GIMA
Geographical Information Management and Applications

Module 7:

MSc Thesis

Multi-Object Tracking livestock method driven by foundation models in computer vision using UAV videos

Final version

Responsible professor:
Kooistra Lammert

Supervisor:
João Valente

Student:
Ricardo Ruiz Sánchez

Multi-Object Tracking livestock method driven by foundation models in computer vision using UAV videos

This thesis was typeset using (R) Markdown, LaTeX and the `bookdown` R-package

An online interactive version of this thesis that includes the used code is available at https://rruiz-s.github.io/thesis-gima-html/, licensed under a to-be-determined license.

Multi-Object Tracking livestock method driven by foundation models in computer vision using UAV videos

Module 7: Thesis GIMA
Faculty of Geosciences

Master thesis report

A thesis submitted in partial fulfilment
of the degree of Master of Science GIMA
Geographical Information Management and Applications
In the presence of committee appointed by the GIMA thesis coordinators,
for public defense in Delft University of Technology
on Thursday 19th June 2024, at 12:30 o'clock

Mr

Ricado Ruiz Sánchez

born in Granada

GIMA Supervisors:

Responsible professor:	prof. dr. K. Lammert	Wageningen University
Supervisor:	prof. dr. J. Pereira Valente	Wageningen University
Reviewer:	– – to-be-determined	to-be-determined

Faculty: Faculty of Geosciences

Contents

Chapter 1

Introduction

The first chapter of this master's thesis introduces the research in four sections. The "Context" and "Field Domain" sections describe how drone services applied to livestock are relevant for resource-efficient management strategies and some of the current problems with their use. The third section, "Research questions", also includes the research framework and its design. The last section, "Literature Review", explores the state-of-the-art supervised and self-supervised machine learning methods, where foundation models are the product of the latest.

1.1 Context

In the face of scarce resources and an increasingly challenging environment, drone services can shape a more resilient society by providing resource-efficient management strategies. These challenging environments, which bring uncertainty, are regarded as an opportunity to grow under the new European Green Deal (Hainsch et al., 2022). In fact, the recent Granada Declaration highlights resource efficiency, resilience to natural disasters, and adaptation to climate as central priorities for a common future for the benefit of all [1].

Initiatives like SAID aim to bring UAV (unmanned aerial vehicles) technology closer to people [2], promoting the application of drone services in real-world scenarios. This pretends to reduce the gap that exists between the rapid pace of technological development and its use in current livestock farm practices (Bewley et al., 2015). Likewise, this project, situated under the framework of ICAERUS, also shares this common objective, to facilitate the application of UAV technology by showcasing the use of drones on several topics [3]

[1] Granada declaration: https://europa.eu/!3c9B84.
[2] SAID initiative: https://saidlab.wur.nl/
[3] ICAERUS programme: https://icaerus.eu%7D%7BICAERUS%20programme

1.2 Field domain

This study pertains to precision livestock farming, where technology provides better management strategies by measuring physiological, behavioural, and production indicators (Bewley et al., 2015). In the past, implementation of oestrus detection or automation of milking robots were some of the earliest applications of technology, which were mostly applied in intensive farms (Belanche et al., 2019). Nowadays, although radio frequency tags along with global positioning systems, and accelerometers are frequently used in pasture-based systems, remote sensing information obtained by UAV technology is becoming more relevant (Aquilani et al., 2022). For example, in the domain of computer vision, images obtained by UAV and processed by artificial intelligence have been used to count livestock, reducing labour costs and significant errors (Sarwar et al. (2018); Shao et al. (2019); Bárbulo Barrios et al. (2024)).

1.2.1 Socio-economic challenge

From an economical point of view, the current manual system to count livestock could be yielding an annual loss of AUD 12,000,0000 (Rančić et al., 2023). Additionally, depredation by large carnivore species in Europa such as wolves (*Canis Lupus* Linneus 1758) is one threat of missing sheep (Gervasi et al., 2021). In fact, domestic animals such as sheep or goats constitute the main prey of wolves in the South Mediterranean Region (Linnell et al., 2020).

From an ethical perspective, lost sheep can be found trapped, which can lead to a long and painful death (Raquel Zarraoa et al., 2022). Employing drones for remote sensing purposes also allows sustainable pasture management, a strategy to face some challenges, such as extreme events like droughts increased by climate change. Therefore, improving livestock monitoring simply by an accurate counting contributes to better management strategies. For example, assigning the right number of livestock depending on the state of the ecosystem encourages more sustainable management. A specific case is found in the Swiss Alpine Commons, where exceeding the allotted grazing rights includes financial penalties (Ostrom, 2012). Nowadays, a positive impact is brought by initiatives such as “Open2Preserve”, which reduces fire risks by guided grazing[4]. Similarly, negative impacts that reduce biodiversity, such as the “wet desert” observed in the Scottish Highlands due to a too high sheep-grazing pressure (Marrs et al., 2020) could be minmised by better pasture management.

[4]Open2Preserve:https://open2preserve.eu/en/noticies/proyecto-open2preserve-desarrollo-de-modelos-de-gestion-sostenible-que-disminuyan-el-riesgo-de-incencios-en-espacios-abiertos-de-montana/.

1.2.2 Technical challenge

On the technical side, the computer vision problem addressed in this project aims to detect and assign a unique ID to each animal for tracking, also known as Multi Object Tracking (MOT) (Vayssade et al., 2023). This task is accomplished through machine learning, a branch of artificial intelligence that builds computer systems capable of adapting and learning from experience (Zhou, 2017). Deep learning networks used for this task include the You Only Look Once (YOLO), region-based convolutional neural network (R-CNN), single-shot multibook detector (SSD) or CNN (Cheng et al., 2022). These detectors are fine-tuned by adding Neural Architecture Search (NAS) (Qiao et al., 2023), using the Mask-R-CNN network to detect objects in low contrast scenarios (Xu et al., 2020b) or employing aerial thermal imagery (Bárbulo Barrios et al., 2024).

Despite the variety of architectures available and advances, when a YOLOv8 model trained on the public dataset "Aerial Sheep Dataset" (Riis, 2022) is applied to the project's dataset, the predictions are far from acceptable, as shown in Figure 1.1. Out of 50 sheep, 278 are counted at the end of the video. These results confirm some previous conclusions from related work, such as the difficulty of counting sheep with UAV technology due to the tiny size of the objects (Sarwar et al., 2018). Similarly, the number of false positives in the test confirms that over-counting, which leads to overestimation of the population, is another common error (Hollings et al., 2018). Even when adding a tracking algorithm to ensure that the object is counted only once (van Gemert et al., 2015), the results are not sufficiently close to the ground truth to satisfy farmers' needs.

However, the rapid development of machine learning technologies is not only attributed to new-state-of-the-art algorithms and architectures (such as YOLOv8), but also to the availability of big data for training the model. To understand the current situation, it is important to recognise that today's novel machine learning age traces back to the 1960s, when the lack of data constrained the development of this technology. In fact, one common current recommendation in computer vision using machine learning is to acquire richer and higher-quality datasets to improve the feature extraction ability for detecting objects (Cao et al., 2023); Sarwar et al. (2018); Bárbulo Barrios et al. (2024); Xu et al. (2020b)).

These richer and higher-quality datasets come with a cost. The cost of these large, annotated datasets is high, as supervised learning requires manual feature labelling that increases labor costs and generalisation errors (Pu et al., 2022). To face this problem, new foundation models trained by self-supervised learning do not require any human annotators as input (Xu et al., 2020a). As they are trained with web-scale datasets, these models are able to perform zero-shot detection.

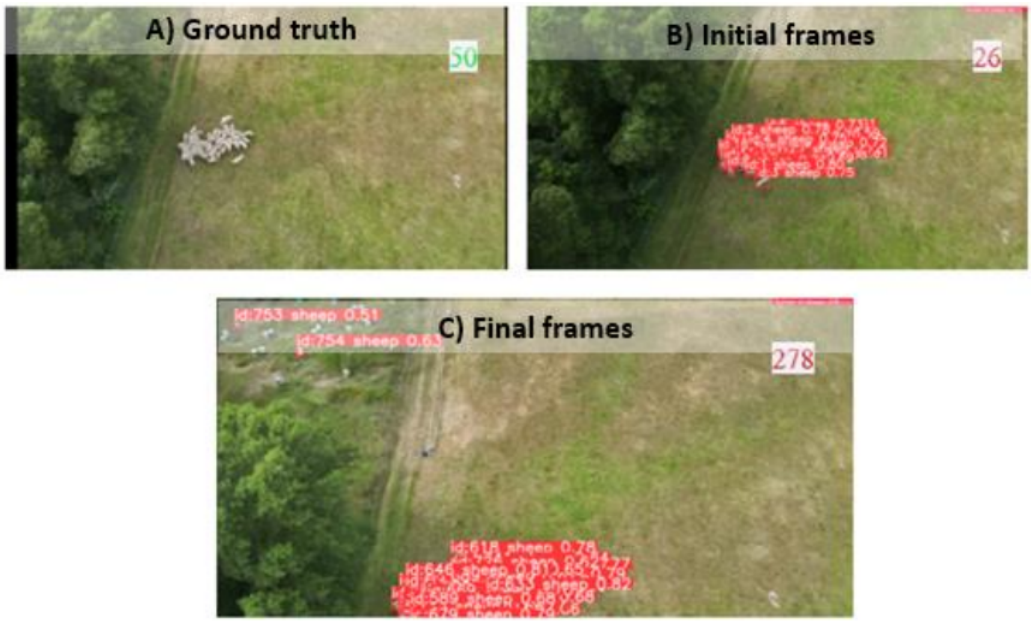


Figure 1.1: The ground truth (A) shows a herd of 50 sheep. As time progresses, the counting algorithm approaches the ground truth with a value of 26 shown in the initial frames (B). In the final frame (C) the counting algorithm overpasses the ground truth value, reaching 278 sheep due to duplicates (over-counting) and false positives identifying honey panels as sheep.

To the best of our knowledge, the application of foundation models in the agricultural sector, specifically for livestock identification and counting, still represents uncharted territory, except for the recent segmentation anything model (SAM) applied to livestock (Noe et al., 2023).

1.3 Research questions

This research aims to develop a machine learning-based system for counting livestock in real-world scenarios using video captured from UAV. To achieve this goal, the research comprises three specific goals: 1) To review the state-of-the-art machine learning models for multi-object tracking and their application in the agriculture field. 2) To assess the performance of the novel foundation model SAM when compared to previous neural networks such as YOLOv8 by using counting the evaluation metrics MAE y NAE 3) to design a strategy for precise livestock counting within the drone service market across heterogeneous real-world scenarios.

Therefore, (a) a literature review of the SOTA supervised and unsupervised methods, (b) yields a range of foundation models comprising three compounds for a machine learning system to count livestock. These components are detection, tracking and counting. (c) This system is applied to different scenarios classified by its metadata, weather conditions, and attributes testing the model generalization. Figure 1.2 illustrates this research framework. These are the research questions that arise from this research framework:

1. What are the opportunities for multi-object tracking systems to count livestock with supervised or self-supervised learning methods?
2. How do the state-of-the-art foundation models (self-supervised) perform when compared to previous supervised neural networks such as YOLOv8 measured by counting metrics such as MAE or NAE?
3. What is the model's performance on livestock counting under different weather conditions and scenarios? How much does the model generalise?

To answer these questions, Figure 1.3 illustrates the conceptual design that describes the logical steps taken in this study:

An initial literature review shows the opportunities for multi-object detection and tracking systems applied to livestock using supervised and self-supervised models. Based on this literature, a set of criteria is applied, resulting in the selection of foundational models to detect livestock on images and then track them on videos. Lastly, the metadata included in the UAV footage classifies the data, creating different scenarios categories according to the weather conditions.

Apart from the metadata, the size of the sheep in the recording, how close they are to each, other and if occlusion is frequent were used to define three scenarios. The

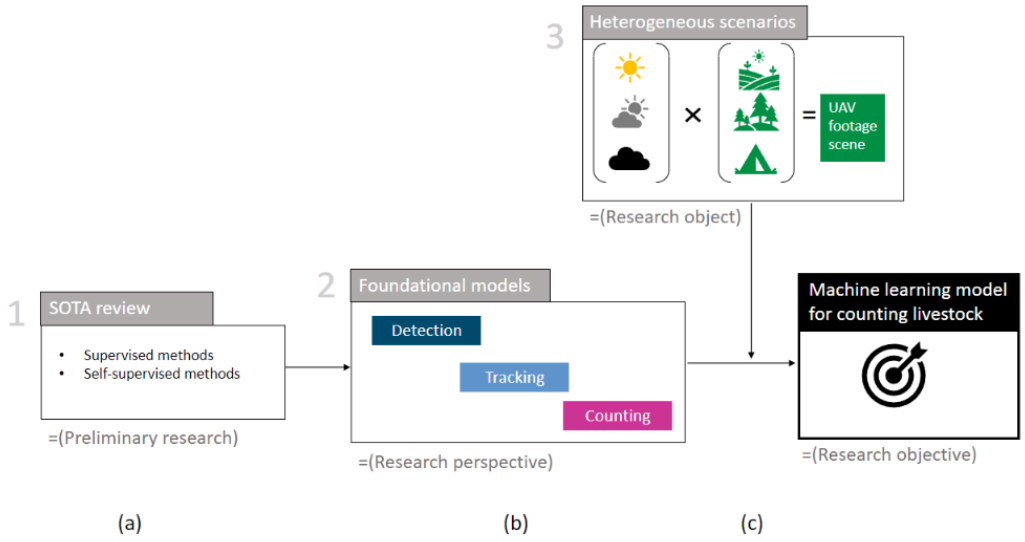


Figure 1.2: The research objective is to design a machine learning model for counting livestock based on preliminary research focusing on a perspective of foundation models applied in different heterogeneous scenarios. The foundational model comprises three tasks: to detect, track, and count livestock.

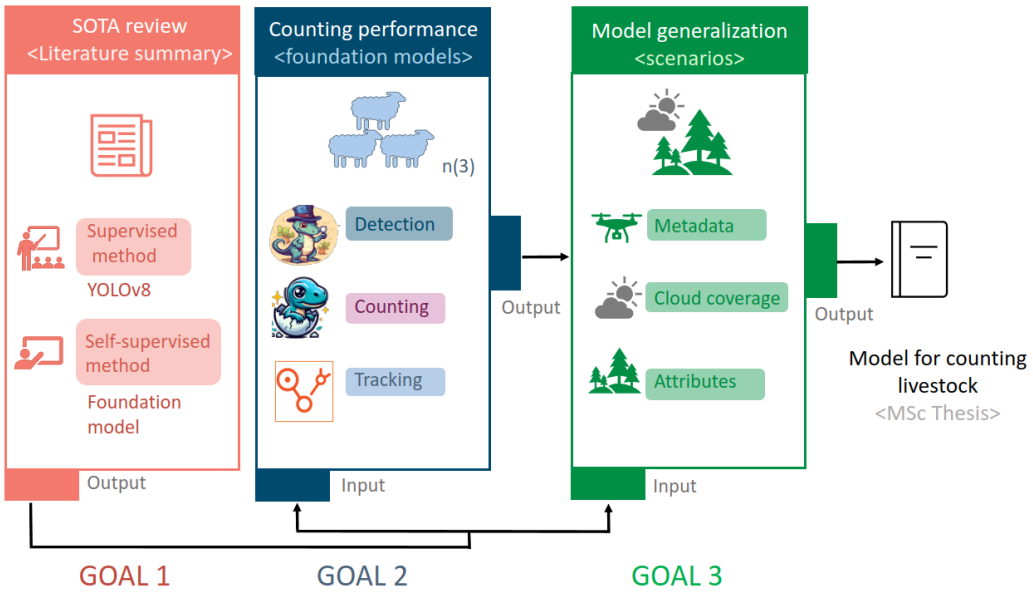


Figure 1.3: A literature summary provides the information to measure the counting performance and generalisation ability of the model for counting livestock. Three foundation models are used across a series of scenarios.

scenario "Small herd" is related to the small target detection (Wang and Jin, 2023), defined as 32x32 pixels objects by MS-COCO metric evaluation (Tong et al., 2020), which can be observed in pasture scenes. When sheep are overlapping, the scenario is categorised as "Dense herd" (Wang and Jin, 2023), which can be observed on crowd pen scenes. The last scenario, "Occluded herd" is related to camouflaged object detection as objects integrate in their surroundings (Tang et al., 2023), which can be observed in forest scenes. These scenarios along with their weather conditions, allow us to make a comparative case study to test the model generalisation performance.

In terms of research design, the study opts for a depth in view of self-supervised methods such as the foundation models applied to precision livestock farming rather than a breadth view of supervised deep learning models and their different architectures. One of the reasons for this decision relies on its potential to overcome one of the barriers to the implementation of deep learning models in different fields, the lack of labelled data (Arweiler et al., 2023). Examples of the implementation of these foundational models range from Zero-shot object counting (Jiang et al., 2023b) to generating pseudo labels for pre-training thermal infrared image segmentation tasks (Chen and Bai, 2023). Therefore, to train the machine learning system for counting livestock in the ICAERUS use case, a foundational model is adapted rather than learning task-specific models from scratch. This approach follows the paradigm shift reported in Geospatial Artificial Intelligence, where foundational models transfer their knowledge across domains without task-specific training data (Mai et al., 2023).

The research takes a quantitative approach to measuring the model's performance, but also a qualitative approach by characterising the UAV footage based on attributes. In fact, some UAV datasets for object detection include qualitative attributes such as weather conditions (Yu et al., 2019). Likewise, the difference in attributes in the scenarios had a quantitative measured effect on the cattle detection, as the evaluation metric F-measure varied its values from 0.952 to 0.713 using the same machine learning model (Shao et al., 2019).

1.4 Literature Review

Multi-Object Tracking (MOT) consists of detecting and tracking all the objects in the video, giving a unique identifier to each individual object (Aharon et al., 2022). There are different strategies to achieve this goal. The most dominant is tracking-by-detection, where targets are detected in a single frame and then associated by their trajectories (Wang et al., 2019). This study also follows this strategy, where the initial object detection is a prerequisite to carrying out the tracking. Supervised and self-supervised are the two categories used to classify machine learning systems in the literature review, being foundation models those created through self-supervised methods.

The next two subsections review the supervised and self-supervised learning methods. In the supervised learning section, there is a classification of two families of object detectors; one-stage models and two-stage models. For the one-stage model, the model YOLO is selected due to its abundance in livestock research, and then different architectures are described. For the two-stage model, R-CNNs models are selected among others and described. For the self-supervised learning section, three foundational models are described. For the detection task, Grounding DINO is explained along Segment Anything Model (SAM) and Grounded-SAM. For counting and tracking, T-Rex and SAM-PT are briefly introduced. These foundation models describe the state-of-the-art machine learning models involved in the multi-object tracking for livestock designed for this study.

1.4.1 Supervised learning method

In this category, the machine learning model is able to learn from data because it was trained on labelled data that includes the solution or target answer (Elgendy, 2021). Human supervision consists of making the necessary annotations (Geron, 2019). Unfortunately, producing these annotations is expensive and time-consuming (Xu et al., 2020b) and the performance of these models is very dependent on this labelled data. It is possible to distinguish between two types of family object detector networks using supervised learning.

1.4.1.1 One-stage object detectors networks

You only look once (YOLO) predicts the coordinates of the bounding boxes for the objects in the image (object localization) and classify their category simultaneously (object classification) (Kaur and Singh, 2023). Although there are a wide range of different architectures, the main components of YOLO are:

- **Backbone**: It extracts image features using a convolutional neural network.
- **Neck**: It combines and reprocesses image features from the backbone with a series of network layers and sends these image features to a prediction layer.
- **Head**: it creates the bounding boxes, prediction categories and assigns them confidence values indicating their precision.

This list contains several machine learning systems for livestock counting found in the literature that use supervised learning methods from the one-stage network family.

- **YOLOv2 + Feature map** (Shao et al., 2019): An entire area is captured by several images, while most counting methods only show a single image. The proposed system in this article optimises the resolution of the UAV images to facilitate

YOLOv2 detection by returning a feature map. It also includes a 3D reconstruction using Structure of Motion (SFM). The network has 23 convolutional layers and 5 pooling layers. The first convolution layers calculate a feature map for the whole image. Then, pooling layers are applied and the image downsampled to create a 'figure map' that is used for a last convolutional layer that generates the positions and sizes of the objects.

- **TPH-YOLOv5** (Pu et al., 2022): Sheep Barns shares challenging scenarios with UAV imagery as changes in the scale of the object and their high density have a negative impact on the detection model. To reduce the effects of tiny objects, four heads are used in this network, including Transformer Prediction Heads (TPH) recommended for high-density scenes. Additionally, the network is improved by adding multiple Spatial Pyramid Pooling (SPP) modules, Bidirectional Feature Pyramid Networks (BiFPN) modules in the neck, and Block Attention Modules (CBAMs). Since this study aims to classify different Chengdu ma goats, an additional self-training classifier helps to distinguish confusing categories. Results showed a better performance with an increase of 1.96% in mAP.
- **YOLOv4 and SSD** (Rančić et al., 2023):This study tested the robustness of YOLOv4 to changes in background and environments using the Verschoor Aerial Cow Dataset that is located in forest and pasture (meadows) scenarios. The result reports how images with different resolutions, different sizes, and small overlapping objects lead to false negative predictions, which means missing the detection of a deer when there was one in this case. Even with a small dataset of 30 images with 169 annotations and using a compressed version of YOLOv4, using pre-trained weights contributed to counting 157 deer in all the images.
- **YOLOv5x-ECA** (Cao et al., 2023): It tackles the difficulty of interference of mutual occlusion when counting sheep at different speeds by using a channel attention mechanism ECA, a sparrow search algorithm, and a deepsort algorithm to obtain real-time detection.

1.4.1.2 Two-Stage object detectors networks

In contrast to the previous one-stage detectors, two-stage detectors first locate the object by region proposals, and then they classify the object in those regions (Kaur and Singh, 2023). This family of object detectors is known as region-based convolutional neural networks (R- CNNs) and includes four main components in its architecture (Elgendy, 2021):

- Extract region of interest or extract region proposals: By selective search, the image finds regions with a high probability of finding an object. This requires a fixed input image size.

- Feature extraction module: A pre-trained convolutional network searches for features in each of the proposed regions.
- Classification module: Based on the features extracted, the object is classified.
- Localization module: This is also known as a bounding-box-regression and it predicts the location and size of the surroundings of the object, providing four numbers indicating the x,y,w,h coordinates of the box.

This list contains a number of machine learning systems for livestock counting found in the literature that use supervised learning methods in a two-stage framework.

- **R-CNN** (Sarwar et al., 2018): Detecting sheep at 80 m from the ground means detecting hundreds of 10x20 pixel objects on a 2048x1080 area representing paddocks. A Region Convolutional Neural Network (R-CNN) with three different architectures was tested along a blob analysis on a brightness thresholded grayscale that takes advantage of the white colour of sheep. The accuracy results were influenced by the training data, which was grouped as Sunny, Overcast and Mixed.
- **Mask R-CNN** (Xu et al., 2020b): Packed herds, visual clutter, strong lighting contrast, and shadow are the scenarios explored in this study, where a Mask R-CNN network is used to detect sheep and cattle. The first of the three modules of this network is a group of convolutional layers that extract the feature using ResNet-101 that acts as a backbone. These layers are arranged in three-layer blocks. Then, Regions Of Interests (ROI) on the feature maps derived from the backbone are generated with a Region Proposal Network (RPN). Subsequently, a RoiAlign operates on these ROI candidates, creating a fixed-size feature. The third module, the head, obtains a bounding-box for the target object by classification and regression, generating a mask prediction applying a Softmax loss. Scores as high as 96% for classification and 92% for livestock counting were reported. Additionally (Xu et al., 2020a) compares this Mask R-CNN algorithm with Faster R-CNN, YOLO v3 and SSD. In the field of tracking objects, the model Track R-CNN extends the Mask R-CNN to Multi-Object Tracking and Segmentation, which was used to monitor mammalian herbivores on thermal UAV imagery (Bárbulo Barrios et al., 2024).

This is just a selection of supervised methods used for livestock. However, in a broader scope that also considers other small object detections, the range of options increases. In fact, a recent survey included 36 different deep learning methods classified into multiscale representation, contextual information, super-resolution and region proposal just for small object detection (Chen et al., 2022).

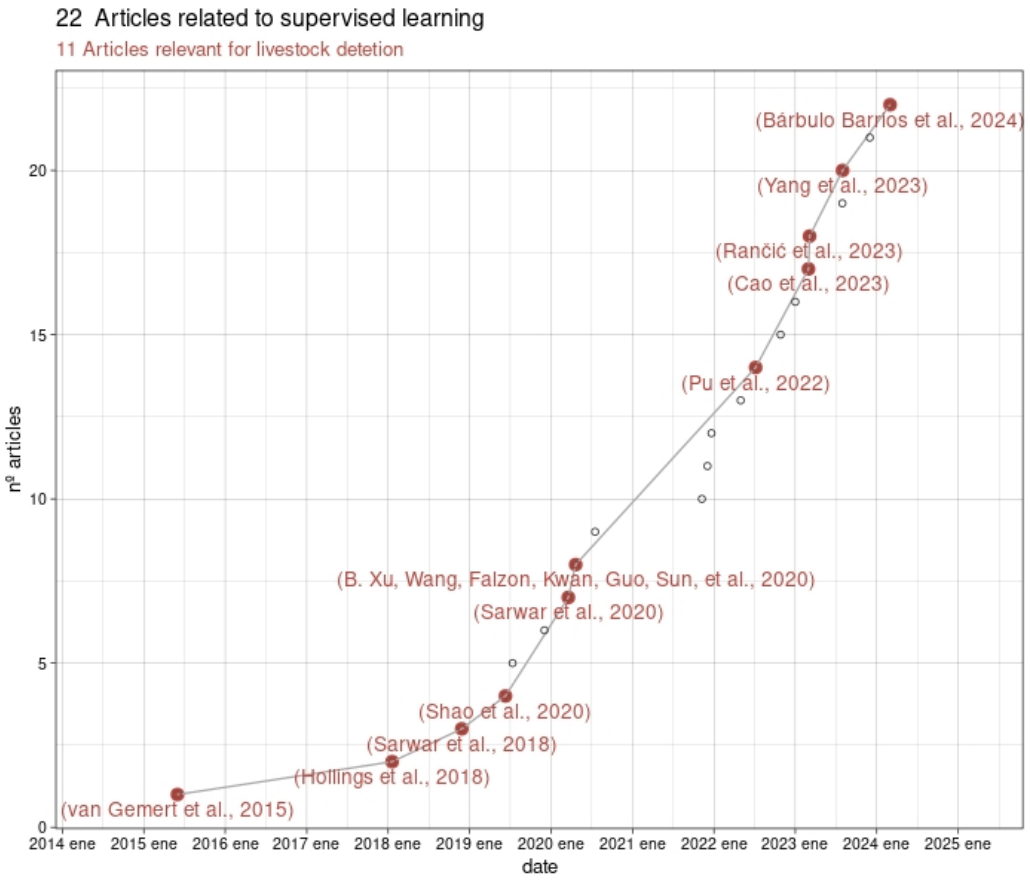


Figure 1.4: The literature review related to supervised learning models sorted by their publication date. More information is available by hovering over the interactive scatter plot.

Aerial object detection, a subcategory of these general small object detection problems, includes specific frameworks for this task such as focus and detect (Koyun et al., 2022) or dense-and-similar object detectors that follow a multistep detection strategy (X. Wang, Yan, et al., 2023). It is important to notice that limits between the one and two stages object detectors networks presented in this literature review are not always clearly delimited, since there are also CNN based one-stage detectors that follow a coarse-to-fine processing paradigm (Zou et al., 2019).

Figure 1.4 and Table 1.1 summarise the relevant literature review for precision livestock using supervised methods:

Table 1.1: Literature related to supervised models reviewed for this study. The field Relevant indicates if the article is specifically related to livestock and DOI facilitates a link

ID	Article	Date	DOI
1	Nature conservation drones for automatic localization and counting of animals	2015-06-01	Link to article: 1
2	How do you find the green sheep? a critical review of the use of remotely sensed imagery to detect and count animals	2018-01-19	Link to article: 2
3	Detecting and Counting Sheep with a Convolutional Neural Network.	2018-11-27	Link to article: 3
4	Cattle detection and counting in UAV images based on convolutional neural networks	2019-06-11	Link to article: 4
7	Towards detection of sheep onboard a UAV	2020-03-19	Link to article: 7
8	Livestock classification and counting in quadcopter aerial images using Mask R-CNN	2020-04-21	Link to article: 8
14	Research on Chengdu Ma Goat Recognition Based on Computer Vision	2022-07-07	Link to article: 14
17	A sheep dynamic counting scheme based on the fusion between an improved-sparrow-search YOLOv5x-ECA model and few-shot deepsort algorithm	2023-03-01	Link to article: 17
18	Animal Detection and Counting from UAV Images Using Convolutional Neural Networks	2023-03-06	Link to article: 18
20	Deformable convolution and coordinate attention for fast cattle detection	2023-08-01	Link to article: 20
22	Monitoring mammalian herbivores via convolutional neural networks implemented on thermal UAV imagery	2024-03-01	Link to article: 22

1.4.2 Sef-supervised learning method

Even when the research object belongs to classes such as humans or cars with available large-scale labelled datasets like COCO, the imbalance in the number of classes affects the detection performance ((Li et al., 2022); (Zhu et al., 2022)). In terms of abundance, some existing labelled datasets for livestock are: 656 images for cattle detection (Shao et al., 2019); 18.356 frames including 30 distinct animals (Agapito et al., 2015); 4133 images for sheep (Riis, 2022); 377 images of Friesian cattle (Burghardt and Andrew, 2016). In contrast to the total sum of 23.522 livestock labelled datasets, the dataset CARPK contains a total of 89.777 images of just cars. To solve the need of datasets, unsupervised methods are able to learn general image and video features without them (Pu et al., 2022), being self-supervised methods that generate the labels from the data itself, creating their own labelled data rather than by human supervision (Geron, 2019).

Figure 1.5 summarises the 23 articles related to the foundational model of self-supervised learning published since the release of Grounding DINO in March 2023. Table 1.2 outlines the most relevant articles for this project.

The criteria for selecting this method are more than the performance of the model. In fact, there are supervised methods such as UNet that have been reported to specifically detect sheep on real-time onboard UAVs with a 93% F1 score (Sarwar et al., 2020) or to count crowds (people) from drone images by CNN-based density estimation models scoring 27.7 on MSE (Nag et al., 2021). The decisions made in this study are justified by applying the following criteria:

- **Suitability**: The problem of detecting small target detection from UAV aerial photography is very complex to solve, with new algorithm improvements scoring an average detection accuracy of 37.88% (Wang and Jin, 2023). The GIMA MSc thesis objective is to demonstrate the ability to use and integrate knowledge acquired in the first six modules, being Python or R programming only a section of the fifth module. Therefore, a pretrained model offers a more suitable opportunity to explore the computer vision field given its easier use and a shorter learning curve appropriate for the length of Python or R in the GIMA MSc programme.
- **Duration**: Including a wider range of livestock production settings or controlling the environment by setting the conditions is relevant to increasing the generalisation capacity of the model or its performance ((Xu et al., 2020b)). However, the duration of the thesis limits the option to planning a UAV operation to gather data from the field. From a legal perspective, EASA drone Regulation EU 2019/947 and 2019/945 require an official drone pilot license, meaning an increase in time and resources for the project. Therefore, the sampling design and dataset are constrained to IDELE's dataset.

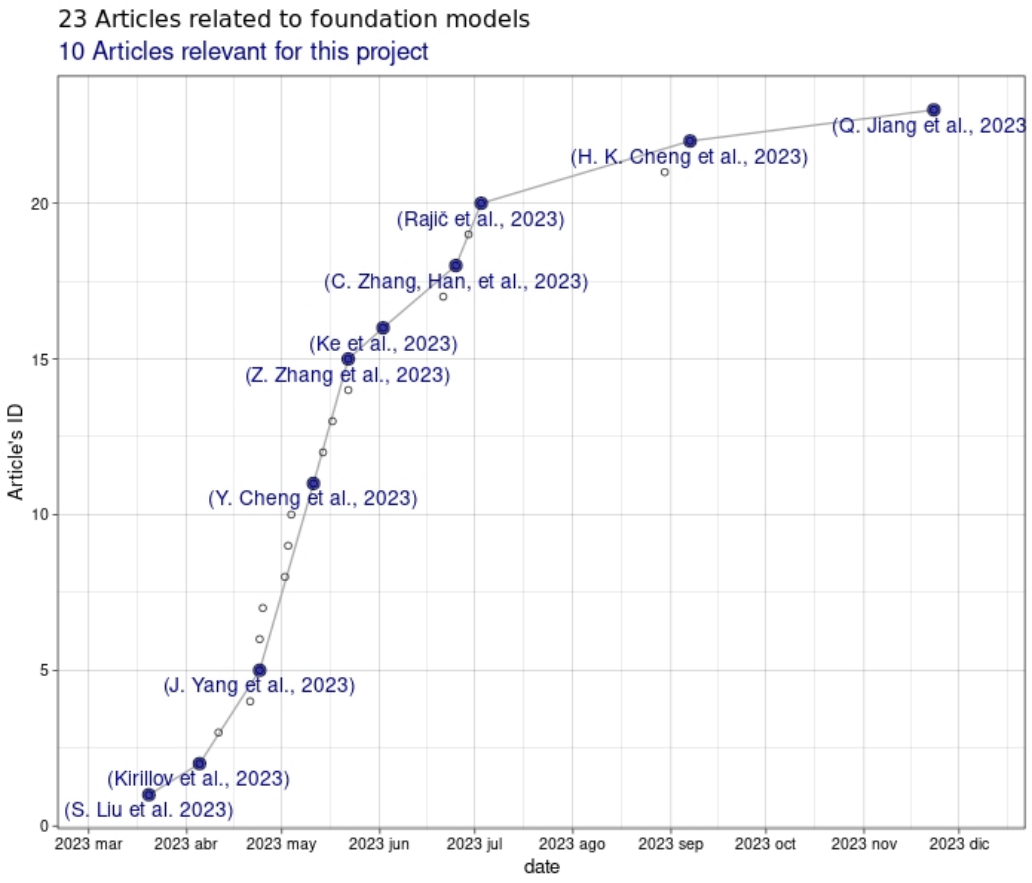


Figure 1.5: The literature review related to self-supervised learning models that generate foundation models is sorted by its publication date. More information is available by hovering over the interactive scatter plot.

Table 1.2: Literature related to foundational models reviewd for this master thesis. Relevant indicates if the article relates specifically to tracking.

id	Article	Date	DOI
1	Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection	2023-03-20	Link to article: 1
2	Segment Anything	2023-04-05	Link to article: 2
5	Track Anything: Segment Anything Meets Videos	2023-04-24	Link to article: 5
11	Segment and Track Anything	2023-05-11	Link to article: 11
15	UVOSAM: A Mask-free Paradigm for Unsupervised Video Object Segmentation via Segment Anything Model	2023-05-22	Link to article: 15
16	Segment Anything in High Quality	2023-06-02	Link to article: 16
18	Faster Segment Anything: Towards Lightweight SAM for Mobile Applications	2023-06-25	Link to article: 18
20	Segment Anything Meets Point Tracking	2023-07-03	Link to article: 20
22	Tracking Anything with Decoupled Video Segmentation	2023-09-07	Link to article: 22
23	T-Rex: Counting by Visual Prompting	2023-11-23	Link to article: 23

- **Availability**: Many of these algorithms have high computational costs (Zhang et al. (2023)) requiring Graphic Processing Units (GPU). Notebook services such as Google Colab offer limited computational time units using NVIDIA T4 Tensor Core GPU , 12.7 GB of ram and a disk of 78.2 GB free of charge. Likewise, API are run on external servers, covering these high computational costs externally.
- **Feasibility**: The decisions on the technical design are based on the availability of code already located in github repositories. On the one hand, the repositories include demos ready to run in notebooks (e.g. Google Colab). On the other hand, API services simplify the interaction with the user, requiring only the input data and a key to identify the user.
- **Ethical**: The co-supervisor João Valente informed IDELE members involved in the ICAERUS project by email and received their consent to use the dataset in this thesis. This communication also explains why this research is undertaken and its objective. Likewise, Ricardo Ruiz Sánchez contacted by email the authors of the models Grounding DINO-T-Rex and SAM-PT to inform them about the purpose of this study.

Additionally, the 96.88% accuracy obtained for a multi-object tracking system on black cattle using Segment Anything Model (SAM), GroundedSAM and Grounding DINO (Noe et al., 2023) supported the decision made. In terms of video tracking, SAM-PT is selected from all the available SAM-based tracking methodologies. One of the reasons is the possibility to combine several point trackers (e.g Cotracker, TAPIT) and SAM variants such as HQ-SAM or MobileSAM.

Although SAM based is selected for this study, there are more models relevant for counting livestock, like the Counting TRansformer (CounTR), a generalised visual object counting system pre-trained by self-supervising learning that uses density maps (Liu et al., 2022) or Cut-and-LeaRn (CutLER), an unsupervised object detection and instance segmentation able to self-learn from its predictions (Wang et al., 2023b).

Finally, autodistill [5] uses these computationally demanding novel foundation models to train small and faster supervised models. For example, it is possible to use GroundedSAM to train a YOLOv8 model. This is done based on unlabeled images and applying a broader range of foundational models not present in this literature review, such as SAM-CLIP, Llava-1.5 or DETIC.

1.4.2.1 Detection with Segment-Anything Model (SAM) and Grounding DINO

In the field of computer vision, Segment Anything Model (SAM) is a foundation model for image segmentation that uses the SA-1B dataset which includes more than

[5] Autodistill: https://docs.autodistill.com/

1 billion masks. SAM was built through self-supervised learning and large-scale supervised training (Kirillov et al., 2023), which means that extensive efforts in the creation of labelled datasets for specific domains could be spared. Despite the use of foundation models such as SAM in fields ranging from digital pathology (Deng et al., 2023) to remote sensing (Wang et al., 2023a), small and dense scenarios (Ma et al., 2023), camouflaged objects (Tang et al., 2023) and concealed scenes (Ji et al., 2023) remain challenging for SAM. Moreover, SAM has been utilized to create large thermal-infrared datasets, such as SATIR, (Chen and Bai, 2023), or the USTC FLICAR, a dataset of LiDAR Inertial camera (Wang et al., 2019). Additionally, recent solutions specifically tailored to the project's task of counting objects have been recently published (Shi et al., 2023b).

The combination of DETR, a novel Transformer-based detection algorithm, that was improved by a DeNoising Anchor Boxes (Zhang et al. (2022)) and a grounded language-image pre-training (GLIP)(Li et al., 2021) results on GroundingDINO (Liu et al., 2023). GroundingDINO is a generic object detector with Referring Expression Comprehension (REC) that allows the user to add attributes to better describe the target object. GroundingDINO detects the object as an image and its class as a text, returning a pair of object boxes and noun phrases. The project Grounded-SAM aims to solve complex problems, such as heterogeneous real-world scenarios, by combining SAM and this GroundingDINO model. For example, to the previous MobileSAM and HQ-SAM a new efficient variant, FastSAM, is added. Fast SAM uses a CNN detector to reduce its computational costs with the purpose of being applied in real-world applications such as detecting anomalies in manufacturing or extracting buildings in urban planning (Zhao et al., 2023).

1.4.2.2 Tracking with SAM-PT

Due to its flexible prompts and real-time mask calculation, SAM has been adapted for various fields, including Video-Object Tracking (VOT) (Cheng et al. (2023b) ; Ke et al. (2023); Yang et al. (2023a)). In fact, previous Multi-Object Tracking and segmentation architectures used in UAV imagery, such as PointTrack (Bárbulo Barrios et al., 2024), have recently been incorporated into SAM creating the model Segment Anything Meets Point Tracking (SAM-PT) (Rajič et al., 2023). SAM-PT does not require training on any video segmentation data. It only requires query points in the first frame (Rajič et al., 2023). This model selects query points in the first step, which are then propagated to all video frames using point trackers in the second step. During the third step, SAM generates segmentation masks based on these propagated points, and the last optional step consists of reinitializing the process (Rajič et al., 2023).

1.4.2.3 Counting with T-Rex

T-Rex is a detection-based counting model that detects all objects with similar patterns given an initial detection drawn by the user (Jiang et al., 2023a). Unlike other models limited by categories, T-Rex perspective focuses on providing an open-set counting system that does not require any categories. Additionally, it is interactive, making it possible to correct errors, such as those caused when a model generalises across heterogeneous scenarios. This is achieved through intuitive visual feedback that facilitates the user's interpretation and by using a visual promptable system. This foundation model can be used as an object counter or automatic-annotation tool in domains such as livestock, but also transportation or biology, among others.

Chapter 2

Methodology

This chapter, separated into three sections, contains exhaustive descriptions to reproduce this use case. By following the section "Heterogeneous scenarios", the UAV footage is classified according to weather and landscape conditions. The following section, "Model performance", describes how to run the foundation models to count and track sheep. Lastly, the section "Evaluation metrics" provides the indicators to assess the counting and tracking performance.

2.1 Heterogeneous scenarios: UAV metadata and weather conditions

The first section, "Heterogeneous scenarios", describes the methods used to obtain the data that was used to establish the different scenarios to assess the model generalization. After defining the study area, the following three steps explain how to import the UAV metadata, join this data with the weather conditions, and how to aggregate the data into three types of scenarios. One last step includes how to measure the ground truth values used to evaluate the models' performance.

Figure 2.1 illustrates this methodology based on these three main steps.

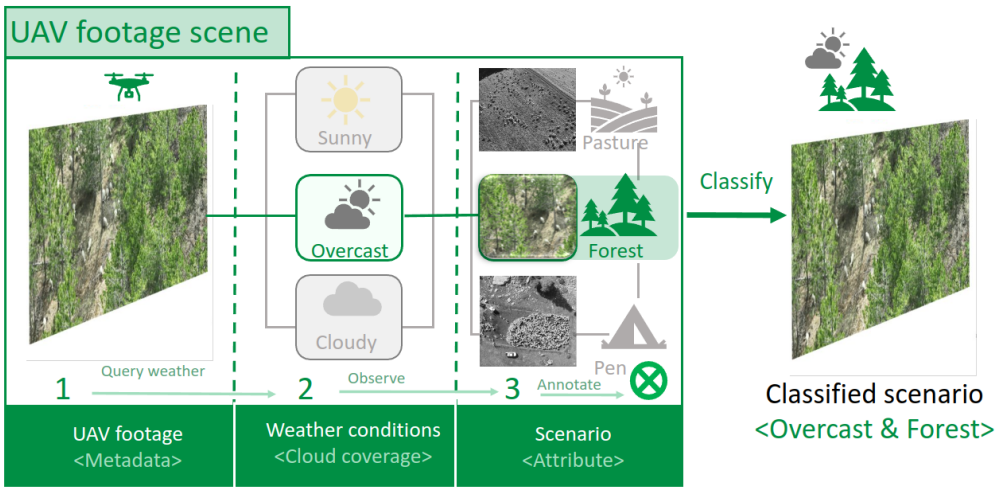


Figure 2.1: The first work package, heterogeneous scenarios, required three activities to classify the UAV footage. (1) The metadata from the UAV video included the date used to query the weather conditions. (2) The weather conditions (derived from the cloud coverage) classified the scenario. (3) The observation from the UAV footage attributes generated three types of scenarios, and they were later annotated in the data. This methodology classified the UAV footage into heterogeneous scenarios to test the model generalisation, addressing the third goal of this research.

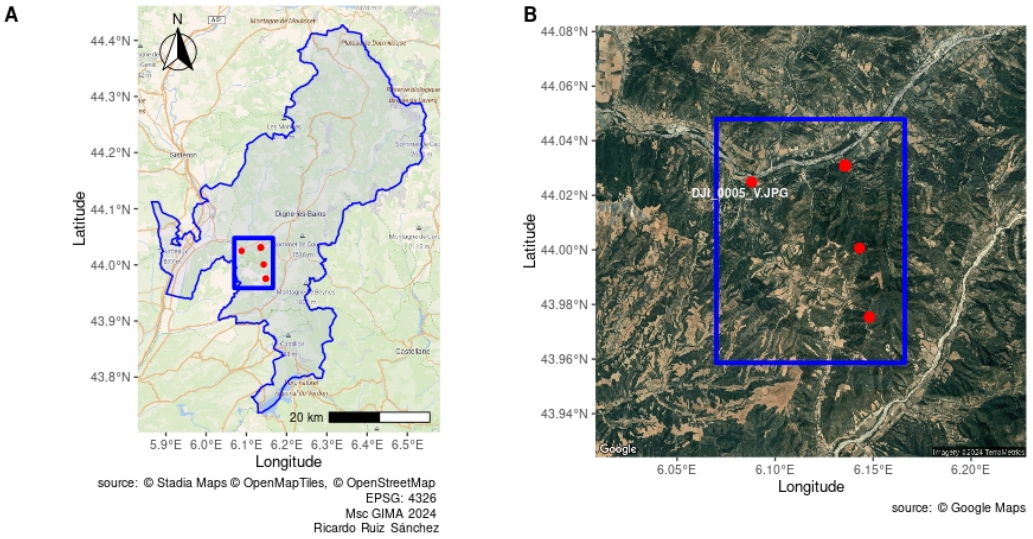


Figure 2.2: The two maps represent the administrative study area and the sampling area. On the left (A), Digne-les-Bains is located in the Provence of Alpes-de-haute, France, where the blue rectangle represents the sampling area. On the right (B), the red points represent where the UAV pictures were taken, and a satellite image from Google Maps shows the surroundings.

2.1.1 Defining the study area

Located between 5º86' - 6º54' N and 43º73' - 44º24' E (Figure 2.2), Digne-Les-Bains is one of the five French districts within the province of Alpes-de-haute. The Digne-Les-Bains district has an area of 1598 km² inhabited by 16 864 people, according to the census in 2020 [1]. Although farmers only represent a 0.5% of the household population, the experimental farm Carmejane is located in this pre-Alpine zone. Several R libraries handled the spatial data to represent the administrative unit and sampling area (Kahle and Wickham (2013) ; Pebesma (2018) ; Dunnington (2023) ; Slowikowski (2024) ; Cheng et al. (2024)).

The following Figure 2.3 shows an interactive map of the study area that is accessible through using the HTML of this thesis. An orthophoto provided by ESRI.WorldImagery and a map offered by OpenStreetMap are the layers of the interactive map. Additionally, a polygon shows the limits of the district of Dinge-les-Bains. In this polygons several points with information about the UAV model, the time when the footage was taken along its GPS position, and a thumbnail

[1] Institut national de la statisque et des études économiques: Commune de Digne-les-Bains: https://www.insee.fr/fr/statistiques/1405599?geo=COM-04070

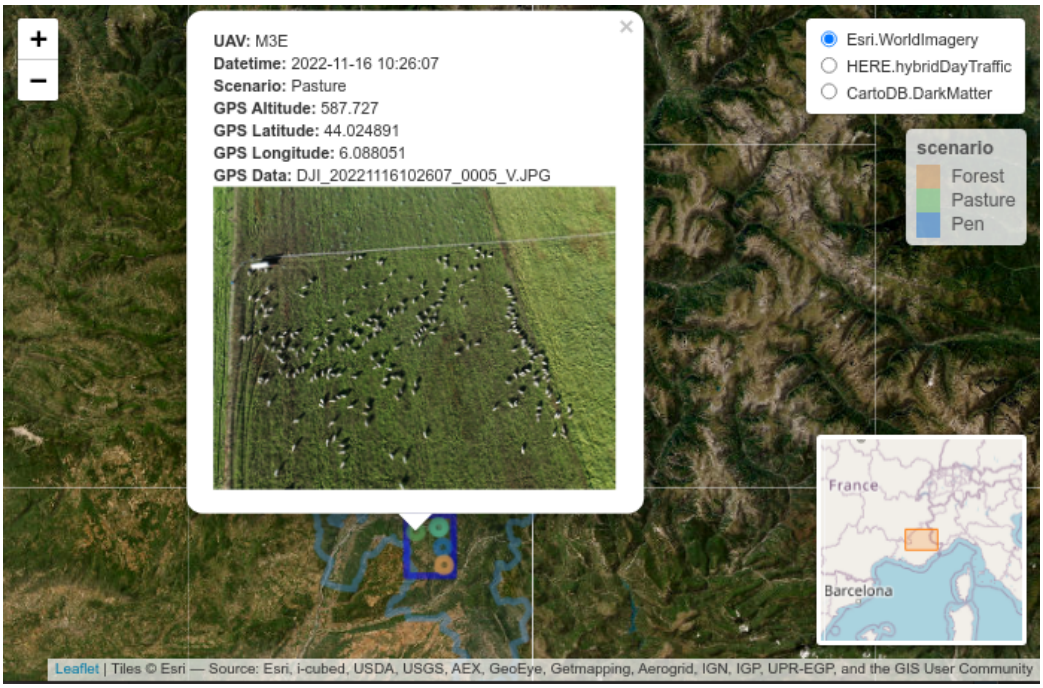


Figure 2.3: The interactive map on the Github page (https://rruiz-s.github.io/thesis-gima-html/chapter_methodology.html#defining-the-study-area) of this study allows you to change the scale, the background layers, and provides information when clicking on the blue circle (samples). Apart from the metadata, the information of the samples includes a thumbnail of the UAV picture, GPS, UAV model, and when it was taken.

of the UAV picture provides more information of the study area and metadata.

This farm comprises 50 ha, mainly dedicated to feeding a herd of 650 pre-alps sheep, 500 ha of forest, and two summer pastures. These sheep's breed, "Préalpes du Sud" is endemic to the regions of the Alp foothills, which includes the study area of Alpes-de Haute-Provecence [2]. Apart from sheep, 8 Bergers des Abruzzes dogs protect the herd against predators. The valorisation of pastoral areas through grazing or the study of the technical and economic performance of these areas aim to provide innovative solutions and decision support systems to tackle the large problems of regional sheep breeding in the experimental farm Digne Carmejane [3]. Figure 2.4 illustrates the livestock in the Carmejane experimental farm.

[2] France Génétique Elevage: Préalpes du Sud: http://en.france-genetique-elevage.org/Prealpes-du-Sud,391.html

[3] Carmejane, La réusse pour tous au coeur des Alpes du Sud: L'explotation de Carmejane: https://digne-carmejane.educagri.fr/exploitation-ovine

Figure 2.4: Shepherds dogs and sheep constitute the livestock in Digne Carmejane. On the left (A), 8 Bergers des Abruzzes, while on the right (B), 650 Préalpes du Sud sheep constitute the herd.

2.1.2 Importing UAV metadata

The software used for this step is the Exiftool through the R library Exiftoolr (O'Brien (2024)). The variable 'metada_uav_video' stored the information from the main object of this research, UAV videos, while 'metadata_uav_pictures' contained auxiliary information. In both cases, the function round() reduced the GPS latitude and longitude to 4 decimals. Likewise, the function lubridate::ymd_hms() transformed the variable CreateDate from "character" to "POSIct" "POSIXt" while lubridate::floor_date() reduced the time to hours. Importing and tidying the metadata were necessary to find weather conditions and ultimately to describe the heterogeneous scenarios. The code to reproduce these steps was written in the Appendix A.1.1.

The function naniar::gg_missupset() assessed the completeness of the metadata checking for missing values. A visual inspection is carried out to verify the consistency of locations among the videos and pictures. Likewise, for similar scenes like the picture "DJI_005_V.JPG" and the video "DJI_0024_V.MP4", the drone model and the date are annotated to check again the consistency of the video locations. Comparing variables related to the spatial resolution is done to identify confounding factors that affects the validity of the comparison between heterogeneous scenarios.

2.1.3 Finding weather conditions

After determining when and where the UAV footage was taken, the Open-meteo API[4] in R used this information to retrieve their weather conditions with the library Openmeteo (Pisel, 2023). The function unique() is applied to the variable 'datetime' from the UAV videos obtaining when the UAV footage was taken. To obtain the location of the UAV video, the UAV metadata from UAV pictures is filtered to obtain the 'GPSLatitude' and 'GPSLongitude' of the picture 'DJI_005_V.JPG", which were similar to the video 'DJI_024_V.MP4'. The function weather_history() connected the Openmeteo API with RStudio.

This function required four arguments: the location of the place for which data will be queried; Start and end dates of the query; "hourly" or "daily" values to return the measurements of the weather conditions accordingly. The main measurement chosen to define the weather condition was the cloud coverage, following previous studies (Sarwar et al., 2018). The variable 'weather_df' stored the query results, and the function left_join() joined the weather conditions with the metadata in the column 'datetime'. Lastly, the weather codes are translated according to the World Meteorological Organization (WMO)[5] with the function recode(). These steps are shown in the code chunk located in the Appendix II.

2.1.4 Defining scenarios

Attributes are referred to characteristics of the scenario, such as the size of the target detection, how occluded with other objects, or how close different targets appear. These three attributes created three different scenario categories: pasture, forest, and pen. Small sparse sheep herd were labelled as "pasture", while scenarios with occluded sheep by trees or bushes in the background of forests were labelled as "forest". Lastly, the label "pen" was assigned to clustered sheep herd. Once all the UAV footage was classified, three categorical variables are created by discretizing the numeric variable 'cloud_coverage' using the function cut(), resulting in the three variables ('Sunny','Overcast', and 'Sunny'). Finally, the UAV metadata is grouped by the drone model (encoder), time, and the categorical cloud coverage to later summarise the scenario distribution by its frequency. Each step is shown in the Appendix III that contains the code chunk for this activity.

[4] Open-meteo API

[5] Codification used in manned weather stations from the World Meteorological Organization (WMO): https://www.nodc.noaa.gov/archive/arc0021/0002199/1.1/data/0-data/HTML/WMO-CODE/WMO4677.HTM

An alluvial chart illustrated the heterogeneous scenarios in the UAV footage. The library ggaluvial added a geom_alluvial to a ggplot object to visualise the distribution of these heterogeneous scenarios by encoder, date, and weather conditions. Lastly, the function cowplot::plot_grid() arranged the three alluvial charts into a single plot. The code that creates this alluvial chart is found in the Appendix IV.

2.1.5 Measuring ground truth

The comparison between the state-of-the-art foundation model T-Rex and the previous supervised method YOLOv8 required the number of sheep in the scenes to measure its model counting performance, since counting evaluation metrics need ground truth values to be computed. For this purpose, a grid of 1200x1200 px using GIMP created different areas for counting to reduce systematic errors. This idea was based on grid-based sampling methods, which are used in fields such as Ecology to assess species richness (Mueller et al., 2021). Moreover, the manual counting is repeated three times with different colours to improve consistency in the measurements, reducing random errors. Repeating observations is also recommended in Ecology to reduce collection errors in the field (Gotelli and Ellison, 2004). Exceptions were images with 932 and 1028 sheep, which were counted only twice due to time constraints, and human errors accumulated. Figure 2.5 illustrates one of these cases and the strategy to reduce random and systematic errors. Similarly, to improve the reproducibility of the experiment, the bounding box used as an input for T-Rex is measured using the software FIJI (Schindelin et al., 2012) and then exported in a text file (csv).The bounding box contained the target object (i.e. a sheep), which T-Rex required to count the herd. Appendix V describes in detail the code used to carry out these steps.

The reason to measure the size of the bounding box used as input for T-Rex, not only its location, was to better understand the model's performance under different operational conditions. Specifically, the size of the sheep is related to the altitude of the UAV, one of the main features of UAV operational conditions in Europe given the EU Regulation 2019/947 and Regulation 2019/945 that establish different flight categories for different altitudes. Different categories are subject to different risk responses and requirements. The GPS gimbal degree that determines the perspective of the UAV image, which also influences the size and therefore the model's performance, is excluded. Only the static images, not the videos, contained the GPS gimbal or GPS altitude, which was decisive in excluding the variable GPS gimbal, since the study object was the UAV videos. Once the measurements were done, they were added to a csv (Comma Separated Value) file to facilitate its use either in R or Python.

The exported text files containing the first bounding box detected, either automatically by Grounding DINO or manually annotated, are imported in an data frame. The variables upper_left_x, upper_left_y, lower_left_x, and lower_left_y required by

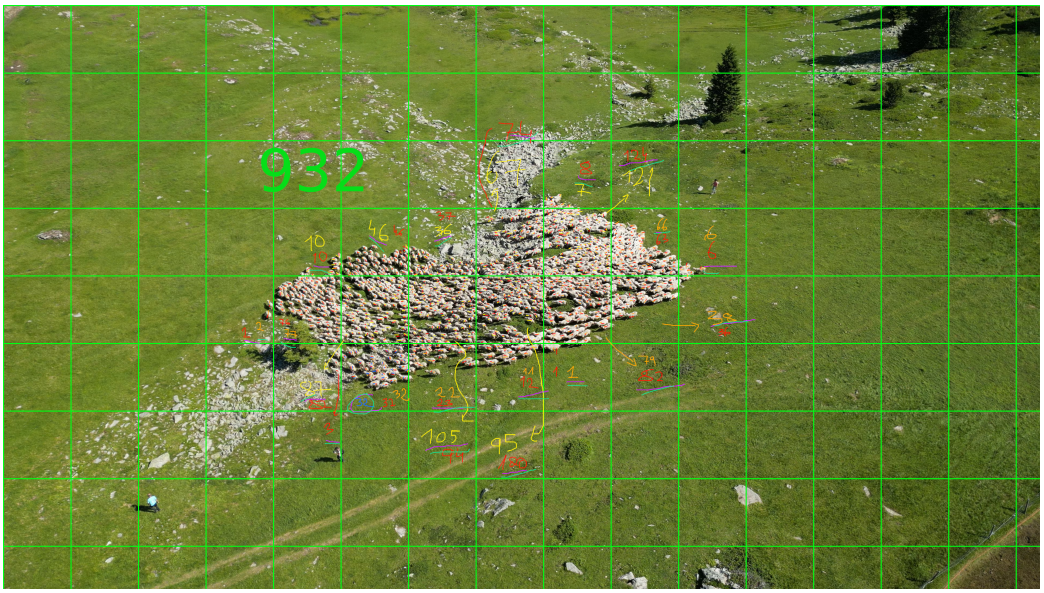


Figure 2.5: A grid and three manual counts to reduce random and systematic errors represented the ground truth.

T-Rex were obtained from the FIJI manual measurements.

2.2 Model design for counting livestock:

This second section describes the methods to identify the first sheep using the foundation model GroundingDINO, which will be later used to count and track the herd. The reason behind this is that the counting model, T-Rex, requires the bounding box of the target object, this sheep detected by Grounding DINO, as an input. Figure 2.6 illustrates the workflow of this design based on three main steps. For the supervised YOLOv8 model, the public YOLOv8 aerial sheep detection and counting model used only required to change the path to the video using the model created in the github repository (Nemati, 2023).Therefore, its methodology was excluded.

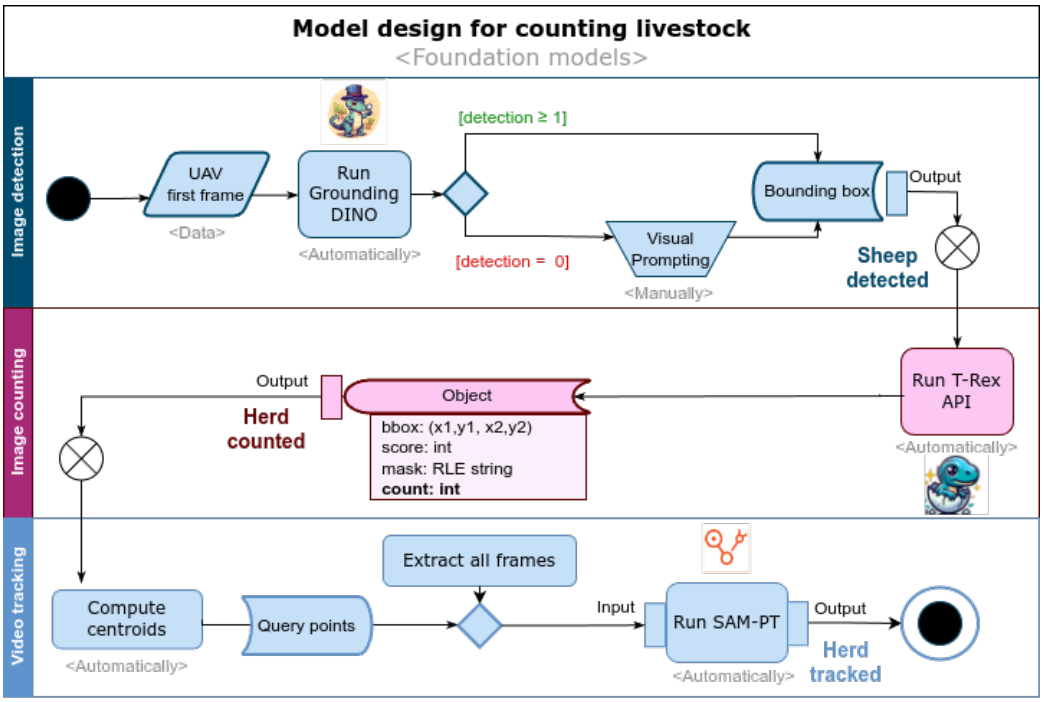


Figure 2.6: The activity diagram represents the workflow required to evaluate the models counting performance and track the livestock. The image detection activity obtained the bounding box of a sheep by automatically running Grounding DINO or manually using visual prompting. The model T-Rex uses this output to count all sheep in the herd during the image counting activity. Lastly, SAM-PT used as an input the centroids calculated from the mask of each sheep and all the frames from the video to track the herd.

2.2.1 Image detection: sheep identification:

Grounding DINO identified one sheep using the first frame of the UAV video as described in Figure 2.7 . The only manual inputs required were the category of the target object, defined in the variable 'TEXT_PROMPT' and the threshold of the object and text stored in the variables OBJECT_TRESHOLD and TEXT_TRESHOLD. For this study case, 0.25 was the value for both thresholds. In general, the accuracy level was low, so higher values did not return any detection. In any case, at the end, all detections were sorted, selecting the one with the highest accuracy level. When Grounding DINO automatically detected at least one sheep, this activity ended. Although Grounding DINO detects multiple objects, this was not the case for most of the dataset. In contrast, T-Rex was able to count multiple objects when Grounding DINO failed. For cases where Grounding DINO did not detect any sheep, it was nec-

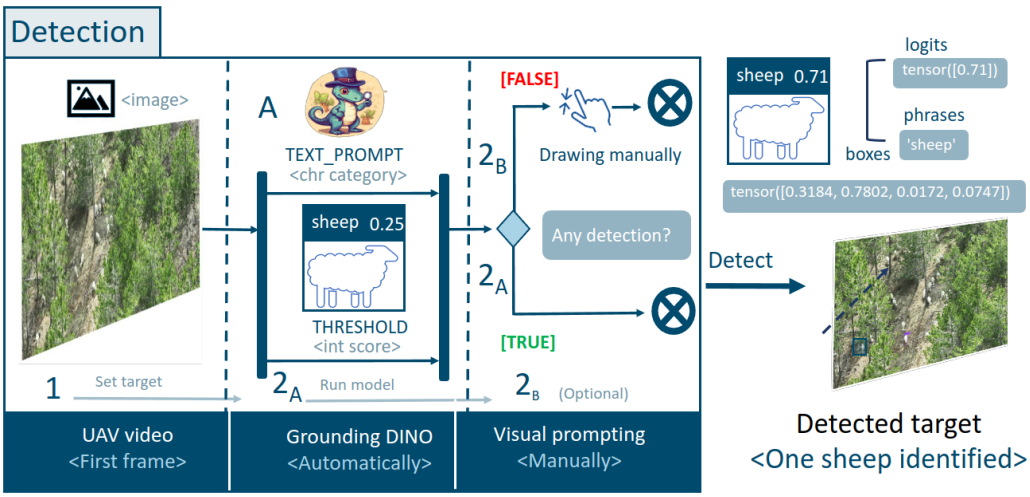


Figure 2.7: The detection step required only two steps, loading the first frame of the UAV video and detecting the target object. This process was automatic when Grounding DINO detected at least one object, but manual when it did not.

essary to manually draw the bounding box to identify the first sheep. The bounding box of the sheep detected was used for counting the herd using T-Rex.

2.2.1.1 Automatically: Grounding DINO

The process started with extracting the first frame from the UAV videos and obtaining an image. The SOTA zero-shot object detection model Grounding DINO is cloned from the official repository (IDEA-Research) and their requirements installed. Since it was run through a Jupyter notebook following the roboflow notebook (roboflow/notebooks), the hardware accelerator is set to a graphical process unit (GPU) to run the detection model. The command nvidia-smi verified the status of the GPU. Before loading Grounding DINO, the weight "groundingdino_swint_ogc.pth" is downloaded and saved in the folder weights. The variable "WEIGHTS_NAME" concatenated to "WEIGHTS_PATH" stored the weight directory. The functions load_model, load_image, predict, and annotate are imported from groundingdino.util.interference and used to run the model. This is explained in detail in the Appendix VI.

Table 2.1: The bounding box from the image detection is stored in the format required for T-Rex, which counted the herd in the activity Image counting.

Manual visual measurement	Bounding box
DJI_0024.csv	[2437,1681,2488,1712]
DJI_0025_v.csv	[1335,966,1385,999]
dji_0040_V_01.csv	[1933,1530,2007,1565]

Once the model was defined, the parameters of the function predict() were adjusted to this use case. "Sheep" was set as text_prompt for the argument caption, while the box and text threshold were set initially to 0.25. Depending on the complexity of the scenario, the box threshold was adusted, setting a lower value for those more challenging. The function predict() ran the model, obtaining the bounding boxes as 'boxes', the accuracy level of each prediction as 'logits' and the category of those predictions as 'phrases'. The function sort() sorted all the predictions in descending order to identify the sheep with the highest accuracy level using its index. The Appendix VII shows these procedures step by step.

2.2.1.2 Manually: Visual prompting

The visual prompting, used when Grounding DINO did not detect any sheep, consisted on drawing manually the bounding box using the FIJI software to store the measurement in a text file (csv). The reason to complement the process with this manual methodology is the lack of any detections using Grounding-DINO in the most complex scenarios. The created bounding-box used the BX, BY, width, and height measurements to create the lower left corner and top-right corner coordinates of the bounding box used in T-Rex. Firstly, the individual measurement of each first frame saved in the same directory is merged into a unique file using RegExr (Regular Expression) and the function list.files(). Secondly, the function basename() created a new variable containing the name of the file using basename(). Lastly, a list named bounding_boxes iterated through each of the records, creating the upper_left_x with the value BX, upper_left_y with the value BY, lower_right_x adding the width of the object to the BX and the lower_right_y adding the height of the object to the BY. Figure 2.8 illustrates how to measure the object using FIJI, while Table 2.1 contains some of the bounding boxes used for the next step: "Image counting: herd identification".

2.2.2 Image counting: herd identification.

T-Rex identified all sheep in the herd with only one required input, a bounding box obtained automatically using Grounding DINO or manually by the user. After set-

Figure 2.8: Under complex scenarios, a manual visual measurement of one sheep provided the bounding box required by T-Rex. When Grounding DINO detected sheep, they were sorted in descending order by its accuracy level and then manually measured to improve reproducibility.

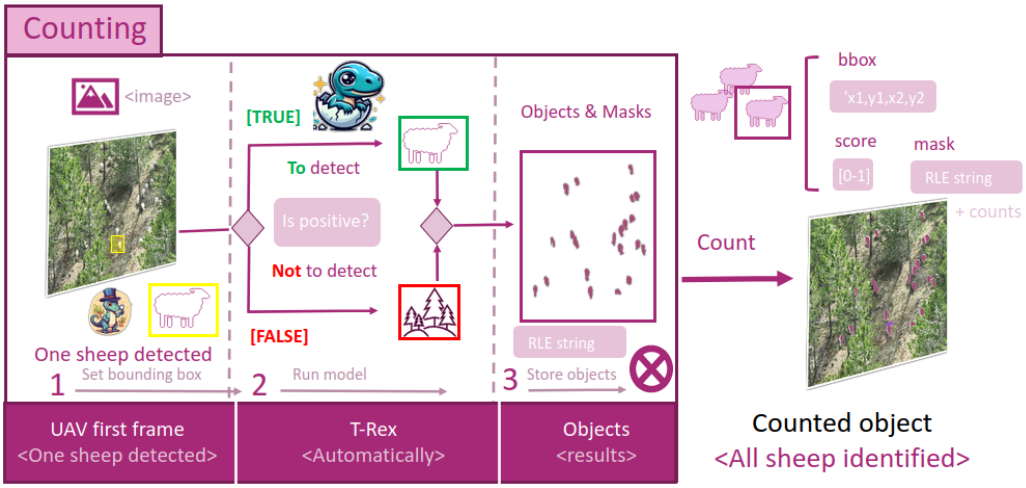


Figure 2.9: Using the T-Rex API given by IDEA team for free made it possible to count the herd automatically when one sheep is detected by Grounding DINO. This API only requires the bounding box from this first detection, generating a global mask with all the detections and individual objects that includes its bounding box, accuracy score, and mask.

ting the bounding box of one sheep in the first frame, T-Rex automatically counted all the objects with a similar pattern, returning a mask with all the sheep and individual object detection. This object detection included its bounding box, accuracy score, and mask, which also had the number of counts. This workflow is illustrated in the next activity diagram (Figure 2.9)

After contacting Lei Zhang, the lead scientist in the Computer Vision and Robotics Department (CVR) of the International Digital Economy Academy (IDEA), he facilitated the contact of Mr WeiQiang Hu. He provided a free API token, the script described in this section to run T-Rex and its patient, constant, and helpful support via email.

The first step was to upload the image to the server using the function upload_image() defined at file_utils.py which required the local path of the image and the API token. The second step triggered an asynchronous task to run T-Rex which returned a Universally Unique Identifier (UUID) used to get the final results.

The data dictionary contained the key-value pair prompt:bounding-box, where bounding-box stored the upper_left_x, upper_left_y, lower_right_x, and lower_right_y points defining the rectangle where the first sheep was found. The third step using the UUID loop inferred the rest of the sheep and saved the results as a string value. The Appendix IX contains the precise instructions to use the T-Rex API facilitated by Mr. WeiQiang Hu.

The objects detected, stored as a string value, are compressed into Run-Length Encoding (RLE) through the defined function string2rle() during step four. Similarly, the mask of the objects was obtained using the defined function rle2mask().In the fifth step, before parsing the results, an accumulator for the masks is defined to store each mask object into an array. A loop iterated over each detection, reading their RLE values, converting these values into arrays, and finally adding them into the mask accumulator. Lastly, the mask accumulator transformed into a RGBA image saved the masks in the alpha channel. Every step is described in detail in the Appendix X.

2.2.3 Video tracking: sheep trajectories.

SAM-PT achieved tracking the sheep identified by T-Rex in the first frame of the detection task. Computing the centroid of every sheep provided the target objects to track, also named as positive points. Optionally, it was possible to add negative points to indicate what not to track, namely, the background or non-target objects. A text file stored these files query points in the specific format described in the public github repository for SAM-PT. Lastly, the non-interactive demo from SAM-PT tracked all sheep identified in the first frame using these predefined query points and the frames from the mp4 video. Figure 2.10 shows the workflow to track the herd, obtaining the sheep trajectories.

2.2.3.1 SAM-PT

The variable “num_objects” stored each sheep detected from the T-Rex object results. A text file named “query_points__DJI_00028.txt” looped for each of the detections, calculating the centroid of the masks and writing it in a text file following the query points format used in SAM-PT. As indicated in the SAM-PT github’s repository(git), the four variables separated by “;” indicated the number of positive points, the timestamp for each mask, and the pair (x,y) coordinates of the positive and negative points. This text file defines the query points required to run the non-interactive SAM-PT demo. The code chunk in the Appendix XI contains these steps.

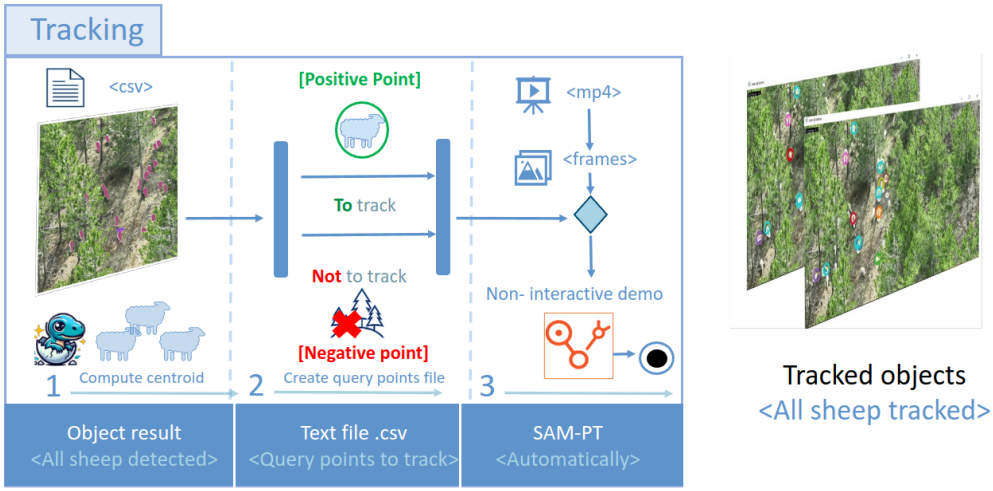


Figure 2.10: SAM-PT allowed to track all sheep detected by T-Rex from the first frame in three steps (https://www.youtube.com/watch?v=L0DTngX_iB4). A computed centroid represented each detected sheep, then a text file stored these points to query, and finally, once the video is split into multiple frames, the non-interactive demo automatically tracks each sheep.

The path to the query point text file made the tracking process automatic, only requiring the path to the frames of the video. It was required to crop some videos, not only because sheep were not present in the first frame, but also due to the high computing costs that caused the system to crash when the video was too long. After this, the audio and manipulation tool ffmpeg (Fast Forward Moving Picture Experts Group) processed the video file (mp4) creating different frames at intervals of time defined by the argument "fps". The platform Hydra created a working directory for each run storing the output from SAM-PT. The description of these procedures in detail can be found in the SAM-PT github repository and in the Appendix XII.

2.3 Evaluation metrics

The third section, "Evaluation Metrics", describes the measurements used to assess the counting performance on images by T-Rex, and the tracking performance on videos by SAM-PT.

2.3.1 Image detection: Sheep identification

The mean absolute error (MAE) and normalised relative error (NAE), defined in the equation (2.2), assessed the counting performance of the models. Where (n), is the total number of images, and (y) and ($\hat{y}$) are the real (ground-truth) and predicted counts, respectively. These metrics were applied individually and grouped by the three types of scenarios. As MAE increased, the counting performance decreased. On the contrary, a low NAE value indicated a high counting performance.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2.1}$$

$$NAE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}|}{y_i} \tag{2.2}$$

Previous studies included these evaluation metrics in the field of object counting and applied them to specialised datasets for this task, such as FSC147 and CARPK (Shi et al. (2023b) ; Liu et al. (2022)).

2.3.2 Video tracking: Sheep trajectories

Although the scope of this study was to evaluate the counting performance of the model, two metrics based on the VisAI Labs (Labs, 2021) post about evaluation of multiple object tracking accuracy were adjusted to be used in four videos. The first metric, Miss Detection (MD) was the errors where SAM-PT did not track the object situated under the query point. The second metric, Partially Tracked (PT) represents those cases where the trajectory is lost. The higher the number of MD and PT, the worse the tracking performance is. Additionally, a qualitative observation added information regarding the scenario.

Chapter 3

Results

This chapter, separated into three sections, contains the results from comparing the counting performance between the supervised model YOLOv8 and the foundation model T-Rex. Firstly, the model generalization section describes the distribution of the heterogeneous scenarios in the UAV footage. Secondly, the image counting shows the counting performance of these models across heterogeneous scenarios based on a quantitative and qualitative analysis. Lastly, results from tracking the sheep are presented.

3.1 Model generalization: Heterogeneous scenarios

The software Exiftool collected 80 variables from the UAV videos. The variable 'CreateDate' was the key variable to obtain the hourly weather condition using OpenMeteo. However, other variables such as ImageSize or Megapixels were key to the validity of the models performance comparisons. These variables are shown in Table 3.1. On the other hand, Exiftool collected 168 variables from the UAV pictures. The variables 'GPSLatitude' and 'GPSLongitude' were the key features to obtain the other required variable for OpenMeteo (Pisel, 2023) weather_history() function, its location.

Table 3.2 includes the location and address of the picture "DJI_0005_V.JPG", obtained through reverse geocoding using tidygeocoder (Cambon et al., 2021).

Table 3.1: Importing metadata from UAV footage (UAV videos). While the metadata CreateDate from the UAV videos were valid, a difference on the ImageSize and Megapixels indicated limitations on valid comparations.

FileName	CreateDate yyyy-mm-dd-hh:mm::ss	ImageSize width height	Megapixels Millions pixels
DJI_0105.MP4	2023-07-13 07:47:02	3840 2160	8.2944
DJI_0104.MP4	2023-07-13 07:46:52	3840 2160	8.2944
DJI_005_S.MP4	2023-04-19 13:22:57	1920 1080	2.0736
DJI_0024_V.MP4	2022-11-16 09:37:41	3840 2160	8.2944
DJI_0026_V.MP4	2023-07-11 08:50:40	3840 2160	8.2944

Source: IDELE dataset used for ICAERUS project

Table 3.2: The location to query the weather condition is based on the GPSLatitude (lat) and GPSLongitud (lon) of the UAV picture. The address obtained through reverse geocoding adds more information for context.

	Location		
FileName	lat	lon	address
DJI_005_V.JPG	44.0249	6.0881	Route du Chaffaut, Malijai, Digne-les-Bains, Alpes-de-Haute-Provence, Provence-Alpes-Côte d'Azur, France métropolitaine, 04350, France

Then, Exiftool imported the metadata from UAV videos and UAV pictures required to retrieve the historical weather conditions. The four selected variables were cloud coverage, wind speed at 100m, diffuse radiation, and the weather code, according to World Meteorological Organization (WMO). Table 3.3 shows some observations with a shorter file name using stringr and RegeX (Regular eXpression). The three values, sunny, overcast, and cloudy, defined the weather conditions based on the cloud coverage, classifying the UAV videos into different heterogeneous scenarios.

The distribution of the UAV videos from IDELE classified according to the weather condition and the scenarios with the purpose of testing the model generalisation is illustrated in Figure 3.1. The three variables used were the type of UAV, the date, and the weather. The variable "datetime" calculated from the "CreateDate" metadata revealed that the UAV videos were recorded during 9 different times from

Table 3.3: Finding weather conditions from UAV footage based on the UAV pictures and time from UAV videos. Hourly weather conditions are returned using Open-Meteo API based on the date and time of UAV video. Cloud coverage greatly varies between the DJI-0024-MP4 and DJI-0105-MP4 videos representing heterogeneous scenarios

		Weather Conditions (hourly)				
Video	Time	Clouds	Wind	Sun	Code	Description
DJI_0105.MP4	2023-07-13 08:00:00	0	18.5	91	0	Cloud development not observed
DJI_0104.MP4	2023-07-13 08:00:00	0	18.5	91	0	Cloud development not observed
DJI_0005_S.MP4	2023-04-19 13:00:00	19	10.0	146	0	Cloud development not observed
DJI_0024_V.MP4	2022-11-16 10:00:00	88	17.7	107	3	Clouds generally forming

Note: Sun was the mean diffuse radiation from the preceding hour W/m_2

2023-07-13, 2023-04-19, 2022-11-16, 2023-07-11, 2023-01-22, 2023-07-12 different days.

Table 3.4 shows the values from Figure 3.1. In general, sunny weather was predominant on the UAV videos in dense herd scenarios named as Pen. The weather conditions obtained from OpenMeteo were coherent with the date stored in the UAV metadata, cloudy weather mostly corresponds to winter months such as November or January. The distribution of the weather conditions was not equally distributed, with the most cloudy conditions found in small scenarios named pasture.

3.2 Image counting: Sheep identification

This section shows the model performance of the different models applied to the first frame of the UAV videos. Firstly, Figure 3.2 represents how different foundation models perform across different scenarios. In the first row, CutLER (Wang et al., 2023c) aggregates three sheep into one single bounding box, reducing the number of sheep detected. In this simple scenario, with a small and sparse herd, Grounding-DINO model performance is similar to T-Rex. In the second row, the scenario complexity increased by adding occluded objects, and T-Rex started to offer a better counting accuracy than Grounding-DINO. Lastly, in the third row, Grounding-DINO detects the large herd as a unique object. It is in the more complex scenes with large and dense herds where T-Rex stands out. This exploratory analysis of these foundation models shows why T-Rex was selected for this study to

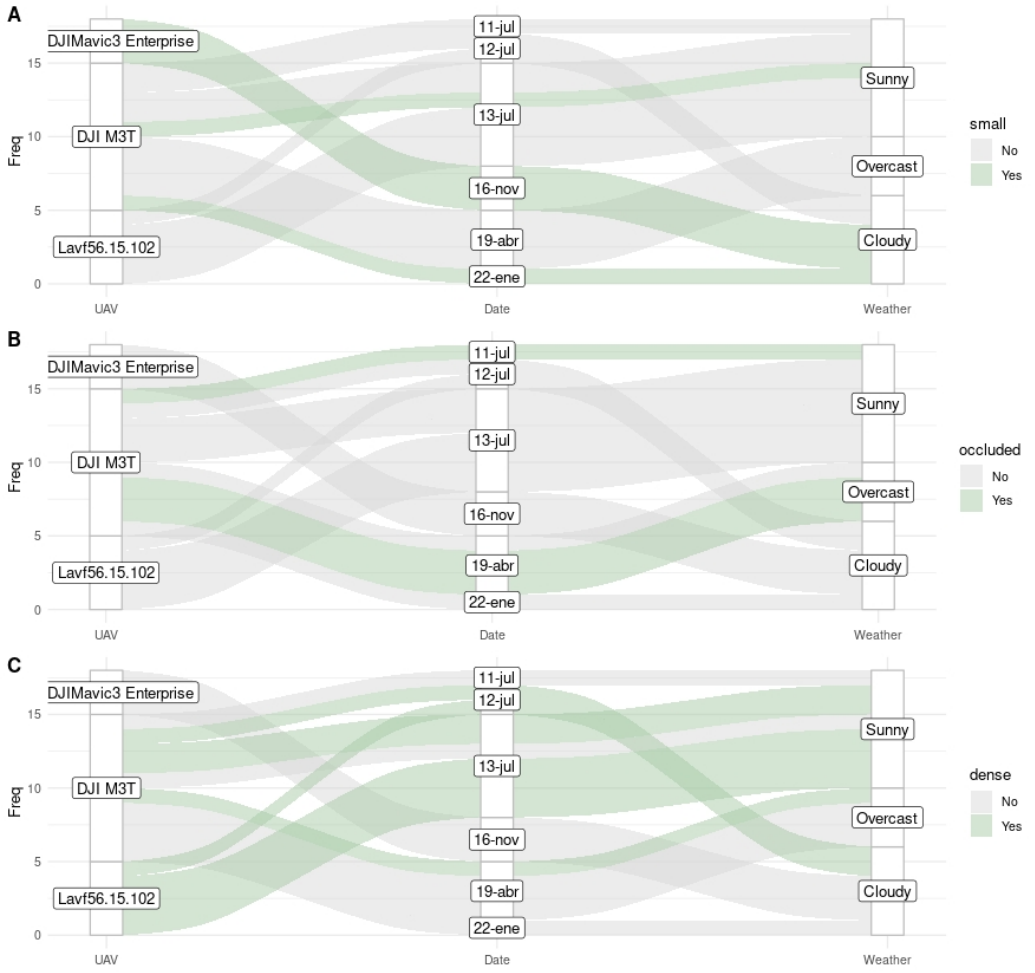


Figure 3.1: Three alluvial charts show how the three scenarios flow into the different weather categories. The chart (A), which represents pasture scenarios with small size of sheep, shows how they were mostly cloudy days. The chart (B) with occluded sheep in forests shows how most of the UAV videos were taken during overcast weather. Lastly, the chart (C) with footage of dense sheep clustered in pen scenarios is mostly sunny.

Table 3.4: A summary of the occurence of these heterogeneous scenarios based on the UAV drone model, date, and weather conditions that is shown in Figure 3.1

UAV model	Date	Weather condition	Frequency
Pasture			
DJI M3T	13-jul	Sunny	1
DJI M3T	22-ene	Cloudy	1
DJIMavic3Entr	16-nov	Cloudy	3
Forest			
DJI M3T	11-jul	Sunny	1
DJI M3T	19-abr	Overcast	3
Pen			
DJI M3T	12-jul	Cloudy	1
DJI M3T	13-jul	Sunny	2
DJI M3T	19-abr	Overcast	1
Lavf56.15.102	12-jul	Cloudy	1
Lavf56.15.102	13-jul	Sunny	4

Note:
Based on the cloud coverage, Sunny (0-7), Overcast (7-33) and Cloudy (33-100) represent a simplified description of the weather condition.

count livestock.

Although Grounded-SAM is based on Grounding-DINO, surprisingly, the results varied between them. For example, using Grounded-SAM in the first row detected the background instead of the herd. In contrast, Grounding-DINO offers an outstanding performance in the same scenario as Figure 3.3 shows. When the shape of the Grounded-SAM detection is queried, only five bounding boxes are returned.

These results justify the motivation for the choice of selecting R-Tex as the state-of-the-art foundation model for counting livestock in this study. The two following analyses present what the main findings of T-Rex counting performance were from a quantitative and qualitative perspective.

3.2.1 Quantitative analysis

A final total of 15 valid UAV videos provided 19 frames where the models T-Rex and YOLOv8 counted herds ranging from 27 to 1028 sheep in 3 real-world scenarios under different weather conditions. The UAV footage obtained from 6 different days shown 32% sunny weather, 32% overcast conditions and 37% cloudy weather. The average wind speed at 100 metres altitude was 16.85 km/h, reaching its maximum on the 2023-01-22 15:00:00 with a speed of 24.9 , and its minimum was on the 2023-07-11 08:00:00 with a value of 7.3 km/h. Forests comprised 3 of the classified scenarios, while pastures represented 6 images, and pens appeared 11 times.

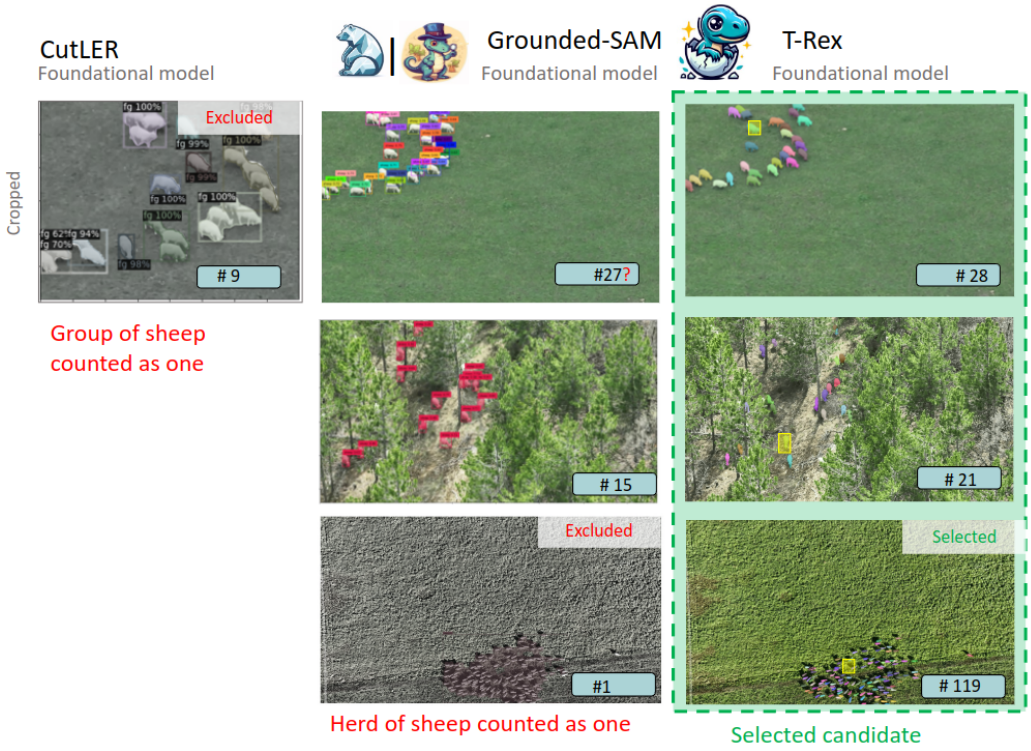


Figure 3.2: The counting performance of the tree foundational models varies depending on the complexity of the scenario. T-Rex performance is remarkable even in complex scenarios with numerous clustered sheep. Other models such as CutLER, Grounded-SAM or Grounding-DINO, were only able to count multiple objects in more simple scenarios.

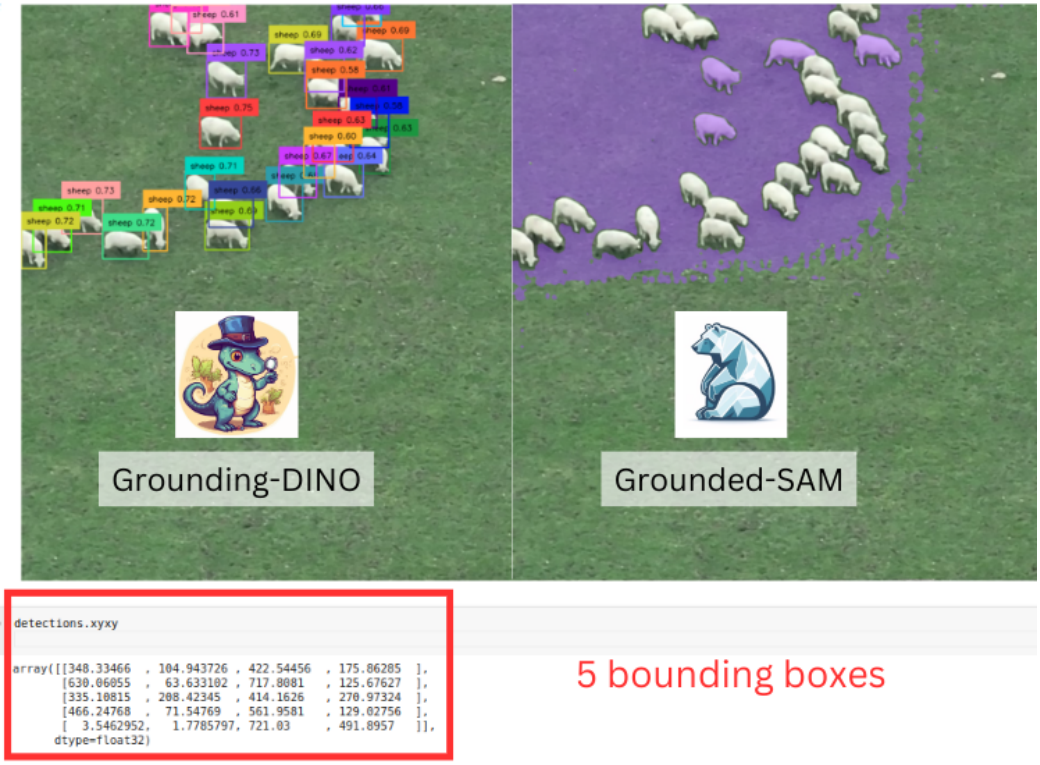


Figure 3.3: Grounding-DINO successfully detects each of the sheep, unlike Grounded-SAM which detects the whole herd. Additionally, when the bounding boxes of the predictions from Grounded-SAM are explored, it returns 5 objects instead of 1 herd.

Table 3.5: The T-Rex model counting performance under different weather conditions tests its model generalization capacity. Pen scenarios with a higher number of sheep clustered present a worse performance with higher MAE values, despite the lower cloud coverage. The images represent the first frame of each video in IDELE dataseet

			T-Rex model performance			Weather conditions	
Scenario	Images	Sheep	MAE	NAE	NAE_min	Clouds	Wind
forest	3	59	28.33	0.41	0.23	21	14
pasture	6	116	36.00	0.22	0.00	62	20
pen	9	165	124.22	0.57	0.04	17	15

Most of the pens classified scenarios are located in pasture, however, its herd density, similar to pen scenes, was the decisive factor in making such a classification. To clarify, the scenario “pen” describes an extremely clustered herd, not necessarily its landscape.

On these classified scenarios, without applying the state-of-the-art T-Rex model, only 1 prediction scored less than 10 MAE using previous neural networks such as YOLOv8. As mentioned in Chapter 2, YOLOv8 was run only by specifying the path of the video using the model shared on Github (Nemati, 2023) representing the performance of a supervised learning model. In contrast, there were 6 predictions with values below 10 MAE using T-Rex. After removing the outliers, the average MAE value showed a difference of 40 points, scoring the T-Rex model 79 MAE, while YOLOv8 scored 119 MAE. The two outliers containing 932 and 1028 sheep were considered anomalies because they did not fit with the rest of the UAV images, which had an average number of sheep of 131. As Figure 3.4 shows, T-Rex MAE were lower than YOLOv8 MAE in all three scenarios.

The forests and pastures scenarios showed a better performance across different weather conditions, presenting lower MAE scores. In fact, pen scenarios scored the worst MAE with a value of 292, while forest an pasture scored 62 and 154 respectively. However, NAE results yielded extreme low values such as 0.04 in the same pen scenarios, meaning almost a perfect prediction. Regarding the weather conditions, similar average cloud coverage with values of 21 for forests and 17 for pen scored extreme different average MAE.

Table 3.5 summarized the T-Rex model performance grouped by the different scenarios and the different weather conditions.

The forests and pastures scenarios shown a better performance across different weather conditions presenting lower MAE scores. In fact, pen scenarios scored the worst MAE with a value of 292, while forest and pasture scored 62 and 154 respec-

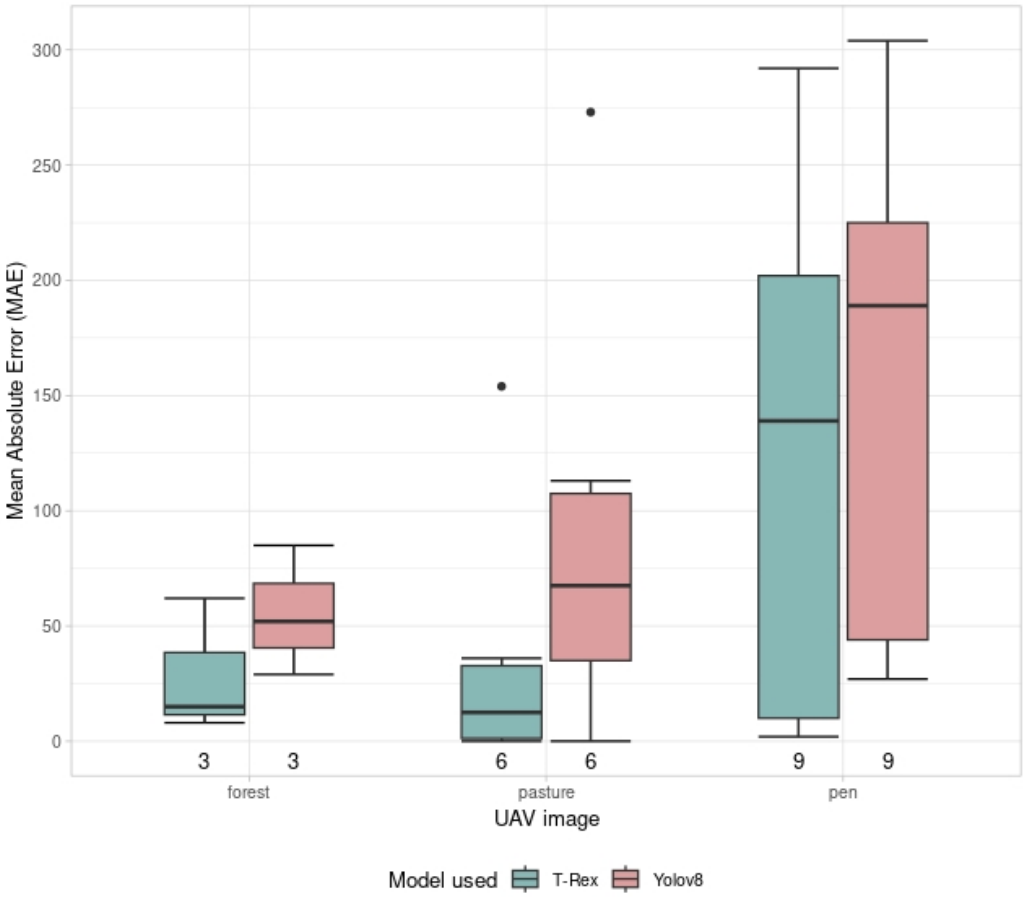


Figure 3.4: The model counting performance of T-Rex and YOLOv8 across the three scenarios is evaluated by the MAE. The foundation model T-Rex scores better with lower MAE than the supervised YOLOv8 model in all three scenarios.

tively. However, NAE results yielded extreme low values such as 0.04 in the same pen scenarios, meaning almost a perfect prediction. Regarding the weather conditions, similar average cloud coverage with values of 21 for forests and 17 for pen scored extreme different average MAE.

Table 3.6: A summary of the ANOVA test shows how each categorical factor affects the model counting performance metric MAE. The weather conditions show a less significant impact than the rest of the factors.

term	df	sumsq	meansq	statistic	p.value
Scenario	2	47755.216	23877.6078	48.357956	0.0001993
Sheep size	2	32262.747	16131.3733	32.669949	0.0005949
Sheep number	2	50696.019	25348.0096	51.335876	0.0001683
Weather	2	3177.440	1588.7199	3.217544	0.1123330
Wind	2	6035.853	3017.9267	6.112035	0.0356876
Residuals	6	2962.608	493.7679	NA	NA

An ANOVA test shown in Table 3.6 shows that the weather conditions had a less significant impact on the MAE predictions using the T-Rex model. The variable "scenario_w" comprised the categories "sunny", "overcast",and "cloudy" depending on the cloud cover. Likewise, "light","moderate", and "strong" were the categories for the "wind_fct" that represented the wind speed. The size and number of sheep along the type of scenario had the highest level of significance, meaning that they could have an effect on the model predictions performance. All the variables, except "scenario_w", scored p-values below 0.05 in the Bartlett test proving their homoscedasticity, an assumption required for ANOVA tests results validity.

In terms of correlations based on numeric variables, a high number of sheep, "ground_truth", was strongly associated with a high MAE showing a strong positive lineal correlation with a Pearson correlation coefficient of 0.91. Therefore, as the number of sheep (ground_truth) increased, also the counting errors (MAE) increased. Conversely, as the cloud cover increased, the MAE decreased following a negative moderate lineal correlation with a Pearson correlation coefficient of -0.4. The wind speed, a feature that characterises a UAV operational condition, did not show any relationship with the model performance.

Figure 3.5 shows the correlations between weather conditions and characteristics of the scenarios, such as the number of sheep, named as "SheepNumber". Variables such as cloud coverage (Clouds) and Sun (short-wave radiation) had a negative relationship. As the cloud coverage increased, the short-wave radiation decreased. Similarly, the number of sheep (SheepNumber) and the number of errors evaluated using NAE or MAE showed a positive correlation. As the number of sheep increased, the errors also increased, leading to worse counting performance.

A comparison of the proportion of the model performance between scenarios and weather conditions showing its distribution across all the UAV images is shown in Figure 3.6.

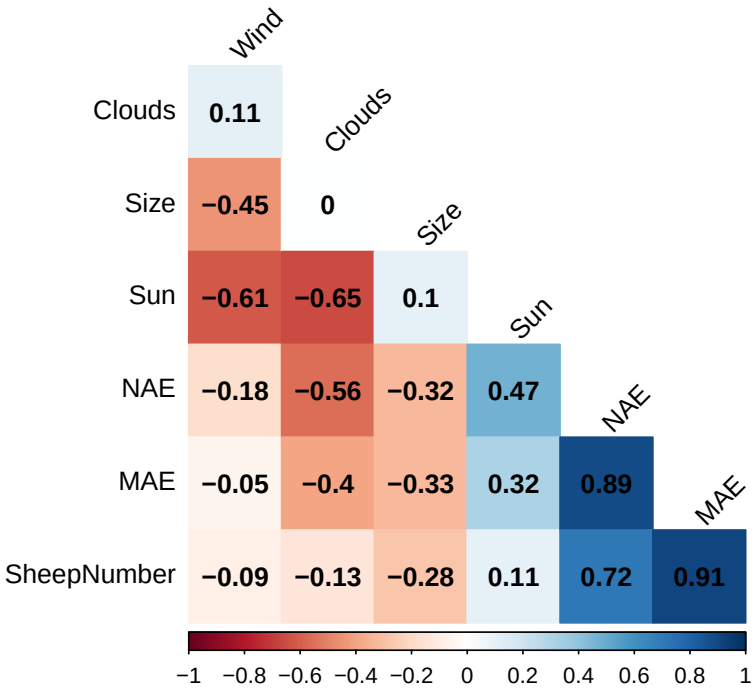


Figure 3.5: The correlation plot displays the relationship between different variables, describing how they are related to each other. The counting performance evaluation metrics, NAE y MAE, show a positive correlation with the number of sheep and a negative correlation with the cloud coverage.

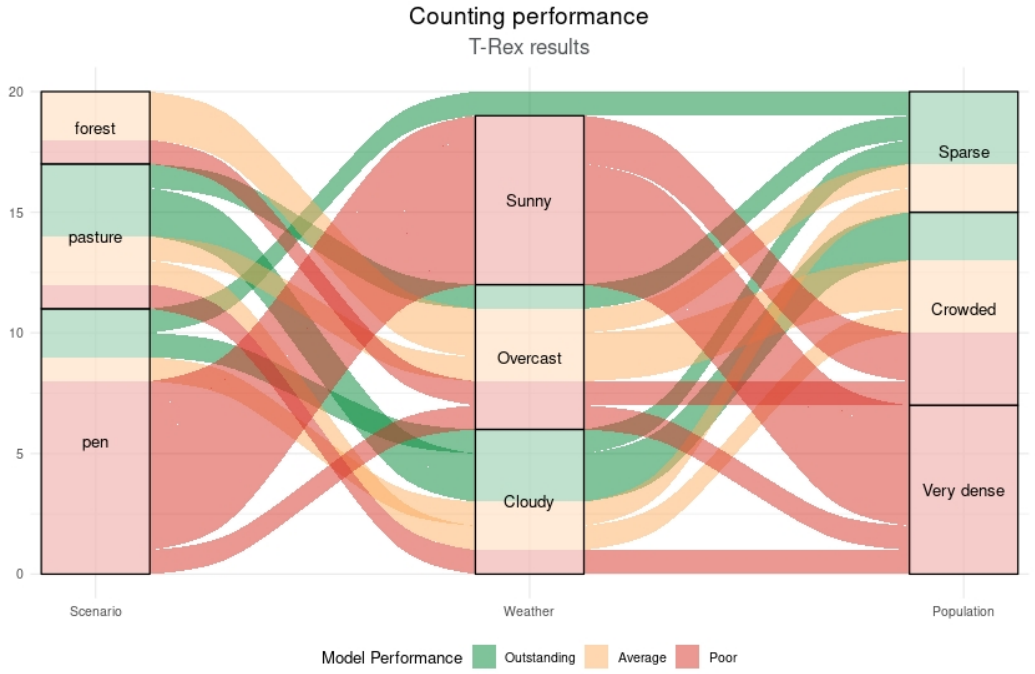


Figure 3.6: The alluvial chart illustrates the distribution of the model performance across heterogeneous scenarios, weather conditions, and the type of sheep population. Herds scattered obtained an outstanding performance overall, while herds with very dense sheep, often in pens scenarios, scored the worst performance.

The first x-axis represented the different scenarios, being "pen" the scenario with poorer performance in contrast with pasture. Most of the poor performance flows from pen scenarios to sunny weather located in the first stratum of the second x-axis. The third x-axis, containing the number of sheep, shows how the "poor" model performance across different weather conditions flows into the stratum "very dense" on this third axis (Population), which had a high number of sheep. Following a similar trend, a sparse population with smaller herds obtained outstanding or average MAE scores across different weather conditions and scenarios. Overall, this alluvial diagram showed higher performance on sparse herds independently of the weather conditions, or the classification of the scenario, except for the sunny days.

Table 3.7 lists the performance of the two compared models, the foundational model T-Rex, and the supervised model YOLOv8 grouped by their scenario classification and including the weather condition. The best cases with the lowest errors for each scenario were the UAV image A, F and M in forest, pasture, and pen, respectively, using T-Rex. In none of those cases, the population or size of the herd

was very dense.In general, a herd with a low number of sheep, for example the UAV image A, offered a better performance compared to other larger herds such as I or K. To compare these cases with the supervised model YOLOv8, T-Rex predicted ($\hat{y}$) 119 sheep out of 119 observed (y) sheep in the UAV image F, while YOLOv8 only predicted ($\hat{y}$) 28 sheep. The worst cases, represented by NAE values close to 1, were on pen scenarios with a very dense population that happens to be mostly sunny. Except for the UAV image D, where YOLOv8 made a perfect prediction, scoring 0 MAE, by and large, T-Rex errors are much lower.

Table 3.7: This table contains the counting performance across the three heterogeneous scenarios using the supervised model YOLOv8, and the foundation model T-Rex. The best performances in each scenario are highlighted in bold.

				T-Rex			YOLOv8		
UAV imagery	Weather	Population	y	$\hat{y}$	MAE	NAE	$\hat{y}$	MAE	NAE
Forest									
A	**Overcast**	**Sparse**	**29**	**21**	**8**	**0.28**	**0**	**29**	**1.00**
B	Overcast	Crowded	85	23	62	0.73	0	85	1.00
C	Overcast	Crowded	64	49	15	0.23	12	52	0.81
Pasture									
D	Overcast	Sparse	27	28	1	0.04	27	0	0.00
E	Overcast	Crowded	60	37	23	0.38	28	32	0.53
F	**Cloudy**	**Crowded**	**119**	**119**	**0**	**0.00**	**28**	**91**	**0.76**
G	Cloudy	Crowded	94	92	2	0.02	50	44	0.47
H	Cloudy	Very dense	275	121	154	0.56	2	273	0.99
I	Cloudy	Crowded	118	82	36	0.31	5	113	0.96
Pen									
J	Overcast	Very dense	306	14	292	0.95	2	304	0.99
K	Sunny	Crowded	192	124	68	0.35	3	189	0.98
L	Sunny	Crowded	143	4	139	0.97	0	143	1.00
M	**Cloudy**	**Sparse**	**50**	**52**	**2**	**0.04**	**23**	**27**	**0.54**
N	Sunny	Very dense	232	30	202	0.87	8	224	0.97
O	Sunny	Very dense	233	31	202	0.87	8	225	0.97
P	Sunny	Very dense	234	33	201	0.86	2	232	0.99
Q	Sunny	Very dense	932	34	898	0.96	5	927	0.99
R	Sunny	Very dense	1028	33	995	0.97	2	1026	1.00
S	Cloudy	Sparse	49	39	10	0.20	5	44	0.90

The bar plot in Figure 3.7 that contains the value from the previous table shows how the state-of-the-art foundation model T-Rex performs better compared to supervised neural networks such as YOLOv8, as errors (MAE) were lower.

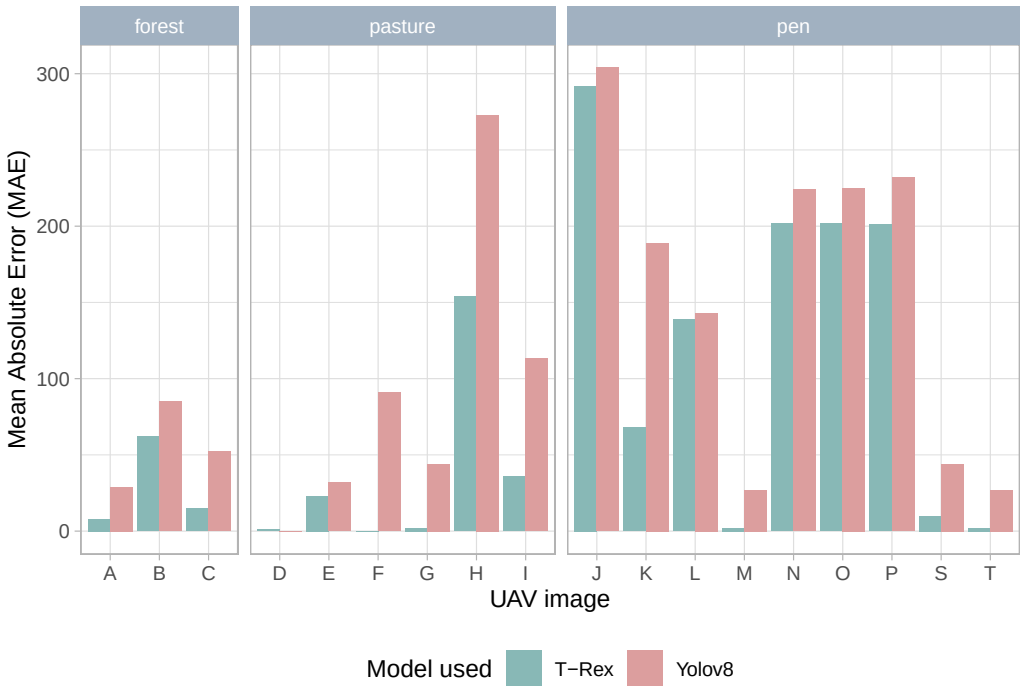


Figure 3.7: An overall view of the variability of the models performance T-Rex and YOLOv8 in the three defined heterogeneous scenarios, pasture, forest and pen. T-Rex counting performance across the heterogeneous scenarios was better than YOLOv8. Additionally, the average number of errors in T-Rex followed the same trend as YOLOv8. The goal of this figure is to compare the counting performance of both models, unlike the Figure 3.6 that only shows the T-Rex performance.

Figure 3.8: The figure shows the counting performance of some selected cases for a qualitative analysis. Row A displays a forest scenario under overcast weather where only T-Rex is able to count sheep blended with the background. Following the same trend, T-Rex made a perfect prediction (NAE 0) while YOLOv8 had a poor performance with NAE 0.76. Lastly, although pen scenarios such as row M pose a challenge, given the bigger size of the sheep, T-Rex almost scored the highest prediction.

3.2.2 Qualitative analysis

This qualitative analysis explored the complexity of heterogeneous scenarios in the UAV videos. Figure 3.8 visualises the highlighted results from Table 3.7, showing the images A,F, and M. The information about the name of the image, its classified scenario, and the weather condition is located on the left side of the figure, while the size or population of the herd is situated on the right side. The Normalized Absolute Error (NAE) measured the models' performance, ranging from 0.00 for a perfect prediction to 1.00 for the worst prediction.

A small yellow rectangle in the images from the T-Rex model indicated where the visual prompt acted, unlike YOLOv8, which did not require any input. This experiment included only a positive rectangle as an input to indicate the object of interest, although it was possible to add more than one and also negative rectangles to exclude other objects. The observed number of sheep considered as the ground truth was an estimation, since they could be hidden by the forest, by the herd itself, or just inaccurate due to human errors. Likewise, the classification of the scenarios

was not always clear to define. The image M is classified as a "pen" scenario with a "sparse" population, however, sheep were grazing on a pasture. Although the size of the herd is small, they appeared very close to each other and were then classified as pen. Despite these errors, in all these images, the foundation model T-Rex performed better than YOLOv8 by a large margin.

One unique specific case where YOLOv8 scored slightly better than T-Rex was the image D, where it predicted 27 sheep out of 27 versus the 28 sheep predicted of T-Rex. Figure 3.9 also shows two cases where preprocessing the data by cropping the area of interest and adding more visual prompting rectangles reduced the NAE metric, increasing the counting model performance. In the image J, T-Rex scored 0.95 indicating a poor performance, whereas after adding one additional positive and negative rectangle, it scored 0.42 NAE reducing its value by 0.53. In terms of observations, this simple adjustment changed the predictions from counting 14 sheep to counting 176 sheep, meaning an improvement of 1157.14%. Conversely, YOLOv8 only improved its performance by counting 4 sheep instead of 2. However, for the image N, YOLOv8 improved its prediction from counting 8 to 195 out of a total of 232. Cropping an area of interest and adding additional visual prompting for T-Rex improved the model's performance in these cases,which are characterised by having a very dense population.

A deeper analysis of the results revealed a series of limitations and errors in counting for both models. The first two columns of Figure 3.10 displayed how both models, T-Rex and YOLOv8, classified dogs, stones, or vehicles as sheep, generating false positives. The third column shows how T-Rex counted one sheep twice, or did not detect a hidden sheep in the forest. These errors revealed how different scenarios affect how much the model can generalise.

During this qualitative explanatory analysis, it was observed that gimbal degree of the UAV had some influence. Figure 3.11 shows a dataset collected locally at Málaga, with the permision of Shepherd "C.". In this occasion, T-Rex was adjusted to count goats changing the category "sheep" for "goat". Each column represents a different gimbal degree, while each row represents a specific height. The counting model performance scored a better performance with a value of 0 NAE at 15m height with 60º degrees in the gimbal. For both heights, a 45º degree had a negative impact on the counting performance.

Figure 3.9: This figure highlights the role of the complexity of the scenario. T-Rex and YOLOv8 counted the herd successfully when only 27 sheep, relatively scattered, appeared on a clear pasture. However, as the number of sheep increased, it was required to crop the image to run both models.

Figure 3.10: False positives and over-counting on both models were errors that compromised the model generalisation ability.

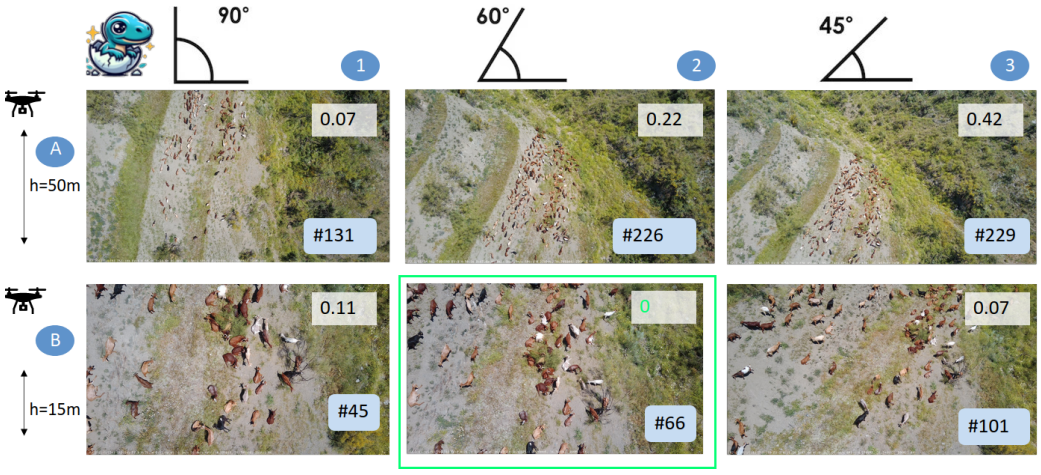


Figure 3.11: T-rex scored a better prediction (NAE=0) when the gimbal degree is set to 60º instead of 90º(NAE= 0.11). However, a gimbal degree on 45º scored a worse performance in both altitudes, at 50m (A) and 15m (B)

3.3 Video tracking: Sheep trajectories

Four videos were selected to show the capacity to track objects using a end-to-end methodology based on foundation models. They are sorted by their performance in descending order, beginning with the better performance first.

3.3.1 Video 1: Small herd in pasture

(https://www.youtube.com/watch?v=L0DTngX_iB4)

This video 1 represents the results from the file "DJI_20230122162230_0019_S.MP4" created on 2023.01.22 from a UAV DJI M3T during a cloudy day in a pasture. All points are tracked successfully with 0 Miss Detection (MD) and 0 Partially Tracked (PT) errors. Unlike most of the UAV videos, the value for the megapixels is 2.0736, suggesting that it was recorded with a zoom. In fact, an observation of the video confirmed it.

3.3.2 Video 2: Large herd in pasture

(https://www.youtube.com/watch?v=OWUjQxoQQkY)

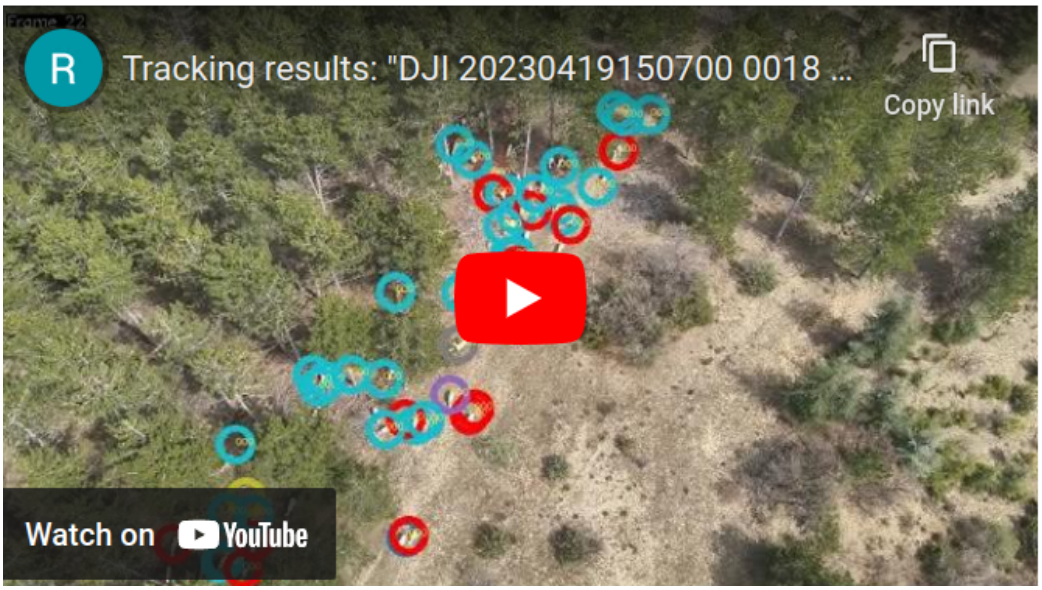
In the same type of pasture scenario, video 2 shows the UAV footage "DJI_20221116104759_0040_V" recorded with a DJI Mavic3 Enterprise on 2022.11.16 also on a cloudy day. On this occasion, the herd is larger than the video 1 with 119 sheep instead of 27. However, there were no missed detections or partially tracking observed. New sheep entering the scenes at the end of the video are not tracked, since they were not present in the first frame. Surprisingly, the occlusions displayed in red did not stop the tracking.

3.3.3 Video 3: Small herd in forest

(https://www.youtube.com/watch?v=K43z7__fn7g)

Unlike the previous two videos, this video 3 from the file "DJI_20230419145636_0003_V_tracked.mp4" includes tracking errors in a forest scenario. There are 3 missed detections, where SAM-PT did not start tracking. One is found in the bottom left corner. On the right corner, 1 partially tracked sheep is observed.

3.3.4 Video 4: Large herd in forest from high altitude.



(https://www.youtube.com/watch?v=OnXlvALXNs4)

Finally, video 4 shows challenging scenarios recorded in the file "DJI_0018_S.MP4". The smaller size of the herd indicates a higher altitude, making it harder to track the small objects. Additionally, unlike the videos 2 and 3, which had 8.29 Megapixels, this video has a value of 2.07 megapixels. It was observed 8 partially tracking. For example, the two sheep in the centre of the footage within the red and orange circles. At least 1 miss detection is observed in the bottom left of the video.

Table 3.8 summarises these tracking performances based on human observations. Since there are no ground truth masks or points to compare with, these values for the tracking performance are not totally accurate and/or precise. The reason not to include dense herd scenarios such as "pen" is the presence of fatal errors in the local machine and system configuration.

Table 3.8: Four videos are sorted by their qualitative tracking performance across different scenarios and conditions. Challenging scenarios such as forests or small objects present in videos 3 and 4 increase the errors, reducing the performance.

Video	FileName	(MD)	(PT)
1	DJI_20230122162230_0019_S.MP4	0	0
2	DJI_20221116104759_0040_V.MP4	0	0
3	DJI_20230419145636_0003_V_tracked.mp4	3	1
4	DJI_20230419150700_0018_S.MP4	1	8

Note:
Manually counting the miss detection in the fourth video is especially challenging, as its true valor is probably higher than one.

Chapter 4

Discussion

This chapter has three sections that answer the three research questions of this study. The first section includes the opportunities in the field of livestock counting based on supervised and self-supervised methods, highlighting T-Rex as a foundation model and YOLOv8 as a self-supervised model. The following section, "Model performance", uses the evaluation metrics MAE and NAE to compare their counting performance. The last section addresses the model generalisations of both models.

This research confirms that a zero-shot machine learning-based system to count livestock achieves better results on the IDELE dataset than a supervised model trained on the Aerial Sheep Computer Vision Project dataset [1]. However, in both cases, specific real-world scenarios still pose a challenge to its implementation. A YOLOv8 supervised model trained on 3609 annotated images accurately predicted only 1 out of the 20 scenes tested. However, without any need for annotated images, the foundation T-Rex model accurately predicted 5 scenes with an error of +2 MAE. The ground truth used to measure the accuracy was a human observation, so although it was carried out systematically and carefully, the ground turth values, and therefore the results, are not free of reasonable biases (Palanisamy and Ratnarajah, 2021). This is especially the case for large and dense herds. In these scenarios, both model counting performances are poor, with high MAE values. The performance of these models illustrates the importance of selecting operational conditions (including altitude, apart from weather conditions) and scenarios to count livestock with some guarantee. In these conditions, the size and density of the herd play a more significant role in the model generalisation capabilities than weather conditions. Therefore, a low UAV height and a sparse herd, such as in the pasture scenarios, facilitate the counting performance of both models.

[1] Dataset containing sheep from birds-eye view

In a broader context, the importance of these findings is that, as long as these conditions are controlled, the adoption of models such as T-Rex in real-world scenarios is feasible. From an economic feasibility perspective, T-Rex does not require high labour costs or a vast amount of time to annotate, images as supervised methods require (Jing and Tian, 2021). In terms of operational feasibility, there is no need for qualified personnel to train T-Rex, which brings the technology closer to the people as SAID Lab aims to. Namely, using T-Rex does not require to apply machine learning from the user since the model is already pre-trained. It could be argued that these costs are lower for classes such as cars or people because there is already manual labelled large-datasets, but this causes imbalanced classes (Emek Soylu et al., 2023). These imbalanced classes limit the model generalisation ability, which means that T-Rex could expand the use of drones on more topics, unlike supervised models. From a technical feasibility point of view, T-Rex authors (especially Lei Zhang and WeiQiang Hu) facilitated on request an API service of T-Rex that could be easily integrated with other technologies. Even when these machine learning-based models offer only a partial solution, a careful design that ensures working within limited conditions makes drone services feasible, shaping a resilient society that manages scarce resources more efficiently.

4.1 Opportunities for livestock counting

A literature review reveals diverse solutions to count livestock based on supervised and self-supervised learning models without a clear best candidate. One of the reasons is that the model performance of a specific model architecture trained with the same dataset varied by 24% when tested on different datasets (Shao et al., 2019). Additionally, different resolutions of the same input images to train the models caused variations in their performance (Rančić et al., 2023). Likewise, the performance of two-stage object detectors against one-stage models is not clear. On the one hand, the two-stage Mask-R-CNN model reported a 4% counting accuracy for livestock UAV images (Xu et al., 2020b). On the other hand, the one-stage YOLOv5x-ECA-SSA had a 5% error rate when counting sheep (Cao et al., 2023). However, in general, it is clear that the current two-stage object detector offers higher accuracy at the cost of slower detection speed compared to one-stage models (Kaur and Singh, 2023).

Independent of the use of a supervised or self-supervised learning methods, their models' performance depends on the complexity of the scene. It was unexpected to read how a one-stage YOLOv3 scored 99.9% accuracy, counting near 1,000 sheep in an hour (Xu et al., 2020b). In this study, the scene was carefully designed by mounting a camera on a horizontal stick under which sheep passed through two races. In contrast, one of the UAV images of this study contained in a single frame 932 sheep located in just one section of the total field of view(Figure 2.5). More recent re-

sults also reported a 99.9% counting accuracy on sheep videos using a registration zone, a count line, and a deregistration zone, as the Plainsight company reported [2] . Moreover, the complexity of one scenario is increased by abrupt changes in recording speed (Bárbulo Barrios et al., 2024). These cases indicate how opportunities to use machine learning-based systems in livestock counting arise when the scene is carefully designed.

However, more complex scenes with shadows and sheep occlusion, such as a forest, lead to tracking and counting errors. For example, the foundation model (Cheng et al., 2023a) DEVA, tested in a forest scenario, detected shadows as sheep and generated duplicate counting. Likewise, the foundation model SAM-PT also reported similar errors. However, they were mitigated when negative points were added, as the authors also indicated (Rančić et al., 2023). For other foundation models such as segment-anything, mitigation measures included new models such as the SAM-Adapter, that increases its performance in camouflaged object detection (Chen et al., 2023). This is also applicable to "pen" scenarios (Figure 3.8), where SAM variations improved its counting accuracy in very dense scenes (Shi et al., 2023b). Apart from selecting state-of-the-art models specially designed to face very dense herds, a nadir view and high environmental temperature causing a low activity of the animals can prevent occlusions (Bárbulo Barrios et al., 2024).

Therefore, a reduction of threats posed by complex scenes, facilitates the implementation of supervised and self-supervised learning models in livestock counting. While providing only partial solutions to the technical problem, there are plenty of opportunities and extensive literature to expand the scientific knowledge on this topic. As this study tested these models in specific real-world scenarios, one of its limitations is that it does not offers an exhaustive comparison using a common standard dataset, which increases the robustness of these comparisons and conclusions.

4.2 Model performance

Previous neural networks such as YOLOv8, a supervised model, offer worse performance than the state-of-the-art model T-Rex, as the higher values in the counting accuracy metric MAE indicate. On average, T-Rex exhibited better counting accuracy on the IDELE dataset, and only in those scenarios where the number of sheep was low and their sizes were large did YOLOv8 offer similar results. For example, both models scored a low MAE value in UAV image D, achieving an outstanding performance. However, the differences in the model performance intensified among scenarios with large herds where sheep also huddled closely. The UAV image H, where T-Rex predicted 121 sheep versus 2 sheep by YOLOv8, was a clear example of these wide differences. The performance of both models drastically dropped with extremely dense

[2] Plainsight case

herds, illustrating how only a partial solution to this technical problem is currently available for some complex scenarios.

Surprisingly, although the number and size of sheep were correlated to the MAE values, in forest scenarios with a small herd of 29 or 85 sheep where a low MAE is expected, YOLOv8 was unable to make any prediction (Figure 3.7) . Specifically, on image A, YOLOv8 did not detect any sheep with a sparse herd of only 29 sheep. In contrast, T-Rex detected 21 sheep. In any case, the ground truth values in forests were more likely to be biassed because sheep are hidden from the human eye, and in some cases only partial parts of their bodies were seen, which made it hard to decide if they should be considered within the sample dataset or discarded. It was also remarkable how T-Rex model performance improved from counting 30 sheep to counting 226 out of 232. Cropping the image and adding two more bounding boxes caused this improvement. For the same image, cropping the image only improved the YOLOv8 performance from counting 8 sheep to counting 195. However, the same cropping process only added 2 sheep to the counter in image J using YOLOv8, while T-Rex added 162 new sheep. To test the influence of the gimbal degree, the small dataset taken with the permission of the local Shepherd "C." indicated that 60º in the gimbal improved the counting performance at a relative height of 15m (3.11). Therefore, these small adjustments heavily influence the model's performance.

In terms of specificity, both models mistakenly identify dogs or vehicles as sheep. In the cropped image J, this is mitigated in T-Rex by including a negative bounding box to indicate what not to count. Without this operation, T-Rex specificity decreases, identifying stones as sheep or even larger objects such as containers or vehicles. Additionally, the reliability of the counting accuracy is at risk when T-Rex counts twice a single sheep. The diversity of shape, pose, and even size of the sheep are obstacles to filtering these irregularities. Related to this, having an oblique view increases the difference in the size of sheep between the background and foreground. This low specificity and the existence of duplicates, where one sheep is counted twice, reduce the reliability of both models' counting accuracy.

Similar to T-Rex, other class-agnostic experimental techniques, which do not rely on classes to count, also struggle with their performance in high density image (Hobley and Prisacariu, 2022). Although these reported papers used the same metrics, such as MAE or NAE, a comparison with our results is not valid as the testing dataset used is FSC-147, CARPK or CA-44 (Jiang et al., 2023a);(Liu et al., 2022);(Ma et al., 2023);(Jiang et al., 2023c) and not in the real-world scenarios explored in this study with the IDELE dataset. However, comparing our results from a qualitative perspective reveals that single objects were also considered multiple separate objects in previous studies (Shi et al., 2023b). Other experimental setups used the ShanghaiTech dataset, made of a "Part_A" mostly dense and a "Part_B" crowded and sparse, and they also obtained a better performance in the "Part_B" following the trend of our results that scored better on the "sparse" and "crowded" scenarios (Shi et al., 2023a).

When comparing YOLOv8 to other related studies, the variety of architectures and training datasets limits the validity of conclusions. For example, studies to detect UAV small targets used YOLOv5 as the baseline, but then they included a small-target-detection (STD) layer, and a multi-scale feature fusion module (MSES) improving the detection of clustered small objects (Cao et al., 2023). In more recent studies using YOLOv8 as a baseline, a deformable convolution and coordinate attention were added to the model to improve cattle detection (Yang et al., 2023b). Apart from the architecture, the amount of training data increases the accuracy of YOLO model (Palanisamy and Ratnarajah, 2021). Even if the same amount of training data is used, the resolution of the input image influences the performance of YOLO (Rančić et al., 2023). However, the difficulties found in the dense "pen" scene is also reported inside barns, where the algorithm TPH-YOLOv5 was specially designed to face these challenges (Pu et al., 2022).

For the "forest" scenarios, similar results were obtained in previous studies. For example, the segment-anything model (SAM) obtained poor performance on camouflaged object detection (COD) benchmark datasets such as CAMO, COD10K and NC4K (Tang et al., 2023). Likewise, the tendency to detect foreground objects in T-Rex, was also observed in studies using SAM (Ji et al., 2023). Additionally, the improvement observed in the cropped images in our dataset is coherent with the adoption of a sliding window prediction strategy that also increased the CountR model performance (Liu et al., 2022). The decrease in the model performance as the occlusions increase is also reported using different techniques, such as convolutional neural networks on thermal UAV imagery (Bárbulo Barrios et al., 2024).

In general, missing sheep in dense herds or identifying stones as sheep validates the limitations found in the original T-Rex study (Jiang et al., 2023a). In terms of methodology, using YOLOv8 with other architectures and training datasets may cause different performance compared to our results. For example, TPH-YOLOv5 is specially designed to tackle one of the limitations of this study: high-density objects and tiny objects. Similarly, the recent model YOLOv5s_MSES obtained better detection on UAV aerial photography images than TPH-YOLOv5 (Cao et al., 2023) and so did the YOLOv5 with ConvMixer prediction (Baidya and Jeong, 2022). Considering this existing literature, implementing other YOLOv8 architectures or increasing its training dataset leads to narrower differences in the model performance compared with T-Rex.

Therefore, the comparison between the YOLOv8 and the state-of-the-art foundational performance models in this study is not generalizable to all YOLOv8 models. This limits the comparison, providing an overview rather than a systematic comparison of every specific YOLOv8 architecture and training possibilities.

4.3 Model generalisation: Weather conditions and scenarios

From the factors that affect the model performance, weather conditions such as wind speed or cloud cover have not had a very significant influence on the predictions. The ANOVA test (Table 3.6) showed that the size of the sheep or their number significantly influenced the model performance measured by the MAE score, however, weather conditions were not significant, and only the wind speed was moderately significant. The correlogram supported the ANOVA test results, specifically, as the number of sheep increased (ground truth) the model performance measured by MAE also increased, exhibiting a strong positive lineal correlation. Although all very dense scenarios exhibits a poor performance, as shown in Figure 3.1, it also reveals poor performance in sunny weather.

Rather than assigning the effect to the sunny weather, it is important to understand the results in a broader context, considering also the type of scenario. Challenging scenarios with a very dense herd are not equally distributed in the testing data. For example, 71% of the scenarios with a very dense herd happen to be in sunny weather. Likewise, 70% of the scenarios classified as "pen" also had sunny weather. Some of the scenarios from the IDELE's UAV footage received were more representative than others, namely, there were 7 images under a scenario classified as "pen" in sunny weather but none in forests or pastures in similar weather conditions. Such a distribution in the UAV images indicates one of the main limitations of this study: the sampling design. This reduces the ability to generalise our findings related to how the weather and types of scenarios affect the model's performance.

Nevertheless, the influence of external factors such as weather conditions, illumination or perspective is confirmed based on reviews of object detection (Kaur and Singh, 2023). In fact, to capture the diversity in illumination, shadow size, and background, this study included UAV footage obtained at different times of day ranging from 07:47 to 15:45 as well as other previous studies did to detect sheep (Sarwar et al., 2020). Likewise, a previous study reported the difficulty of pasture scenarios where the livestock tends to form tightly packed herds (Xu et al., 2020b). In terms of the weather conditions, a study in the southern region of China included rain and fog conditions to evaluate the performance of DCA-YOLOv8 to detect cattle, reporting a significant decrease in the model performance under fog weather (Yang et al., 2023b). Similarly, previous studies to count sheep using Convolutional Neural Network (CNN) to count sheep using UAV imagery reported differences in the model accuracy using training data sets on cloudy, sunny, and mixed weather (Sarwar et al., 2018).

Based on the results of this study, the model T-Rex generalises more than YOLOv8 in all scenarios, offering better model performance by scoring lower MAE values. Table 3.7 from the results section shows how T-Rex obtained a lower average MAE in all scenarios, meaning a lower number of errors. Likewise, the bar plot in Figure 3.7 displayed how YOLOv8 predictions for each single UAV image contained a higher MAE value, meaning a higher number of errors. However, it remains unknown what the results would be if YOLOv8 was trained using the UAV imagery received or a larger dataset taken in the same location. In fact, having a sufficient number of samples is critical to improving the generalisation ability of supervised trained models (He et al., 2021).

In theory, state-of-the-art foundation models such as Grounding DINO (Liu et al., 2023), T-Rex (Jiang et al., 2023a) or Grounded-SAM based on segment-anything (Kirillov et al., 2023) aim to offer a zero-shot generalisation that other supervised models struggle to achieve. However, both types of models are not excluded between them. Foundation models can be used to reduce the cost of annotating images, increasing the availability of larger training datasets for supervised models in smart livestock farming (Tedeschi et al., 2021). Based on the observed results, defining limits on the scenarios and operational conditions by establishing a maximum density or setting a maximum altitude (reasonable size of the target object) for the UAV reduces the barriers to implementing a machine learning-based system for counting livestock. To better describe these operational conditions, one of the limitations of this study was the lack of spatial information, which was only available on the images and not in the videos.

Chapter 5

Conclusions

The last chapter of this master's thesis comprises two sections; the importance of the findings and future implications. While the importance of the findings describes the contribution of this research to solving the problem, the future implications define its limitations and offer suggestions for further research.

5.1 Importance of the findings

Drone services that use foundation models to count sheep are not only useful for the better management of livestock but also for other valuable and scarce goods that demand a resource-efficient management strategy. It could facilitate adjusting the grazing pressure to preserve ecosystem services, and also to give value to nature spaces by quantifying their resources.In fact, the number of units (RU5) is one of the variables to manage the resources of a social-ecological system according to Elinor Ostrom (Ostrom, 2012). Although this study focused on the detection of sheep, the zero-shot capabilities of these foundation models offers flexibility to cover a wide range of applications without an annotated data set or models trained for a specific task.

This study included a literature review of the state-of-the-art self-supervised models, applying some models released in less than a year, such as T-Rex. Additionally, this methodology is able to act as an annotator tool, increasing annotated datasets for supervised models. In fact, the carried out literature review also describes a variety of architectures and different types of models applied in precision livestock farming, which could be used with a richer and larger dataset annotated by foundation models. Lastly, some of the observed results that clearly demonstrate how the model generalisation capacity is limited in situations with very small, dense, or occluded sheep were also found in previous studies finding common barriers.

This brings us to its application in real-world scenarios, which were defined using the metadata obtained from the UAV footage and joined to information from weather stations. This information allows us to find those scenarios where foundation models performed remarkably well, scoring 0.00 NAE. Moreover, it was found that in other challenging scenarios, simple pre-processing procedures such as cropping or zooming reduced the MAE score considerably. Similarly, access to the metadata using Exiftoolr provided abundant information regarding the UAV flight that can be used to create protocols.A limitation for the conclussions drawn from the 3.11 is the different number of sheep in each scene, which should be the same to avoid cofounding factors regarding the effect of the gimbal degree in the counting performance.. These protocols could set a series of parameters to define a safe operational envelope within the foundation model, and counting performance is guaranteed to some extent. For example, avoid taking UAV footage at 120 metres from the ground with a 5x optical zoom and in a forest.

From a technical point of view, this methodology offers a end-to-end solution for counting livestock using these pre-trained models that do not require any training from the user. This is feasible for those scenarios where Grounding DINO is able to detect at least one target object and the object belongs to a pre-defined category. In terms of tracking capability, a very simple methodology, including only one centroid as a query point, allowed us to track the detected sheep along all the video. Regarding this aspect, as kindly suggested by Frano Rajič (one of the main authors of SAM-PT), using trained pose trackers for specific animals and more than one query point, such as in joints, could improve its performance. In general, the results obtained by combining innovative methods such as Grounding DINO, T-Rex and SAM-PT were surprising, since the user needs very little technical knowledge in the challenging field of computer vision.

Regarding the elaboration of the master' thesis, this study involved working with repositories in Github, LaTeX, or Quarto (R) (Allaire and Dervieux, 2024) to increase its reproducibility. The use of code chunks in Quarto, visible to the reader in the github page of this thesis, includes instructions to repeat the study. For this printed version, the use of bookdown (Xie, 2016) with the libraries thesisdown (Ismay and Solomon, 2024) and amasterdown (Reteig, 2024) involved a better knowledge of LaTeX and Rmarakdown. Similarly, the possibility to run the Grounding DINO model using notebooks, but also the countless errors with Python libraries and dependencies, encouraged me to explore reliable options to replicate my work in the future (e.g. Docker). These tools facilitate a clear, transparent methodology and an easier way to collaborate with other peers, ensuring reproducibility.

5.2 Future implications

One of the main limitations of this study is the lack of comparability between the different UAV footage to make valid comparisons. For example, since most of the UAV footage with sunny weather recorded a large herd of clustered sheep, its performance on sunny days is poor. This could be acting as a confounding factor, suggesting that as the sun's activity (short -wave radiation) increase, also the errors (NAE or MAE) increase. Similarly, the UAV footage was taken at very different heights and on occasions with different zooms, affecting the model performance by reducing the size of the object, which makes it harder to detect. Likewise, the conclusions regarding the scenarios are very limited because the number of samples is not equally distributed in each of them. For all these reasons, a sampling protocol for the UAV pilot controlling the height, angle, zoom, or time of the day could reduce the number of confounding factors and make the dataset more representative among the three scenarios.

Following this direction, there could be errors regarding ground truth value, especially in scenes with a large herd (approximately 920 sheep). Due to these factors, it is possible that in certain scenarios where both models scored similarly, a human mistake could have been decisive. Apart from the data, some aspects of the analysis, such as using a qualitative indicator based on the size of the sheep ("crowded","sparse", and "very crowded") reduced the chances of finding meaningful relationships between the different variables. Instead, a statistical analysis such as K-means clustering or the Clark Evans test could define how clustered the centroids of the sheep were from a quantitative point of view. Related to the analysis as well, previous studies indicated how the application of Structure from Motion (SfM) reduced double counts (Shao et al. (2019)). Similarly, defining a polygon zone to count sheep entering the area and adding it to a counter could reduce double counting.

Another important limitation of the tracking process was using only the sheep detected in the first frame. This limitation in the tracking by detection method used in this study could be solved by repeating the process over an interval of frames and introducing new query points. However, another interesting approach could be using a "Detect and Track" approach that increases the accuracy by using tracklets given two or more frames as input. This alternative would add new objects as they appear in the video and improve the model's accuracy by using tracking to detect, instead of tracking the detected objects. Also related to the tracking process, the tracking evaluation metric is limited due to its qualitative perspective. To provide a quantitative evaluation metric, line 168 of the SAM-PT demo contains the trajectories of the SAM-PT predictions. This code from the demo can be modified to store the values and use them in quantitative metrics.

In terms of computing costs, tracking hundreds of query points caused several crashes in the local computer. The solution at that moment was to reduce the length of the video; however, if longer videos or real-time detection are required, another al-

ternative should be considered. Since SAM-PT includes several versions of Segment Anything Model (SAM), one alternative could be using Light HQ-SAM or Mobile-SAM instead of HQ-SAM , both lighter versions of SAM. Lastly, as suggested by Frano Rajič, more query points on joints specifically trained for sheep could be explored to improve their performance.

From the perspective of supervised models, comparing a YOLOv8, trained on a small dataset, with these foundation models could be too specific to make generalisations between all supervised models and all self-supervised models. Moreover, a better explanation of anomalies found in Grounded-SAM (Ren et al., 2024) shown in Figure 3.2, as well as a deeper understanding of the model CutLER (Wang et al., 2023c) could have generated other results. Therefore, future studies could also make a more in depth comparison of these foundation models. Additionally, the search for a benchmark dataset in the field of precision livestock from the literature review of this study and websites such as Paper with Code [1] was unsatisfactory. In the future, a benchmark dataset could also facilitate the evaluation of different models for counting livestock.

Regarding the use of drone services to offer resource-efficient management, this method could be improved by designing a web application where the user could define which category object wants to be detected. Additionally, post-processing procedures could refine the detected masks, not only increasing the model performance by deleting overcounting or false positives, but also filtering specific masks. For example, filtering the masks by their size could detect lambs when detecting livestock. Moreover, adapting this foundation model to UAV on-board technologies could alert the Shepherd when the herd is dispersing too much during grazing. The zero-shot detection of these foundation models combined with UAVs as providers of remote sensing information offers numerous possibilities to build a more resilient society in the future, for example, by creating updated inventories of valuable and scarce resources.

[1] Papers with code-Machine learning datasets: https://paperswithcode.com/datasets

Appendix

Appendix I: Code in R to import and tidy the data from UAV photos and videos.

```
library(exiftoolr)
library(lubridate)
library(tidyverse)
# Photos from UAV footage
## Import UAV metadata from pictures to provide auxiliary
## information
metadata_uav_picture <-exiftoolr::exif_read(
  "/home/ricardo/spatial_data/photo_gps/",
                        recursive =TRUE)
## Tidying data applying format to later be used in
## finding the weather conditions by describing spatial location.
metadata_uav_picture <- metadata_uav_picture %>%
  mutate(
    GPSLatitude=round(GPSLatitude,4),
    GPSLongitude=round(GPSLongitude,4),
    CreateDate = lubridate::ymd_hms(CreateDate),
    datetime = floor_date(CreateDate, unit="hour"),
    date = date(floor_date(CreateDate, unit="hour"))
  )
# Videos
## Import UAV metadata from videos providing the
## research object required to describe
## heterogeneous scenarios.
metadata_uav_videos <- exiftoolr::exif_read(
  "/home/ricardo/spatial_data/video/",
    recursive=TRUE)
## Tidying data applying format to be later be used in
## finding the weather conditions by describing
## temporal location.
```

```
metadata_uav_videos <- metadata_uav_videos %>%
  mutate(CreateDate=lubridate::ymd_hms(CreateDate),
         datetime=round_date(CreateDate, unit="hour"),
         date = date(floor_date(CreateDate, unit="hour")))
## store results
write_csv(metadata_uav_videos,
    "~/thesis_gima/results/metadata_uav_videos.csv")
write_csv(metadata_uav_picture,
    "~thesis_gima/results/metadata_uav_picture.csv")
```

Appendix II: Code in R to find weather conditions using Openmeteo

```
library(openmeteo)
# Define when the UAV footage was taken
vector_date <-  metadata_uav_videos$date |> unique()

# Define where the UAV footage was taken
location <- metadata_uav_picture |>
  filter(FileName == 'DJI_20221116102607_0005_V.JPG') |>
  select(c(GPSLatitude,GPSLongitude)) |>
  as.matrix() |>
  t() |>
  as.vector()

# Define weather conditions of interest
hourly_data <- c("cloud_cover","wind_speed_100m",
                 "shortwave_radiation","diffuse_radiation",
                 "weathercode")
# Apply the openmeteo::weather_history() function
# to the previous date,
# location and retrieve the conditions of interested defined.

## Define function
query_weather <- function(location, date, hourly) {
  result <- openmeteo::weather_history(
                  location=location,
                  start=date,
                  end=date,
                  hourly=hourly
                  )
```

```
  return(result)
}
## Iterate through the dates stored in vector_data()
result_list <- lapply(vector_date, function(date){
  query_weather(location,date,hourly_data)
})
weather_df <- do.call(rbind, result_list)

## Join meta_data_videos_df with weather_df
uav_videos_weather <-left_join(metadata_uav_videos,
                               weather_df,
                               by ="datetime")
uav_videos_weather <- uav_videos_weather %>%
 mutate(weather_description =recode(hourly_weathercode,
  "0" ="Cloud development not observed or not observable",
  "1" ="Clouds generally dissolving or becoming less developed",
  "2" ="State of sky on the whole unchanged",
  "3" ="Clouds generally forming or developing",
  "51"="Drizzle, not freezing, continuous",
  "53"="Drizzle, not freezing, continuous moderate
  at time of observation",
    "61"="Rain, not freezing, continuous slight
  at time of observation"))
## store results
write_csv(uav_videos_weather,
  "~/thesis_gima/results/uav_videos_weather.csv")
```

Appendix III: Joining the UAV footage data with the hourly weather history from openmeteo and defining scenarios.

```
# Import labels manually:
uav_videos_weather_scenarios <-uav_videos_weather |>
    select(FileName, Encoder, CreateDate,
            hourly_weathercode, weather_description,
            hourly_cloud_cover,hourly_wind_speed_100m,
            datetime) |>
    mutate(date = date(CreateDate),
      month_day = format(as.Date(
        uav_videos_weather$date,"%Y%m%d"),"%d-%b"),
          scenario = c('Dense herd','Dense herd',
```

```
                        'Dense herd','Small herd',
                        'Occluded herd','Dense herd',
                        'Small herd','Dense herd',
                        'Dense herd','Occluded herd',
                        'Occluded herd','Dense herd',
                        'Small herd','Dense herd',
                        'Dense herd','Small herd',
                       'Small herd','Occluded herd')) |>
                 rename(cloud_coverage = hourly_cloud_cover)
# Tidy UAV footage grouping by variables and transforming
# into frequency
scenarios_freq <- uav_videos_weather_scenarios   |>
                      mutate(cloud_cat = cut(
                        cloud_coverage,
                        breaks = c(-Inf,7,33,100,Inf),
                        labels = c("Sunny",
                                   "Overcast",
                                   "Cloudy",
                                   "Error"))) |>
                      group_by(Encoder, scenario,
                      month_day,cloud_cat) %>%
                        summarise(Freq = n())

# Discretization of numeric cloud coverage into three
# categorical values

# Dense
scenarios_freq$dense <- with(scenarios_freq,
          ifelse(scenario=="Dense herd","Yes","No"))
# Small

scenarios_freq$small <- with(scenarios_freq,
          ifelse(scenario=="Small herd","Yes","No"))

# Occluded
scenarios_freq$occluded <- with(scenarios_freq,
         ifelse(scenario=="Occluded herd","Yes","No"))

## store results
```

```
write_csv(scenarios_freq,
"~/thesis_gima/results/uav_videos_scenarios.csv")
```

Appendix IV: How to analyse the heterogeneous scenarios using an alluvial chart

```
# Visualize with ggplot
library(ggalluvial)
library(cowplot)
## Small herd (pasture)
p1 <- ggplot(as.data.frame(scenarios_freq),
       aes(y = Freq,
           axis1 = Encoder, axis2 = month_day,
           axis3=cloud_cat)) +
  geom_alluvium(aes(fill = small), width = 1/12) +
  geom_stratum(width = 1/12, fill = "white", color = "grey") +
  geom_label(stat = "stratum",
    aes(label = after_stat(stratum))) +
  scale_x_discrete(limits = c("UAV", "Date","Weather"),
                   expand = c(.05, .05)) +
  scale_fill_manual(values = c("Yes" = "#aacbaa",
                               "No" = "grey85")) +
  theme_minimal()
## Occluded herd (forest)
p2 <- ggplot(as.data.frame(scenarios_freq),
       aes(y = Freq,
           axis1 = Encoder, axis2 = month_day,
           axis3=cloud_cat)) +
  geom_alluvium(aes(fill = occluded), width = 1/12) +
  geom_stratum(width = 1/12, fill = "white", color = "grey") +
  geom_label(stat = "stratum",
    aes(label = after_stat(stratum))) +
  scale_x_discrete(limits = c("UAV", "Date","Weather"),
                   expand = c(.05, .05)) +
  scale_fill_manual(values = c("Yes" = "#aacbaa",
                               "No" = "grey85")) +
  theme_minimal()
## Dense herd (Pen)
p3 <- ggplot(as.data.frame(scenarios_freq),
       aes(y = Freq,
    axis1 = Encoder, axis2 = month_day, axis3=cloud_cat)) +
```

```
  geom_alluvium(aes(fill = dense), width = 1/12) +
  geom_stratum(width = 1/12, fill = "white", color = "grey") +
  geom_label(stat = "stratum",
    aes(label = after_stat(stratum))) +
  scale_x_discrete(limits = c("UAV", "Date","Weather"),
          expand = c(.05, .05)) +
  scale_fill_manual(values = c("Yes" = "#aacbaa",
                               "No" = "grey85")) +
  theme_minimal()
## Arranging small, occluded and dense plot into one plot
## and store the result.
png("~/thesis_gima/results/scenarios_distribution.png",
    width=740, height=820)
cowplot::plot_grid(p1,p2,p3, ncol=1, labels="AUTO")
dev.off()
```

Appendix V: How to obtain all the bounding boxes in a unique file.

```
library(tidyverse)
library(kableExtra)
# Pre-processing individual measurements
# to import them as an unique dataframe
csv_directory <- "~/dimensions_bbox"
list_of_files <- list.files(path = csv_directory,
                            recursive = TRUE,
                            pattern = "\\.csv$",
                            full.names = TRUE)
# Importing all individual individual measurements
# and selecting the relevant variables for T-Rex
df_measurement_first_bbox <- read_delim(list_of_files, id = "file_name", delim =
  mutate(file_name = basename(list_of_files),
         upper_left_x = BX,
         upper_left_y = BY,
         lower_left_x = BX+Width,
         lower_left_y = BY+Height) |>
  select(c(file_name,upper_left_x,upper_left_y,
           lower_left_x,lower_left_y, Area))
write.csv(df_measurement_first_bbox,
  "~/measurements_first_sheep.csv")
```

Appendix VI: Python used in Google Colab from Roboflow notebook to load

Grounding DINO model.

```
%cd {HOME}
!git clone https://github.com/IDEA-Research/GroundingDINO.git
%cd {HOME}/GroundingDINO
!pip install -q -e .
!pip install -q roboflow
!pip install -r
requirements.txt --use-deprecated=legacy-resolver
## Check that the hardware accelerator is set to GPU
!nvidia-smi
## Download weights and saved them in weights folder.
%cd {HOME}
!mkdir {HOME}/weights
%cd {HOME}/weights
!wget -q https://github.com/IDEA-Research/GroundingDINO
/releases/download/v0.1.0-alpha/
groundingdino_swint_ogc.pth
## Import functions to define the model
%cd {HOME}/GroundingDINO
from groundingdino.util.inference
import load_model, load_image, predict, annotate
from groundingdino.util.inference import Model
grounding_dino_model = Model(model_config_path=
"~/config/GroundingDINO_SwinT_OGC.py",
model_checkpoint_path=GROUNDING_DINO_CHECKPOINT_PATH)
model = load_model(CONFIG_PATH, WEIGHTS_PATH)
```

Appendix VII: Python used in Google Colab from Roboflow notebook to load Grounding DINO model.

```
# Import auxiliary modules for visualization purposes
import os
import supervision as sv
# Load the image of interest
IMAGE_NAME = "DJI_0003_V.png"
IMAGE_PATH = os.path.join("/content/data/", IMAGE_NAME)
# Set parameters to identify sheep with different thresholds
# for the category (text) and the object (box)
TEXT_PROMPT = "sheep"
BOX_TRESHOLD = 0.25
```

```
TEXT_TRESHOLD = 0.25
# Assign values using load_image()
image_source, image = load_image(IMAGE_PATH)
# Run the model to obtain boxes, accuracy level
# and the category.
boxes, logits, phrases = predict(
    model=model,
    image=image,
    caption=TEXT_PROMPT,
    box_threshold=BOX_TRESHOLD,
    text_threshold=TEXT_TRESHOLD
)
# Visualize the result
annotated_frame = annotate(image_source=image_source,
boxes=boxes, logits=logits, phrases=phrases)
%matplotlib inline
sv.plot_image(annotated_frame, (16, 16))
# Sort the bounding boxes by its accuracy to obtain
# the sheep with a higher logits score.
indices_sorted= torch.sort(logits, descending=True)
boxes[indices_sorted.indices[0]]
```

Appendix VIII: Adapting the bounding box to T-Rex API format.

```
library(tidyverse)
# Merge individual measurement into an unique file
csv_directory <- "~/thesis_gima/results/uav_videos_first_frame/
dimensions_bbox"
list_of_files <- list.files(path = csv_directory,
                            recursive = TRUE,
                            pattern = "\\.csv$",
                            full.names = TRUE)
# Import the unique file and create the variable file_name()
# to store the name of each file as records
df <- read_delim(list_of_files, id = "file_name",
                 delim = ",") |>
  mutate(file_name = basename(list_of_files)) # this is
# the name of the images populated in a new column
# named "file_name"
```

```
# Create a list to store bounding box coordinates from
# ChatGPT + personal adjusting.
bounding_boxes <- as.character(nrow(df))
# Loop through each row of the dataframe
for (i in 1:nrow(df)) {
  # Calculate bounding box coordinates for each observation
  bounding_box <- paste("[", df$BX[i], ",", df$BY[i], ",",
  df$BX[i] + df$Width[i], ",", df$BY[i] + df$Height[i], "]",
  sep = "")
  # Assign bounding box coordinates to the corresponding row
  bounding_boxes[i] <- bounding_box
}
# Create a dataframe with the file name
# and bounding box coordinates
bounding_boxes_df <- data.frame(name = df$file_name,
bounding_box = bounding_boxes)
```

Appendix IX: T-Rex code facilitated by WeiQiang Hu from the Lei Zhang's team in IDEA. This chunk obtains the results.

```
# Loading modules
import requests
from PIL import Image
from rle_utils import rle2mask
from rle_utils import string2rle
from file_utils import upload_image
# Import image and token
token = "token_from_IDEA"  # The token included in the email
image = "path_to_image.png"
# Step 1: Upload local image to server
image_url = upload_image(image, token)
# Step 2: Trigger an async task and get it's uuid
api = "https://apitest.deepdataspace.com/tasks/ivp"
data = {
    # The image to be inferred, the image containing the
    # objects you want to count
    # The image url must be directly visible on the internet
    "infer_image" : image_url,
    # The image the prompts are acting on, the image you draw
    #a rect on
```

```
    # The image url must be directly visible on the internet
    "prompt_image": image_url,
    # The label types to be inferred,
    # possible values are "bbox" and "mask"
    "label_types" : ["bbox", "mask"],
    # The prompts to be used in the inference
    "prompts"     : [
        {
            "type"       : "rect",  # The type of the prompt,
            # only "rect" is supported now
            "is_positive": True,  # the prompt is positive,
            # only True is supported now
            "rect"       : [  # the rect area, upper_left_x,
              #upper_left_y, lower_right_x,
              #lower_right_y
                1190,
                1610,
                1259,
                1756
            ],
        }
    ]
}
rsp = requests.post(api, json=data, headers={"Token": token})
rsp_data = rsp.json()  # {'code': 0, 'data':
  # {'task_uuid': '77fe4952-4ce7-4830-89a8-9f60a02d1a59'},
  # 'msg': 'ok'}
task_uuid = rsp_data["data"]["task_uuid"]
# Step 3: Query task result by uuid in loop until it's done
task_result = None
while True:
api = f"https://apitest.deepdataspace.com/task_statuses/{task_uuid}"
    rsp = requests.get(api, headers={"Token": token})
    rsp_data = rsp.json()
    if rsp_data["data"]["status"] == "waiting":
        print(f"task {task_uuid} is waiting")
    elif rsp_data["data"]["status"] == "running":
        print(f"task {task_uuid} is running")
    elif rsp_data["data"]["status"] == "failed":
        print(f"task {task_uuid} is failed")
```

```
        print(rsp.text)
        exit(1)
    elif rsp_data["data"]["status"] == "success":
        print(f"task {task_uuid} is success")
        task_result = rsp_data["data"]["result"]
        break
    time.sleep(1)
```

Appendix X: The code from IDEA was slightly modified to store every result from T-Rex using an accumulator array.

```
# Step 4: Parse result and get mask
## String values transformed into RLE
def string2rle(rle_str: str) -> List[int]:
    p = 0
    cnts = []

    while p < len(rle_str) and rle_str[p]:
        x = 0
        k = 0
        more = 1

        while more:
            c = ord(rle_str[p]) - 48
            x |= (c & 0x1f) << 5 * k
            more = c & 0x20
            p += 1
            k += 1

            if not more and (c & 0x10):
                x |= -1 << 5 * k

        if len(cnts) > 2:
            x += cnts[len(cnts) - 2]
        cnts.append(x)
    return cnts

## RLE values transformed into masks
def rle2mask(cnts: List[int], size: Tuple[int, int], label=1):
    img = np.zeros(size, dtype=np.uint8)
```

```
    ps = 0
    for i in range(0, len(cnts), 2):
        ps += cnts[i]

        for j in range(cnts[i + 1]):
            x = (ps + j) % size[1]
            y = (ps + j) // size[1]

            if y < size[0] and x < size[1]:
                img[y, x] = label
            else:
                break

        ps += cnts[i + 1]

    return img
# Step 5. Creating the RBA image that contained the objects
# detect in the alpha channel.
if task_result is not None:
    objects = task_result["objects"]
    print(f"Detected {len(objects)} objects")

    # Initialize accumulator array
    accumulator = np.zeros((2160, 3840), dtype=np.uint8)

    # Process every detected object
    for idx, obj in enumerate(objects):
        score = obj["score"]
        print(f"[{idx + 1}/{len(objects)}]',
        "Processing object of score {score}")
        bbox = obj["bbox"]
        print(f"Bbox of object: {bbox}")

        # Read the compressed rle string, convert it to a
        # numpy 1/0 array
        mask = obj["mask"]
        counts = mask["counts"]
        rle = string2rle(counts)
        shape = (2160, 3840)  # height, width
        mask_array = rle2mask(rle, shape)
```

```
        # Accumulate the mask
        accumulator = np.logical_or(accumulator, mask_array)

    # Convert the accumulated array to a 4-channel RGBA image
    mask_alpha = np.where(
      accumulator == 1, 255, 0).astype(np.uint8)
    mask_rgba = np.stack((255 * np.ones_like(mask_alpha),
                          255 * np.ones_like(mask_alpha),
                          255 * np.ones_like(mask_alpha),
                          mask_alpha),
                         axis=-1)

    # Save the accumulated mask to an image file
    image = Image.fromarray(mask_rgba, "RGBA")
    image_path = "DJI_003V.png"
    image.save(image_path)
    print(f"Saved combined mask image to PNG file at {image_path}")
```

Appendix XI: Code to compute centrois from object detection

```
num_objects = len(task_result["objects"])
import sys

# Redirect standard output to a file
sys.stdout = open('query_points__DJI_00028.txt', 'w')

# Print the initial line with index 1
print("1")
for i in range(num_objects):
    if 'bbox' in task_result["objects"][i]:
        bbox_value = task_result["objects"][i]['bbox']
        x_min, y_min, x_max, y_max = bbox_value

        # Calculate centroid
        centroid_x = (x_min + x_max) / 2
        centroid_y = (y_min + y_max) / 2

        # Format the output
        print(f"{0}; {centroid_x:.1f}, {centroid_y:.1f}")
    else:
```

```
        print(f"{i + 1}; No bounding box found")
# Restore standard output
sys.stdout.close()
sys.stdout = sys.__stdout__
```

Appendix XII: The code from SAM-PT github public repository to run the non-interactive demo.

```
# Code copied from SAM-PT public repository
# Convert bees.mp4 to png frames
mkdir data/demo_data/DJI_20230419145636_0003_V
ffmpeg -i data/demo_data/DJI_0003_V_cropped.mp4
-vf fps=5 data/demo_data/DJI_0003_V_cropped/frame-%05d.png

# Run non-interactive demo on DJI_0003_V.mp4
export HYDRA_FULL_ERROR=1
python -m demo.demo \
  frames_path='${hydra:runtime.cwd}/data/demo_data/DJI_0003_V/' \
  query_points_path='${hydra:runtime.cwd}/data/demo_data
  '/query_points_DJI_0003_V.txt' \
  longest_side_length=1024 frame_stride=1 max_frames=-1
```

Acknowledgments

I would like to express my gratitude to my Supervisor, Dr. João Ricardo Pereira Valente, whose words of encouragement from the beginning to the end helped me gain the confidence that I lacked throughout this endeavour. Likewise, I want to express my gratitude to Dr. Kooistra Lammert, who accepted the position of Responsible Professor without hesitation and was always available to provide feedback.

I am extremely grateful to Manuel López Antequera, who, as an expert, introduced me to the concept of this research on foundation models by sharing the latest developments in SAM, Grounding DINO, or SAM-PT with me. As a friend, his guidance was essential in my decision to pursue a master's degree in Europe. Without him, I wouldn't have had the unique experience of undertaking this GIMA master's programme. The numerous collaborative projects with my colleagues have been incredibly enriching, as I always learned from them and from the programme professors invaluable feedback. Additionally, I also had colleagues like Mór Grommers, who helped me as a friend to face challenges, I wouldn't have managed on my own. This thesis is also the result of our periodic meetings, where we shared the joys and struggles of working on our theses.

It is essential to recognise that the literature review of this study is not the only part built upon the contribution of researchers. In December, I found myself utterly surprised for hours by the results from the recent model T-Rex. After contacting Mr Lei Zhang, the lead scientist at IDEA, my astonishment grew even larger. He not only offered me the opportunity to test its model, but also connected me with Mr. WeiQiang Hu. Apart from providing a script for testing T-Rex, he was always there to solve any problems I encountered, no matter how small they were. Mr. Lei Zhang and Mr. WeiQiang Hu showed authentic interest in my humble thesis by inquiring me about my needs, so they could offer greater assistance. Moreover, Mr. WeiQiang Hu generously offered tailored solutions for me without seeking anything in return, for which I will always be thankful. The credit for the T-Rex API methodology section belongs to him. This genuine concern for a master's student, while participating in events such as the Computer Vision and Pattern Recognition Conference (CVPR) or engaging in meetings with national government leaders [2], is

[2] Mr Lei Zhang, the lead scientist of IDEA and the T-Rex project, meets the Spanish President in

for me an undoubted proof of the incredible human value of the team led by Mr. Lei Zhang and himself.

Likewise, I was very fortunate to contact Frano Rajič, one of the authors of SAM-PT. Apart from providing specific answers to my questions, he shared his expertise with extensive commentaries and suggestions. Parts of the conclusions are attributed to him. Similarly, I would like to recognise that the idea of using Exiftool came from Inti Luna Aviles, who has extensive knowledge on remote sensing as well as an outstanding willingness to collaborate and share ideas. Piotr Skalski was the author of several Google Colab, Youtube videos, and blogs about Grounding DINO [3] and SAM. Lastly, this printed version was originally made by Leon Reteig, who kindly offered his support, making it possible to use his library amsterdown. The Richard J Telford blog "Enough Markdown to Write a Thesis" [4], and its github repository "biostats" [5] were the foundation for this thesis's online interactive version.

Finally, words cannot express my gratitude to my family and close friends, who are always there offering their unconditional support. My father, Manuel Ruiz Rivas that found himself installing Ubuntu, Windows, or Manjaro operating systems countless times (i.e more than five times) without complaining, even when my clumsiness and near-zero knowledge on Python libraries and graphical drivers caused the problems. My mother, and brother, María del Carmen Sánchez Sierra, and Adrián Ruiz Sánchez, with whom I am very lucky to share my time. Unfortunately, I do not know if my friend Jorge Calabuig Bartual still thinks the same after helping me so many times (i.e close to five times) with the github repository of this thesis.

2024: https://www.lamoncloa.gob.es/multimedia/galeriasfotograficas/presidente/Paginas/2024/280224-sanchez-encuentro-ceo-envision.aspx?galf1er=0&galf2r=1

[3] Piotr Skalski Google Colab on how to use Grounding DINO

[4] The blog post "Enough Markdown to Write a Thesis

[5] Biostats repository by Richard Telfrod

Bibliography

Sam-pt: Extending sam to zero-shot video segmentation with point-based tracking. https://github.com/SysCV/sam-pt. [Accessed 22-05-2024].

Lourdes Agapito, Michael M Bronstein, and Carsten Rother, editors. *Computer vision - ECCV 2014 workshops*. Lecture notes in computer science. Springer International Publishing, Cham, Switzerland, 2015 edition, March 2015.

Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking, 2022. URL https://arxiv.org/abs/2206.14651.

JJ Allaire and Christophe Dervieux. *quarto: R Interface to 'Quarto' Markdown Publishing System*, 2024. URL https://CRAN.R-project.org/package=quarto. R package version 1.4.

C. Aquilani, A. Confessore, R. Bozzi, F. Sirtori, and C. Pugliese. Review: Precision livestock farming technologies in pasture-based livestock systems. *Animal*, 16(1): 100429, January 2022. ISSN 1751-7311. doi: 10.1016/j.animal.2021.100429. URL http://dx.doi.org/10.1016/j.animal.2021.100429.

Joel Arweiler, Cihan Ates, Jesus Cerquides, Rainer Koch, and Hans-Jörg Bauer. Similarity-based framework for unsupervised domain adaptation: Peer reviewing policy for pseudo-labeling. *Machine Learning and Knowledge Extraction*, 5(4): 1474–1492, October 2023. ISSN 2504-4990. doi: 10.3390/make5040074. URL http://dx.doi.org/10.3390/make5040074.

Ranjai Baidya and Heon Jeong. Yolov5 with convmixer prediction heads for precise object detection in drone imagery. *Sensors*, 22(21):8424, November 2022. ISSN 1424-8220. doi: 10.3390/s22218424. URL http://dx.doi.org/10.3390/s22218424.

Alejandro Belanche, A. Ignacio Martín-García, Javier Fernández-Álvarez, Javier Pleguezuelos, Ángel R. Mantecón, and David R. Yáñez-Ruiz. Optimizing management of dairy goat farms through individual animal data interpretation: A case study of smart farming in spain. *Agricultural Systems*, 173:27–38, July 2019. ISSN

0308-521X. doi: 10.1016/j.agsy.2019.02.002. URL http://dx.doi.org/10.1016/j.agsy.2019.02.002.

J M Bewley, R A Russell, K A Dolecheck, and M R Borchers. 1.1. precision dairy monitoring: what have we learned? In *Precision livestock farming applications*, pages 13–24. Wageningen Academic Publishers, The Netherlands, May 2015.

Tilo Burghardt and Will Andrew. Friesiancattle2015, 2016. URL http://data.bris.ac.uk/data/dataset/wurzq71kfm561ljahbwjhx9n3/.

Diego Bárbulo Barrios, João Valente, and Frank van Langevelde. Monitoring mammalian herbivores via convolutional neural networks implemented on thermal uav imagery. *Computers and Electronics in Agriculture*, 218:108713, March 2024. ISSN 0168-1699. doi: 10.1016/j.compag.2024.108713. URL http://dx.doi.org/10.1016/j.compag.2024.108713.

Jesse Cambon, Diego Hernangómez, Christopher Belanger, and Daniel Possenriede. tidygeocoder: An r package for geocoding. *Journal of Open Source Software*, 6(65): 3544, 2021. doi: 10.21105/joss.03544. URL https://doi.org/10.21105/joss.03544. R package version 1.0.5.

Yuanyang Cao, Jian Chen, and Zichao Zhang. A sheep dynamic counting scheme based on the fusion between an improved-sparrow-search yolov5x-eca model and few-shot deepsort algorithm. *Computers and Electronics in Agriculture*, 206:107696, March 2023. ISSN 0168-1699. doi: 10.1016/j.compag.2023.107696. URL http://dx.doi.org/10.1016/j.compag.2023.107696.

Guang Chen, Haitao Wang, Kai Chen, Zhijun Li, Zida Song, Yinlong Liu, Wenkai Chen, and Alois Knoll. A survey of the four pillars for small object detection: Multiscale representation, contextual information, super-resolution, and region proposal. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(2): 936–953, February 2022. ISSN 2168-2232. doi: 10.1109/tsmc.2020.3005231. URL http://dx.doi.org/10.1109/TSMC.2020.3005231.

Junzhang Chen and Xiangzhi Bai. Learning to "segment anything" in thermal infrared images through knowledge distillation with a large scale dataset satir, 2023. URL https://arxiv.org/abs/2304.07969.

Tianrun Chen, Lanyun Zhu, Chaotao Ding, Runlong Cao, Yan Wang, Zejian Li, Lingyun Sun, Papa Mao, and Ying Zang. Sam fails to segment anything? – sam-adapter: Adapting sam in underperformed scenes: Camouflage, shadow, medical image segmentation, and more, 2023. URL https://arxiv.org/abs/2304.09148.

Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking anything with decoupled video segmentation. 2023a. doi: 10.48550/ARXIV.2309.03903. URL https://arxiv.org/abs/2309.03903.

Joe Cheng, Barret Schloerke, Bhaskar Karambelkar, and Yihui Xie. *leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library*, 2024. URL https://CRAN.R-project.org/package=leaflet. R package version 2.2.2.

Man Cheng, Hongbo Yuan, Qifan Wang, Zhenjiang Cai, Yueqin Liu, and Yingjie Zhang. Application of deep learning in sheep behaviors recognition and influence analysis of training data characteristics on the recognition effect. *Computers and Electronics in Agriculture*, 198:107010, July 2022. ISSN 0168-1699. doi: 10.1016/j.compag.2022.107010. URL http://dx.doi.org/10.1016/j.compag.2022.107010.

Yangming Cheng, Liulei Li, Yuanyou Xu, Xiaodi Li, Zongxin Yang, Wenguan Wang, and Yi Yang. Segment and track anything, 2023b. URL https://arxiv.org/abs/2305.06558.

Ruining Deng, Can Cui, Quan Liu, Tianyuan Yao, Lucas W. Remedios, Shunxing Bao, Bennett A. Landman, Lee E. Wheless, Lori A. Coburn, Keith T. Wilson, Yaohong Wang, Shilin Zhao, Agnes B. Fogo, Haichun Yang, Yucheng Tang, and Yuankai Huo. Segment anything model (sam) for digital pathology: Assess zero-shot segmentation on whole slide imaging, 2023. URL https://arxiv.org/abs/2304.04155.

Dewey Dunnington. *ggspatial: Spatial Data Framework for ggplot2*, 2023. URL https://CRAN.R-project.org/package=ggspatial. R package version 1.1.9.

Mohamed Elgendy. *Deep learning for vision systems*. Manning Publications, New York, NY, January 2021.

Busra Emek Soylu, Mehmet Serdar Guzel, Gazi Erkan Bostanci, Fatih Ekinci, Tunc Asuroglu, and Koray Acici. Deep-learning-based approaches for semantic segmentation of natural scene images: A review. *Electronics*, 12(12):2730, June 2023. ISSN 2079-9292. doi: 10.3390/electronics12122730. URL http://dx.doi.org/10.3390/electronics12122730.

Aurelien Geron. *Hands-on machine learning with scikit-learn, keras, andTensorFlow*. O'Reilly Media, Sebastopol, CA, 2 edition, October 2019.

Vincenzo Gervasi, John D.C. Linnell, Tomaž Berce, Luigi Boitani, Rok Cerne, Paolo Ciucci, Benjamin Cretois, Daniela Derron-Hilfiker, Christophe Duchamp, Adrienne Gastineau, Oksana Grente, Djuro Huber, Yorgos Iliopoulos, Alexandros A. Karamanlidis, Ilpo Kojola, Francesca Marucco, Yorgos Mertzanis, Peep Männil, Harri Norberg, Nives Pagon, Luca Pedrotti, Pierre-Yves Quenette, Slaven Reljic,

Valeria Salvatori, Tõnu Talvi, Manuela von Arx, and Olivier Gimenez. Ecological correlates of large carnivore depredation on sheep in europe. *Global Ecology and Conservation*, 30:e01798, October 2021. ISSN 2351-9894. doi: 10.1016/j.gecco.2021.e01798. URL http://dx.doi.org/10.1016/j.gecco.2021.e01798.

Nicholas J. Gotelli and Aaron M. Ellison. *A primer of ecological statistics*. Sinauer Associates Publishers, Sunderland, Mass, 2004. ISBN 9780878932696. OCLC: ocm54865386.

Karlo Hainsch, Konstantin Löffler, Thorsten Burandt, Hans Auer, Pedro Crespo del Granado, Paolo Pisciella, and Sebastian Zwickl-Bernhard. Energy transition scenarios: What policies, societal attitudes, and technology developments will realize the eu green deal? *Energy*, 239:122067, January 2022. ISSN 0360-5442. doi: 10.1016/j.energy.2021.122067. URL http://dx.doi.org/10.1016/j.energy.2021.122067.

Xiaowei He, Rao Cheng, Zhonglong Zheng, and Zeji Wang. Small object detection in traffic scenes based on yolo-mxanet. *Sensors*, 21(21):7422, November 2021. ISSN 1424-8220. doi: 10.3390/s21217422. URL http://dx.doi.org/10.3390/s21217422.

Michael Hobley and Victor Prisacariu. Learning to count anything: Reference-less class-agnostic counting with weak supervision, 2022. URL https://arxiv.org/abs/2205.10203.

Tracey Hollings, Mark Burgman, Mary van Andel, Marius Gilbert, Timothy Robinson, and Andrew Robinson. How do you find the green sheep? a critical review of the use of remotely sensed imagery to detect and count animals. *Methods in Ecology and Evolution*, 9(4):881–892, February 2018. ISSN 2041-210X. doi: 10.1111/2041-210x.12973. URL http://dx.doi.org/10.1111/2041-210X.12973.

IDEA-Research. Official implementation of the paper "grounding dino: Marrying dino with grounded pre-training for open-set object detection". https://github.com/IDEA-Research/GroundingDINO. [Accessed 22-05-2024].

Chester Ismay and Nick Solomon. *thesisdown: An updated R Markdown thesis template using the bookdown package*, 2024. R package version 0.2.0.9000.

Wei Ji, Jingjing Li, Qi Bi, Tingwei Liu, Wenbo Li, and Li Cheng. Segment anything is not always perfect: An investigation of sam on different real-world applications. 2023. doi: 10.48550/ARXIV.2304.05750. URL https://arxiv.org/abs/2304.05750.

Qing Jiang, Feng Li, Tianhe Ren, Shilong Liu, Zhaoyang Zeng, Kent Yu, and Lei Zhang. T-rex: Counting by visual prompting, 2023a. URL https://arxiv.org/abs/2311.13596.

Ruixiang Jiang, Lingbo Liu, and Changwen Chen. Clip-count: Towards text-guided zero-shot object counting. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23. ACM, October 2023b. doi: 10.1145/3581783.3611789. URL http://dx.doi.org/10.1145/3581783.3611789.

Ruixiang Jiang, Lingbo Liu, and Changwen Chen. Clip-count: Towards text-guided zero-shot object counting. 2023c. doi: 10.48550/ARXIV.2305.07304. URL https://arxiv.org/abs/2305.07304.

Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4037–4058, November 2021. ISSN 1939-3539. doi: 10.1109/tpami.2020.2992393. URL http://dx.doi.org/10.1109/tpami.2020.2992393.

David Kahle and Hadley Wickham. ggmap: Spatial visualization with ggplot2. *The R Journal*, 5(1):144–161, 2013. URL https://journal.r-project.org/archive/2013-1/kahle-wickham.pdf.

Ravpreet Kaur and Sarbjeet Singh. A comprehensive review of object detection with deep learning. *Digital Signal Processing*, 132:103812, January 2023. ISSN 1051-2004. doi: 10.1016/j.dsp.2022.103812. URL http://dx.doi.org/10.1016/j.dsp.2022.103812.

Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality, 2023. URL https://arxiv.org/abs/2306.01567.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023. URL https://arxiv.org/abs/2304.02643.

Onur Can Koyun, Reyhan Kevser Keser, İbrahim Batuhan Akkaya, and Behçet Uğur Töreyin. Focus-and-detect: A small object detection framework for aerial images. *Signal Processing: Image Communication*, 104:116675, May 2022. ISSN 0923-5965. doi: 10.1016/j.image.2022.116675. URL http://dx.doi.org/10.1016/j.image.2022.116675.

VisAI Labs. Evaluating multiple object tracking accuracy and performance metrics in a real-time setting. https://visailabs.com/evaluating-multiple-object-tracking-accuracy-and-performance-metrics-in-a-real-time-setting/, 2021. [Accessed 22-05-2024].

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei

Chang, and Jianfeng Gao. Grounded language-image pre-training, 2021. URL https://arxiv.org/abs/2112.03857.

Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection, 2022. URL https://arxiv.org/abs/2203.16527.

John D.C. Linnell, Benjamin Cretois, Erlend B. Nilsen, Christer M. Rolandsen, Erling J. Solberg, Vebjørn Veiberg, Petra Kaczensky, Bram Van Moorter, Manuela Panzacchi, Geir R. Rauset, and Bjørn Kaltenborn. The challenges and opportunities of coexisting with wild ungulates in the human-dominated landscapes of europe's anthropocene. *Biological Conservation*, 244:108500, April 2020. ISSN 0006-3207. doi: 10.1016/j.biocon.2020.108500. URL http://dx.doi.org/10.1016/j.biocon.2020.108500.

Chang Liu, Yujie Zhong, Andrew Zisserman, and Weidi Xie. Countr: Transformer-based generalised visual counting, 2022. URL https://arxiv.org/abs/2208.13721.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

Zhiheng Ma, Xiaopeng Hong, and Qinnan Shangguan. Can sam count anything? an empirical study on sam counting, 2023. URL https://arxiv.org/abs/2304.10817.

Gengchen Mai, Weiming Huang, Jin Sun, Suhang Song, Deepak Mishra, Ninghao Liu, Song Gao, Tianming Liu, Gao Cong, Yingjie Hu, Chris Cundy, Ziyuan Li, Rui Zhu, and Ni Lao. On the opportunities and challenges of foundation models for geospatial artificial intelligence, 2023. URL https://arxiv.org/abs/2304.06798.

Robert H. Marrs, HyoHyeMi Lee, Sabena Blackbird, Leslie Connor, Susan E. Girdwood, Michael O'Connor, Simon M. Smart, Robert J. Rose, John O'Reilly, and Richard C. Chiverrell. Release from sheep-grazing appears to put some heart back into upland vegetation: A comparison of nutritional properties of plant species in long-term grazing experiments. *Annals of Applied Biology*, 177(1):152–162, April 2020. ISSN 1744-7348. doi: 10.1111/aab.12591. URL http://dx.doi.org/10.1111/aab.12591.

Lothar Mueller, Viktor G Sychev, Nikolai M Dronin, and Frank Eulenstein, editors. *Exploring and optimizing agricultural landscapes*. Innovations in Landscape Research. Springer Nature, Cham, Switzerland, 1 edition, June 2021.

Subhrajit Nag, Yash Khandelwal, Sparsh Mittal, C. Krishna Mohan, and A. Kai Qin. Arcn: A real-time attention-based network for crowd counting from drone images. In *2021 IEEE 18th India Council International Conference (INDICON)*. IEEE, December 2021. doi: 10.1109/indicon52576.2021.9691659. URL http://dx.doi.org/10.1109/INDICON52576.2021.9691659.

Mostafa Nemati. Yolov8 aerial sheep detection and counting. simulated on gazebo. https://github.com/monemati/YOLOv8-Sheep-Detection-Counting, 2023. [Accessed 22-05-2024].

Su Myat Noe, Thi Thi Zin, Pyke Tin, and Ikoub Kobyashi. Efficient segment-anything model for automatic mask region extraction in livestock monitoring. In *2023 IEEE 13th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*. IEEE, September 2023. doi: 10.1109/icce-berlin58801.2023.10375624. URL http://dx.doi.org/10.1109/ICCE-Berlin58801.2023.10375624.

Joshua O'Brien. *exiftoolr: ExifTool Functionality from R*, 2024. URL https://CRAN.R-project.org/package=exiftoolr. R package version 0.2.4.

Elinor Ostrom. *The Future of the Commons: Beyond Market Failure and Government Regulation*. Number 148 in Occasional Paper / Institute of Economic Affairs. Inst. of Economic Affairs, London, 2012. ISBN 978-0-255-36653-3 978-0-255-36681-6.

Vigneshwaran Palanisamy and Nagulan Ratnarajah. Detection of wildlife animals using deep learning approaches: A systematic review. In *2021 21st International Conference on Advances in ICT for Emerging Regions (ICter)*. IEEE, December 2021. doi: 10.1109/icter53630.2021.9774826. URL http://dx.doi.org/10.1109/icter53630.2021.9774826.

Edzer Pebesma. Simple Features for R: Standardized Support for Spatial Vector Data. *The R Journal*, 10(1):439–446, 2018. doi: 10.32614/RJ-2018-009. URL https://doi.org/10.32614/RJ-2018-009.

Tom Pisel. *openmeteo: Retrieve Weather Data from the Open-Meteo API*, 2023. URL https://CRAN.R-project.org/package=openmeteo. R package version 0.2.4.

Jingyu Pu, Chengjun Yu, Xiaoyan Chen, Yu Zhang, Xiao Yang, and Jun Li. Research on chengdu ma goat recognition based on computer vison. *Animals*, 12(14):1746, July 2022. ISSN 2076-2615. doi: 10.3390/ani12141746. URL http://dx.doi.org/10.3390/ani12141746.

Yongliang Qiao, Yangyang Guo, and Dongjian He. *Deep Learning-Based Autonomous Cow Detection for Smart Livestock Farming*, page 246–258. Springer International

Publishing, 2023. ISBN 9783031261183. doi: 10.1007/978-3-031-26118-3_19. URL http://dx.doi.org/10.1007/978-3-031-26118-3_19.

Frano Rajič, Lei Ke, Yu-Wing Tai, Chi-Keung Tang, Martin Danelljan, and Fisher Yu. Segment anything meets point tracking. *arXiv:2307.01197*, 2023.

Kristina Rančić, Boško Blagojević, Atila Bezdan, Bojana Ivošević, Bojan Tubić, Milica Vranešević, Branislav Pejak, Vladimir Crnojević, and Oskar Marko. Animal detection and counting from uav images using convolutional neural networks. *Drones*, 7(3):179, March 2023. ISSN 2504-446X. doi: 10.3390/drones7030179. URL http://dx.doi.org/10.3390/drones7030179.

Raquel Zarraoa, Miguel Cazorla Quevedo, and Francisco Gomez-Donoso. *Animal Detection by Drone*. PhD Thesis, Universitat d'Alacant Escuela Politécnica Superior, Alicante, January 2022.

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024.

Leon Reteig. *amsterdown: An RMarkdown PhD thesis template for the University of Amsterdam*, 2024. URL https://github.com/lcreteig/amsterdown. R package version 0.1.1.9000.

Riis. Aerial sheep dataset. https://universe.roboflow.com/riis/aerial-sheep, jun 2022. URL https://universe.roboflow.com/riis/aerial-sheep. visited on 2024-05-17.

roboflow/notebooks. Automated dataset generation with grounding dino + segment anything model (sam). https://colab.research.google.com/github/roboflow-ai/notebooks/blob/main/notebooks/automated-dataset-annotation-and-evaluation-with-grounding-dino-and-sam.ipynb#scrollTo=LoD2bIptG-qO. [Accessed 22-05-2024].

Farah Sarwar, Anthony Griffin, Priyadharsini Periasamy, Kurt Portas, and Jim Law. Detecting and counting sheep with a convolutional neural network. November 2018. doi: 10.1109/avss.2018.8639306. URL http://dx.doi.org/10.1109/avss.2018.8639306.

Farah Sarwar, Anthony Griffin, Saeed Ur Rehman, and Timotius Pasang. Towards detection of sheep onboard a uav, 2020. URL https://arxiv.org/abs/2004.02758.

Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld,

Benjamin Schmid, Jean-Yves Tinevez, Daniel James White, Volker Hartenstein, Kevin Eliceiri, Pavel Tomancak, and Albert Cardona. Fiji: an open-source platform for biological-image analysis. *Nature Methods*, 9(7):676–682, June 2012. ISSN 1548-7105. doi: 10.1038/nmeth.2019. URL http://dx.doi.org/10.1038/nmeth.2019.

Wen Shao, Rei Kawakami, Ryota Yoshihashi, Shaodi You, Hidemichi Kawase, and Takeshi Naemura. Cattle detection and counting in uav images based on convolutional neural networks. *International Journal of Remote Sensing*, 41(1):31–52, June 2019. ISSN 1366-5901. doi: 10.1080/01431161.2019.1624858. URL http://dx.doi.org/10.1080/01431161.2019.1624858.

Zenglin Shi, Pascal Mettes, and Cees G. M. Snoek. Focus for free in density-based counting, 2023a. URL https://arxiv.org/abs/2306.05129.

Zenglin Shi, Ying Sun, and Mengmi Zhang. Training-free object counting with prompts, 2023b. URL https://arxiv.org/abs/2307.00038.

Kamil Slowikowski. *ggrepel: Automatically Position Non-Overlapping Text Labels with 'ggplot2'*, 2024. URL https://CRAN.R-project.org/package=ggrepel. R package version 0.9.5.

Lv Tang, Haoke Xiao, and Bo Li. Can sam segment anything? when sam meets camouflaged object detection, 2023. URL https://arxiv.org/abs/2304.04709.

Luis O Tedeschi, Paul L Greenwood, and Ilan Halachmi. Advancements in sensor technology and decision support intelligent tools to assist smart livestock farming. *Journal of Animal Science*, 99(2), February 2021. ISSN 1525-3163. doi: 10.1093/jas/skab038. URL http://dx.doi.org/10.1093/jas/skab038.

Kang Tong, Yiquan Wu, and Fei Zhou. Recent advances in small object detection based on deep learning: A review. *Image and Vision Computing*, 97:103910, May 2020. ISSN 0262-8856. doi: 10.1016/j.imavis.2020.103910. URL http://dx.doi.org/10.1016/j.imavis.2020.103910.

Jan C. van Gemert, Camiel R. Verschoor, Pascal Mettes, Kitso Epema, Lian Pin Koh, and Serge Wich. *Nature Conservation Drones for Automatic Localization and Counting of Animals*, page 255–270. Springer International Publishing, 2015. ISBN 9783319161785. doi: 10.1007/978-3-319-16178-5_17. URL http://dx.doi.org/10.1007/978-3-319-16178-5_17.

Jehan-Antoine Vayssade, Xavier Godard, and Mathieu Bonneau. Wizard: Unsupervised goats tracking algorithm. *Computers and Electronics in Agriculture*, 209:107831, June 2023. ISSN 0168-1699. doi: 10.1016/j.compag.2023.107831. URL http://dx.doi.org/10.1016/j.compag.2023.107831.

Gaihua Wang and Peng Jin. Research on uav small target detection algorithm based on improved yolov5s. In *2023 19th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. IEEE, July 2023. doi: 10.1109/icnc-fskd59587.2023.10280829. URL http://dx.doi.org/10.1109/ICNC-FSKD59587.2023.10280829.

Xiaobin Wang, Ye Yan, Haohui Sun, and Dekang Zhu. Dense-and-similar object detection in aerial images. *Pattern Recognition Letters*, 176:153–159, December 2023a. ISSN 0167-8655. doi: 10.1016/j.patrec.2023.10.028. URL http://dx.doi.org/10.1016/j.patrec.2023.10.028.

Xudong Wang, Rohit Girdhar, Stella X. Yu, and Ishan Misra. Cut and learn for unsupervised object detection and instance segmentation, 2023b. URL https://arxiv.org/abs/2301.11320.

Xudong Wang, Ishan Misra, Ziyun Zeng, Rohit Girdhar, and Trevor Darrell. Videocutler: Surprisingly simple unsupervised video instance segmentation. *arXiv preprint arXiv:2308.14710*, 2023c.

Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking, 2019. URL https://arxiv.org/abs/1909.12605.

Yihui Xie. *bookdown: Authoring Books and Technical Documents with R Markdown*. Chapman and Hall/CRC, Boca Raton, Florida, 2016. ISBN 978-1138700109. URL https://bookdown.org/yihui/bookdown.

Beibei Xu, Wensheng Wang, Greg Falzon, Paul Kwan, Leifeng Guo, Guipeng Chen, Amy Tait, and Derek Schneider. Automated cattle counting using mask r-cnn in quadcopter vision system. *Computers and Electronics in Agriculture*, 171:105300, April 2020a. ISSN 0168-1699. doi: 10.1016/j.compag.2020.105300. URL http://dx.doi.org/10.1016/j.compag.2020.105300.

Beibei Xu, Wensheng Wang, Greg Falzon, Paul Kwan, Leifeng Guo, Zhiguo Sun, and Chunlei Li. Livestock classification and counting in quadcopter aerial images using mask r-cnn. *International Journal of Remote Sensing*, 41(21):8121–8142, April 2020b. ISSN 1366-5901. doi: 10.1080/01431161.2020.1734245. URL http://dx.doi.org/10.1080/01431161.2020.1734245.

Jinyu Yang, Mingqi Gao, Zhe Li, Shang Gao, Fangjing Wang, and Feng Zheng. Track anything: Segment anything meets videos, 2023a. URL https://arxiv.org/abs/2304.11968.

Wenjie Yang, Jiachun Wu, Jinlai Zhang, Kai Gao, Ronghua Du, Zhuo Wu, Eksan Firkat, and Dingwen Li. Deformable convolution and coordinate attention for

fast cattle detection. *Computers and Electronics in Agriculture*, 211:108006, August 2023b. ISSN 0168-1699. doi: 10.1016/j.compag.2023.108006. URL http://dx.doi.org/10.1016/j.compag.2023.108006.

Hongyang Yu, Guorong Li, Weigang Zhang, Qingming Huang, Dawei Du, Qi Tian, and Nicu Sebe. The unmanned aerial vehicle benchmark: Object detection, tracking and baseline. *International Journal of Computer Vision*, 128(5):1141–1159, December 2019. ISSN 1573-1405. doi: 10.1007/s11263-019-01266-1. URL http://dx.doi.org/10.1007/s11263-019-01266-1.

Chunhui Zhang, Li Liu, Yawen Cui, Guanjie Huang, Weilin Lin, Yiqian Yang, and Yuehong Hu. A comprehensive survey on segment anything model for vision and beyond, 2023. URL https://arxiv.org/abs/2305.08196.

Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection, 2022. URL https://arxiv.org/abs/2203.03605.

Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything, 2023. URL https://arxiv.org/abs/2306.12156.

Zhi-Hua Zhou. Machine learning challenges and impact: an interview with thomas dietterich. *National Science Review*, 5(1):54–58, May 2017. ISSN 2053-714X. doi: 10.1093/nsr/nwx045. URL http://dx.doi.org/10.1093/nsr/nwx045.

Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Heng Fan, Qinghua Hu, and Haibin Ling. Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7380–7399, November 2022. ISSN 1939-3539. doi: 10.1109/tpami.2021.3119563. URL http://dx.doi.org/10.1109/TPAMI.2021.3119563.

Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey, 2019. URL https://arxiv.org/abs/1905.05055.