de millor manera que ByteTrack. En canvi, en la Fig. E.2 de l'Apèndix, es pot observar que en la mètrica MOTP, presenten resultats similars. Tot i això, és important tenir en compte que la capacitat de seguir els objectes en un vídeo pot dependre també de l'arquitectura de cada model. Per exemple, en les gràfiques es pot observar com BoT-Sort no és capaç de seguir cap objecte en el vídeo 6, mentre que ByteTrack no ho aconsegueix en els vídeos 5, 6, 7 i 9. Aquests vídeos presenten pocs objectes molt petits amb canvis de moviment de càmera ràpids i bruscos, circumstàncies que poden dificultar el seguiment. En la Taula 10 es pot veure la comparativa més clara, com BoT-Sort obté un millor rendiment en general que ByteTrack.

Taula 10. Mitjana de les Mètriques MOTA i MOTP pels Vídeos del Conjunt de Test.

TRACKER	MOTA	MOTP
ByteTrack	0.228	0.562
BoT-Sort	0.299	0.68

En el treball on es proposa el conjunt de dades BIRDSAI, els autors obtenen un mAP@50 de 0.459 sobre el conjunt de test amb el detector *Domain Adaptive Faster R-CNN* [33] amb la funció de pèrdua *WCE*. Utilitzen el conjunt d'imatges reals i sintètiques, tenint en compte totes les classes. En aquest treball s'ha utilitzat solament el conjunt de dades reals i no s'ha tingut en compte les classes *lion* i *dog*. Per intentar fer la comparativa justa es suposarà que en les classes *lion* i *dog* s'ha obtingut un mAP@50 de 0, per tant, obtenim un mAP@50 general de 0.47 en aquest treball. Tot i que aquesta comparativa no és del tot representativa, ja que les condicions són lleugerament diferents, els valors són pròxims, permetent afirmar que hem obtingut un rendiment similar al mAP@50 de 0.459 que presenten els autors de BIRDSAI.

Quant al seguiment, en el treball de BIRDSAI els autors han obtingut un MOTP de 0.66 utilitzant l'algorisme de seguiment IoU [34], mentre que en aquest treball l'algorisme de seguiment BoT-Sort ha assolit un MOTP de 0.68, resultats pràcticament iguals, ja que en la part de detecció s'ha obtingut un rendiment similar. No obstant això, es pot observar una diferència en la mètrica MOTA: en l'estudi amb el qual es compara s'obté una MOTA de -1.23, mentre que en aquest treball el BoT-Sort presenta una MOTA de 0.299. Tot i que aquesta comparació pot no ser del tot representativa, els resultats de la mètrica MOTA indiquen un millor rendiment dels algorismes de seguiment BoT-Sort i ByteTrack, que mostren una MOTA positiva.

10 CONCLUSIONS I TREBALL FUTUR

En aquest treball s'ha implementat un sistema de seguiment d'animals en imatges tèrmiques aplicant la tècnica *track by detection*. S'ha utilitzat YOLOv8 per la part de detecció i BoT-Sort i ByteTrack per al seguiment, que són models populars dins l'estat de l'art. No obstant això, els resultats obtinguts no són molt bons, principalment a causa del repte que presenta el conjunt de dades BIRDSAI. Aquest conjunt de dades proporciona vídeos d'animals de diferents mides en imatges tèrmiques, amb moviments bruscos de càmera i elements de l'escena molt similars als animals, fet que dificulta el seguiment d'aquests. S'ha demostrat que el sistema amb animals de gran mida obté millors resultats, com per exemple la classe *elephant*.

S'han realitzat diversos experiments on s'ha obtingut un rendiment similar en la part de detecció en comparació amb els resultats presentats en el treball de BIRDSAI [26], amb el detector YOLOv8. No obstant això, pel que fa al seguiment, s'ha aconseguit una MOTA de 0.299 i 0.228 amb els algorismes de seguiment BoT-Sort i ByteTrack, en comparació amb la MOTA de -1.23 que es presenta en el treball de BIRDSAI amb l'algorisme de seguiment IoU. Per tant, els algorismes de seguiment BoT-Sort i ByteTrack mostren un rendiment superior en el seguiment d'objectes en comparació amb IoU. Cal destacar que BoT-Sort ha obtingut resultats millors que ByteTrack en el seguiment d'objectes de diferents mides en imatges tèrmiques.

Com a possibles extensions d'aquest treball, es proposa utilitzar el conjunt d'imatges sintètiques per avaluar el rendiment amb YOLOv8. També es proposa realitzar una estratificació agrupant objectes de diferents mides amb l'objectiu de fer una distribució més equilibrada.

S'espera que aquest treball ajudi en la detecció i seguiment d'objectes de diferents mides en imatges tèrmiques, sigui en alguna institució o en un entorn específic.

AGRAÏMENTS

En aquest apartat vull agrair al meu tutor, Daniel Ponsa Mussara, per ser la meua guia en aquest treball i ajudar-me en tot moment. També volia agrair als meus amics i familiars, i sobretot a la meua parella, per donar-me suport i estar quan més ho necessitava.

BIBLIOGRAFIA

- [1] A. V. Sayagavi, T. S. B. Sudarshan, and P. C. Ravoor, "Deep Learning Methods for Animal Recognition and Tracking to Detect Intrusions," *Information and Communication Technology for Intelligent Systems*, pp. 617–626, Oct. 2020.
- [2] T. Jintasuttisak, A. Leonce, M. Sher Shah, T. Khafaga, G. Simkins, and E. Edirisinghe, "Deep Learning based Animal Detection and Tracking in Drone Video Footage," *Proceedings of the 8th International Conference on Computing and Artificial Intelligence*, Mar. 2022.

- [3] Y. A. Roopashree, M. Bhoomika, R. Priyanka, K. Nisarga, and S. Behera, "Monitoring the Movements of Wild Animals and Alert System using Deep Learning Algorithm," *IEEE Xplore*, Aug. 01, 2021.
- [4] S. Guo *et al.*, "A Review of Deep Learning-Based Visual Multi-Object Tracking Algorithms for Autonomous Driving," *Applied Sciences*, vol. 12, no. 21, p. 10741, Jan. 2022.
- [5] Jumabek Alikhanov and H. Kim, "Online Action Detection in Surveillance Scenarios: A Comprehensive Review and Comparative Study of State-of-the-Art Multi-Object Tracking Methods," *IEEE Access*, vol. 11, pp. 68079–68092, Jan. 2023.
- [6] B. Mirzaei, H. Nezamabadi-pour, A. Raoof, and R. Derakhshani, "Small Object Detection and Tracking: A Comprehensive Review," *Sensors*, vol. 23, no. 15, p. 6887, Jan. 2023.
- [7] S. Hassan, G. Muftaba, A. Rajput, and N. Fatima, "Multi-object tracking: a systematic literature review," *Multimedia Tools and Applications*, Oct. 2023.
- [8] L. Kalake, W. Wan, and L. Hou, "Analysis Based on Recent Deep Learning Approaches Applied in Real-Time Multi-Object Tracking: A Review," *IEEE Access*, vol. 9, pp. 32650–32671, 2021.
- [9] M. Bashar, S. Islam, K. K. Hussain, M. B. Hasan, Rahman, and M. H. Kabir, "Multiple Object Tracking in Recent Times: A Literature Review," *arXiv (Cornell University)*, Sep. 2022.
- [10] Gal Hyams and Dan Malowany, "The battle of speed vs. accuracy: Single-shot vs two-shot detection meta architecture," Apr 2020.
- [11] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," *Neural Information Processing Systems*, 2016.
- [12] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," *Computer Vision – ECCV 2016*, vol. 9905, pp. 21–37, 2016.
- [13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [14] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, Dec. 2023.
- [15] "Ultralytics | Revolutionizing the World of Vision AI," *www.ultralytics.com*. <https://www.ultralytics.com>.
- [16] N. Aharon, R. Orfaig, and B.Z. Bobrovsky, "BoT-SORT: Robust Associations Multi-Pedestrian Tracking," *arXiv (Cornell University)*, Jun. 2022.
- [17] Y. Zhang *et al.*, "ByteTrack: Multi-object Tracking by Associating Every Detection Box," pp. 1–21, Oct. 2021.
- [18] G. Maggolino, A. Ahmad, J. Cao, and K. M. Kitani, "Deep OC-SORT: Multi-Pedestrian Tracking by Adaptive Re-Identification," *arXiv (Cornell University)*, Feb. 2023.
- [19] J. Cao, X. Weng, Rawal Khirodkar, J. Pang, and K. M. Kitani, "Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking," Mar. 2022.
- [20] Y. Du *et al.*, "StrongSORT: Make DeepSORT Great Again," in *IEEE Transactions on Multimedia*, vol. 25, pp. 8725–8737, 2023.
- [21] G. F. Welch, "Kalman Filter," *Computer Vision*, pp. 1–3, Dec. 2020.
- [22] Y. Zhu *et al.*, "Tiny Object Tracking: A Large-Scale Dataset and a Baseline," *IEEE transactions on neural networks and learning systems*, pp. 1–15, Jan. 2023.
- [23] Ryota Yoshihashi, R. Kawakami, S. You, Tu Tuan Trinh, M. Iida, and Takeshi Naemura, "Finding a Needle in a Haystack: Tiny Flying Object Detection in 4K Videos using a Joint Detection-and-Tracking Approach," *arXiv (Cornell University)*, May 2021.
- [24] C. Aguilar, M. Ortner, and Josiane Zerubia, "Small Object Detection and Tracking in Satellite Videos With Motion Infor-med-CNN and GM-PHD Filter," *HAL (Le Centre pour la Communication Scientifique Directe)*, vol. 2, Apr. 2022.
- [25] J. Fan, J. Wei, H. Huang, D. Zhang, and C. Chen, "IRSdT: A Framework for Infrared Small Target Tracking with Enhanced Detection," *Sensors*, vol. 23, no. 9, pp. 4240–4240, Apr. 2023.
- [26] E. Bondi *et al.*, "BIRDSAI: A Dataset for Detection and Tracking in Aerial Thermal Infrared Videos," *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Mar. 2020.
- [27] E. Bondi *et al.*, "AirSim-W," *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, Jun. 2018.
- [28] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A Survey on Performance Metrics for Object-Detection Algorithms," *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, vol. 1, no. 1, Jul. 2020.
- [29] Ultralytics, "ultralytics: Ultralytics YOLOv8 for SOTA object detection, multi-object tracking, instance segmentation, pose estimation and image classification.," *PyPI*. <https://pypi.org/project/ultralytics/>.
- [30] "COCO - Common Objects in Context," *cocodataset.org*. <https://cocodataset.org/#home>.
- [31] Ultralytics, "Consejos para obtener los mejores resultados de entrenamiento," *docs.ultralytics.com*. https://docs.ultralytics.com/es/yolov5/tutorials/tips_for_best_training_results.
- [32] Nitish Shirish Keskar, Dheevatsa Mudigere, J. Nocedal, Mikhail Smelyanskiy, and P. Tak, "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima," Sep. 2016.
- [33] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain Adaptive Faster R-CNN for Object Detection in the Wild," *openaccess.thecvf.com*, 2018.
- [34] E. Bochinski, T. Senst, and T. Sikora, "Extending IOU Based Multi-Object Tracking by Visual Information," *IEEE Xplore*, Nov. 01, 2018.

APÈNDIX

A. PLANIFICACIÓ: DIAGRAMA DE GANNT

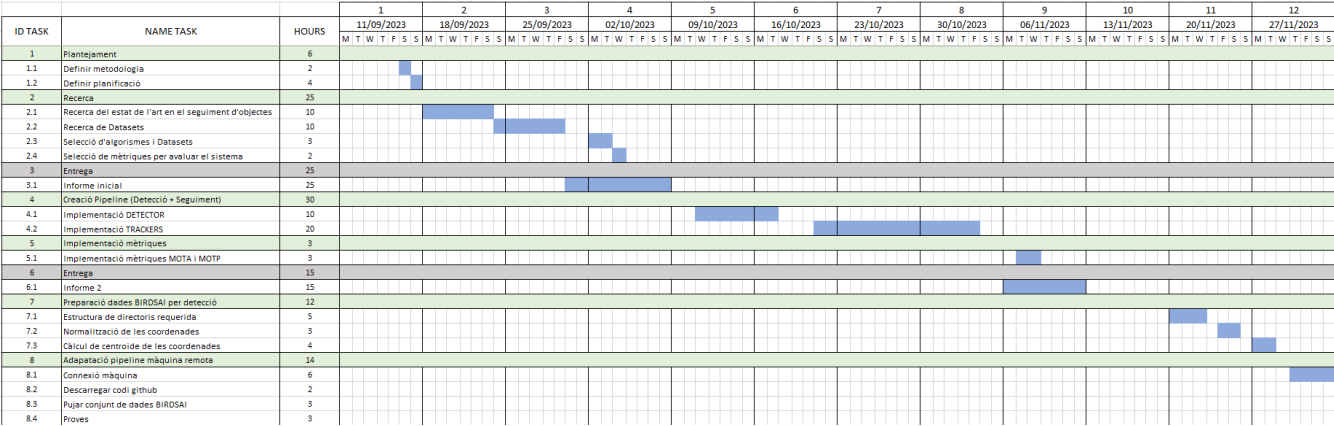


Fig. A.1. Primera part del diagrama de Gantt on es reflecteix la planificació de les tasques i les seves respectives hores dedicades.

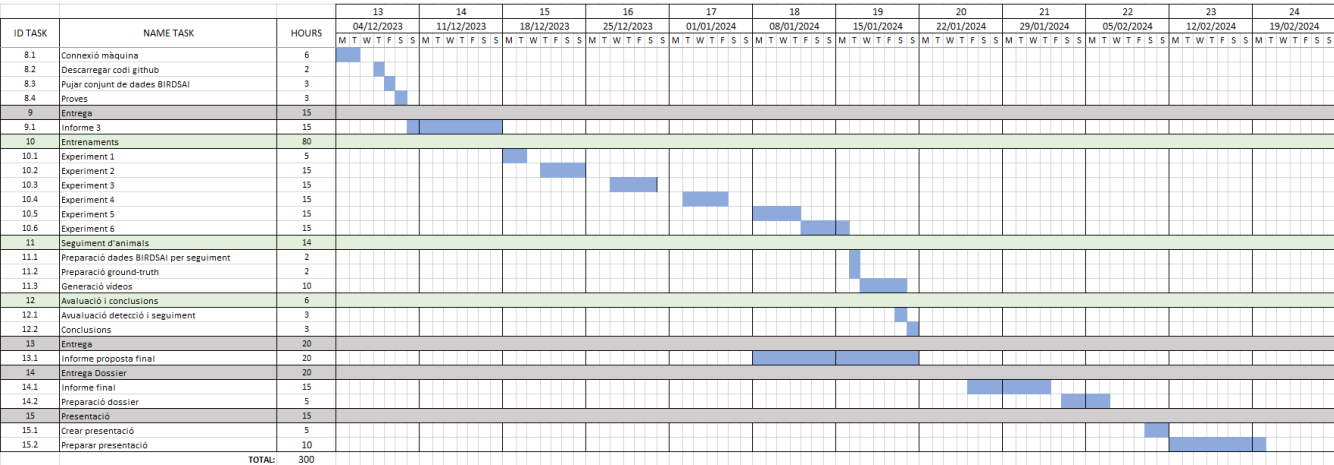


Fig. A.2. Segona part del diagrama de Gantt on es reflecteix la planificació de les tasques i les seves respectives hores dedicades.

B. ENTRENAMENTS AMB YOLOv8

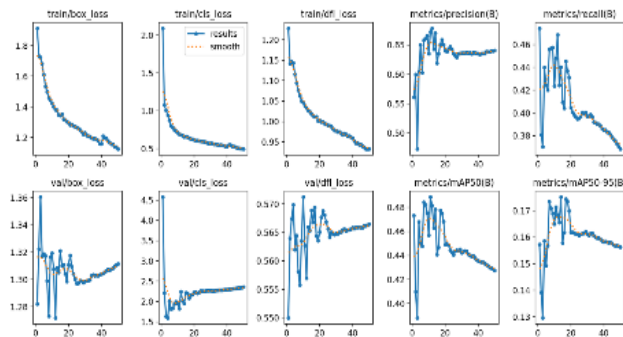


Fig. B.1. Entrenament de l'experiment 1 (imgsz de 640).

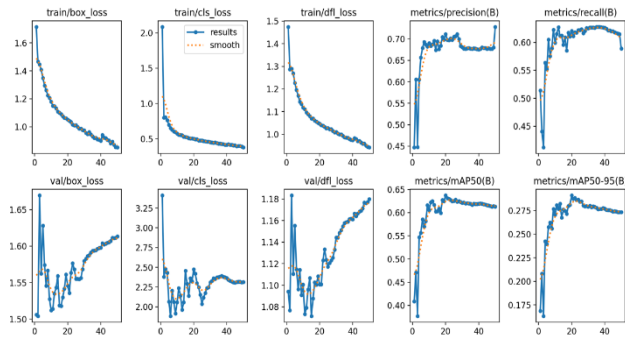


Fig. B.5. Entrenament de l'experiment 4 (model Small).

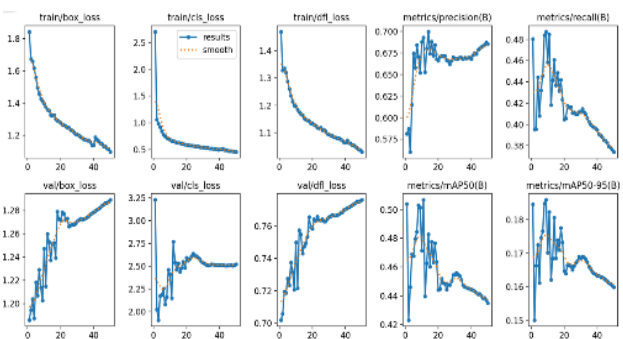


Fig. B.2. Entrenament de l'experiment 1 (imgsz de 1280).

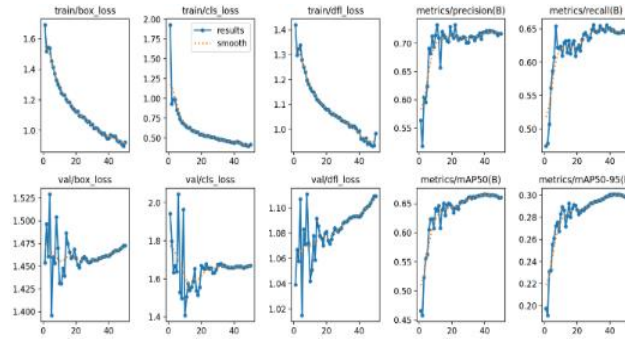


Fig. B.6. Entrenament de l'experiment 4 (model Medium).

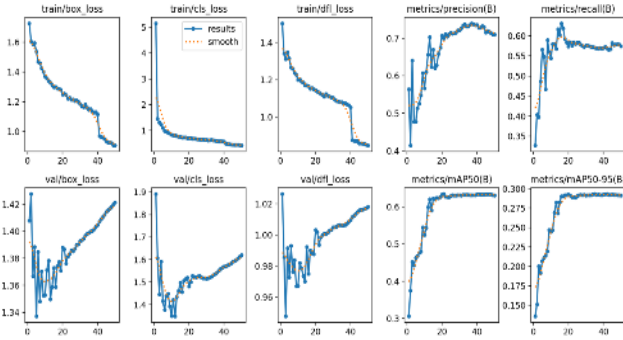


Fig. B.3. Entrenament de l'experiment 3 (batch de 4).

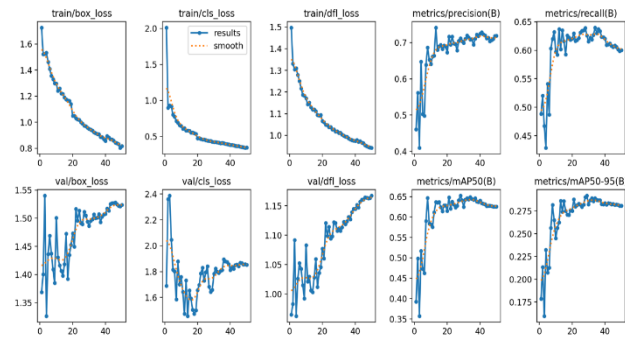


Fig. B.7. Entrenament de l'experiment 5.

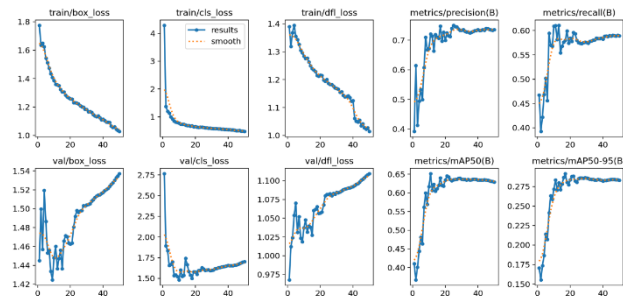


Fig. B.4. Entrenament de l'experiment 3 (batch de 8).

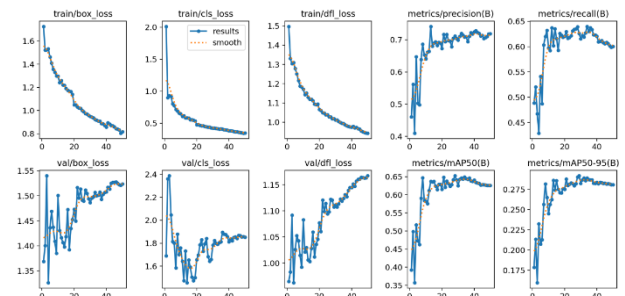


Fig. B.8. Entrenament de l'experiment 6.

C. EXPERIMENT 1

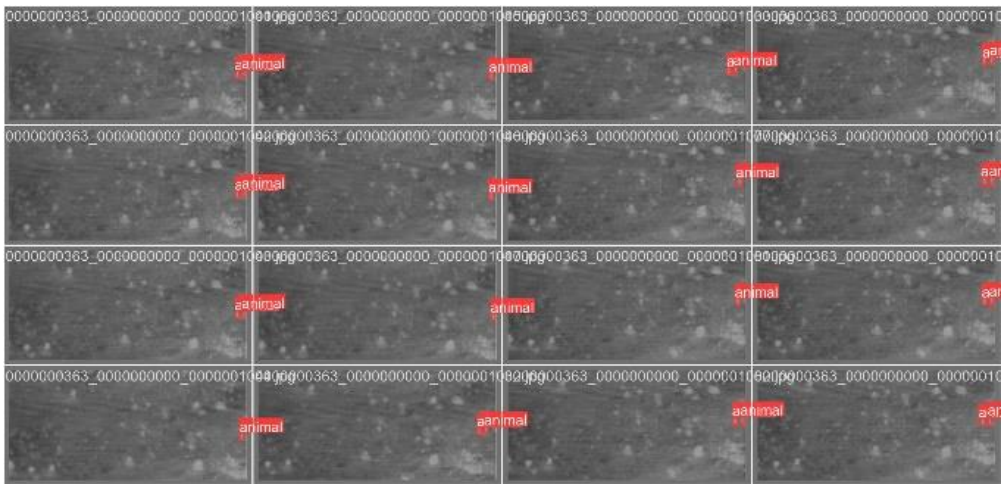


Fig. C.1. Imatges del conjunt de validació amb les seves respectives caixes delimitadores.

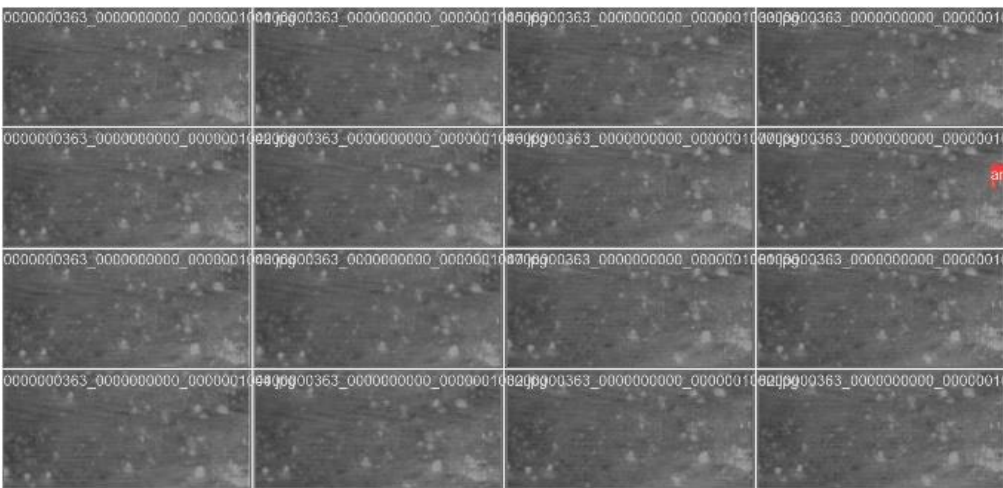


Fig. C.2. Imatges del conjunt de validació amb les prediccions que ha fet el model entrenat amb un imsgz de 1280 px.

D. EXPERIMENT 2



Fig. D.1. Imatge on hi ha girafes.



Fig. D.2. Imatge on hi ha elefants.

E. GRÀFIQUES D'AVALUACIÓ TRACKERS

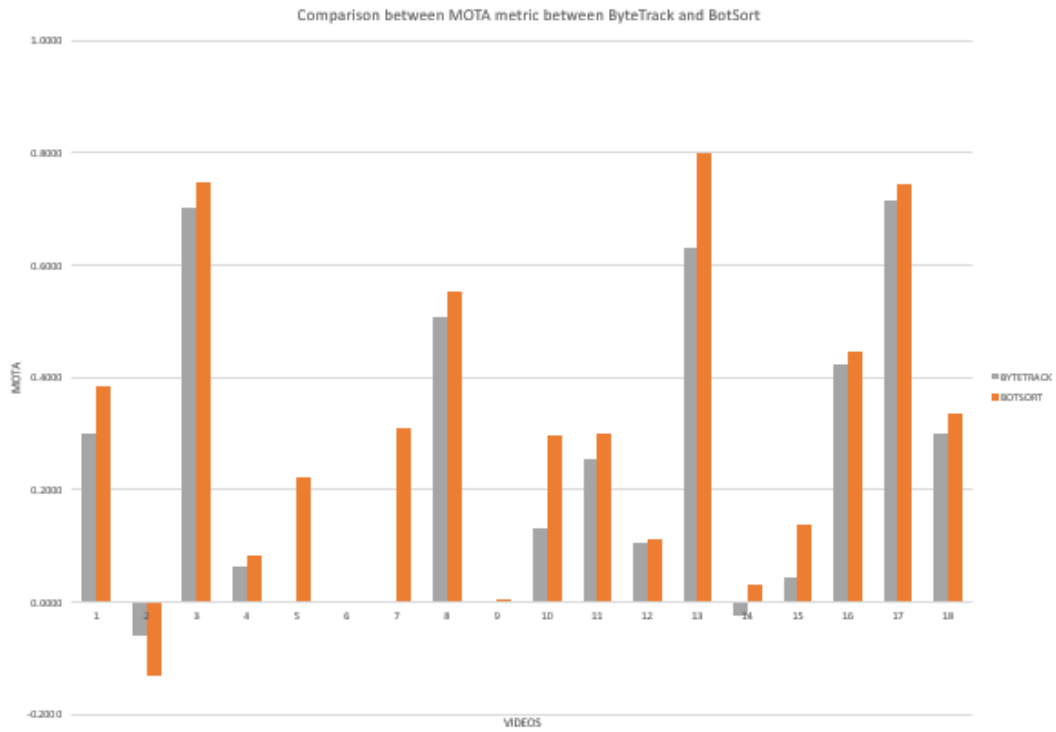


Fig. E.1. Comparació de la mètrica MOTA entre els trackers ByteTrack i BoT-Sort.

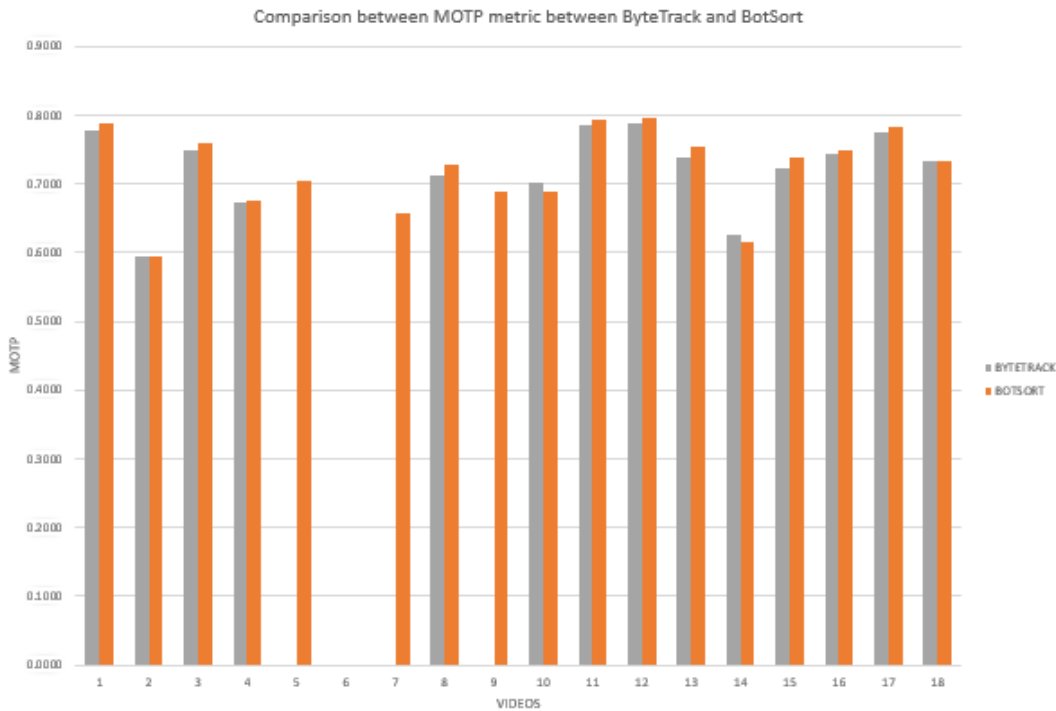


Fig. E.2. Comparació de la mètrica MOTP entre els trackers ByteTrack i BoT-Sort.