UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

telecos
BCN

Ant Tracking

Bachelor's Degree Thesis
Submitted to the Faculty at the
Escola Tècnica Superior
d'Enginyeria de Telecomunicació de Barcelona
de la Universitat Politècnica de Catalunya
by

Pol Serra Montes

In partial fulfillment of the requirements for the
DEGREE IN TELECOMUNICATION TECHNOLOGIES AND SERVICES ENGINEERING

Advisor: Ramon Morros Rubio

Barcelona, 01 2024

Resum

Aquest Treball de Fi de Grau continua el treball realitzat per Ignasi Nogueiras en el desenvolupament d'una eina de visió per computador per a la detecció i seguiment de formigues. S'han seguit dues línies de millora: la primera, optimitzar el model de detecció, que ha aconseguit avenços significatius en la identificació de formigues en situacions de solapament, beneficiant-ne així el seguiment. La segona, implementar un sistema de seguiment offline que ha millorat els resultats en termes de HOTA (Higher Order Tracking Accuracy). S'ha pogut concloure que millores en el sistema de detecció d'hormigues, en concret en situacions de superposició, conseqüentment milloren el sistema de seguiment, a més a més, per millorar el sistema de seguiment offline per millorar el model d'aparièncía que usa.

Resumen

Este Trabajo de Fin de Grado continúa el trabajo realizado por Ignasi Nogueiras en el desarrollo de una herramienta de visión por computador para la detección y seguimiento de hormigas. Se han seguido dos líneas de mejora: la primera, optimizar el modelo de detección, que ha logrado avances significativos en la identificación de hormigas en situaciones de solapamiento, beneficiando así su seguimiento. La segunda, implementar un sistema de seguimiento offline que ha mejorado los resultados en términos de HOTA (Higher Order Tracking Accuracy). Se ha podido concluir que mejoras en el sistema de detección de hormigas, en concreto en situaciones de solapamiento, consecuentemente mejoran el sistema de seguimiento, además de que para mejorar el sistema de seguimiento offline cabe mejorar el modelo de apariencia que usa.

Summary

This Final Degree Thesis advances Ignasi Nogueiras initial work on developing a computer vision tool for ant detection and tracking. Two main lines of improvement were pursued: firstly, enhancing the detection model, which has achieved significant progress in addressing ant overlap scenarios, consequently improving ant tracking; secondly, proposing an offline tracking system that has enhanced results in terms of Higher Order Tracking Accuracy (HOTA). It has been concluded that improvements in ant overlap situation in the ant detection system consequently improve the tracking system, and that to improve the offline system it is necessary to improve the appearance model it uses.

Acknowledgements

I would like to express my deepest gratitude to Ramon Morros for the excellent guidance and consistent monitoring throughout the course of my project. The wealth of knowledge imparted to me has been invaluable and will accompany me in both my professional and personal future endeavors. Special recognition also goes to Ignasi Nogueiras, whose previous work not only made my transition into the project seamless but also his steady presence and support during our follow-up meetings have been invaluable help in overcoming the challenges encountered.

Revision history and approval record

Revision	Date	Description
1.0	18/12/2023	Document creation
1.1	08/01/2024	Review State of the art and Methodology
2.0	18/01/2024	Document Revision
3.0	20/01/2024	Final Version

Document distribution list

Role	Surname(s) and Name
Student	Pol Serra Montes
Project Supervisor	Ramon Morros Rubio

Written by:	
Date	18/12/2023
Name	Pol Serra Montes
Role	Project Author

Reviewed and approved by:	
Date	20/01/2024
Name	Ramon Morros Rubio
Role	Project Supervisor

Contents

1 Introduction

Computer vision has become an indispensable tool in many fields where skills are required that until recently only humans had. One of this fields could be modern biology, allowing scientists to observe and analyze animal behaviors with reliable precision. In the study of social insects like ants, these technologies offer unique insights into their complex social dynamics and collective behaviors. Ants, with their societies and efficient communication methods, present a fascinating field of study to understand complex social systems. Detailed tracking of their movements and behaviors can reveal key patterns that enhance our understanding of collective intelligence and decision-making in the animal kingdom.

Tracking individual ants in a natural environment poses significant challenges due to their small size and the great similarity between them. This is where advanced computer vision tools become essential, enabling detailed analysis of movement patterns and behaviors in large group of ants.

This project was born as the continuation of the work carried out by the student Ignasi Nogueiras in his master's thesis and its main objective is the acquisition through computer vision tools of the ant tracks. Also this project is conducted in close collaboration with the Theoretical and Computational Ecology group at the Centro de Estudios Avanzados de Blanes (CEAB), part of the Consejo Superior de Investigaciones Científicas (CSIC) [1]. All the data that this project required had been supplied by CSIC's researchers. This project has consisted of understanding all the fabulous work already realized by Ignasi and develop, test and anlayze the results obtained on the purposed improvements in both the tracking and ant detection systems.

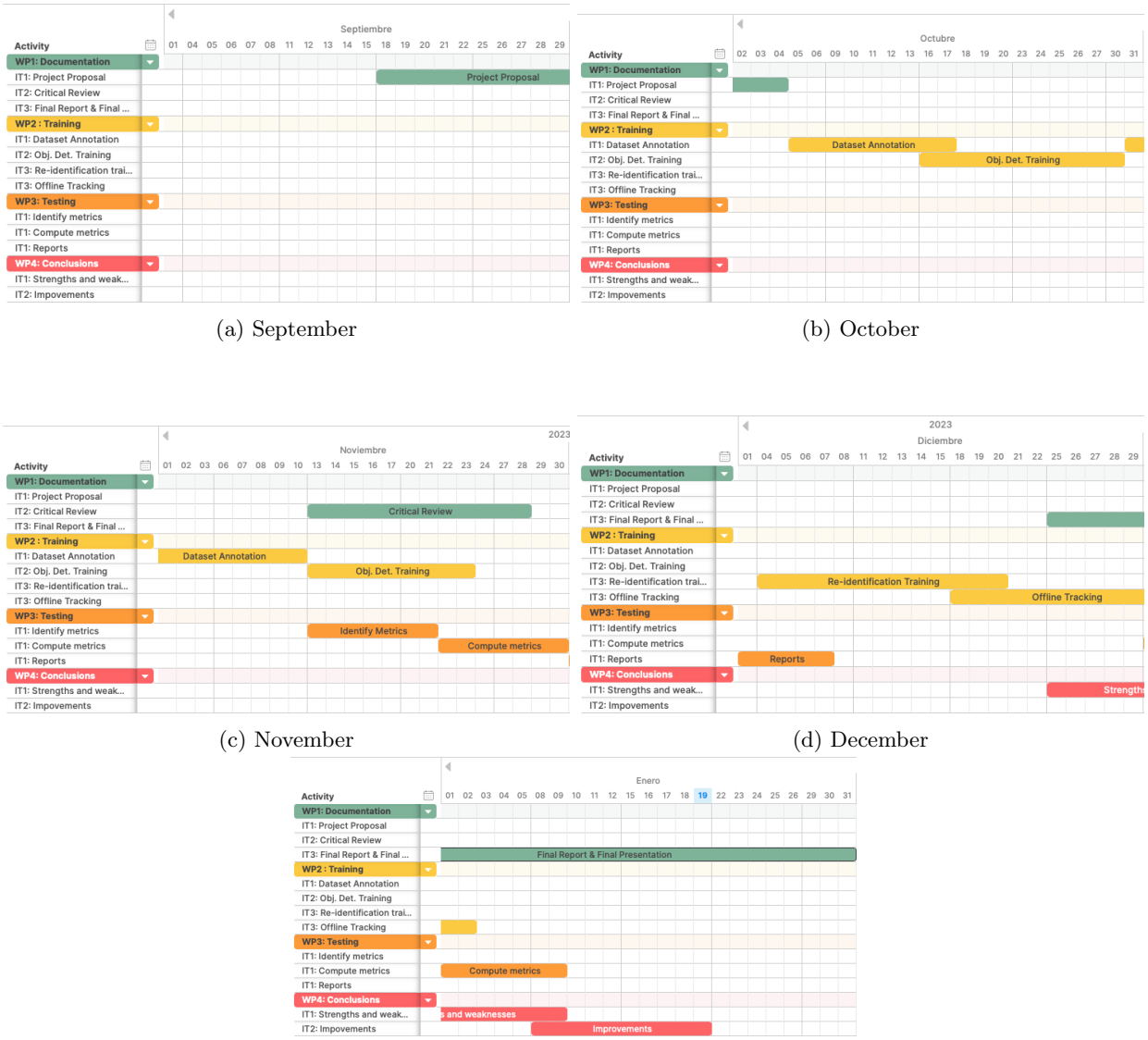
Below, both project requirements and specifications established at the beginning of the project will be detailed:

Project Requirements
- The project aims to enhance an AI-based system's capability to capture and track ant movement from various recordings effectively.
- The system should utilize computer vision algorithms for object detection and object tracking with a considerable degree of accuracy.
- Consider the creation of a dataset for algorithm training by annotating existing ant recordings.

Project Specifications
- The system should deliver high tracking accuracy for ant movement.
- The system should improve tracks in situation of ants crossing themselves.
- Object detection algorithms should effectively identify ants.
- Creation of a comprehensive dataset for algorithm training.
- The dataset should cover situation under the presence of a track where two ants are crossing.

1.1 Work plan

In this section the final Gantt diagram belonging to the complete development of the project will be presented. The primary adjustment involves the extension of the duration for Work Package 1, which focuses on the object detector training. After an initial augmentation of the dataset, it became evident that the ob-



(a) September

(b) October

(c) November

(d) December

(e) January

Figure 1: Final Gantt diagram of the project development

tained results for the Object Detector training were not satisfactory and could be improved. The challenges mainly arose in scenarios involving overlapping ants, and it became apparent that further enhancements were essential to improve the tracker's operation. To address this, additional efforts were directed towards annotating more images to refine the model's understanding of situations with overlapping ants. The aim is to bolster the Object Detector's performance in scenarios that are crucial for achieving substantial improvements in tracker results.

This modification on the work plan became essential to ensure the overall success of the project as addressing these specific challenges will contribute significantly to the efficacy of the Object Detector and, subsequently, the entire tracking system.

Another consideration regarding the established work plan was due to the challenges presented by the training of the re-identification model, having to extend the time previously established for these task, yet all of these setbacks mentioned have not had any notable impact on the development of the project.

2 State of the art

In the rapidly evolving field of computer vision, the development of object detection models, tracking algorithms, and re-identification models represents a cornerstone of progress. Object detection models, exemplified by advanced architectures like YOLOv8 [2], have continually pushed the boundaries of speed and accuracy. Concurrently, tracking algorithms have become more sophisticated, efficiently tracking objects across frames in diverse conditions. Meanwhile, re-identification models have emerged as an innovative approach, enhancing model performance through strategic modifications and techniques. This section delves into the state-of-the-art advancements in these areas, highlighting those systems that that can be correctly adapted to the nature of the project and its needs.

2.1 Object detection models

The field of object detection has seen significant advancements, with YOLOv8 being one of the latest and most notable models. YOLOv8, an iteration in the "You Only Look Once" series, is known for its exceptional speed and accuracy, making real-time detection feasible. It improves upon its predecessors like YOLOv4 and YOLOv5 [3] in terms of both precision and processing speed, thanks to advancements in network architecture and training techniques. Another significant model is Faster R-CNN [4], which introduced the concept of Region Proposal Networks for efficient and accurate object detection. It's known for its high accuracy, especially in cases where detailed detection is critical. Mask R-CNN [5] extends Faster R-CNN by adding a branch for predicting segmentation masks, making it suitable for instance segmentation tasks.

Traditionally, many object detection models, including earlier versions of YOLO, used anchor boxes to predict the location of the targeted objects. These anchors are boxes of different sizes and proportions that the model uses as a reference to detect objects. However, this approach may be less efficient, especially for detecting small or unusually shaped objects, as anchor boxes must be chosen carefully and may not fit well on all types of objects. In contrast, an "anchorless" approach does not depend on these predefined boxes. Instead, the model learns to detect objects directly from the data, which can lead to more accurate and efficient detection. It is especially useful for small objects, since the model is not restricted by the limitations of the predefined anchor boxes and can better adapt to the wide variety of shapes and sizes that these objects can have. For this reason, the YOLOv8 architecture has been established as the object detector for this project.

2.2 Tracking algorithms

Tracking, in our case Multiple Object Tracking (MOT) have as main objective to identify the target objects over time in a video sequence. To carry out this type of tracking there are three strategies to follow that will be introduced below.

Detection-Free Tracking: This technique is based on the manual initialization of the targets that must be tracked and the model searches in the following frames for the target specified by the user.

End-to-End Tracking: Refers to a methodology where the problem of detecting and tracking objects is addressed in an integrated manner, optimizing the entire process together. Unlike traditional deep learning

methods, which break the problem into separate components and combine them heuristically during inference, the end-to-end approach seeks to handle as many aspects of the problem as possible through a machine learning process integrated.

Tracking by Detection: In this case, detection and tracking are treated as optimized processes independently of each other. On the one hand, the detections are obtained by typically an object detection model, and than an algorithm, as the following ones, manages out those detections to predict as accurately as possible the tracks. This is the strategy on which this project focuses to be able to track ants.

2.2.1 SORT

SORT (Simple Online and Realtime Tracking) [6] model had been published in 2016, was born out of the necessity for faster, more efficient tracking systems, SORT revolutionizes the way objects are tracked in video sequences, striking an optimal balance between speed and accuracy.The sytem could be divided in four diferenciated modules:

Detector: In the SORT model, the developers employ a model based on Convolutional Neural Networks (CNNs), which offers superior detection capabilities compared to earlier models.

Estimation Model: On SORT architecture is used as estimator a Kalman filter, which is a recursive filter.

Associator: Resolves the matching of the current frame detections with the track estimations. Is made by a Intersection over Union (IoU) distance matrix and the Hungarian algorithm.

2.2.2 Deep SORT

Deep SORT [7] is a model published in 2017 as an upgrade of SORT. Most relevant innovations were the use of a new associator containing an appearance model. Extracting feature vectors from the detections crops Deep SORT defines a feature space, each track is represented with the set of feature vectors of the last 100 detections of the track. Each new detection is compared with each of the last 100 feature vectors of the each existing tracks and is associated with the track corresponding to the detection with the minimum distance. This enables the associator to combine two distances, as the location distance and the appearance distance. It is one of the first uses of appearance features in tracking algorithms.

2.2.3 OC-SORT

Observation-Centric (OC-SORT) [8], an improvement over the SORT tracking algorithm, was developed to enhance robustness and reduce noise in tracking applications, particularly in situations involving occlusion and non-linear object motion. The OC-SORT model addresses several limitations of the SORT algorithm by being observation-centric rather than estimation-centric.

2.2.4 Strong SORT

Published in 2022, Strong SORT [9] improves the Deep SORT model explained above including several notable features and improvements as could be the use of advanced module for the detection tasks, as

improves the appearence model replacing a simple CNN model used in Deep SORT with the much more efficient Bag of Tricks [10] model which will be mentioned and detailed later in the re-identification section. Instead of keeping in memory the last 100 feature vectors of the detections as is done in Deep SORT, Strong SORT proposes to represent each vector associated with the track as a moving average of the previous feature vectors, which means that when performing the new associations only a comparison must be made for each track instead of the 100 needed in Deep SORT. Also are introduced some improvements to the Kalman filter.

2.2.5 Deep OC-SORT

Deep OC-SORT [11] is an evolution of the OC-SORT model designed to enhance multi-object tracking by incorporating re-identification. It leverages the strength of the pure motion-based tracking approach of OC-SORT and introduces new methodologies to integrate appearance information. This allows Deep OC-SORT to improve tracking accuracy, especially in challenging scenarios such as crowded scenes or when objects are in non-linear motion.

2.3 Re-identification

The re-identification models are an appearance-based components used in much many object tracking algorithms as, for example, in Deep OC-SORT mentioned above.

A notable option for these re-identification models could be: Bag of Tricks (BoT) [10], this model is composed of a backbone, whose function is to extract a feature vectores that then will be passed through a neck which output is a classification output. also propose a set of tricks for training the mentioned network that potentially boost the results. Although the basis of this paper is focused on the re-identification of people, it is considered a potentially good tool for re-identification ants with a proper training.

A considerable implementation of the model discussed just above is the one presented as FastREID [12] used by the Deep-OC-SORT model.

2.4 Tracking Software

Multiple solutions at the software level have already been proposed and there are several software available that propose a solution to the callenge that is treat on this project, although some are more generalized and not so focused on ants. Next, some interesting options will be raised.

2.4.1 AnimalTA

Implemented in 2023 AnimalTA [13] is a application that allows the user get the tracks of the videos uploaded. It relies on methods as background extraction as a kind of object detector which tries to isolate the objects considered as foreground. For the targets association it uses a minium distance criteria above the last observation. Although they show considerable results, in terms of technology tracking models such as SORT or the rest presented in the previous section, theoretically should be more efficient in predicting tracks.

2.4.2 AnTraX

AnTraX [14] published in 2020, is another software solution to the problem presented in this project. Using techniques as optical flow techniques for for those tracks with high certaintly, and than it tries to join the tracks by using appearence descriptors obtained from a CNN-based model.

3 Methodology

In this section all the information related to the development of this project will be detailed. The type of data (video) with which the project was carried out will be detailed first, and the specific way in which these data have been structured so that it could fit into the difference data structure needed by the diferent kind of models that have been used. It will also detail the different metrics / algorithms that have made it possible to quantitatively evaluate the operation of the different proposed solutions. Finally, this section is going to talk about the three core components of this research:

Detection: This component, as its name suggests, corresponds to the "Detection model (YOLO + SAHI)" block shown in Figure 2 and is responsible for locating frame by frame the position of all the ants participating in the scene.

Tracking: This crucial component aims to associate and manage the detections obtained by the detector with the objective of obtaining the different tracks of the ants that take part of the scene. As can be seen from Figure 2, the algorithm used for this component has been OC-SORT.

Re-Identification: This component is based on the concept of appearance and tries to exploit the different appearance between the elements that must be tracked to try to re-identify them. This concept has been tried to exploit to improve the output provided by the Tracker with the "Offline Tracking Method" the last block in Figure 2.

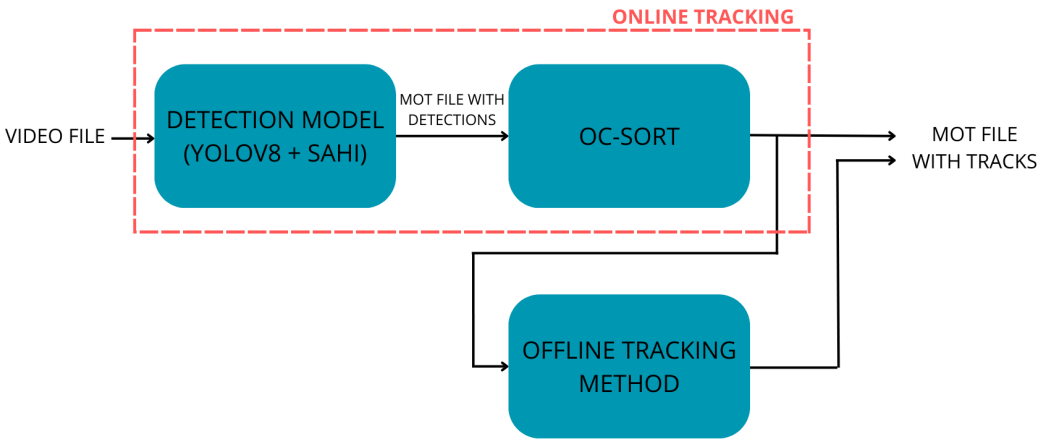


Figure 2: Block diagram of the tracking pipeline

3.1 Data and Datasets

All the data used in the project had been acquiesced by the members of the Research group Theoretical and Computational Ecology of the CEAB-CSIC. It consists of videos of specific characteristics that will be explained below. On Table 1 are shown the characteristics of the video files. Also it is important notice that all the videos are recorded using top-view cameras, this location reduces in a significant way the occlusions which benefice our tracking purpose.

Table 1: Video specifications

Frame width	4000 px
Frame height	2992 px
Channels	Gray scale as RGB
Frame rate	15 fps
Encoding	MPEG4

The **content of the videos** was not the same across the different sets of videos. On one hand, there were six one-hour videos and another six 45-minute videos set in a sort of hexagonal maze where only a single ant appeared. These videos were made to train an appearance model, as it was straightforward and unequivocal to assign IDs to different ants to create a dataset. Since the ants were manually introduced by the researchers and only one ant appeared in each frame, it was easily detectable when an ant had been replaced by another, as one could observe a hand removing it and subsequently introducing a new ant. Additionally, due to their easy and rapid annotation, they have served to create a solid base of annotated images for the detection model's dataset.

The second set of videos featured a significantly higher number of ants and could be divided into two subsets. In one subset, there were 87 ants inside a white box, and in the second subset, there were 46 ants, also within the same white box. The total duration of each of these subsets was one hour, but for easier logging of the annotated content, the original videos were divided into 72 shorter videos. These videos, with their higher density of ants, have been very useful in enhancing the model's performance in situations where ants overlapped, as they allowed us to annotate a considerable number of images containing overlapping ants. On the following Figures 3, 4,5 are shown examples of frames of each set or the detailed videos content in this section.

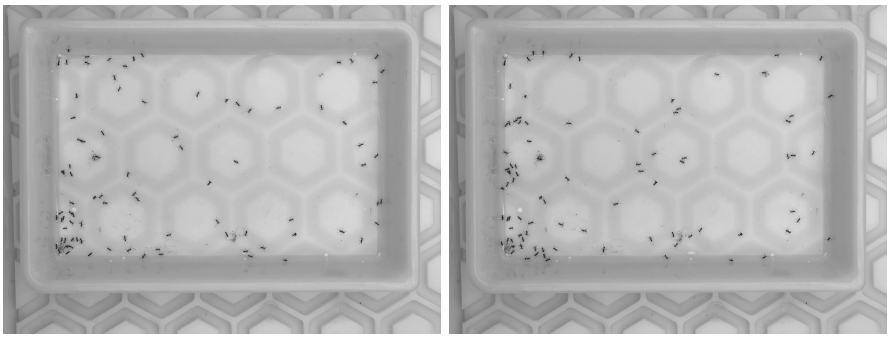


Figure 3: Frames of the set of videos containing 87 ants

3.1.1 Dataset Formats

This project integrates a variety of dataset structures to facilitate the training and application of diverse models. These structures include MOT Challenge format, the YOLO format for image-based object detection and the Market-1501 format for appearance models.

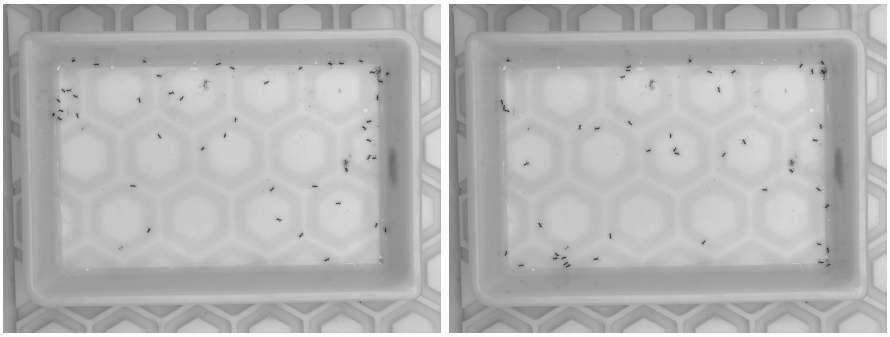


Figure 4: Frames of the set of videos containing 46 ants

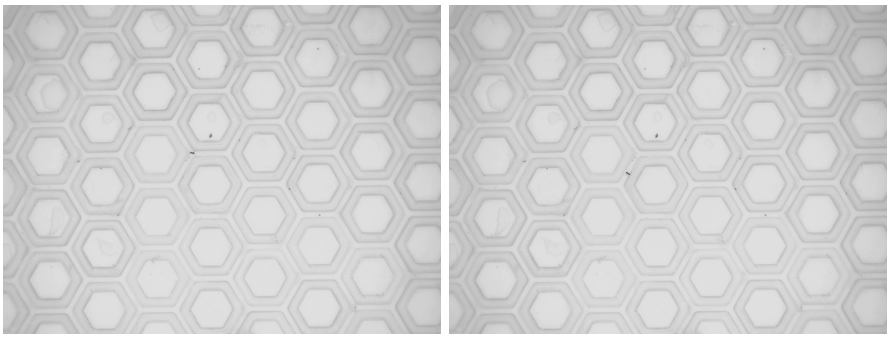


Figure 5: Frames of the set of videos containing a single ant

MOT Challenge

The predominant method of annotation employed is based on the format used in the **MOT Challenge**. This approach is particularly effective for video-based object tracking and detection. The format is operationalized using a CSV (Comma-Separated Values) file for each video. Each file comprises lines with 10 distinct fields:

"Frame Number, Identifier, Left, Top, Width, Height, Confidence Level, 3D-World X Coordinate, 3D-World Y Coordinate, 3D-World Z Coordinate."

For 2D datasets, the 3D-World coordinates are replaced with a -1 value, and similarly, for detection purposes, the Identifier is also set to -1.

YOLO

The YOLO structure is specifically employed for labeling individual images extracted from the video content. Each image correlates with a CSV text file of the same name, containing columns for each detection:

"Category Identifier, Center X-Axis, Center Y-Axis, Width, Height."

The dimensions in these files are adjusted relative to the image size. Furthermore, the YOLO framework necessitates a unique directory structure, dividing images and their corresponding labels into separate subfolders originating from a common root directory. A YAML file is used within this structure to outline the data paths and classify indices for each dataset.

Market-1501

The Market-1501 format is commonly used in re-identification tasks and is a format organized by means of folder structures and number of files. In this format, a crop is made to the ground-truth detections which are identified by the file number. Forming three datasets which are train, test and test query.

3.1.2 Data Annotation

For the annotation of new images, the CVAT (Computer Vision Annotation Tool) has been utilized. This tool is an interactive platform, designed specifically for annotating images and videos in computer vision models. In the scope of this project, the web-based version of the tool was used.

CVAT offers a variety of annotation types, including for object detection purposes or for tracking purposes. It allows for the creation, deletion, and modification of annotations with ease. For object detection, it enables frame-by-frame annotation of different classes (although in our case, only one class was used). In terms of tracking, it also supports frame-by-frame annotation, allowing for efficient management of the IDs of different tracks. Furthermore, CVAT provides the capability to import and export annotated files in a wide range of formats, including MOT and YOLO, which have been used in this project.

Figure 6: Example of CVAT annotation tool for object detection annotation

3.1.3 Datasets for Object Detection

The final dataset for the training of YOLOv8n was originated from the initial dataset of the project, with additional images containing a high degree of overlapping. To acquire these images, it has been used frames from the video set featuring 87 ants and those containing 46 ants.

To select suitable images for augmenting the datasets, a Python script was used. This script required input parameters such as a video, the number of frames desired, the position from the top-hight and top-left point from which a 640x640 crop would be made, and the frame gap. The reason for adding a frame gap,

which indicates how many frames to skip before capturing the next frame, was to ensure a greater variety of images. This was necessary because consecutive frames often showed little variation. These new frames were also annotated using CVAT.

The first dataset and which was the base for the new annotation includes content from the set of videos which include one single ant and also from the subset of 46 ants with a lower frequency.

Dataset	Num. Images	Percentge
TRAIN	8374	69,9%
VALID	3622	30,2%

Table 2: Training - Validation initail dataset distribution

As mentioned above, the initial dataset contained frames either of individual ants or, in much smaller quantities, crops from videos of 46 ants. It is observed that the model trained by Igansi in his thesis could not separate well the instances in which there were ants in overlapping situations, to avoid this it was decided to annotate new images that contained these situations. For this purpose, the script described on the section bottom was used, and a total of 460 new 640x640 annotated images were added to the initial dataset. These images were frames from videos featuring 87 and 46 ants, and the cropping position was manually selected to try to capture a wide variety of overlapping situations among the ants.

Dataset	Num. Images	Percentge
TRAIN	8765	70,4%%
VALID	3691	29,6%%

Table 3: Training - Validation final dataset distribution

To ensure that both training and validation datasets had a balanced number of overlaps and not over or under train the model with overlapping situation, it was calculated the percentage of overlaps in relation to the total number of annotated ants in both train and valid datasets. It is considered as overlaps those bounding boxes with a greater IoU score between them than 0.1.

Dataset	Num. Overlaps	Total Num. Ants	Percentage
TRAIN	1364	19838	6,9%
VALID	517	8158	6,3%

Table 4: Balance of overlaps between training and validation

3.2 Metrics

3.2.1 Object Detection Metrics

An object detector concurrently addresses the challenges of locating and identifying objects. The performance of such a system is commonly evaluated using metrics like precision, recall, F1-Score, and Mean Average Precision (mAP). Before the metrics just mentioned are detailed, it is necessary to pay attention to the following concepts that are used to compute the metrics.

Intersection Over Union (IoU): is a metric used in object detection to evaluate how closely a predicted bounding box matches with a ground truth box. It's calculated by dividing the area of overlap between the predicted and actual boxes by the area encompassed by both boxes. IoU values range from 0 (no overlap) to 1 (perfect match).

True Positives (TP) occur when the detected objects are correctly identified and matched with the corresponding ground truth objects of the same class. This matching often, in object detection context, involves using an Intersection over Union (IoU) threshold. **False Positives (FP)** arise when there is a misclassification of the detected object. This includes cases where an incorrect class is assigned to an object or when the background is mistakenly identified as an object. **False Negative (FN)** refer to situations where detections fail to match with existing ground truth objects, indicating missed detections.

Precision (P) is a metric evaluating the accuracy of the detections made; a high precision indicates strong confidence in the objects identified, yet it doesn't account for objects that were missed. **Recall (R)** assesses the effectiveness in identifying targets, where a high recall signifies a robust capability in spotting relevant objects, but it overlooks the aspect of unwanted detections. **F1-Score** metric is the harmonic mean between precision and recall. This symmetrically encapsulates both metrics, providing a comprehensive assessment of a model's performance.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad \text{F1-Score} = \frac{2 * \text{P} * \text{R}}{\text{P} + \text{R}}$$

Mean Average Precision (mAP): is a metric commonly used to gauge the accuracy of models in computer vision tasks such as object detection. It is particularly prevalent in computer vision due to its ability to provide a comprehensive assessment of a model's performance at various levels of granularity.

"Average Precision" (AP), which is the average precision for a specific object class. Precision is calculated as explained above. However, in practice, precision is assessed at various decision thresholds, yielding a precision-recall curve. For a given class, the AP would be the area under this precision-recall curve. Then mAP is the average of the AP calculated for all classes in a dataset. In the case of object detection tasks, where it might have multiple classes (like 'cats', 'dogs', 'cars'), the mAP gives a single figure summarizing the performance of the model across all these classes. In this project case, AP and mAP are the same due to the fact that this project only treats with only one class.

A crucial aspect of calculating mAP is the use of the "Intersection over Union" (IoU) threshold. Only predictions with an IoU above a certain threshold (commonly 0.5, or 50%) are considered true positives. To calculate AP, it has to be averaged out the precision at various recall points along the curve. Practically, this is approximated by a weighted mean, which can be broken down to a straightforward average of each detected item's precision, assuming all false negatives are assigned a precision value of zero:

$$\text{AP} = \frac{\sum_{i \in TP} P(k_i)}{|TP| + |FN|}$$

Hence, when it's referred to mAP@50, it is talked about the mAP calculated with an IoU threshold of 50%. Similarly, mAP@50-95 refers to the average of the mAPs calculated at various IoU thresholds, from 50% up to 95%, in 5% increments.

Precision and Recall of overlapping situations: This metric aims to determine the precision and recall in scenarios where there is overlap among elements of the same class. It is currently implemented for datasets with a single class, as was the case in this project, but it could be easily scaled for multi-class situations. The primary focus is on detecting instances where there is overlap in both the predictions and the ground truth. This is identified by assuming that if two bounding boxes (bbox) share an Intersection over Union (IoU) score greater than 0.1, an overlap is present.

After thoroughly analyzing the ground truth data and the model's predictions, and identifying overlaps among their bounding boxes, these overlapping cases are compared between the ground truth and the predictions. If the overlapping bounding boxes from both ground truth and predictions have an IoU score of more than 0.8, the overlap is considered to be effectively predicted and thus classified as a True Positive (TP). Situations where the ground truth and predictions have an IoU score of less than 0.8 are classified as False Positives (FP). Finally, cases where an overlap is not detected in the predictions but is present in the ground truth are determined as False Negatives (FN), see Figure 7 for graphic examples.

This metric has been useful in determining whether the model shows improvement in cases of overlap, which it was suspected that would also enhance the results of the tracking metrics.

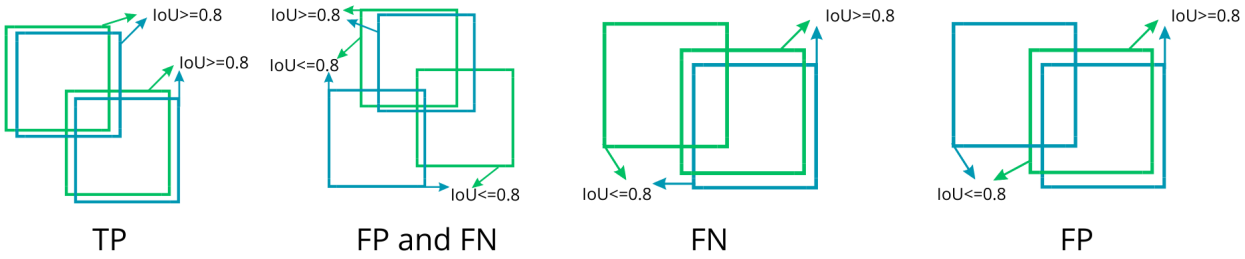


Figure 7: Example of TP and FP, green bbox is considered as ground-trurh and blue bbox as predicted bbox

3.2.2 Tracking Metrics

Measuring tracking performance is difficult task. The three following factors must be taken into account:

Localization: measures the spatial alignment from the predicted detections and the ground-truth ones. Localization IoU (Loc-IoU) is defined as the ratio of intersection from two detections and the whole are covered for both. Localization Accuracy (LocA) can be defined as the average of all the Localization Intersection over Union (Loc-IoU) scores for all matching pairs of predicted and ground-truth objects.

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

Detection: measure the target objects which are found and its alignment with the ground-truth detections. It had to be defined previously a localization threshold to consider that two detections intersect (for example could be used the Loc-IoU). A one-to-one optimal matching had to be done using the Hungarian algorithm because predicted detections may match with more than one of the ground-truth. Finally the matching detections are considered as True Positives (TP), those that are only predicted and are not found in the ground-truth are considered as False Positive (FP) and the ground-truth ones that not match with any predicted will be considered as False Negatives (FN).

$$\text{Det-IoU} = \frac{|\text{TP}|}{|\text{TP}| + |\text{FP}| + |\text{FN}|}$$

Association: measures how well the found objects are temporally aligned with their target identity. This can be done by considering the number of TP matches between two tracks, called True Positives Associations (TPA). Those remaining detections on the predicted track are considered False Positives Associations (FPA). Finally, any remaining detections on the ground-truth are considered as False Negative Associations (FNA). As can be seen, Figure 8 tries to exemplify this concept visually.

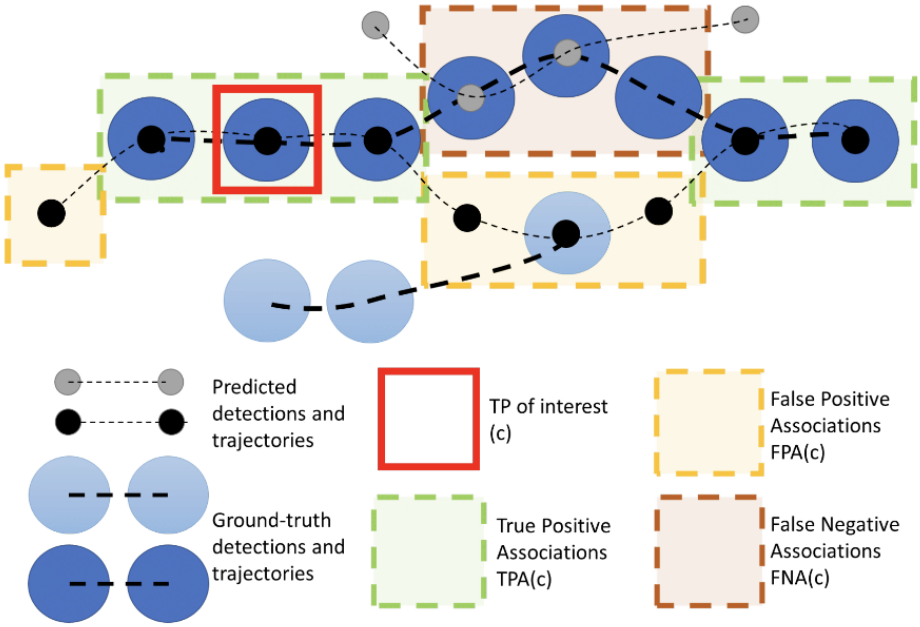


Figure 8: Visual example of TPA, FNA and FPA

The overall Association Accuracy (AssA) can be determined by calculating the mean Ass-IoU for all corresponding pairs of predicted and actual detections across the dataset.

$$\text{AssA} = \frac{1}{|\text{TP}|} \sum_{c \in \text{TP}} \text{Ass-IoU}(c)$$

$$= \frac{1}{|\text{TP}|} \sum_{c \in \text{TP}} \frac{|\text{TPA}(c)|}{|\text{TPA}(c)| + |\text{FNA}(c)| + |\text{FPA}(c)|}$$

HOTA (High Order Tracking Accuracy)[15] [16]: All three components—localization, detection, and association—are crucial for successful tracking, thus their measurement is essential. Nevertheless, for an overall performance comparison, a singular metric is often preferred. HOTA serves this purpose by integrating all three of the IoU scores previously defined.

$$\text{HOTA}_\alpha = \sqrt{\text{DetA}_\alpha \cdot \text{AssA}_\alpha} = \sqrt{\frac{\sum_{c \in \text{TP}_\alpha} \text{Ass-IoU}_\alpha(c)}{|\text{TP}_\alpha| + |\text{FN}_\alpha| + |\text{FP}_\alpha|}}$$

$$\text{HOTA} = \int_0^{\alpha \leq 1} \text{HOTA}_\alpha \, d\alpha \approx \frac{1}{19} \sum_{\alpha=0.05}^{\alpha=0.95} \text{HOTA}_\alpha; \quad \alpha_{t+1} = \alpha_t + 0.05$$

We calculate the HOTA over a range of diffetent Loc-Iou thresholds represented by α. For each threshold vaule it is calculate the geometric mean of the DetA and AssA which ensure that they are evenly weighted in the final score.

3.2.3 Re-Identification Metrics

To evaluate the appearence model behaiviour can be used the **mAp** which has been seen before and also the rank metrics.

Rank metrics: The k-rank metrics assess true positives (TPk) based on whether a query ant has any correct match within its k closest appearances. The ultimate measure derived from this is the accuracy for a specified set of queries.

$$\text{Rank}_k = \frac{|\text{TP}_k|}{|\text{queries}|}$$

3.3 Detection model

3.3.1 YOLOv8n

The YOLOv8n architecture is characterized by its compact design with approximately 3.2 million parameters, making it ideal for environments with limited computational resources. It incorporates a multi-level backbone that improves the detection of objects of different sizes, and uses pyramidal processing to increase accuracy by integrating contextual information from features at different levels. Each level has an independent detection head, allowing more precise localizations, and the architecture stands out for its efficiency in deployment and training, supported by the Ultralytics library, which makes it accessible to a wide range of users in the field of computer vision.

3.3.2 YOLOv8n training

For the training of YOLOv8n a script was developed using Ultralytics library and its methods which allow to train easily the model. This methods work with a YAML file where were specificated the paths for the train and validation dataset as information about the number of classes (only one for this project) and the name of each class. On Table 5 are going to be detailed the values of the hyper-parameters for the training.

Hyper-parameter	Value
Epochs	80
Patience	20
Batch size	16
Optimizer	auto
Learning rate	0.01
Momentum	0.937
Dropout	0
Scale	0.5
Flip left-right	0.5
Mosaic	1.0
Hue	0.015
Saturation	0.7
Value (Brigthness)	0.4

Table 5: YOLOv8n hyper-parameters

The first group of hyper-parameters are related to the training process and to the model itself. The training had a duration of 80 epochs with the fact that if during 20 epochs the validation metrics did not show any improvement would it be produced an early to avoid any over-training of the model on the training data. Also Ultralytics selects the most suitable optimizer for training based on the dataset characteristics, enhancing efficiency and adaptability without manual tuning. This feature simplifies usage and can potentially improve model performance. For instance, it may choose an optimizer that converges faster for one dataset or better handles local minima for another. The use of dropout was not necessary as the model provided convincing results without it.

The second group of hyper-parameters detail the data augmentation which was realised based on the real dataset images, as referring as data augmentation to a modification of the real images of the dataset applying them some transformation as by scaling, flipping or not use in this case but rotation. This technique is used with the purpose of expanding the variety of data with a reduced data size to obtain new images from those already available. Some examples of images and augmented images are shown in Figure 9.

Once the YOLOv8n was trained, it was used SAHI (Slicing Aided Hyper Inference) library [17] which applies tot the original image a slicing window. This method it allows to divide the original image into overlapped sub-images of it, without having to reshape the images. In this projects cases this was util due to the fact that original frames of the videos had a shape of 4000x2992 pixels, any downscaling of this images would probably make disappear most of the ants from the image.

3.4 Appearence model

The Bag of Tricks (BoT) was the elective as appearance model for this project, utilized in advanced algorithms like Strong SORT and Deep OC-SORT. It consists of two parts: a backbone and a neck. The backbone, a 50-layer Residual Network (ResNet50) enhanced with non-local blocks, extracts features. These features are then processed in the BoT neck through a sequence of operations, including batch normalization and generalized mean pooling, to produce appearance and classification metrics. This architecture is chosen for its ability to capture complex spatial and semantic relationships, underscoring its importance in state-of-the-art tracking solutions. The implementation discussed in the State of the art section of FastREID has

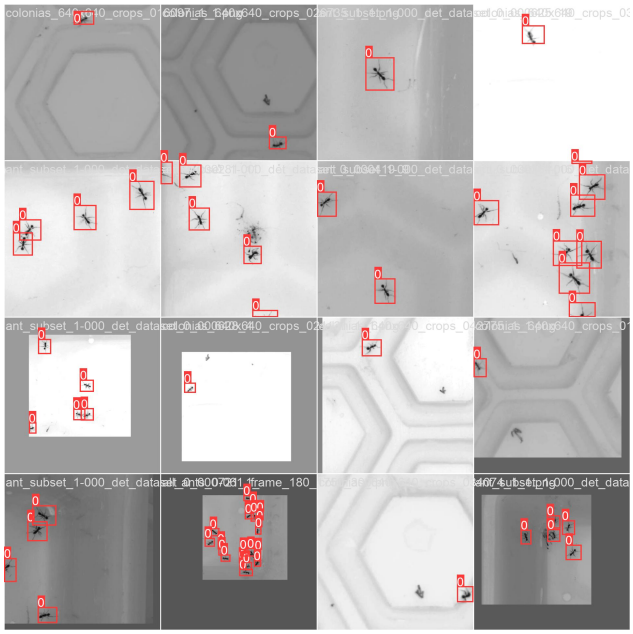


Figure 9: Samples of the augmented traing data

been used, with the only modification applied to the source code to allow images in the '.png' format since it was the format in which they were found in the dataset.

3.4.1 Bag of Tricks training

The most extensive ant appearance dataset utilized already existed and comprised 186 distinct identities. These were split equally with 93 designated for training and the remaining 93 for validation purposes. Within the validation set, 20% of the identities were categorized as unknown, serving as true negatives. Ultimately, the training set encompassed a total of 6730 images. The validation query set, on the other hand, consisted of 489 images that were to be allocated among a larger pool of 5795 images as is shown in Table 6.

Dataset	Identities	Images	Percentage
Train	93	6730	51.1%
Validation - Test Images	93	5795	44.0%
Validation - Query Images	74	489	3.7%

Table 6: Bag of Tricks Market1501 Distribution

In Table 7 it can see the configuration of parameters and hyper parameters used among the possibilities allowed by the source code.
The most notable thing to comment on is the second group of parameters, which refer to data augmentation, the REA [18] parameter (Random Erasing Augmentation), which consists of randomly erase a rectangular section of the images. This technique is proposed as a training trick in the BoT paper mentioned in the state of the art section.

Parameters	Value
Epochs	120
Images per batch	64
Optimizer	Adam
Learning rate	0.00035
Momentum	0.9
Loss	Cross-Entropy Loss, Triplet Loss
Backbone	Resnet50
Flip (probability)	0.5
REA (probability)	0.5

Table 7: Bag of Tricks hyper-parameters

3.5 Tracking models

3.5.1 OC-SORT

OC-SORT improves the Kalman filter in its motion model by freezing its state until new observations are assigned, reducing estimation noise in skipped frames. The location score is refined into a global direction metric, calculated by considering the most relevant older observation and the newest observation to determine a track's global direction. This direction helps compute a normalized angular score, crucial for track-detection alignment. In the association stage, OC-SORT categorizes detections into high and low score groups and uses a combination of Intersection over Union (IoU) and angular score for assignment, applying a threshold on IoU. OC-SORT's multi-stage approach effectively refines detection and tracking, especially in complex scenarios.

The scripts to use OC-SORT were developed yet on the base-line GitHub of the project.

3.5.2 Offline Tracking

Leveraging the fact that the tracking system did not have to be Online, meaning real-time tracking, so could be Offline tracking, which allows to apply more algorithms and post-processing techniques to the tracks. This approach forms the basis of the system detailed in Figure 10. Taking as input the MOT file gived by the OC-SORT model and utilizing the existing appearance model, the following technique was proposed to avoid ID changes in ants during their crossings.

Figure 10: Block diagram for the offline tracking system proposed

Initially, taking advantage of improved detection of overlapping ants with the new training of the YOLO, all tracks where ants crossed would be split into two, this would be detected because two bounding boxes (bboxes) or more would have an IoU score between them higher than a certain threshold in the input MOT

file as exemplified in the Figure 11. This MOT output file of the "Split Tracks" block maintains the same format as the MOT input file, the only thing to note is that instead of using an Identifier, as the original format does, the Track ID is used in order to give a unique identifier to all the tracks that have been split. A side effect of this splitting of the tracks will be that all those detections/tracks that overlap with other ants will be lost. But given the nature for which this tool should be used by CSIC researchers, this effect is not harmful since the main thing is to know how ants evolve and avoid identity changes.

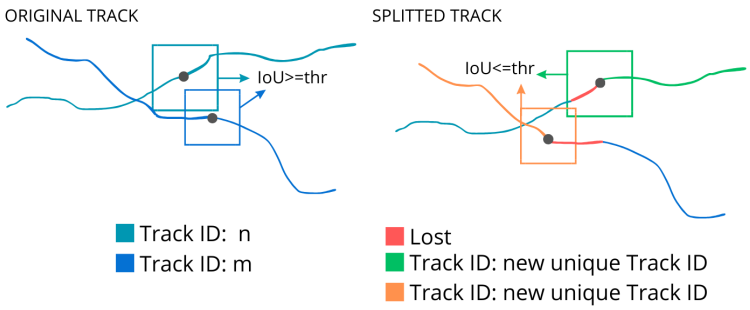


Figure 11: Example split tracks and new Track ID association

Once the MOT file with split tracks have been obtained and as now the detections are only containing ants without overlap, their feature vectors would be extracted using the appearance model trained. With this nuance it is possible to obtain an improvement to solutions proposed by algorithms such as Strong-SORT since in this proposed case, for the association of tracks only the feature vectors belonging to images where only one ant appears are considered. In situations where two or more ants appear the feature vectors are not significant since the appearance model is trained with full one-ant content images. For example in the case of Strong SORT these feature vectors extracted from overlap situation negatively affect the moving average between all the previous feature vectors of the track. Finally is computed the average of all those feature vectors associated with the same TrackID. When all Track ID had associated a single vector, the final block joins the Track ID following a minimum Euclidean distance criteria.

It is important to note that since the appearance model was not sufficiently reliable, a spatial distance threshold between the track association was also added to prevent that tracks with large distances between them but similar-looking ants being joined. Also another parameter of the system was a Euclidean distance threshold between tracks with different TrackID from which it no longer joined said tracks with a higher Euclidean distance between their feature vectors. To optimize the value of these hyper parameters of the system, a bash script was created which return the metrics such as HOTA, DetA, AssA, LocA for relevant values of this two parameters.

4 Results

On the following section are going to be presented the results obtained by using the best models trained both in the detection and tracking tasks. This results are going to be analyzed by the values obtained from the metrics explained in the previous section. The final tracking results are based on 150 frames annotated a subset of the video which contains 87 ants. Just note that this video was not part neither the training or validation dataset of any trained model.

These results will break down the training results of both the YOLOv8 and the re-identification model. As well as going into more detail about the operation of the YOLOv8 on overlap situations, which was essential to validate its training. And finally, in the tracking part, quantitative and qualitative analyzes will be presented to try to compare the operation of the two systems that have been proposed.

4.1 YOLOv8n Training

Two training sessions were conducted using the YOLOv8n model: the first with the project's base dataset and the second with a dataset containing a greater number of images featuring overlaps. The first training was performed on CALCULA, the "Teoria del Senyal i Comunicacions" computing server and it used 1 GPU NVIDIA GeForce GTX 1080 Ti with 11GB of memory and 16 cores of CPU and it lasts 7 hours and 15 minutes. The second one was also performed in CALCULA with the same amount and type of resources and it lasts 6 hours and 17 minutes.

The following figures 12, 13 and 14 are referring to the second train, the one with more overlapping cases.

From curves shown in Figure 12 can be appreciated a first fluctuating zone for the validation set on the first epochs due to the lack of optimization that the model have. As epochs advanced validation loss seems to stabilize as both training and validation losses decrease little by little. Finally at epoch 80 the training ended up. It can be seen how in the epochs close to 80 a small distension begins to be observed between the validation loss and the train loss, with the train loss continuing to decrease while the validation loss seems to stagnate. This phenomenon could mean some first symptoms of overfitting of the model on the training data. Therefore, it was correct to not continue training for additional epochs. Moreover, with the 'patience' hyperparameter set to 20 epochs, the training would have encountered an early stop to prevent overfitting in case the validation metrics did not improve during these 20 epochs.

On Figure 13 it is shown the mAP@50-90 for the validation set, also contains a fluctuation on the first epochs as the losses, but then stabilizes and continuous increasing until reaching a value a little higher than 85% in the last epoch.

Finally Figure 14 present the F1-Score metrics for different confidence thresholds, confidence is understood as a parameter returned by the model associated with each detection that aims to express, as its name indicates, the confidence with which each prediction is made. From this graphics it can be determined that the optimal confidence threshold for the inference would it be nearby 0.5 which is the value that maximizes F1-Score which consequently jointly maximizes Precision and Recall.

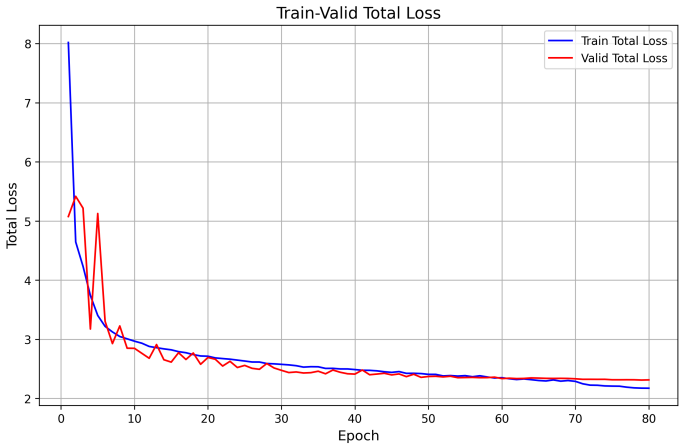


Figure 12: Train and validation loss curves of YOLOv8 training.

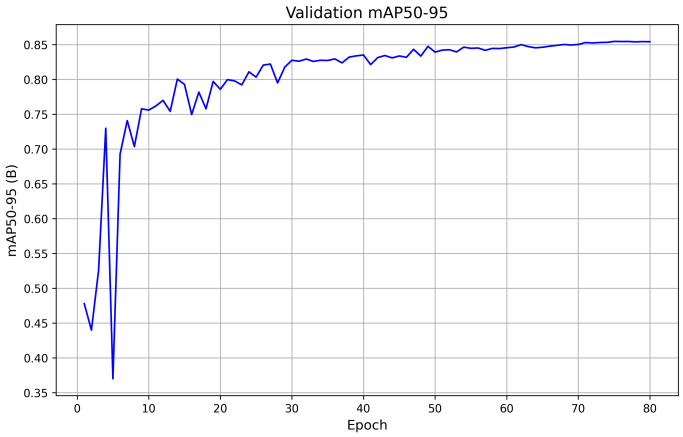


Figure 13: Validation mAP@50-95 of the final YOLOv8n

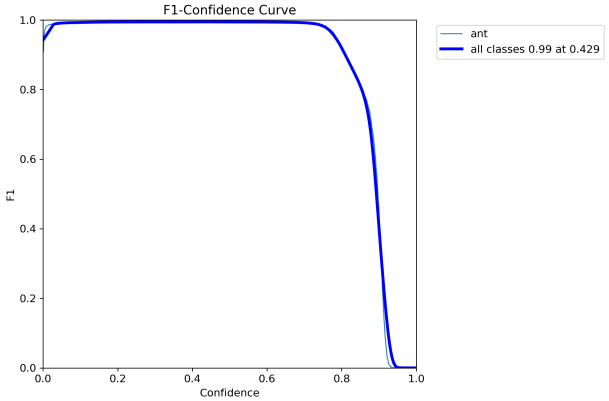


Figure 14: Validation F1-Score - Confidence curve of final YOLOv8n

Overlap situations metrics

All of these metrics are obtained on the same validation dataset and try to make a comparison between the two models trained and its behaviour in overlapping situations. The dataset used in the two results is the validation of the final ones because it contains more overlapping situations and that is the focus of our study.

Sum. GT Overlaps	517
Sum. Pred Overlaps	588
TP	388
FP	198
FN	129
Precission	0.75
Recall	0.66

Table 8: Model trained with original dataset

Sum. GT Overlaps	517
Sum. Pred Overlaps	502
TP	428
FP	74
FN	89
Precission	0.83
Recall	0.85

Table 9: Model trained augmented dataset

As it can observed, the instances of correctly detected overlaps have increased considerably, while the occurrences of both non-detections (FN) and false detections (FP) have decreased. This is reflected in the significant improvement of Precision and Recall, as shown in Table 9 compared to Table 8.This enhancement could potentially boost the tracking results.

4.2 Bag of Tricks training

In this section the result of the training of the appearance model will be shown. It can be seen in Figure 15 that a rapid convergence can be seen in the training data. Regarding the metrics of the validation set, a maximum peak of 72% can be seen in Rank-1, while what refers to mAP is 26%, as shown by the curves in Figure 16. Analyzing together the behavior of the metrics in validation and the progress of the train weights, it can be seen that the metrics in validation do not decrease as the model is optimized on the training data, so it can be concluded that training has not suffered overfitting on training data.

It should be noted that in several extra trainings that were done with different optimizations of the hyperparameters where longer trainings (more epochs) were used, they ended up showing the presence of overfitting as well as worse metrics on the validation dataset.

The BoT training, as the YOLO ones, was performed on CALCULA servers. Using a total of 4 GPUs NVIDIA 1080 Ti and 8 cores of CPU. With this resource the training lasts less than 20 minutes.

4.3 Tracking Results

Finally, it is time to analyze the global operation of the different proposed tracking systems. The following table will present the metrics obtained by on the annotated test video.

It can be seen in Table 10 how the increase in cases of overlaps between ants in the object detection trainer has, as proposed, greatly improved the HOTA score of the OC-SORT tracker. This improvement is probably

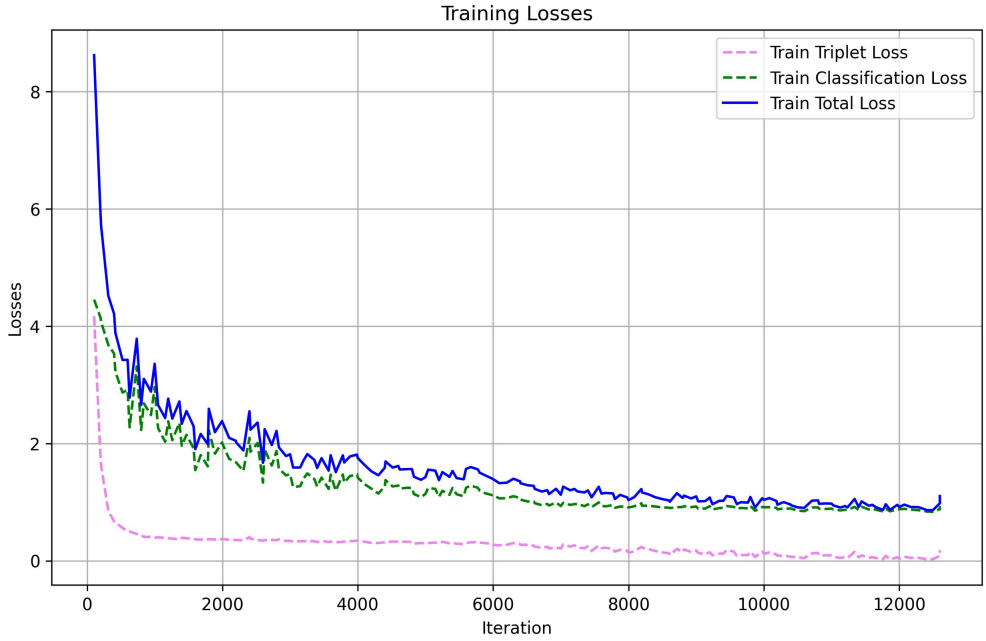
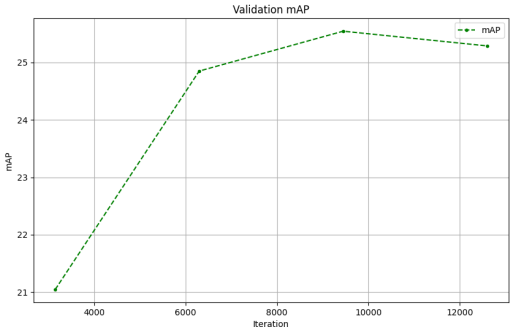
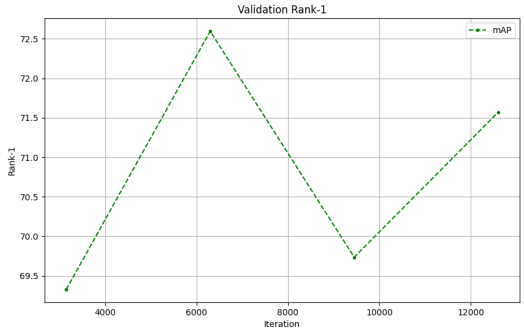


Figure 15: Train Losses for the BoT model training.



(a) Validation curves for mAP of BoT training



(b) Validation curves for Rank-1 of BoT training

Figure 16: mAP and Rank-1 in validation set

due to the fact that always detecting cases in which ants come together is much easier for OC-SORT to approximate their trajectories based on the previous ones and obtain better results in situations of crossings between ants. Also is relevant to see how the tracker using the predictions of the YOLO trained with more overlapping situation had decrease in a nototrious way the number of ID swictches (**IDSW**), which refers to the number of ants which had change the ID during the tracking process in realtion to the GT tracks.

Regarding the case of the offline tracking system proposed in the methodology section, which proposed dividing the tracks by the overlaps and joining them by the criterion of minium Euclidean distance between the average of the feature vectors that the appearance model provided, it can be first seen that the results obtained to date have not managed to improve the result of the system's own input which is the OC-SORT predictions. It should be noted that, using this method, HOTA is penalized by the elimination of said

System	IDSW	LocA	DetA	AssA	HOTA
YOLO (Trained with less overlap situations) - OC-SORT	60	72%	42%	53%	46%
YOLO (Trained with more overlap situations) - OC-SORT	21	48%	61%	74%	53%
Offline tracker	34	77%	37%	56%	45%

Table 10: Tracking metrics

tracks in an overlap situation, so metrics such as DetA or AssA are penalized even if the tracks are joined correctly after the overlap, during the time in which the ant is in a situation of overlapping detections and consequently the track is not predicted by the model in those frames

System	IDSW	LocA	DetA	AssA	HOTA
Offline tracker	22	78%	46%	61%	52%
YOLO - OC-SORT	9	76%	39%	59%	48%

Table 11: Tracking metrics non considering Overlaps in GT

First of all, it must be clarified that the two results shown in Table 11 belong to the same systems that in Table 10. However, they differ in the way in which the ground truth is determined. In Table 10 results are shown with the original ground truth containing full tracks for the ants, while in Table 11 all those detections that suffered overlap have been extracted from the ground truth making a partition of the track as the Offline Tracking does. This modification has been made, as commented above, to avoid possible effects on the metrics due to the fact that in moments of overlap this system eliminates the detections for those frames. In addition, it also intended to combat the nature of the test video itself, which contained about 4-5 ants which remained immobile and were never detected as they were always overlapping between them. With this considerations in the ground-truth can be seen that HOTA had increased in a considerable way, up to 52%, for Offline Tracking surpassing that of the YOLO - OC-SORT model which has suffered a decrease of up to 48%, which makes sense since the HOTA of 53% previous YOLO - OC-SORT where all ground-truth detections and tracks were considered, takes advantage of all those detections and tracks of immobile ants which for the final purpose of our tool are of little interest. From these new metrics in both systems also it can be seen a decrease of 12 in the case of ID switches, which indicates that these 12 ID switches that now did not occur were happening during the interval time which the occlusion between the ants lasts, but once occlusion ended up, if ended, the track was once again associated with the corresponding ID. Related to the ID-Switches it can also be seen how YOLO-OC-SORT still produces a lower HOTA than Offline Tracking is changing fewer IDs, which should be analyzed in depth to determine what this factor is due to.

A qualitative visual analysis will be conducted to compare some potential differences between OC-SORT and the Offline tracking model. For this analysis, a smaller segment of the test video was selected. This segment notably features a significant reduced number of ants that were not in overlapping and stationary positions as some of the complete video were. It's important to note that, for this analysis, it had not been used the final track files from the complete video. Instead, it has been specifically processed the chosen video segment to ensure that only the subset of ants in it is considered for track association via feature vectors. During this small-scale analysis, it was observed that the system can be effective in some cases. However, splitting the tracks means temporarily disregarding the ants' movement estimation and observation. As a

result, the system's reliability heavily depends on re-identification. This approach works well when feature vectors can clearly distinguish between different ants. But, in cases where they do not, it might lead to significant errors, resulting in completely incorrect track associations even when trying to mitigate them using a spatial distance threshold for track associations.

5 Sustainability Analysis and Ethical Implications

On the following section is going to be evaluated the environmental, social and economic impacts that arise from the development of the project, as well as the possible ethical implications that its implementation entails. This critical analysis guarantees the alignment of the project with the principles of sustainability and ethical responsibility in scientific research.

5.1 Sustainability matrix

The sustainability matrix presents the result of the evaluation of the Ant Tracking project in three large blocks: Environmental, Economic and Social. For each of the blocks, an assessment of the development, project execution and risk and limitations processes has been carried out.

	Development	**Project Execution**	**Risk and Limitations**
Environmental	Environmetnal impact	Environmetnal impact	Environmental risks and limitations
Economic	Total cost of the development	Viability	Economic risks and limitations
Social	Personal impact	Social impact	Social risks and limitations

Table 12: Sustanability Matrix

5.1.1 Environmental impact

Development

To quantify the ecological impact of carrying out the project first of all it is going to estimate the total amount of energy consumed in kWh. It represents the energy consumption of the resources involved in the development of the project. Where the energy consumed is equal to the power of the resource times the time of use or work.

$$Ec = P * T$$

First, a calculation of the hours dedicated to the project by all the resources involved will be carried out, in this project's case computers and graphic processing units. It is assumed the use of two computers, one for the Developer with a workload of 6 hours a day, while one for the Computer Vision Expert with a workload of two hours per week. Also doing an approximate calculation of the hours that the GPU has been running, it can be estimated them to a total of 37 hours.

Device	**Power**	**Hours**	**Consume**
Computer Expert	60 W	40 Hours	2,4 kWh
Computer Developer	60 W	600 Hours	36 kWh
GPU	180 W	37 Hours	6,66kWh
Total	-	-	**45,06kWh**

Table 13: Energy consume calculated for the project development

In order to reduce energy consumption, it has carried out term energy practices with responsible consumption. To avoid having computers in use when they are not working, use the GPU's olny when it has been

required and in obvious cases that new training could improve the model's operation or adapt the brightness of the screen to the needs of the environment.

In terms of carbon footprint, through some data collected on said footprint in the activities related to this work, the following calculations have been made:

Through data collected from Mike Berners-Lee's book "How Bad are Bananas?: The Carbon Footprint of Everything" [19], the following metric has been established for the estimation of the footprint of sent email:

- Email without attachments: 0,3 g CO2eq/email
- Email with attachments;: 50 g CO2eq/email

Taking into account the overall number of emails send in the development of this project which approximately have been 80 without attachments and 10 with attachments a **total estimation of 534 g CO2eq** have been generated in relation with this activity.

Another aspect to take into account is the video calls made, through data collected from the book mentioned in the previous estimate, an average cost of 0.164gCO2 [**empty citation**] per minute per person involved in the meeting is determined. Taking into account that the average duration of the calls has been about 30 minutes, in which a total of 3 people have participated and that they have been made once a week, bringing the total to 20 video calls. **The total estimate has been about 295.2 g CO2eq**.

Assuming the use, as has been in this project case, of a MacBook Pro to carry out this project, which entails a production cost at the level of CO2 emissions of a total ranged between 208/227 kg CO2eq [20]. Making the following estimate assuming the average value of 217.5kg CO2 and a useful life for the device of 7 years:

$$217.5 \text{ kg CO2} \ / \ 7 \text{ years} = 15{,}5 \text{ kg CO2eq} \ / \ \text{year}$$
$$15{,}5 \text{ kg CO2} \ / \ \text{year} = 1{,}77 \text{ g CO2eq} \ / \ \text{hour}$$

Assuming the hours detailed in the power calculation for the use of computers, which amounted to a total of 640 among all participants, it is estimated that the generation of emissions related to the production of the devices would raise **total of 1,13 kg CO2**.

Finally for the use of electrical resources that it has been consumed, and taking into account an estimate of 273g CO2eq/kWh according to data from the "Generalitat de Catalunya" [21]. Therefore, relating this to the estimate made of a consumption of 45.06kWh consumed gives a **total of 12,3 kg CO2eq**.

Execution

Since this is a project with a clear research purpose and not focused on the development of a final product, it can be assumed that the final product is the complete Pipeline through which CSIC researchers will be able to obtain tracks from the ant videos they want. Since the pipeline can operate without the need for a GPU, only the hours of computer consumption expenditure should be computed. Assuming an average execution

Activity	CO2 generated
Emails	534 g CO2eq
Online meetings	295,2 g CO2eq
Device production	1,13 kg CO2eq
Power consumed	12,3 kg CO2eq
Total	**14,26 kg CO2eq**

Table 14: Carbon footprint

time of the best pipeline (YOLO-OC-SORT) of approximately 0.03 frames per second without consuming GPU resources and with a video ingestion as the provided for the project development. Taking into account that the system can run without the need to use GPU resources, the savings compared to the development phase of the project will already be significant.

If the analysis is carried out considering only using computer resources and taking into account the carbon footprints and the power consumes of the development section it has been obtained the following expression which can be estimated the CO2 emissions per frame processed by the system:

60 W * 33,3 seconds/frame * 273 g CO2eq/kWh = **0,15 g CO2eq / frame**

Risks and limitations

Environmentally there is the risk of increasing the ecological protection of the project as the use of the tracking pipeline expands or grows, even if is want to use a GPU device for a faster execution of the pipeline to obtain the tracks, the ambiental footprint will be harmed. It must also be taken into account, the context of constant development within the framework of the project, such as the field of computer vision, where new solutions and algorithms are constantly being considered and if it wanted to be updated with these new milestones it would be necessary to keep the pipeline in constant evolution, which would mean new training, new research, which would lead to the consumption of many more resources at a similar level to those used for the development of this project.

5.1.2 Economical Impact

Development

To estimate the total cost of the project, it will be based on the data used in the environmental section, which specified the hours dedicated by each participant in the project. Additionally, it should be noted that the estimated hourly rates for the different ranks of those involved in the project are as follows: Computer Vision Expert at 60 euros per hour, and Developer at 10 euros per hour.

Personal	Rate	Hours	Cost
Computer Vision Expert	60 €/hour	40 Hours	2400€
Computer Vision Developer	10 €/hour	600 Hours	6000€
Total	-	-	**8400€**

Table 15: Cost of the Human Resources

Applying a lineal amortization to the costs detailed in Table 16, and assuming a useful life for computer devices of 7 years and for the GPU of 5 years, applying the following expression:

Device	Units	Unit Cost	Total Cost
Computers	2	1100€	2200€
GPU NVIDIA GTX 1080 Ti	1	760€	760€
Total	-	-	**2960€**

Table 16: Cost of the Device Resources

Anual Amortization = Total Cost / Useful life
Anual Computer Amortization = 157,14 € / year
Anual GPU Amortization = 152 € / year

Applying to the amortizations calculated above the fact that approximately 6 months have been dedicated to the project, the updated cost taking into account the amortization of the device resources is updated in the Table 17.

Device	Units	Amortization	Total Amortization
Computers	2	78,57€	157,14€
GPU NVIDIA GTX 1080 Ti	1	152€	152€
Total	-	-	**309,14€**

Table 17: Costs of the Device Resources with Amortizations

Finally, as indirect costs related to the project, the option of renting a table in a coworking space is considered, with a monthly price for a table of €165. Taking into account this amount and a duration of 6 months of the project, **the total indirect costs amounts to 990€**

Therefore, merging the human costs showed in Table 15 with the devices amortizations detailed in Table 17 and the inderect costs the total cost of the development of this projects rises to a **total amount of 9699,14€**.

Execution

In this project, the scenario of incurring costs related to adjustments, updates, and repairs of pipeline scripts has not been a primary consideration, as the project is focused more on research rather than on the development of an optimal software for the manage of the tracks. The primary objective is to facilitate research and data analysis, with less emphasis on creating a software product that would require ongoing maintenance and updates. Given this research-oriented approach, the software developed is considered more as a tool to achieve specific research outcomes rather than a continuously evolving or commercially deployable product.

Risks and limitations

The main risk and limitation that could affect this project could be the rapidly evolving in the field of computer vision which means that the technology might become outdated quickly, impacting long-term economic sustainability. Since this is a research-oriented project rather than the commercialization of the tool, the future viability of the project is not a primary issue.

5.1.3 Social Impact

Development

This project, focusing on computer vision, emphasizes the importance of upholding high personal, professional, and ethical standards, ensuring respect for diversity and equity. The language used throughout the research is inclusive and non-sexist, reflecting our commitment to respectful and mindful communication.

In the field of computer vision, there is currently rapid technological advancement, coupled with an increasing debate on ethics in artificial intelligence and responsible data usage. This work is situated within this context, trying to balance innovation with relevant ethical and social considerations.

Execution

This project's exploration of ant behavior through advanced computer vision tools could help CSIC's researcher to offers valuable insights for understanding and improving ant social dynamics. By analyzing the cooperative and organized nature of ants, it can be drawn parallels to human societal structures and behaviors. The findings could potentially inform strategies to enhance community collaboration, resource management, and organizational efficiency. Such insights, gleaned from a seemingly distant natural world, reinforce the importance of interdisciplinary research in addressing complex social challenges in human society.

Risks and limitations

Given that the project deals exclusively with the analysis of ant behavior through video data, the likelihood of it being detrimental to any specific segment of the human population is minimal.

5.2 Ethical Implications and Sustainable Development Goals

Within the scope of this Bachelor's thesis, which focuses on the development of a computer vision tool for ant tracking, some of the main objectives aligned directly or indirectly with the Sustainable Development Goals (SDGs), specifically SDG 9 (Industry, Innovation, and Infrastructure), SDG 11 (Sustainable Cities and Communities), and SDG 15 (Life on Land). The needs addressed by this work include the advancement of scientific research methods, the improvement of ecological monitoring techniques, and the support of biodiversity conservation efforts. These needs have been defined by the CSIC's researchers, and this project aims to meet them by providing innovative computer vision tools to enhance the understanding of the ants ecosystem.

In terms of professional ethics, this project upholds the principles laid out in the code of ethics associated with ecological research and computer science. This includes ensuring that the tracking of ants is non-invasive and does not harm the subjects or their habitat (SDG 15), maintaining integrity in data handling and analysis, and ensuring that all outcomes are transparently reported for peer review and public knowledge.

The tool developed could be used to inform sustainable urban development (SDG 11) by understanding ecological balances within urban environments. Additionally, the advancement of research infrastructure (SDG 9) through technological innovation could have applications in other fields of ecological and environmental research.

By addressing these ethical considerations, the project not only contributes to the field of computer vision and ecology but also aligns with the ethical pursuit of science and technology for a sustainable future.

6 Conclusions

This project, as is said on the introduction, was born as the continuation of the work previously carried out by Ignasi in his thesis with the aim of improving a computer vision tool capable of tracking ants. Analyzing the excellent work carried out previously, two intertwined lines of new developments were established, on the one hand the improvement by the detection model of occlusions between ants and on the other an Offline tracking system was proposed taking advantage of the virtues of re-identification models.

To address this entire strategy, it was necessary first of all to obtain a fundamental part, if not the most, for the future of the project, new annotated data that was used to train the YOLOv8n.

To address this entire strategy, it was necessary first of all to obtain a fundamental part, but what is more than just new annotated data that met the condition of containing overlapping ants that were used to train the YOLOv8n. With this new training, it was able to verify that although the difference between the two models trained with or without the new data, although the mAP@50-95 in validation did not improve very noticeably, going from 83.44% to 85.97% but when was quantified the behavior of both in cases of overlap with the development of a metric that determined the Precision and Recall of the overlaps, it was verified that with an approximate increase of 400 images with high content of overlaps the model was capable of detecting them with much greater reliability. With this new detection model was tested together with OC-SORT and it was found that by improving the detections of ants crossing each other the tracking result presented a notable improvement, going from a HOTA of 46% to 53%.

Once this first development was implemented, it was the moment to start with the next one. The first thing consisted of training a re-identification model like BoT. Once the training results were analyzed and the model with the best performance was chosen. After that the project continued with the implementation of the Offline Tracker that has been detailed in this document. Although at the level of metrics it has not presented improvements with respect to the YOLO - OC-SORT system, it has been determined that if discriminating feature vectors between ants are achieved it is an option to consider.

The entire base code of the original project plus all my contributions developed within the framework of this thesis can be found in the Git Hub of the project called AntTracking [22]

In conclusion, my journey through the exploration of computer vision in the context of my final degree thesis has been both intellectually enriching and personally fulfilling. This field, which captivates my interest, has presented numerous challenges, each of which has been a stepping stone towards a greater understanding and proficiency. The process of overcoming these challenges has not only deepened my knowledge but also enhanced my problem-solving and analytical skills. It has been an immense pleasure to immerse myself in a subject that is both fascinating and evolving, offering a window into the future of technology. The experience gained through this project has been invaluable, contributing significantly to my academic and professional growth. I am eager to continue contributing to and evolving within this dynamic area of study.

7 Future Work

Finally, and as the last section of this report, some guidelines for future work will be detailed based on the work carried out and the conclusions obtained from the results.

- With this project it has been determined that an increase in overlap situations in the training data of the object detector has greatly improved the performance of said detector in the detection of overlapping ants, which has consequently allowed the Tracker to also improve notably its results. With this it can be concluded that a good line of progress would be to obtain new annotated data to consequently re-train the model to suggestively continue improving the tracking result.
- Although the result of the Offline tracking system has not managed to improve in terms of ID-Switches the YOLO - OC-SORT model, and it has been determined that with good discrimination between the feature vectors of different tracks, it could give more optimal results, with which the focus of the study should be on improving the Re-identification model. With this objective in mind and taking advantage of the fact that the videos are recorded at the same point, a reshape could not be applied to the crops of the detections to try to exploit the differences between the size of the ants in the feature vectors. Another option to consider would be to prepare a recording configuration where less space will be covered in order to obtain closer images of the ants, which would allow the model to be trained with higher quality images and more detail about the ants, which could conclude with richer features and greater discrimination between the appearance of different ants.
- Another key point would be the acquisition of data, referring to videos, in situations similar to those that the system must operate for the relevant investigations that they want to carry out. Which will not be made in the context of the ants inside the white box like the videos available, but will be made in the context of the videos where a single ant appears. It should be verified how it is suspected that the proposed system presents an improvement due to the non-massification of immobile ants, which today is where the current system presents its main defects.
- In parallel to the offline Tracking system, the emergence of new Tracking algorithms descended from OC-SORT could be investigated, with an improvement in re-identification since it is used. In some models such as Strong SORT or Deep-OC-SORT they could further exploit the functionality obtained with OC-SORT. Obviously with a constant review of the State of the art in relation to these technologies due to the growing research and research on these technologies.

List of Tables

List of Figures

List of Acronyms

AI Artificial Intelligence.

AssA Association Accuracy.

BoT Bag of Tricks.

CEAB Centro de Estudios Avanzados de Blanes.

CNN Convolutional Neural Network.

CPU Central Processing Unit.

CSIC Consejo Superior de Investigaciones Científicas.

CVAT Computer Vision Annotation Tool.

DetA Detection Accuracy.

FN False Negatives.

FP False Positives.

GPU Graphics Processing Unit.

HOTA Higher Order Tracking Accuracy.

IoU Intersection Over Union.

LocA Localization Accuracy.

mAP Mean Average Precision.

MOT Multiple Object Tracking.

P Precision.

R Recall.

REA Random Erasing Augmentation.

SDG Sustainable Development Goals.

SORT Simple Object and Realtime Tracking.

TP True Positives.

YAML Yet Another Markup Language.

YOLO You Only Look Once.

References

[1] *Centro de Estudios Avanzados de Blanes (CEAB), Consejo Superior de Investigaciones Científicas (CSIC*. https://www.ceab.csic.es/es/.

[2] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. *YOLO by Ultralytics*. Version 8.0.0. Jan. 2023. URL: https://github.com/ultralytics/ultralytics.

[3] Zheng Ge et al. *YOLOX: Exceeding YOLO Series in 2021*. 2021. arXiv: 2107.08430 [cs.CV].

[4] Shaoqing Ren et al. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2016. arXiv: 1506.01497 [cs.CV].

[5] Kaiming He et al. *Mask R-CNN*. 2018. arXiv: 1703.06870 [cs.CV].

[6] Alex Bewley et al. "Simple online and realtime tracking". In: *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, Sept. 2016. DOI: 10.1109/icip.2016.7533003. URL: http://dx.doi.org/10.1109/ICIP.2016.7533003.

[7] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. *Simple Online and Realtime Tracking with a Deep Association Metric*. 2017. arXiv: 1703.07402 [cs.CV].

[8] Jinkun Cao et al. *Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking*. 2023. arXiv: 2203.14360 [cs.CV].

[9] Yunhao Du et al. *StrongSORT: Make DeepSORT Great Again*. 2023. arXiv: 2202.13514 [cs.CV].

[10] Hao Luo et al. *Bag of Tricks and A Strong Baseline for Deep Person Re-identification*. 2019. arXiv: 1903.07071 [cs.CV].

[11] Gerard Maggiolino et al. *Deep OC-SORT: Multi-Pedestrian Tracking by Adaptive Re-Identification*. 2023. arXiv: 2302.11813 [cs.CV].

[12] Lingxiao He et al. *FastReID: A Pytorch Toolbox for General Instance Re-identification*. 2020. arXiv: 2006.02631 [cs.CV].

[13] Violette Chiara and Sin-Yeon Kim. "AnimalTA: A highly flexible and easy-to-use program for tracking and analysing animal movement in different environments". In: *Methods in Ecology and Evolution* 14.7 (2023), pp. 1699–1707. URL: https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/2041-210X.14115.

[14] Asaf Gal, Jonathan Saragosti, and Daniel Kronauer. "anTraX, a software package for high-throughput video tracking of color-tagged insects". In: *eLife* 9 (Nov. 2020), e58145. DOI: 10.7554/elife.58145.

[15] Jonathon Luiten et al. "HOTA: A Higher Order Metric for Evaluating Multi-object Tracking". In: *International Journal of Computer Vision* 129.2 (Oct. 2020), pp. 548–578. ISSN: 1573-1405. DOI: 10.1007/s11263-020-01375-2. URL: http://dx.doi.org/10.1007/s11263-020-01375-2.

[16] Jonathon Luiten. "How to evaluate tracking with the HOTA metrics". In: (2020).

[17] Fatih Cagatay Akyon, Sinan Onur Altinuc, and Alptekin Temizel. "Slicing Aided Hyper Inference and Fine-Tuning for Small Object Detection". In: *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, Oct. 2022. DOI: 10.1109/icip46576.2022.9897990. URL: http://dx.doi.org/10.1109/ICIP46576.2022.9897990.

[18] Zhun Zhong et al. *Random Erasing Data Augmentation*. 2017. arXiv: 1708.04896 [cs.CV].

[19] Mike Berners-Lee. *How Bad are Bananas?: The Carbon Footprint of Everything*. https://carbonliteracy.com/the-carbon-cost-of-an-email/.

[20] Apple Inc. *Product Environmental Report, 13-inch MacBook Pro*. Tech. rep. 2020.

[21] Generalitat de Catalunya. *Factor de emisión de la energía eléctrica: el mix eléctrico*. https://canviclimatic.gencat.cat/es/actua/factors_demissio_associats_a_lenergia/.

[22] Ramon Morros, Ignasi Nogeuiras, and Pol Serra. *AntTracking*. Jan. 2023. URL: https://github.com/imatge-upc/AntTracking.