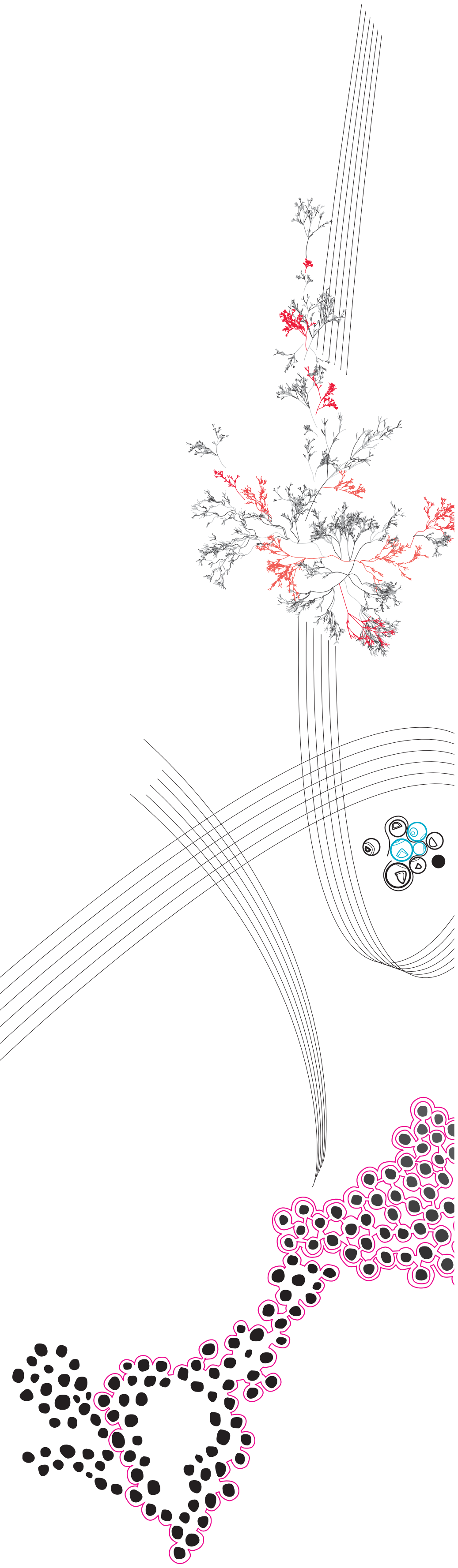


BSc Thesis Advanced Technology

Detecting, Tracking, and Identifying Horses across Heterogeneous Videos

Max M. Lievense

Assignment committee:

Chairman	A. Chiumento
Supervisor	J. Kamminga
External Member	A. Keemink

June 2021

Pervasive Systems
Faculty of Electrical Engineering, Mathematics and Computer Science
University of Twente, Enschede

UNIVERSITY OF TWENTE.

Abstract

Being able to localize unique subjects across a collection of heterogeneous videos in an unsupervised manner is a challenging task. A task that humans are able to perform quite accurately. We can analyse and understand scenes, easily follow subjects through a video, and are able to re-identify lost subjects based on their appearance. Implementing these skills into a computer system crosses various computer vision and artificial intelligence fields. State of the art applications have been created for use cases where humans are the subject. This thesis aims to tackle the research question: Which state of the art applications can be utilized to extract the information about the occurrence of unique subjects from heterogeneous videos and what are the limitations of these existing applications. Based on these applications and limitations, a pipeline was created with YoloV4+DeepSORT and FairMOT that can detect, track and re-identify, adapted for non-human subjects to finally output the desired information. The subject type used in this thesis are horses, however, this thesis is applicable to any other subject type with a suitable training set.

The two limitations that were found in the re-identification task are 1) the incapability to extract long-term information from horses resulting in insufficient accuracy when attempting to re-identify subjects and 2) the online method of tracking resulting in undesired identify transfers. Suggestions on how to improve these limitations are given. The final pipeline was able to detect 93% of the horses within the evaluation frames and was able to minimize the number of identity transfers to 5 within the evaluation fragments.

Contents

Chapter 1

Introduction

Developments in computer vision have been trying to simulate the basic biological system, such as the ability to recognize movements, understand scenes, detect, follow and identify an object. Humans can glance at an image and instantly be able to processes all these abilities accurately, however -in a world of automation- the human is to be replaced with a computerized counterpart. The ability to automate the collection, analysis and processing of data with the use of Artificial Intelligence is an objective where much of research has been devoted to. A successful method of replacing the human actor can be beneficial in many use cases such as video analysis [1, 2], video surveillance [3–5], activity recognition [6, 7] and animal habitat preservation [8–12]. In these use cases, the ability to single out subjects from a video collection automatically can aid in the performance of tasks. For example, when labelling the activities of a subject, following that particular subject through multiple videos would eliminate the need to search for that subject.

This thesis attempts to develop an entirely unsupervised (without the need for human interaction) pipeline that is able to process multiple raw video recordings with unknown configurations, unknown duration, unknown placements and an unknown number of subjects. From such a heterogeneous input, the analysis and processing would refer to the extraction of useful information per unique subject that appears in the entire collection of input videos. Given the multiple input videos, Video to Video (V2V) Re-Identification (re-ID) will have to performed.

The proposed pipeline would need to perform 3 distinct tasks: Detection objects, tracking objects and re-ID subjects. The detection of objects localizes the subjects in each individual frame of the video, defining Bounding Boxs (BBs) around the objects. This process can be compared to the understanding of a scene, deciding where the relevant information is. The defined BBs are linked to another by the tracker, creating sequences of detections that follow a unique subject. Lastly, these subjects within sequences need to be linked to subjects in the entirety of the input collection using re-ID'ing. re-ID'ing corresponds to an associations problem that uses information from the sequences like the appearance of the subject in the sequence, the location and movement direction of the subject in the video, and the timings that the sequences are active.

There exist State of the Art (SOTA) applications that can handle one or multiple of these tasks. Detectors often use Convolutional Neural Network (CNN) to localize the desired subjects [13–18]. Tracking applications often use mathematical algorithms to link detections from frame to frame, resulting in a matrix of most-probable links [1, 2, 19–21]. Additionally, trackers often use parts of the re-ID'er to correctly link frames into sequences, in the form of temporal feature extraction. The feature extraction for re-ID identifies what parts of the subject is significant and creates a comparable key for each subject [3–5, 22–28]. It should be noted that the re-ID of that the tracker uses and the re-ID for V2V re-ID are different, as the tracker can use more temporal information of the subject (only valid for a short amount of time), whereas the V2V re-ID defines global features of entirety of subject [29]. Some applications can perform all tasks end-to-end [30, 31].

Given the above context, this thesis aims to tackle the research question: Which SOTA applications can be utilized to extract the information about the occurrence of unique horses from heterogeneous videos and what are the limitations of these existing applications?

This thesis will approach the question from a practical point of view, attempting to identifying and solving problems of existing applications rather than researching and creating an entirely new pipeline. To limit the scope of the approach, the subjects will be horses from a dataset explained in section 3 on page 8. It should be noted that the proposed pipeline will be able to handle wide applicability of subject types, not being exclusive for horses. A change of subject would only require the retraining of the models for that particular subject or multiple subjects. To obtain the aforementioned database from raw video, the application should perform 3 distinct steps. Another restriction that determines the approach of this thesis is the limitation of the allowed training data of the pipeline. As the proposed pipeline is to be unsupervised, the training of Neural Network (NN) is not allowed to be performed on the input data. Lastly, the applications that will be considered have to be free and open-source due to the proof-of-concept nature of this thesis. Figure 1.1 illustrates the 3 steps mentioned above which the pipeline should perform to obtain the necessary information from a collection of raw heterogeneous videos.

Firstly, this thesis explores the field in which this assignment exists, describing various methods that existing works have implemented. Secondly, an analysis is done on the evaluation dataset and the creation of the training dataset is explained. This is done in the three different chapters for each distinct task: Horse detection, Horse tracking and Horse re-Identification. In each of these chapters, the challenges and requirements are given, followed by a description of each used and otherwise considered applications. Each task has its own evaluation and discussion. Lastly, the final chapters discuss the results of the entire pipeline, suggest the possible solutions to the ensued issues and concludes this thesis.

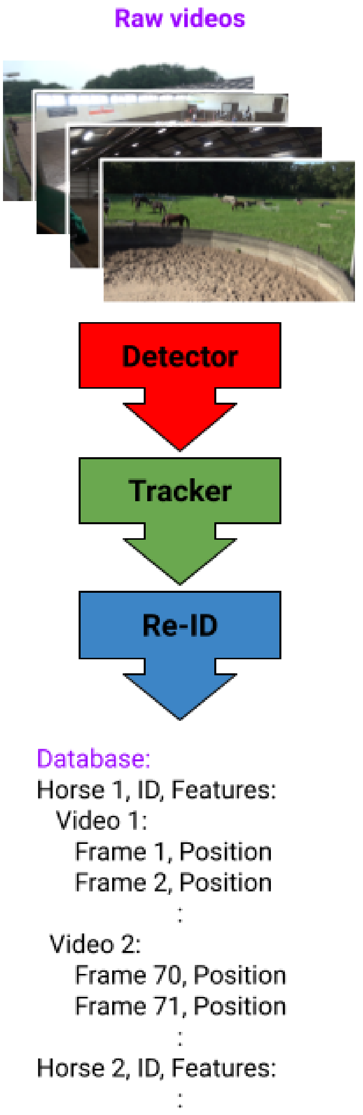


Figure 1.1: An illustration of the flow of the thesis. The raw video footage input is passed through the 3 distinct tacks: Detection, Tracking and Re-Identification to finally output a database containing when and where each individual horse is displayed in a video, per each individual horse.

Chapter 2

Related work

2.1 Object Detection Research

Classification of objects within an image is a well-researched topic. With the increasing number of images that are uploaded to the internet, a need for easy but accurate classification of images has emerged. Several database frameworks have been created with labelled images that can be used to train custom NNs. ImageNet [32], OpenImages [33], PASCAL VOC [34] and COCO [35] are opensource databases with associated challenges which are used to evaluate multi-class detectors [13, 14, 17, 36, 37]. The above databases all have the particular class ’Horse’. However, there is not much published about horse detection specifically. Nonetheless, using the YOLO framework, Máster et al. [38] is able to detect horses within an enclosed environment. The goal was to aid in the care of horses by automatically localizing them in the camera footage. The paper focuses on the bad quality of the videos (low resolution and lighting conditions) and the ability to train a CNN with such a dataset.

2.2 Subject Tracking Research

A simple approach on linking BBs from a frame to the next is by only using the information that comes with the BBs [20] (position, size and velocities). This method works for datasets that have only a few and non-occluding objects. In use cases where occlusions do occur, Soleimanitaleb et al. [39] and Gayki et al. [40] propose that there are four additional methods for subject tracking when using more information from the detection:

- Feature-based: Matching unique features of the subject from one frame to the next.
- Segmentation-based: Separating the background from the subject by assuming the subject is moving and the background is static.
- Estimation-based: Using state vectors to estimate the future location of the subject in the next frame.
- Learning-based: Using Machine Learning (ML) models to extract features and predictions of the subject.

Although idtracker.ai [2] is made for top-view video footage under laboratory conditions - and will likely not work under non-laboratory conditions, which is the case in this thesis - their method of handling occlusions is worth mentioning. idtracker.ai is a python based application that allows for tracking of larger amounts of subjects through a video, and also using CNN to able to identify subjects. They approach the issue of occlusions with an additional CNN to detect when crossings occur by training it to distinguish between touching and single individuals. They use frames before and after these occurrences to determine the trajectory of the subjects. With this information, idtracker.ai estimates the probability of which subject is which after the crossing. The downside to such an approach is that the NN need to be retrained when changing species or even when changing between laboratory setups.

Walter et al. [41] argues that analysing every frame (compared to only when occlusions occur) allow for a more flexible algorithm; matching subjects from frame to frame and maximizing probabilities of the trajectories. They tackle the problem of occlusions by removing them out of consideration until the involved subjects are again separate blobs in the view. Both approaches aims to decrease the amount of Identity Transfer (IDT) made when tracking subjects.

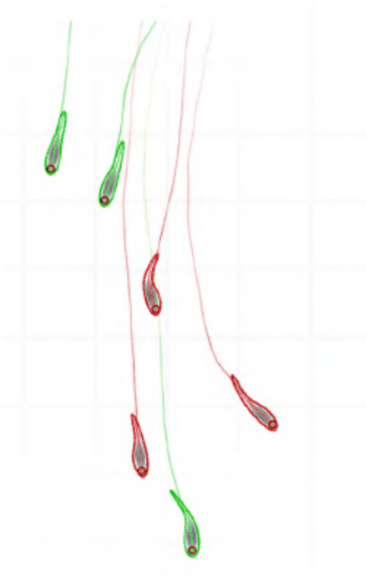


Figure 2.1: Example of trajectory tracking with groupies using TGrabs and TRex [41].

Human tracking in surveillance footage is often achieved with short-term feature extractors for associating the previous frame to the current frame [23, 31, 42–45]. Characteristic features need to be identified when trying to re-identify a subject. McLaughlin et al. [45] approached the tracking challenge with a combination of CNN, Recurrent-NN and Temporal Pooling. The input of the CNN is unique features of the subjects appearance and optical flow that represents the short-term motion of the subject from a single frame.

Another short-term method is using dictionary learning [4, 24]. Using the textures and colours of the subject, their algorithm computes histograms in vector form that can be compared to associate the subject. The advantage of this type of method is that there is less training needed compared to NN-based approaches.

2.3 Subject (re-)Identifying Research

McLaughlin et al. [45] continues its paper by introducing Recurrent-NN and Pooling layers to save information between frames, enabling re-identification of the subject after leaving the view and after occlusions. An adjustment on this method would be to extract feature maps from the subject, Gu et al. [46] notes that this technique can also be used for image-to-sequence mapping, allowing to re-identify subjects in a video from a single/multiple images. Zhu et al. [44] approaches the same problem with a combination of Spatial- and Temporal Attention Networks. Where the Spatial extracts and compares combined features from the detection and links detections to a sequence of a unique subject, the Temporal links sequences to another using the same features data.

This method can be improved by adding labels to the subject. This was done by Lin et al. [47], they proposed a system that would be able to identify the clothing of the subject, the sex of the subject [5] and the additional accessories the subject is wearing, like hats or bags. Such a long-term feature extractor can be used for horses as well (i.e. "The horse has white spots in the neck", "On the left front leg, the horse has a white sock"). This would require the system to be able to distinguish the body parts of the horse. Object skeleton extraction [6] uses edge detection algorithms linked to scale-associated side outputs to estimate the location of the skeleton of the subject. With the estimation of the skeleton, limbs can be segmented from the subject (see figure 2.2). This method can be used to train other methods (i.e. the dictionary) more efficiently, making profiles for segmented parts of the body instead of its entirety.

Some animal species have unique marks with which the animal can be visually recognized. Visual features which are not prone to deformation due to changing perspectives. Crall et al. [9] have made HotSpotter, which is an application made just for those cases. In their paper, they show promising results for giraffes, leopards, lionfish and zebra. Other researches have used this application on turtles [10], whale tales [12] and many other animals. Using the pattern on the skin of the animal as key points or hotspots, a query can be linked to that particular animal to be compared against when re-identifying (much like the histogram dictionary). Horses, however, are not a strictly patterned species with which they can be identified, hindering an identical method to be used on the entire horse species. A wider range of long-term features needs to be extracted from horses to be able to distinguish subtle differences in this type of subject.

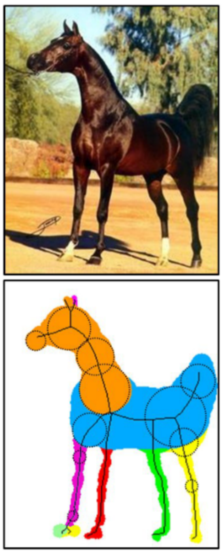


Figure 2.2: The top image is the input frame of the Deep-NN which outputs the bottom image, an estimation of the location of the skeleton and segmentation of limbs [6].

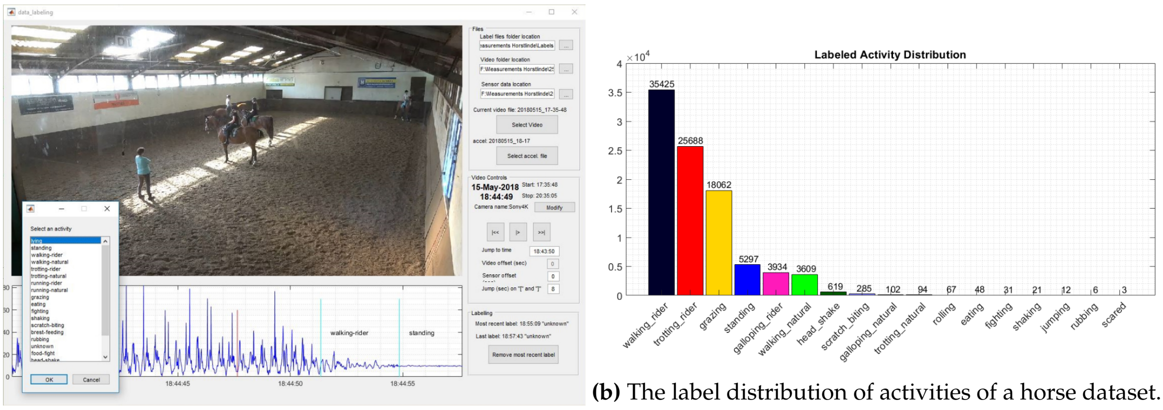
In this thesis, the footage is pre-recorded and can be processed as a whole, which allows for a post-processing application to utilize the non-existing restriction of time (compared to the live processing of incoming footage). This method would refer to ’offline’ tracking. Almost all the tracking and re-identification applications cited till now have been online applications, which is the result of the current direction of the field focusing on live implementations of these methods. Tang et al. [26] uses the definition of the offline problem as ’lifted multicut’. Where lifted refers to association being dependent on several comparisons across time, instead of the single future frame, and multicut refers to the possibility of a subjects trajectory being spread over several sequences. They approach the association problem by creating feasible sets (hypotheses per trajectory) which related papers do as well [31, 48]. The final association is done by combining information of the appearance (short- and long-term features), position and sequence timing. Peng et al. [42] defines how various problems in the matching of subject sequences and features can be handled with deleting, merging and interpolating trajectories, and how to extract useful re-identification frames using feature similarity calculations.

2.4 Use cases

The discussed use cases consist of related work that all could benefit from the ability to distinguish subjects based on visual appearance and/or track subjects over camera networks or on an individual camera view.

Video Labelling Applications

One field where ML is actively researched is animal activity recognition. Large amounts of data has been collected with an ever-expanding collection of monitoring devices waiting to be processed into various forms of extracted information. Typical methods for monitoring animals are achieved with unobtrusive devices strapped on or implanted in the animal. The device can take many forms, such as a camera mounted above a dogs’ head [49], a Global Position System (GPS) on the fins of marine wildlife [50], or a collar with sensors around the neck of cattle [51]. The goal of such devices are generally the same: collect data to uncover *what activity* the subject is doing, *when* the subject is doing it and (in cases where it is applicable) *where* the activity is performed.



Files
Label files folder location: easurements Horstlinde\Labels
Video folder location: F:\Measurements Horstlinde\2!
Sensor data location: F:\Measurements Horstlinde\2
Current video file: 20180515_17-35-48
accel: 20180515_18-17

Video Controls
15-May-2018 18:44:49
Start: 17:35:48
Stop: 20:35:05
Camera name:Sony4K
Jump to time: 18:43:50
Video offset (sec): 0
Sensor offset (sec): 0
Jump (sec) on "[" and "]": 8

Labelling
Most recent label: 18:55:09 "unknown"
Last label: 18:57:43 "unknown"

Select an activity
lying
standing
walking-rider
walking-natural
trotting-rider
trotting-natural
running-rider
running-natural
grazing
eating
fighting
shaking
scratch-biting
brest-feeding
rubbing
unknown
food-fight
head-shake

(a) Screenshot of AiSensus GUI.

(b) The label distribution of activities of a horse dataset.

Figure 2.3: Illustrating the current form of the application and the significance of the non-uniform distribution of activities [8].

Complex ML models should theoretically have the capability to classify what activity is linked to any form of data. This would imply that data can be automatically processed without the need for human interaction. However, ML models need to be trained with data. In the case of supervised learning ML models, the training is done using Ground-Truth (GT) datasets. The GT data-set is a collection of data samples that have been annotated with (to be assumed) true labels and is often created by a human actor. Utilizing the GT datasets, the model adjusts its parameter to optimize for the correct classification of activities.

AiSensus This thesis is an assignment from AiSensus [52], a labeling application that creates the GT data-set for the mentioned training of ML models, see Figure 2.3a. The current application allows the user to synchronize sensor data (i.e. accelerometer, gyroscopes, magnetometer, temperature) to reference data (i.e. video-, sound-recordings). The supervisor (user of the application) goes through the reference data and labels the activity of the subject, simultaneously labelling the synchronized sensor data. The most basic form of labelling data is to process the reference data by labelling all the activities. This is a labour-intensive task and will most likely lead to an imbalance of training data due to a non-uniform distribution of activities made by the subject (i.e. when walking occurs more than running). As a result, the amount of labelled data for one activity could be insufficient to make an accurate classification whilst other activities could have an abundance of labelled data, see Figure 2.3b.

AiSensus is looking to improve on this basic form of labelling by implementing Active Learning (also called query learning) [53]. The extension will ask the supervisor to label specific sensor data that have uncertainties in the dataset, which could be boundary points between or outliers when classifying. Computer vision is considered to analyse the video footage to provide the supervisor with relevant video reference data of the subject with as input the query of the Active Learning algorithm. This would allow the application to show the supervisor short fragments of footage that need to be labelled for activities. The goal of the addition of Active Learning is to make the labelling task is more efficient where the amount of labels for a comparable accuracy with the basic form is decreased.

Wildlife camera traps

Placing a camera in wildlife to better understand a ecosystems - with the goal to better manage and protect them - is a common practice. Again, information extraction with human actors with these vast amounts of data is to be replaced with automated computer systems [11]. High accuracy and significantly faster extraction have been achieved with Deep-NN. Although the goal of such a system is less of the identification of individual animals and more of classifying species and activities, this is to be considered a use case.

Human tracking and (re-)identification

In the field of security, the ability to identify and follow a subject through a network of video streams has been a well-researched topic. With an ever-growing network of cameras in both Federal agencies and private firms, the need to replace the human operator who constantly monitors the streams has grown with it [3].

Focusing on the appearance of a human through a camera tends to result in the focus of bigger surfaces of the human, like hair, clothes, bags. Using CNN and Recurrent-NN [45] consisting of many steps that include processing multiple layers (containing different information between frames) with convolution, pooling, and non-linear activation functions, the system is able to re-identify humans across time-steps. Likewise, invariant dictionaries [4] can be extracted at different orientations of the human, focusing on recognizable features and their vectors at certain viewpoints. Both methods allow for the training of a model that can be used across multiple camera views and is reusable as long as the appearance of the subject does not change drastically. In this thesis, the equivalent would be a change of gear or rider. However, these surfaces are less significant than humans, where most of the horse is not covered.

Chapter 3

Dataset

In this chapter, both the evaluation and training datasets will be explained. The evaluation dataset is extracted from a dataset provided by the faculty. The created evaluation dataset consists of frames and fragments, explained in sections 3.2 and 3.3 respectively. The training dataset consists of images taken from the internet and is explained in section 3.4.

The provided dataset this thesis uses contains 39 hours of horse recordings [54, 55]. They have been categorized into 3 groups: Outside Arena, Inside Arena and Field with subcategories representing different view points and video quality. The distribution of these categorizes can be found in Figure 3.1, previews of the camera views can be found in Figure 3.2, and in Table 3.1 more information can be found.

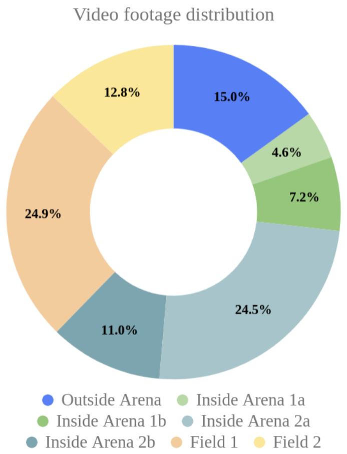
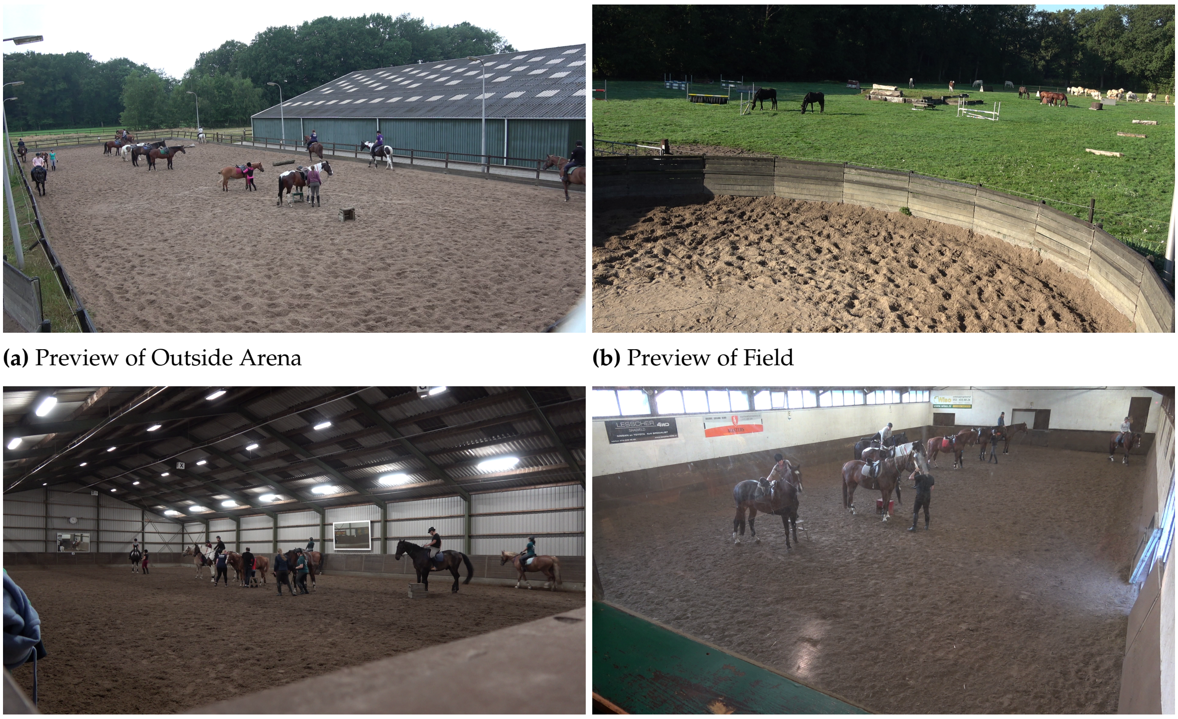


Figure 3.1: Distribution of video footage of provided dataset.

Table 3.1: Information about the provided dataset of video footage of the horses.

Category	Duration (H:MM:SS)	FPS	Resolution (Pixels)	Notes
Outside Arena	5:51:24	25	1920x1080	50 minutes of the video are obstructed with a plastic bag and raindrops. (Figure 3.3e) 16 minutes are empty. 60 minutes has solar flare. (Figure 3.3a)
Inside Arena 1a	1:48:37	25	1920x1080	There are mirrored windows that show a reflection of the horses.
Inside Arena 1b	2:47:38	48	1280x960	8 minutes are empty. 30 minutes have only a single horse. Bright sunlight spots that change the appearance of the horses. (Figure 3.3d) There are mirrored windows that show a reflection of the horses. (Figure 3.3c)
Inside Arena 2a	9:34:22	25	1920x1080	The window is not homogeneous due to reflections and stains. (Figure 3.3b)
Inside Arena 2b	4:16:30	25	1280x720	The window is not homogeneous due to reflections and stains. (Figure 3.3b) 157 minutes are empty.
Field 1	9:41:56	25	1920x1080	Horses are mostly in the shade and far away from the camera. There are cows in the background. (Figure 3.3f)
Field 2	5:00:30	25	1280x720	Footage is interrupted by manually zooming and movement of the camera.



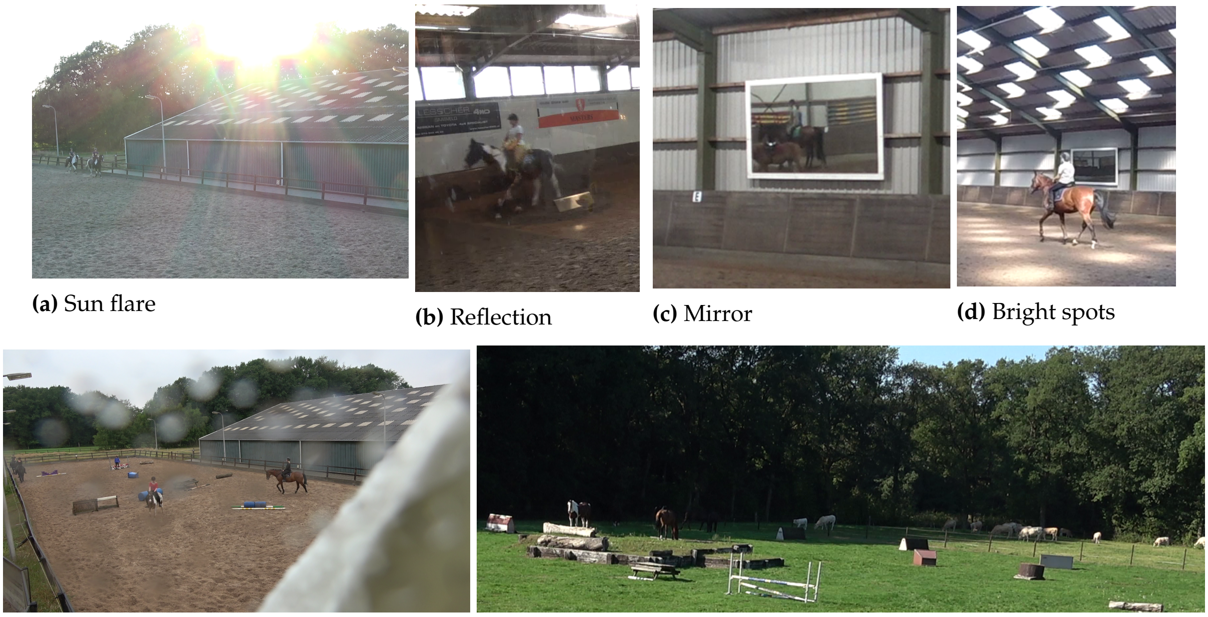
(a) Preview of Outside Arena

(b) Preview of Field

(c) Preview of Inside Arena 1

(d) Preview of Inside Arena 2

Figure 3.2: Snapshots taken from video footage of the provided dataset.



(a) Sun flare

(b) Reflection

(c) Mirror

(d) Bright spots

(e) Obstructed

(f) Far away and cows

Figure 3.3: Snapshots taken from video footage of the provided dataset showing difficult situations.

3.1 Input sizes

For NNs the training dataset must be a representation of what the input of the network will be. This also includes the size of the detections that will have to be made. The size of a detection can be expressed in the area coverage of the BB in respective to the image size, as this metric does not change when rescaling the image. It should be noted that some backbones use aggregation [56] of the input images (training and/or testing) to aid in reducing the size dependence by automatically varying the input sizes [19, 30, 36]. Nevertheless, trying to match the training and testing detection sizes will increase the accuracy of the detectors (this will be explained in practice in chapter 4.1 on page 18).

To simplify the evaluation of the sizes of the testing versus training size the metric of the percentage area coverage was defined. Objects are classified in 4 categories based on their percentage area coverage; where >5% are for big objects, >0.5% for medium, >0.1% for small and <0.1% for tiny objects (see Figure 3.4 to see the sizes in a real image).

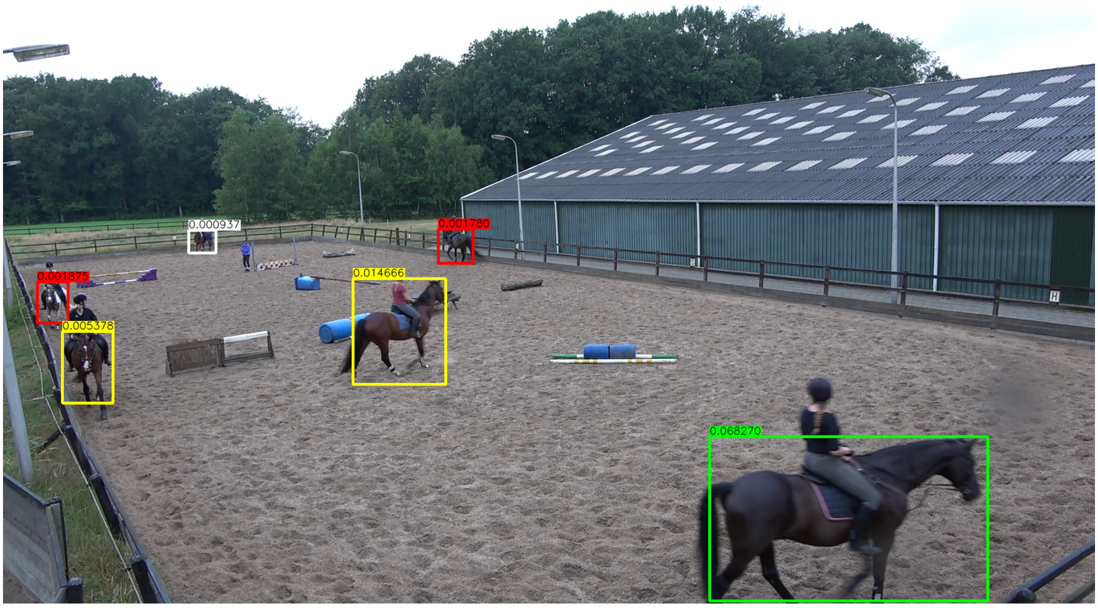


Figure 3.4: Illustration of the size definition in a frame from Outside Arena. Green represents a big object, yellow medium, red small and white tiny. The value on the left corner of the BB represents its coverage.

This metric will be used throughout this chapter to compare the training dataset and the evaluation dataset on their BB sizes, as the sizes of the training dataset should resemble the sizes of the evaluation dataset.

3.2 Evaluation frames

To be able to evaluate the object detectors on is use case, evaluations frames were taken from the provided videos. The frames were not used for training and only as test dataset. This evaluation dataset consisted of 261 frames and 1229 unique annotations from every dataset and were picked at random from the multiple videos within the same category.

Table 3.2: Distribution of evaluation frames from the evaluation dataset. Size is a metric of percentage area coverage of the BB.

Category	Frames	Anno- tations	Avg size [% coverage]	Big [>5.0%]	Medium [>0.5%]	Small [>0.1%]	Tiny [<0.1%]
Outside Arena	60	434	0.746	8	137	248	41
Inside Arena 1a	25	135	1.695	9	67	55	4
Inside Arena 1b	51	192	0.512	6	20	109	57
Inside Arena 2a	28	144	1.612	8	109	27	0
Inside Arena 2b	11	26	1.463	2	22	2	0
Field 1	39	122	0.529	1	33	52	36
Field 2	48	176	1.401	12	45	58	61
Total (Partition)	**261**	**1229**	**1.002**	**46 (3.7%)**	**433 (35.2%)**	**551 (45.3%)**	**199 (16.2%)**

In Table 3.2 the distribution of annotation from the video are shown, with their distribution of BB sizes. From this table, it can be concluded that the categories: Outside Arena, Inside Arena 1b and Field 1, have the smallest objects to detect. In addition, Inside Arena 1b has a resolution of 1280x960 resulting in even fewer pixels for the detector to work with.

3.3 Evaluation fragments

In order to evaluate the tracking and re-ID applications for this use case, evaluation fragments were cut from the original videos. In total, 4 fragments of 30 seconds were extracted from each category that was defined in Figure 3.2. These fragments were chosen as evaluation fragments for their complexity and their ability to test a certain aspect of the model and were annotated ensuring that all appearing horses hold the same ID (even after leaving the video for an extended period of time).

Table 3.3: Distribution of evaluation clips from the evaluation dataset. Size is a metric of percentage area coverage of the BB.

Frag- ments	Category	Frames	Anno- tations	Avg size [% coverage]	Big [>5.0%]	Medium [>0.5%]	Small [>0.1%]	Tiny [<0.1%]
1	Outside Arena	750	5675	0.797	82	2325	2838	430
2	Field 1	750	4575	0.446	0	1426	2987	162
3	Inside Arena 1	750	4236	1.964	385	1835	2014	2
4	Inside Arena 2	750	6000	1.658	66	4141	1793	0
	Total (Partition)	**3000**	**20486**	**1.212**	**533 (2.6%)**	**9727 (47.5%)**	**9632 (47.0%)**	**594 (2.9%)**

Table 3.4: Additional information about the 4 extracted fragments, regarding the number of unique horses that appear in the video (excluding cross video appearances), the fragments difficult aspects and references to images where these fragments (or an image of the same camera view) is shown.

Fragment	Horses	Description	Figures
1	8	High camera placement Almost full coverage of entire walkable space Multiple partial occlusions	3.2a on page 9 3.4 on page 10 A.5 on page 44
2	7	Multiple horses of the same breed A horse rolling on the ground 2 difficult occlusions A long duration partial occlusion 2 partially visible horses	3.2b on page 9 A.3 on page 42 A.4 on page 43
3	8	Low camera placement Low coverage of the walkable space High amount of partial and full occlusions Low lighting levels 2 mirrors showing reflections	3.2c on page 9 3.3c on page 9 A.1 on page 40
4	8	Almost full coverage of entire walkable space Multiple partial and full occlusions Sun flare and spots Dirty window blurring the cameras' view	2.3a on page 6 3.2d on page 9 3.3b on page 9 A.2 on page 41

In Table 3.3 the details of the fragments are given, indicating from which category they were extracted, the number of annotations it holds and the sizes of the BBs. In these clips there are 31 horses, not considering the reappearance of a horse in another clip. A textual description of the fragments is given in Table 3.4 future explaining the contents of the fragments and indicating difficult their aspects.

3.4 Training Dataset

A training dataset was created by collecting horse images from the internet. This dataset was used to train the detection models for horses and were downloaded from two different sources: OpenImages and ImageNet.

OpenImages

OpenImages [33] is a database of other 9 million annotated images with 600 unique object classes. Users are able to freely download images with their associated BB, segmentation mask and visual relations (i.e. "person is walking") annotations. They differentiate the images with the following tags:

- The annotation is for a group of objects (one annotation can hold multiple objects)
- The object is occluded by another object in the image
- The object is truncated in the image and extends beyond the boundaries of the image
- The image is a depiction of the object (i.e. a drawing or illustration)
- The image is taken from inside of the object

Using OIDv4 ToolKit [57], 1507 images of the class 'Horse' were downloaded from OpenImages that were not: a group or a depiction. 16 examples of this image set can be seen in Figure 3.5.

Terms of use The images from OpenImages are licensed by Google LLC under Creative Commons (CC) BY 4.0 license, and the annotations are under the CC BY 2.0. With these licenses, users are allowed to freely share and adapt the material with the condition of giving the appropriate credit.

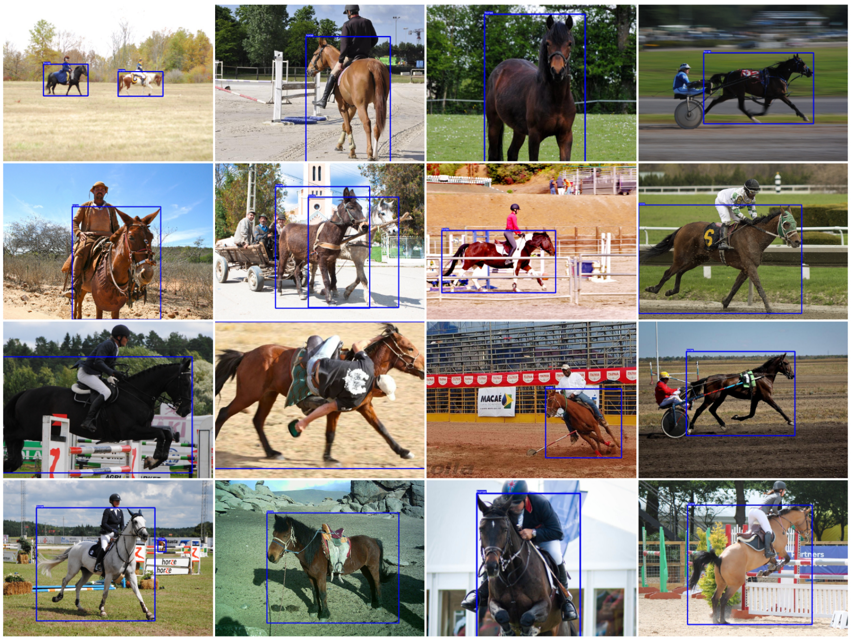


Figure 3.5: Example of images taken from the OpenImages dataset with associated bounding boxes in blue. Images are cropped to fit the aspect ratio in the grid.

ImageNet

ImageNet [32] is a accurate collection of web images organized according to the WordNet hierarchy. Each concept in WordNet has associated images liked to it by ImageNet. Unlike OpenImages, ImageNet does not provide annotations within the images and annotates the entire images.

The majority of the downloaded images from OpenImages were close-ups of horses and often only displaying a single horse (without occlusions), which is the opposite of the evaluation videos. Therefore, using a downloader, images from the following class concepts were obtained:

- Cross-country riding, 507 images
- Horse racing, 506 images
- Race horse, 505 images
- Riding, 499 images
- Trotting horse, 501 images.

16 examples of this image set can be seen in Figure 3.6.

These images were manually annotated using AlexeyAB's annotator 'Yolo mark' [58].

Terms of use ImageNet is free to use only for non-commercial research and educational purposes, as stated in their terms of access.

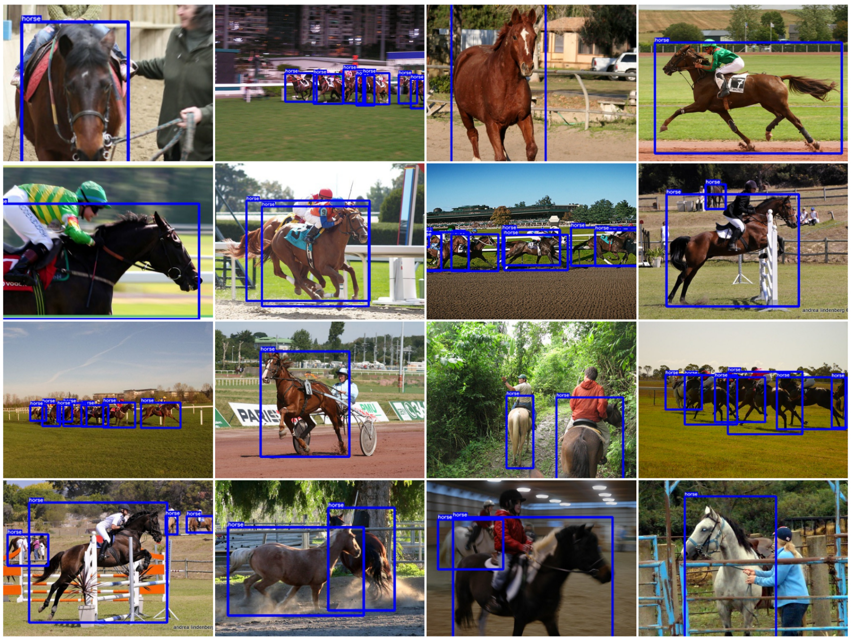


Figure 3.6: Example of images taken from the ImageNet dataset with associated bounding boxes in blue. Images are cropped to fit the aspect ratio in the grid.

Training dataset augmentations

An analysis was made on the BB sizes from the training dataset and can be seen in Table 3.5. These sizes do not resemble the sizes from the evaluation dataset (see Table 3.2 on page 11 and 3.3 on page 11). In the following sections, the augmentations that have been performed on the training dataset to match the input sizes of the evaluation dataset are explained.

Table 3.5: Distribution of evaluation frames from the evaluation dataset. Size is a metric of percentage area coverage of the BB.

Category	Images	Anno- tations	Avg size [% coverage]	Big [>5.0%]	Medium [>0.5%]	Small [>0.1%]	Tiny [<0.1%]
OpenImages	1507	2599	17.27	1734	736	114	15
ImageNet	2509	4880	22.83	3508	1139	210	23
Total (Partition)	**4016**	**7479**	**20.9**	**5242 (70.1%)**	**1875 (25.1%)**	**324 (4.3%)**	**38 (0.5%)**

RGB Augmentations

To increase the effectiveness of a training dataset, colour augmentation is often used. With slight RGB adjustments to the images, synthetic data is created that help reduce overfitting when training the NN [15]. Examples of such methods are changing adjusting the contrast, brightness and hue values of the image, thereby eliminating the significance of colours from in the training dataset. These augmentations are often automatically performed by the backbones of the NNs.

Square input images

It is often the case with object detection that the input image of the network is made square. In the case of YOLOv4 (an application that will be used in this thesis) [17], the aspect ratio is kept and the image is scaled down to the input size of the network. For example, a 1920x1080 image is given as input for a 416x416 network. The input image is re-scaled to 416x234. This re-scale has a ratio of 4.6 to 1 pixels. This decreases the number of pixels in the detection. This influences the training dataset as well as the evaluation dataset. The new sizes of both datasets are computed can be seen in Table 3.7 on the following page.

Mosaic

A commonly used image augmentation is making mosaics of images. This is a combination of down-scaling, cropping, repositioning and increasing the number of detections in a single image. To counteract the larger detection sizes in the training set, the mosaic method is used, making 3x3, 4x4 and 5x5 mosaics with a random combination of images from the training images. To increase the size of the training set, the images were horizontally flipped and again randomly combined into the mosaics. The new sizes can be seen in the Table 3.6, which describes the type of mosaic's that have been created and their respective detection sizes.

Table 3.6: Distribution of training images when augmenting the dataset by making mosaics

Type	Frames	Anno- tations	Avg size [% coverage]	Big [>5.0%]	Medium [>0.5%]	Small [>0.1%]	Tiny [<0.1%]
3x3	892	14946	1.659	852	8882	3303	1909
4x4	502	14958	0.933	26	7992	4036	2904
5x5	320	14909	0.597	0	6413	4641	3855
Combined (Partition)	**1714**	**44813**	**1.063**	**878 (2.0%)**	**23297 (52.0%)**	**11980 (26.7%)**	**8668 (19.3%)**

3.5 Datasets size comparison

The final sizes that will be used in the training and evaluation of the thesis can be seen in Table 3.7. The training mosaic dataset is the combination row in Table 3.6. The combination is chosen as removing the 3x3′ed mosaics from the training would decrease the number of training images by too much. These sizes assume a square NN input where the image keeps its aspect ratio.

Table 3.7: Adjusted distribution of the evaluation and training images considering square input images.

Dataset	**Images**	**Anno-tations**	**Avg size [% coverage]**	**Big [>5.0%]**	**Medium [>0.5%]**	**Small [>0.1%]**	**Tiny [<0.1%]**
Evaluation Frames	261	1229	1.002	21 (1.7%)	281 (22.9%)	539 (44.0%)	388 (31.6%)
Evaluation Fragments	3000	20486	0.682	278 (1.4%)	6798 (33.2%)	11901 (58.1%)	1509 (7.4%)
Training Normal	4016	7479	14.93	4736 (63.3%)	2201 (29.4%)	460 (6.2%)	82 (1.1%)
Training Mosaic	1714	44813	1.063	878 (2.0%)	23297 (52.0%)	11980 (26.7%)	8668 (19.3%)

The augmentations made to the training dataset decreased the average coverage of the BBs from 14.93% to 1.063% which better resembles the 1.002% and 0.682% from the evaluation frames and fragments respectively. The effect of this augmentation will be further explained in section 4.1 on page 18.

Chapter 4

Horse Detection

The first step in the pipeline is to recognize where the horses are in the video. This should return a BB around the objects that have been classified as a horse. The object detection of horses will be achieved using and retraining a pre-trained NN. Figure 4.1 shows the flow of what the detector should do. The input for this part is a raw video. Using a trained NN the detector locates the position of the horses in the video and outputs these detections per frame. The pre-trained NN is to be used as a benchmark to evaluate improvements done to the NN by additional training. Using transfer learning the classification layer of SOTA will be re-trained to only classify for horses. The additional training would improve the accuracy of detecting horses. The NN will be trained on open-source images from the internet (as explained in section 3.4 on page 13) and will be evaluated on frames from the real dataset provided by the research group. The final evaluation of the object detection will compare the tested applications with each other and their Out-of-the-Box (OotB) version of said application. The training setup used in thesis thesis is explained in the appendix in the paragraph "Training setup" on page 40.

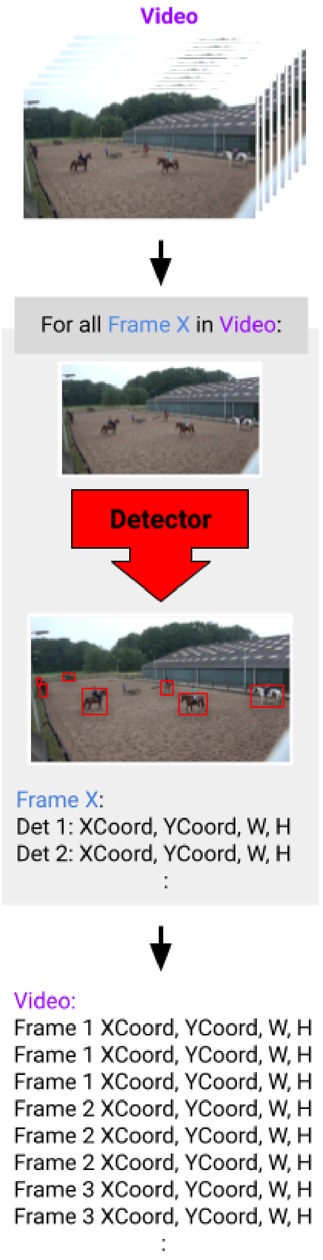

Video
For all Frame X in Video:
Detector
Frame X:
Det 1: XCoord, YCoord, W, H
Det 2: XCoord, YCoord, W, H
:
Video:
Frame 1 XCoord, YCoord, W, H
Frame 1 XCoord, YCoord, W, H
Frame 1 XCoord, YCoord, W, H
Frame 2 XCoord, YCoord, W, H
Frame 2 XCoord, YCoord, W, H
Frame 2 XCoord, YCoord, W, H
Frame 3 XCoord, YCoord, W, H
Frame 3 XCoord, YCoord, W, H
:

Figure 4.1: Process of the detecting task of the pipeline. The detector uses an NN to locate the subjects in the video.

Challenges and Requirements The challenges in this process are:

- *Resolution*: The resolution of video recordings can vary where objects are limited to a certain amount of pixels.
- *Far away objects*: Objects at a bigger distance of the camera have fewer pixels.
- *Illumination conditions*: Varying light conditions, where brightness, contrast, and colours can influence the appearance of the object.
- *Shadows and reflections*: Objects might cast shadows and have reflections that can be incorrectly classified as an object.
- *Varying perspectives and deformation of the object*: An object can be viewed from different angles where it can be deformed requiring the object detector to be trained for the various perspectives.
- *Obstruction*: The object can behind another object, where only a portion of the object is visible.
- *Cluttered or textured backgrounds*: Objects can blend into the background due to it having similar colours to the background.
- *Varying subject-class appearance*: Objects can come in various shapes, sizes, and colours.

The detector should be able to fulfil the following requirements:

- The amount of False Negatives (FNs) (number of missed detections) should be minimized. False Positives (FPs) (false detections) should be low as well, however, these detections can be filtered out in a later step, whilst FNs can not be added in a later step. The metric that will be used for this is the recall score.
- The speed of the detections can not be too slow as the input of the pipeline is a video and not individual photos.
- The input of the network should be big enough to handle high definition frames with detections that are in the background. As explained in 3.4 on page 13, if the input size is smaller than the original image, the image will get cropped to fit the input size.
- The implementation of the detection support in tracking applications. The goal of this thesis is not to create an entire application from scratch, rather look for existing applications and tweak them to work for this use case.

4.1 Detection Applications

YOLOv4

You Only Look Once (YOLO) [15] uses the Darknet framework to achieve its object detection. The goal of YOLO is to achieve faster image recognition than their current competitors and is actively being developed. They gain faster detections using multiple layers of CNN and dense with sparse prediction [17] to require only a single scan of the image.

Two pre-trained - but re-trainable - weights for the network are provided by the authors (note that the 2 weights can be differentiated by the tag 'Tiny'):

- YOLOv4 convolutional-layer 137, which is the original NN without the last YOLO-layer, the classification layer. It consists of 64 million parameters (66.3k not trainable) and has a size of 256MB (after re-adding a custom classification layer).
- YOLOv4 Tiny convolutional-layer 29, is a more compact version of the original YOLOv4. It consists of 5.9 million parameters (6.2k not trainable) and has a size of 23.5MB (after re-adding a custom classification layer).

The input layer of these models consists of a resize image to fit the input size of the model (if the model has an input size of 416x416, the image is resized to 416x416 keeping its aspect-ratios and adding black borders). The input size can be altered to any square input, where higher resolutions take longer to process.

With these pre-trained models, 10 unique YOLOv4 models (including Tiny) were trained with different input sizes and training datasets, and are given nametags that indicate their properties. These properties refer to if they are a Tiny, their input size, which training set they were trained with if the mosaic augmentation was used and for how many iterations it was trained (see Table 4.1). The standard parameters for YOLO are an input size of 416, with colour augmentation (saturation, exposure and hue) with 128 images for 6000 iterations. The backbone Darknet provides automatic save points of the model every 1000 iterations and when at the 100th iteration the mean Average Precision (mAP) score has been improved. Only these models will be evaluated.

The Tiny variant of YOLOv4 was also tested due to its significant faster speeds in training and detecting. The downside of was the lower accuracy in the detections.

Table 4.1: All custom trained YOLOv4 models. The Training dataset letters O and I represent OpenImages and ImageNet.

Tag	Tiny	Input Size	Training Dataset	Mosaic	Training Iterations
Y		416	O		4000
YT	x	416	O		5000
Y_v2		416	O+I		6000
YT_v2	x	416	O+I		6000
Y960		960	O+I		4000
YT960	x	960	O+I		6000
YT960_M	x	960	O+I	x	6000
Y640		640	O+I		6000
YT640	x	640	O+I		6000
Y640_M		640	O+I	x	6000

Table 4.1 is ordered according to which the training process was performed first. During this process, the observation that the detector could not handle far away detections ensued the testing of methods to solve the issue. A learning process occurred whilst training the different models, which resulted in the collection of additional training images, an increase of the input size, and the mosaiced training set (discussed in chapter 3 on page 8).

FairMOT

CenterNet [36] uses a different method for creating BB than, for example, YOLO. CenterNet can be trained through TensorFlow. Instead of using the commonly used sliding window detection method, CenterNet extracts a heatmap obtained from the overlapping BB, marking the centre of the object.

Build on the CenterNet detection method with heatmaps, FairMOT [30] detects objects and then tracks them with a NN trained with feature extraction. This is done in two different branches, similar to DeepSORT. The main difference its inference speed due to the anchor-free (by heatmap) method of detections.

The authors have provided models for feature extraction and object detection, which can be trained with still images. However, there only is support for training their DLA34 [56] model with a size of 259.4MB

A FairMOT model (DLA34) was trained with the entire (OpenImages, ImageNET, Mosaic'ed 3-5) due to it utilizing the pyramid input scaling. This model has a set input size of 1088x608 and resizes the image to fit this condition (deforming the image). The training method was to iterate through every training image for 30 epochs, giving save points every 5 epochs and the 5 last epochs.

Other Applications

The mentioned applications under this and future 'Other Applications' sections list applications that were considered, but were not successfully implemented and evaluated.

EfficientNet An acknowledged neural architecture that is often used as a comparison by other detection applications is EfficientNet [37] and can be trained through TensorFlow. They focus on improving performance by optimizing the balance of the networks depth, width and resolution.

EfficientNet provides 8 sizes of pre-trained models ranging from EfficientNet-B0 with 5.3 million parameters and a size of 15MB, to the slower EfficientNet-B7 with 66.2 million parameters and a size of 244MB. They released a new model in 2020 called, EfficientNet-L2 [59] which currently holds numerous SOTA benchmarks and has a size of 480 million parameters.

EfficientNet models can be used as an additional check for if a BB does or does not contain a horse. This filter FPs or noisy detection with the first detector makes. The output of EfficientNet is not a BB as anchor creation is not part of the application, however, only classifies an image.

The smallest model (B0, input size of 224) has been trained and was able to get 100% accuracy on the test dataset (combination of images with and without horses). This evaluation is not complete and not continued as the usage of this model was not yet required, nevertheless showed promise.

RetinaNet With the backbone of ResNet, RetinaNet is a BB detector that should be able to compete with YoloV4. Due to time restriction, the application could not be fully trained and tested.

4.2 Detection Evaluation

To evaluate the horse detections, the following metrics were obtained from evaluating on the evaluation frames:

- Recall, $\frac{TP}{TP+FN}$, the percentage of detected objects compared to the GT. An indication of how many horses in the image have been detected. This metric does not include FP.
- Intersection over Union (IoU), $\frac{A_{overlap}}{A_{union}}$, the overlap between the detection BB and the GT BB. An indication of the ability to correctly placing the BB around an object.
- mAP at 50% IoU, a score calculated by taking the mean average precision (a measure of how many detections were correct) overall classes (in this thesis' case, only horses). Calculated by finding the area under the precision-recall curve. This score is a way to summarize the evaluation and is commonly used throughout detection evaluations.

In Table 4.2 shows the evaluation of every trained model with the best recall scored and highest mAP scored savepoints will be given (if only a single was given per model, it implies that that savepoint was had the best recall and mAP). FairMOT did not allow for the extraction of the IoU:

Table 4.2: All custom trained detection models evaluations. Limiting Table to the best recall-, mAP-scored and general savepoints. The values in the Table are given in percentages. The mAP(0.5) referece to a minimum of 50% IoU needs to be covered to be considered an detection. The highlighted models were used in future evaluations.

Detector	Precision	Recall	average IoU	mAP(0.5)
Y (Best)	87	71	64.6	79.3
Y (2k)	89	67	67.3	79.0
YT (Best)	82	58	59.27	63.1
Y_v2 (5k)	91	71	69.54	79.3
YT_v2 (Best)	82	58	59.27	63.1
YT_v2 (5k)	80	58	57.89	62.6
Y960 (Best)	87	71	64.6	79.3
YT960 (Best)	90	74	66.75	81.5
YT960_M (2k)	79	83	59.4	85.0
YT960_M (5k)	76	83	58.01	83.4
Y640_M (Best)	**90**	**87**	**68.85**	**88.9**
Y640 (Best)	89	80	64.68	83.4
YT640 (2k)	81	78	60.88	82.8
YT640 (4k)	78	80	58.52	81.9
FairMOT (5)	3.58	93.5	-	90.4
FairMOT (10)	3.52	91.8	-	87.7
FairMOT (15)	3.56	93.0	-	90.2
FairMOT (20)	3.48	91.2	-	87.3
FairMOT (25)	**3.56**	**92.9**	**-**	**88.8**
FairMOT (26)	3.50	91.5	-	87.1
FairMOT (27)	3.50	91.2	-	86.9
FairMOT (28)	3.51	91.2	-	86.8
FairMOT (29)	3.50	91.3	-	86.7
FairMOT (30)	3.48	90.7	-	86.3

4.3 Detection Discussion

During the training of the YoloV4 model, the aforementioned learning process showed an improvement on the recall, average IoU and mAP when adding the additional training images. Then again when increasing the input size. And one more when training on the mosaiced dataset. The mosaiced dataset outperformed every other trained YoloV4 model, from which can be concluded that the training dataset better resembled the evaluation dataset.

From the evaluation (table 4.2) it can be concluded that the YoloV4 detector and the FairMOT have approximately the same output. The precision of FairMOT (at around 3.5%) seems to be an error in the evaluation script as the number of detections do not show an indication of a high amount of FPs nor can they be seen when using the detector in a video. The recall of FairMOT out performs YoloV4, indicating that FairMOT is able to detect 6% more horses than the best-trained YoloV4 model.

The model Y640_M (Best) was chosen to be used in the next evaluations as they show the best results from this evaluation. FairMOT (25) was chosen with an additional manual evaluation of detections in the video. Before the 25th model, the IoU was not as stable and covering. After the 25th, the model began to be overfitted with the training set.

In Figure 3.3 on page 9 several difficult scenarios have been identified. Whilst not all of these scenarios have caused issues in the evaluation, the sun flare, reflection and mirror have cause FPs and FN (see Figure A.2 on page 41 and A.1 on page 40 respectively).

Chapter 5

Horse Tracking

The tracking of horses will be achieved with a tracker that associates detections from frame to frame into sequences. The flow of the tracker is illustrated in Figure 5.1. With the detections of every frame, a tracking algorithm is used to link BBs from one frame to the next. The input of the tracker are the outputs of the detector, therefore the performance of the tracker is subject to the performance of the detector. The tracker is able to use the information from the video as well as the BBs by referencing the video. This information could be the appearance of the subject and the direction (velocity) from frame to frame.

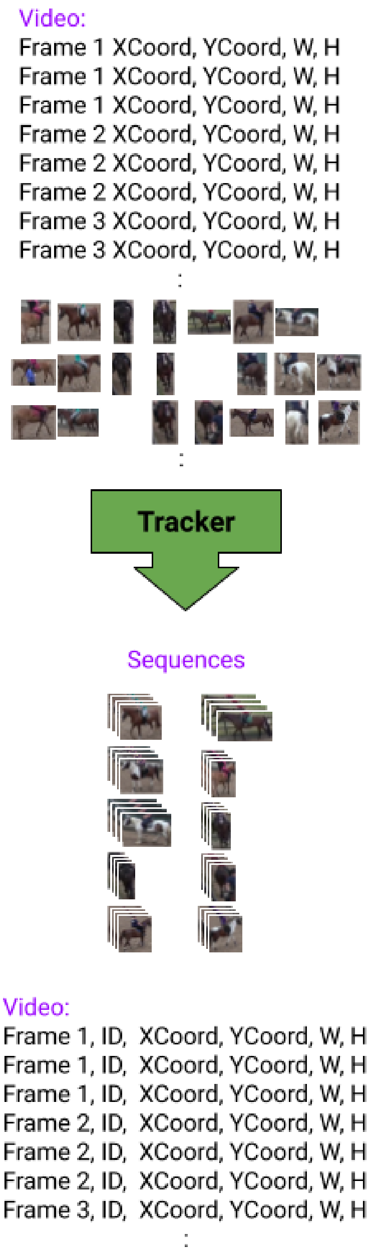

Video:
Frame 1 XCoord, YCoord, W, H
Frame 1 XCoord, YCoord, W, H
Frame 1 XCoord, YCoord, W, H
Frame 2 XCoord, YCoord, W, H
Frame 2 XCoord, YCoord, W, H
Frame 2 XCoord, YCoord, W, H
Frame 3 XCoord, YCoord, W, H
Frame 3 XCoord, YCoord, W, H
:
:
Tracker
Sequences
Video:
Frame 1, ID, XCoord, YCoord, W, H
Frame 1, ID, XCoord, YCoord, W, H
Frame 1, ID, XCoord, YCoord, W, H
Frame 2, ID, XCoord, YCoord, W, H
Frame 2, ID, XCoord, YCoord, W, H
Frame 2, ID, XCoord, YCoord, W, H
Frame 3, ID, XCoord, YCoord, W, H
:

Figure 5.1: Process of the tracking task of the pipeline. The tracker uses an algorithms to find the best associations for a subject to an existing track.

The different applications will be evaluated on their ability to handle the 4 datasets and specifically their ability to correctly handle short occlusions. Within the tracking application, parameters will be tweaked and this effect will be evaluated. The final evaluation of the subject tracking will compare the tested applications with each other and their OotB version of said application.

Challenges and Requirements The challenges in this process are:

- *Occlusion*: Subjects overlap in the camera view. The implementation should recognize that the overlap is not a new object, nor that after the overlap the objects are new subjects (identity switches).
- *Appearing and Disappearing*: Subjects can leave the camera view.
- *Missing frames*: Using frame to frame comparison may have issues when the time steps are not constant. This can occur with data loss of the video file.

The tracker should be able to fulfil the following requirements:

- An ID should be given to a subject (BB) and that ID should stay the same throughout the sequence of frames.
  - Situations where a single object has multiple IDs in a sequence are allowed
  - Situations where a single ID has multiple subjects in a sequence are not allowed
- When occlusions occur, the tracker should not switch the IDs of the concerning subjects. Identity Switch (IDSW) would be detrimental for the output of the pipeline, therefore a new ID would be better than having subjects take over each over IDs.
- Faulty detections where a single subject has more than one BB at the same time should be considered as a single ID.

Figure 5.2 illustrates these metrics. In this example, the detector has missed an detection due to an occlusion as see in the most left image.

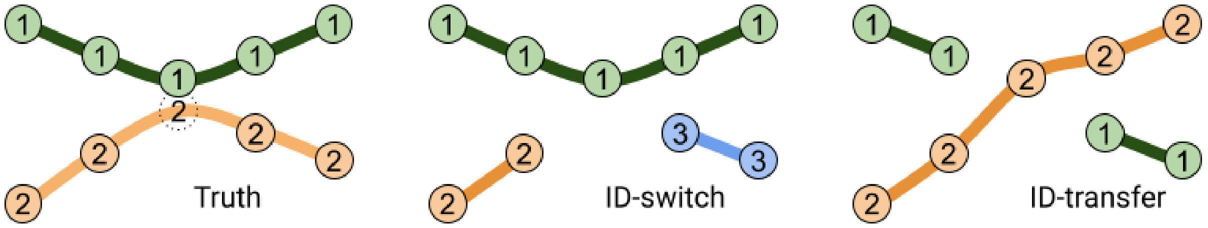


Figure 5.2: Illustration showing the difference between IDSWs and IDTs.

5.1 Tracking Applications

DeepSORT

Simple Online and Realtime Tracking (SORT) [60] looked to improve the performance of tracking subjects through longer periods of occlusions. The application uses a pre-trained CNN to detect the objects, also allowing support of TensorFlow and YOLO. Then, using recursive Kalman filtering and frame-by-frame association using the Hungarian method, it is able to track the subject through the frames. In another paper of the same authors [22], they propose an extension to improve the method by adding subject re-identification using a soft-max classification regime and cosine metric learning, re-calling the application to DeepSORT.

This application is able to re-identify and track with the Deep and SORT respectively. For tracking, only the SORT will be discussed. The matching of objects to frames is done with a Nearest Neighbor Distance cost matrix. The SORT part has the following customize parameters:

- Distance Metric **(T)**: Either cosine or euclidean [60]
- Maximum matching distance **(C)**: Threshold where larger distances are considered invalid matches. Lowering the threshold will limit the distance at which the match is allowed to be made.
- Maximum IoU overlap **(O)**: Threshold where larger overlaps should be considered as a single object. Lowering the threshold will allow for less overlapping IoUs. By default this value is set to 100%, assuming that the detector does not make mistakes. It should be considered that occluded objects have overlapping IoU as well, therefore, a too low threshold will affect how occlusions are handled.
- Position prediction **(P)**: The option to use velocity to predict the future position which aids in the cost determination of the subject to a track, can be disabled, where only the static position from the previous frame is used.

To ensure that the issues from the re-ID of DeepSORT would not negatively effect the evaluations of the tracking part, the capabilities of the re-ID part were limited. This can be achieved by allowing the tracker to re-use the features of a track that is younger than several frames, ensuring it only uses short-term features. This results in the tracker only using creating a new trajectory instead of trying to link a lost trajectory to a newly detected subject. This, however, does not disable the usage of the Deep part when associating BBs, and the tracker will use short-term features to improve the accurate creation of sequences.

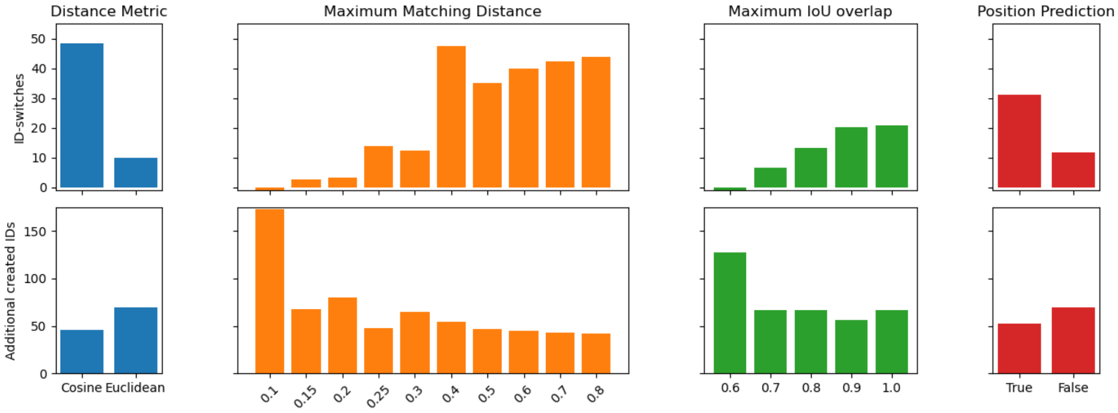


Figure 5.3: Results of the parameter sweep of DeepSORT. The number of IDSW needs to minimized whilst not significantly increasing the additional creation of IDs.

A parameter-sweep was performed, from which the results are shown in Figure 5.3. Note a clear relation where the amount of IDSW is decreased with: the Euclidean distance metric, minimizing the maximum distance cost (>0.1), minimizing the maximum overlap (>0.6), and with position prediction with velocity disabled. Decreasing below the mentioned limit, causes the tracker to lose the ability to associate correctly, causing the tracker to make create large amounts of extra ID's, which is undesirable.

FairMOT

Similar to DeepSORT, FairMOT is an application that is able to detect, track and re-identify. Likewise these tasks have influence on each other. For tracking, FairMOT has the following adjustable parameters:

- Confidence Threshold for Tracking **(C)**: Threshold where only larger confidence matches are considered when associating subjects to tracks. Increasing this value will ensure more confident associations.
- Confidence Threshold for Detection **(D)**: Threshold where only larger confidence detection is considered. Decreasing this value will increase the amount of FPs.
- Maximum IoU overlap **(0)**: Expressed in non-maximum suppression, threshold at which overlapping entities are considered one. Increasing this value will allow more overlapping BBs.

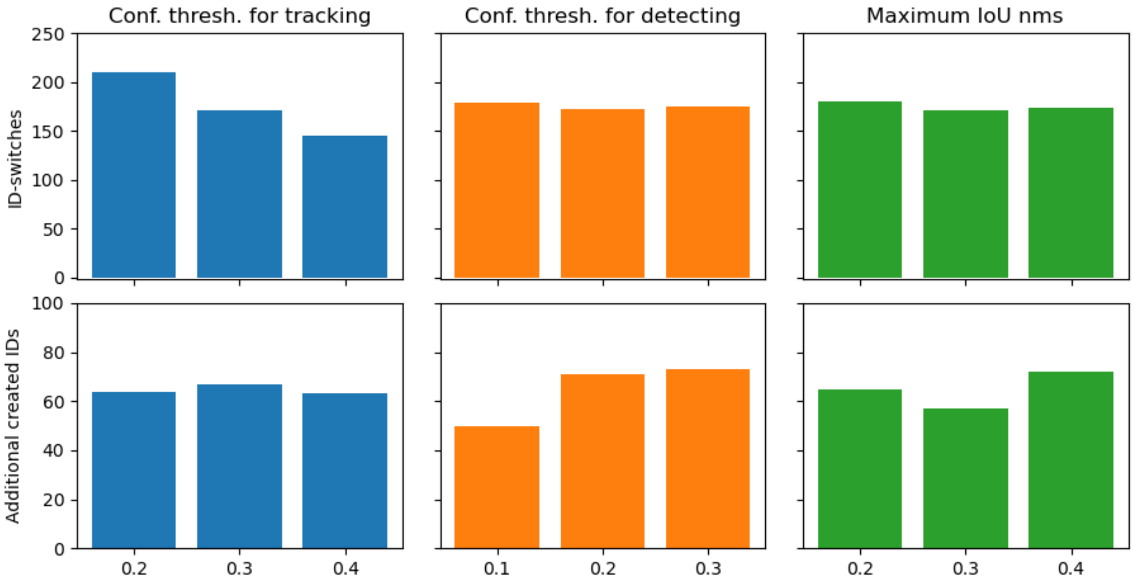


Figure 5.4: Results of the parameter sweep of FairMOT. The number of IDSW needs to minimized whilst not significantly increasing the additional creation of IDs

Over these values, a parameter-sweep was performed, from which the results are shown in Figure 5.4. Only a change in the Tracking confidence appears to have an influence on the number of IDSWs, where increasing the confidence results in less IDSW.

Other applications

CenterTrack Another application that is based on CenterNet is CenterTrack [19]. It utilizes the displacement of pixels to track subjects and is trained in that manner. If the model is trained with still images, the backbone will displace the input to simulate a moving subject. A model was trained for CenterTrack using the non-mosaiced training set and with the same settings as FairMOT. The final model did not work properly, the predictions of future heatmap locations were off, resulting in no associations but only new ID's. Even with tweaking the settings, this problem was not solved. Additionally, the generated BBs were often not covering the entire horse.

5.2 Tracking Evaluation

The evaluation of the tracker is done with the same application, TrackEval, that evaluates the re-identification task. The following metrics will be used to evaluate the trackers:

- IDSW: Count of the number of identity switches that have happened. Occurs when:
  - A track has been lost and a new ID is initialized (detected ID's minus GT ID's)
  - 2 subjects wrongfully switch ID
- IDT: Count of the number of identity transfers that have happened. Based on the IDSW, the wrongfully switch of IDs.

A problem with this evaluation method is that the number of IDT uses the number of created IDs which also includes FP and multiple BBs for the same object (see Figure 5.5). This results in the evaluation being unreliable when the detector creates too many IDs, where the value for IDT becomes negative.

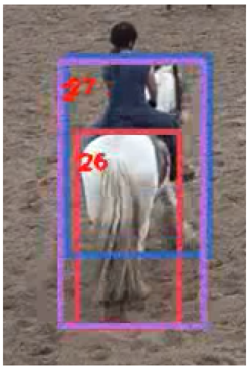


Figure 5.5: Example where the evaluation method starts failing. One subject has multiple BBs, creating additional IDs. Image is from fragment 1, using FairMOT

Table 5.1 shows the evaluation of the tracking based on the metrics IDSW and IDT. The shown data is not all the data that has been obtained, but only the top 3 ranking results from each tracker.

Table 5.1: Results of the evaluation of the Tracking. The Out-of-the-Box (OotB) models are also displayed to compare to. The tag holds configured value for the parameters explained in section 5.1.

Tracker	Tag	IDSW	IDT
DeepSORT	(YoloV4) OotB	69	53
DeepSORT	(YoloV4-Tiny) OotB	75	44
DeepSORT	(Y640_M_B) C15-O90-PF-T_e	63	7
DeepSORT	(Y640_M_B) C25-O90-PF-T_e	55	6
DeepSORT	(Y640_M_B) C15-O70-PF-T_e	48	5
FairMOT	OotB is trained for humans		
FairMOT	(DLA34) D30-C30-O30	177	76
FairMOT	(DLA34) D30-C30-O40	188	77
FairMOT	(DLA34) D10-C30-O40	187	53

5.3 Tracking Discussion

Whilst FairMOT outperformed DeepSORT as a detector, DeepSORT was better at handling occlusions and holding stable tracks as can be concluded from the smaller IDSWs and IDTs from the evaluation. DeepSORT was able to be configured to eliminate many difficult occlusions where FairMOT failed. With the online method of handling tracking, the significance of such errors is catastrophic in this use case.

Chapter 6

Horse re-Identification

The last step of the pipeline is the (re-)identification of the horses. With the sequences made by the tracker, the re-ID application should identify which horse is which and create tracks that represent a single subject for the duration of the video. Figure 6.1 illustrates the flow for this part. It uses the output of the tracker to associate sequences to one another into tracks and associate tracks with existing subjects from the video collection.

The re-ID would involve identifying which long-term features the feature extractor should focus on with horses to be able to differentiate horses from one another. The re-ID is used in 2 different situations: V2V re-identification and re-ID after long occlusions. For both situations, some type of information extractor is used to extract long-term data to compare subjects with. The type of extractor can differ from application to application. Both situations come down to the same thing, comparing feature data to the detected features against existing ID's, however, when approaching V2V feature might look different due to camera settings and lighting conditions.

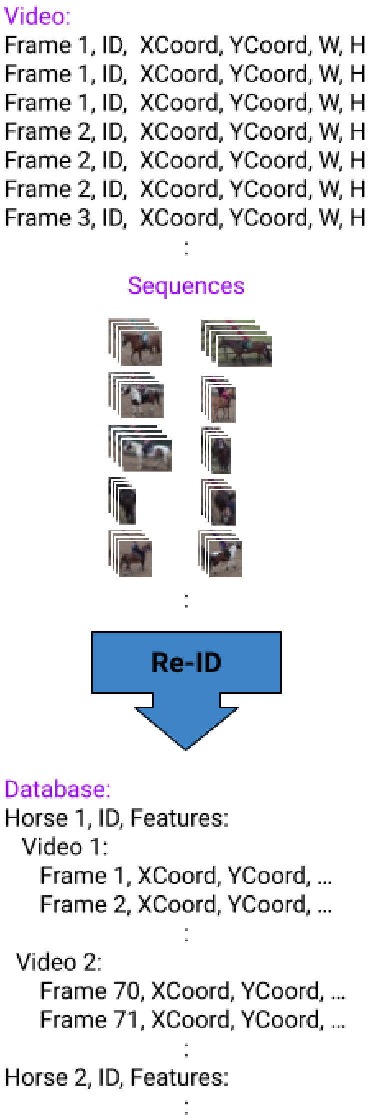

Video:
Frame 1, ID, XCoord, YCoord, W, H
Frame 1, ID, XCoord, YCoord, W, H
Frame 1, ID, XCoord, YCoord, W, H
Frame 2, ID, XCoord, YCoord, W, H
Frame 2, ID, XCoord, YCoord, W, H
Frame 2, ID, XCoord, YCoord, W, H
Frame 3, ID, XCoord, YCoord, W, H
⋮
Sequences
⋮
Re-ID
Database:
Horse 1, ID, Features:
Video 1:
Frame 1, XCoord, YCoord, ...
Frame 2, XCoord, YCoord, ...
⋮
Video 2:
Frame 70, XCoord, YCoord, ...
Frame 71, XCoord, YCoord, ...
⋮
Horse 2, ID, Features:
⋮

Figure 6.1: Process of the re-identification task of the pipeline. The re-ID uses an feature extractor to be able to compare the appearance of the subjects, but can also use information like location, time of appearances to determine which associations are likely.

Challenges and Requirements The challenges in this task are:

- *Minor visual differences between subjects*: Horses can look identical with minor visual differences in certain parts of the body.
- *Varying perspectives*: Like object detection, comparing appearances over time with a moving object will make feature identification a challenge.
- *Changing appearance*: Horses can have changing appearances (due to gear changes) and horses can have different riders on them at different times.
- *Lighting*: Horses can look different in varying light conditions, this can change colours but also shadows can interfere with referencing between colours.
- *Motion blur*: When looking from frame to frame, inaccuracy in recordings due to motion blur can cause the subject to look different.

Re-ID'ing subjects after longer occlusions can be considered as an extension of the tracking, where sequences are associated to one another, resulting in tracks containing a single subject. However, the difference is in the short- and long-term features that are required to perform this with high accuracy. The challenges and requirements stated in tracking can be used for such situations. In the second situation, the application should attempt to associate the subject in the track to the subject that is present in the entire collection of input videos. In this process the following aspects are important:

- Matching subjects based on appearance should consider multiple frames of the past and future, this should reduce the chance of wrongly associating subjects to trajectories. This is the difference between the 'Online' and 'Offline' tracking explained in section 2.3 on page 4.
- When subjects look alike and no difference can be identified, the subjects should be considered as one or as a subgroup. An example is the black horses in the evaluation set, which to an untrained eye seem identical.

6.1 Re-Identification Applications

DeepSORT

To continue on the DeepSORT application, the Deep part has the following parameters that can be tweaked:

- Feature Budget **(B)**: The number of older frames' features from a track are used for associating a subject. Decreasing this value results in the association based on the fewer older frames. If subjects turn, older frames can have outdated features which do not compare with the new angle.
- Maximum age of the Track **(A)**: DeepSORT has 3 track states: Alive, Lost, Dead. If a Track is found in a frame, the track is Alive. If the Track is not associated in a frame, the track is Lost. If the track is not associated for longer than the maximum age, the track is presumed dead and will not be considered anymore. Decreasing this value will force the algorithm to make new IDs instead of re-using older ones.
- Minimum initialization frames **(N)**: The amount of frame a new track needs to be assumed as new before the track is made. Increasing this value will allow more time to find an appropriate match.
- The type of feature extractor is used. An interchangeable model that returns a feature matrix of the subject.

DeepSORT allows for interchanging the feature extractor, however has already implemented a few MARS [43] which looks at the change of colours between frames. 4 different MARS models were tested: small128 (default), triplet, magnet, and cosine [22].

A parameter sweep was performed, from which the results are shown in Figure 6.2. It can be concluded that changing these settings, do not have a significant effect on the HOT-score with only a change of less than 2%. However, in the amount of IDSW change in settings can be noted: Setting a limit in the number of features that can be used, taking a maximum age of around 20 frames and an initialization time of around 5 frames, with the triplet setting, show the best results in terms of IDSWs

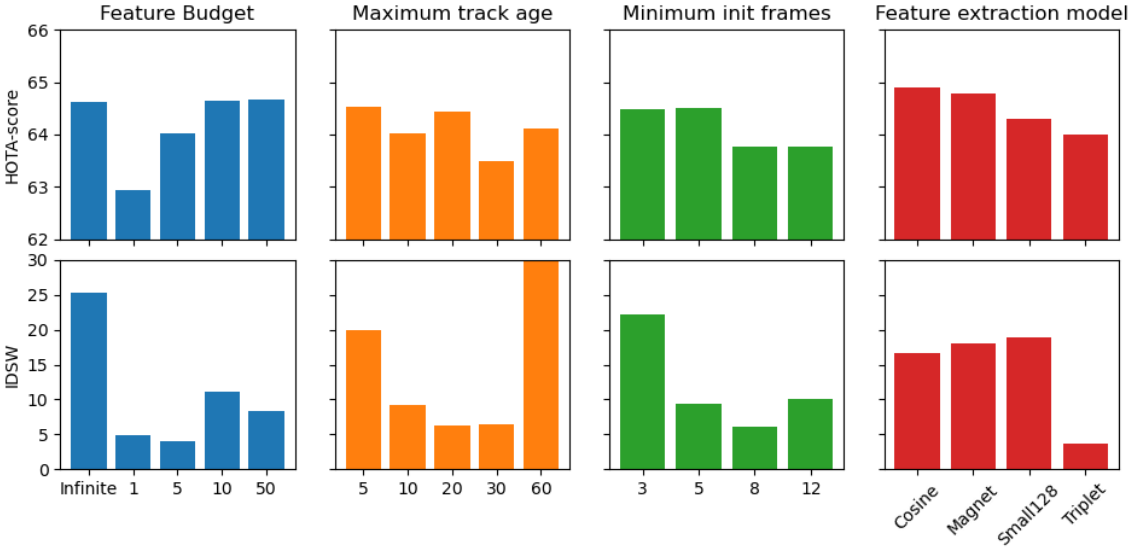


Figure 6.2: Results of the parameter sweep of DeepSORT. The number of HOTA needs to be maximized, whilst the amount of IDSW needs to minimized.

FairMOT

Continuing on the FairMOT applications, the following parameters can be adjusted:

- Tracking Buffer **(B)**: The number of older frames' features from a track are used for associating a subject. Decreasing this value results in the association based on the fewer older frames. If subjects turn, older frames can have outdated features which do not compare with the new angle.
- Re-Identification matrix dimensions **(R)**: The dimensions of extracted features matrix. Increasing the dimensions will allow for more information to be extracted and compared to, but will require higher match confidence.

A parameter sweep was performed, from which the results are shown in Figure 6.3. The dimensions of the re-ID matrix do aid in the accuracy of the tracking. This is probably true due to the extractor being trained on humans and not on horses. The buffer does however increase the HOTA-score but also increases the inaccuracy in IDSWs.

No further applications were considered for re-ID.

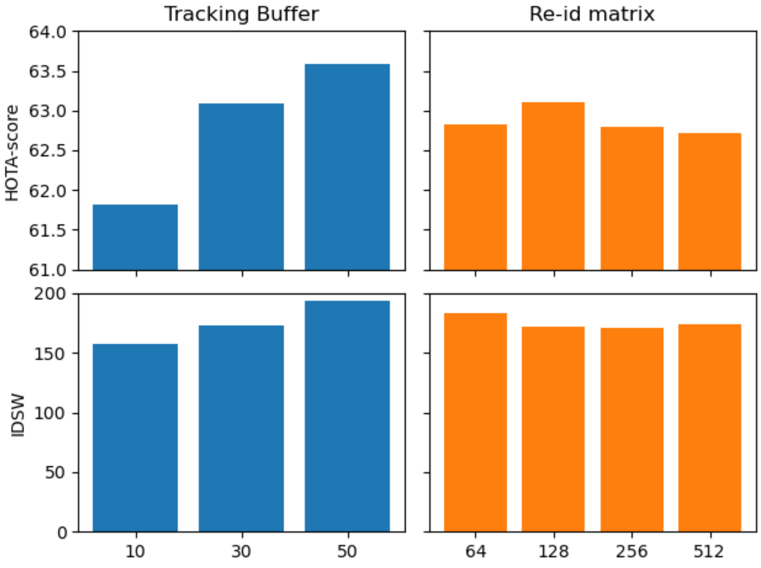


Figure 6.3: Results of the parameter sweep of FairMOT. The number of HOTA needs to be maximized, whilst the amount of IDSW needs to minimized.

6.2 Re-Identification Evaluation

Evaluating the re-identification is done using the HOTA [61] and CLEARMOT [62] metrics. These show the complete evaluation from detection to ID'ing, as all tasks before influence the last task. The following metrics are selected to evaluated the re-identification:

- Higher Order Tracking Accuracy (HOTA): The geometric mean of the DetA and AssA scores, which are:
  - Detection Accuracy (DetA): Percentage of GT detections were correct. An combination of the recall and the precision of the detections.
  - Association Accuracy (AssA): Percentage of GT associations were correct. An combination of:
    - The recall: measure of how well the predicted trajectories compare to the GT, a low percentage results in an object that is split into multiple predicted tracks
    - The precision: measure of how well the trajectories keep to the same GT trajectories, a low percentage results in a trajectory split into multiple objects.
- Multi-Object tracking accuracy (MOTA): Summary of over- all tracking accuracy in terms of false positives, false negatives and identity switches.
- Mostly tracked (MT): GT tracks that have the same label for at least 80% of their life span.
- Partly tracked (PT): GT trackes that have the same label between 80% and 20% of their life span.
- Mostly lost (ML): GT tracks that are tracked for at most 20% of their life span.
- IDSWs: Number of times the reported identity of a GT track changes.

Table 6.1 shows the evaluation of the re-ID using the metrics stated above. The shown data is not all the data that has been obtained, but only the top 3 ranking results from each application.

Table 6.1: Results of the evaluation of the re-ID. The Out-of-the-Box (OotB) models are also displayed to compare to. The tag holds configured value for the parameters explained in sections 5.16.1.

Re-Id	Tag	HOTA	MOTA	MT	PT	ML	IDSW
DeepSORT	OotB (YoloV4)	25.54	31.34	0	20	11	69
DeepSORT	OotB (YoloV4-Tiny)	44.74	56.78	10	19	2	75
DeepSORT	(Y640_M_B) PF-e-B10-A30-N8-magnet	55.89	75.72	15	16	0	55
DeepSORT	(Y640_M_B) PF-e-B10-A10-N8-magnet	55.74	75.63	15	16	0	54
DeepSORT	(Y640_M_B) PF-e-BN-A20-N3-triplet	55.53	74.55	17	14	0	72
FairMOT	OotB is trained for humans						
FairMOT	(DLA34) D30-C30-O30-B50-R64	56.73	73.49	14	17	0	177
FairMOT	(DLA34) D30-C30-O40-B50-R64	56.26	73.49	14	17	0	188
FairMOT	(DLA34) D10-C30-O30-B50-R128	56.12	73.54	14	17	0	187

6.3 Re-Identification Discussion

The problem with FairMOT's tracking can be seen in the evaluation of the re-ID where the amount of stable tracks (concluded from MT, PT and ML) is outperformed by DeepSORT. This is due to the amount of IDSWs FairMOT incorrectly makes.

The re-ID part of the applications have only aided in solving occlusions and not creating tracks from sequences. The result of the re-ID task is not decisive as V2V re-ID was not attempted with the SOTA due to the lack of the good feature extractor for horses. The feature extractors used by YoloV4 and FairMOT were both trained for humans and did not have the accuracy required for the determination of association confidences for V2V re-ID. The next chapters will discuss a solution to this problem.

Chapter 7

Results & Discussion

The current output of the pipeline contains all the desired information stated in the aim, sorted on frames of the video. With a conversion script the output of the applications was converted to a table containing (per unique subject):

- Path of the video
- Frame number
- Left side of the BB (minimum x-pixel)
- Top side of the BB (minimum y-pixel)
- Right side of the BB (maximum x-pixel)
- Bottom of the BB (maximum y-pixel)

With this information, a sequence of images of a subject can be extracted from the video. The missing part in the pipeline is an extended re-ID that is able to link these sequences.

The current pipeline performs different between the 4 fragments (explained in section 3.3 on page 11). In Table 7.1 it can be seen that both applications performed significantly better on fragment 1, and significantly worse on fragment 3. This is due to the placement of the camera, where fragment 1 is placed on higher ground than fragment 3, and the density of subjects and their number of occlusions (see Figure 3.2 on page 9). The final result from fragment 1 would satisfy the requirements of this thesis without the addition of the extra re-ID'ing and V2V re-ID'ing as can be seen in Figure A.5 on page 44.

YoloV4 and FairMOT show promise in their ability to localize horses from video with high accuracy whilst minimizing FPs. These detectors however have been made for speed to satisfy the requirement of live video processing, which is not a requirement of this use case. The trained detectors were able to correctly localize 87% and 93% of the horses respectively, where the missed detection were occluded, far away and blurred horses. YoloV4's BB creation outperformed FairMOT's, where FairMOT tends to see multiple objects in a single object as can be seen in Figure 5.5 on page 27 and A.2 on page 41.

Table 7.1: Analysis of the applications on each individual evaluation fragment. Taking the average of the 3 best performing tracks per application from the re-ID evaluations in section 6.2 on page 31. In the 'App' row, 'DS' refers to DeepSORT and 'FM' refers to FairMOT.

Fragment	**HOTA [%]**		**MOTA [%]**		**MT**		**PT**		**ML**		**IDSW**		**IDT**	
App	DS	FM	DS	FM	DS	FM	DS	FM	DS	FM	DS	FM	DS	FM
1	89.6	88.0	91.7	88.8	7	6	1	2	0	0	2	15	0	6
2	65.4	61.6	75.3	61.7	3	2	4	5	0	0	9	26	3	12
3	43.5	38.2	52.2	62.4	0	1	8	7	0	0	27	67	5	10
4	70.7	68.2	75.8	75.8	5	5	3	3	0	0	18	68	6	36

DeepSORT and FairMOT (using YoloV4 and FairMOT as detector respectively) were used for tracking, where a limitation was identified. With the online method of tracking (only using the current and previous frame instead of multiple past and future frames) in the combination of incorrect re-IDs when associating lost tracks, results in IDT that are catastrophic in this use case.

The re-ID part of the applications continued on this problem. An additional part needs to be added to the pipeline that would extend the re-ID part of the thesis, as the current SOTA applications can not solve this problem. The proposed solution to the problem is to ensure that amount of IDT is minimized, which can be achieved by promoting new IDs instead of re-ID after difficult occlusions. This results in sequences (shorter track outputs) that need to link into real tracks of each subject. In Figure 7.1 an illustration can be found that shows the output of the current pipeline.

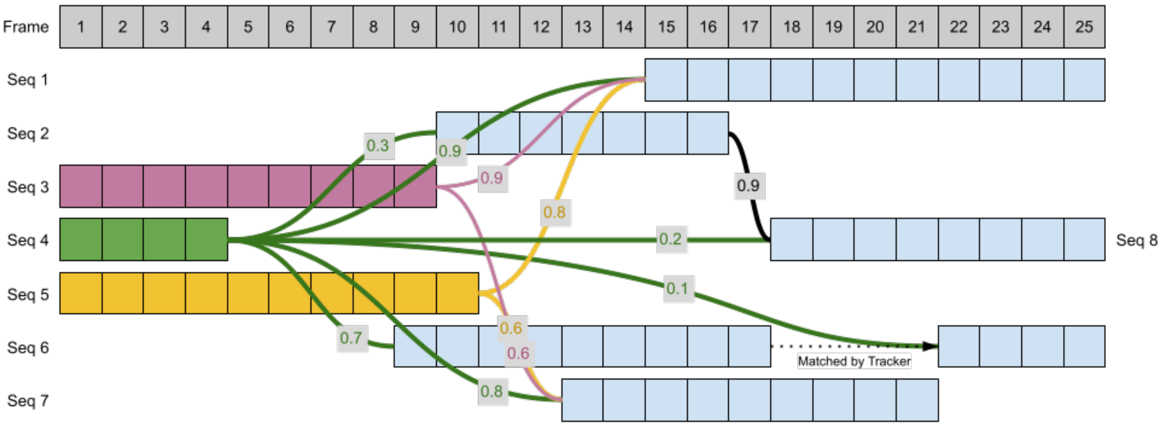


Figure 7.1: An simplified example of associating sequences, the output of the tracker, to another. The boxes represent detection and are linked in sequences. The lines between the boxes are hypotheses of association, with a probability based on details of the sequence. Without considering past and future associations of other sequences, the solution can become illogical resulting in incorrect associations due to the first-come-first-serve type of association. However, allowing these considerations and making hypotheses, should allow for a more optimal solution.

The second limitation of the DeepSORT and FairMOT is their incapablities to extract long-term features that can be used when re-ID'ing. The additional part of the re-ID would need to link by associating (short- and long-term) features, location, velocities and time bounds of the sequences. This problem can be classified as a lifted, multicut or multi-camera association problem and will be further explained in section 8.3 on page 36.

All in all, the full pipeline could not be completed with the current open source SOTA applications. The used SOTA applications are not designed for this use case, where IDT errors are not as catastrophic in the evaluation dataset that these applications use. These datasets mostly consist of a short appearance of a unique subject, where the only re-ID that needs to be done is after short occlusions, and the track can be considered dead when the subject leaves the video. In this use case, re-ID'ing is required after a subject leaves the video and even from V2V.

Chapter 8

Future work

In this chapter, suggestions on improvements are given in where for each part of the thesis. These suggestions are concluded from the work done during this thesis and were not implemented or tested due to time restraints or that suggestion being outside of the scope of this thesis. The order at which the suggestions are listed refers to the significance the suggestion can influence the output of the task, the first being most significant.

8.1 Object Detection suggestions

Input from previous detections When an object is detected in a prior frame, that object is likely to be present in the next frame. Using the position, velocity and size of the previous detection(s), an expected location can be predicted which can aid in the localization in the current frame. This could help stabilize detections and solve issue shown in Figure A.2 on page 41.

A bigger CNN One application used is this task is YoloV4. This application was built for high-speed detections which penalizes the accuracy of the detections. CNNs like EfficientNet [59] which has 480 million parameters (EfficientNet-L2) compared to Yolos' 64 million parameters.

Adjusting input size Commonly the input size of a CNN is square as the aspect ratio of images can vary. However, in the case of this use case, the input size should have the aspect ratio of 16:9 as standardize resolutions are 1920x1080 and 1280x720. This would decrease information loss due to resizing the video to the required input size.

Adjusting the training dataset The final result from the trained models show limitations in the detection of horses that are moving closer to the camera. The algorithm is trained on still images and not trained on moving objects. The augmentation blur could be used to increase the accuracy in detecting moving objects.

Additional Classes To reduce the chances of a detection being a FP, the detector can be trained to also classify the other objects in the videos. In this use case, horses and humans appear in the video, where the detector has issues handling occlusions with humans. With training for humans as well, the detector should improve on such situations.

8.2 Subject Tracking suggestions

Backwards confirming of sequences When associating subjects to sequences, the association check is only performed once, after which the association is kept. If an association is wrongfully linked, a later check could correct the sequence by re-associating the previous frames of the sequence and the current subject. With this information, an extra cut in the sequence could be made to correct the miss-association.

Detect Occlusions One of the biggest flaws of the approach is the inability to correctly handle occlusions, where IDT happen. Recognizing when an occlusion happens could aid in correctly handling the occlusions like Romero-Ferrero et al. [2] have done.

Better use of trajectories The current implementation does not utilize the direction of the subject as much as it could. The significance of this propriety of a sequence is not used to its full extent when determining associations. Referring back to section 2.2 on page 3, the current applications only utilize the Feature-based tracking in combination with previous and current position probability.

8.3 Re-identification suggestions

This section discusses the requirements of the additional part that would be needed to be added to the pipeline to satisfy the ultimate goal of this thesis. It would require an accurate feature extractor and an accurate method of solving the sequence association into tracks (as illustrated in Figure 7.1 on page 34).

Feature extraction for horses As stated, the feature extraction model used in the SOTA tracking applications are trained for humans. To bypass this issue, the solution was to use other feature extractors like MARS which looks at the motion of RGB streams [43]. This approach allowed to re-ID from frame to frame, how did not show promise in its ability to re-ID from sequence to sequence.

To be able to re-ID from sequence to sequence and track to database, a more accurate feature extractor needs to be trained. Multiple methods have been proposed in the related works that are used for humans and in SOTA applications:

- Training on a dataset comparable to Market1501 [25] (contains 1501 unique identities, from 6 camera's with 33K annotations) NNs can be trained to identify which features are important to differentiate horses from another [5, 26, 44–46].
- Segmentation of body parts [6] in combination with attribute extraction [47] might be an improvement on the previous suggestion. This also allows for better identification of the subjects within the database with a textual description of the subject.
- When sequences are merged into tracks, the subject can be seen from multi-angles. Identifying how a subject looks at each angle and associating a possible association with that same angle could aid in the correctly re-ID of sequences and tracks to subjects in the entire collection of input video's [63].

Without an accurate feature extractor, neither the sequence to sequence nor the track to database associations can be completed.

The current hurdle to implement this suggestion is the lack of an appropriate training dataset that consists of many images of many unique horses and labelled as such. A comparable dataset for humans would be Market1501 [25] consisting of 1501 unique subjects with 32.6k images, averaging at 21.8 images per subject.

Offline lifted sequence association DeepSORT and FairMOT both are applications that approach re-identification in an online matter, with a single association and by only associating subjects to sequences using the current and previous frame(s). This online method limits an association application when the video footage is pre-recorded. An example of this limitation is when the application incorrectly associates subjects to tracks, resulting in a continuation of that incorrect association due to the application not being able to correct the past associations. Considering more frames before making an association should generally aid in solving long-term occlusions and false or missed detections [26, 29, 42, 64]. Additionally, it could filter out FP, for example, sequences like shown in Figure A.1 on page 40. The solution used in this thesis was to ensure that the number of incorrect associations was minimized, extracting sequences uninterrupted of subjects. An application should be able to solve the sequence association the arises from these sequences (see Figure 7.1). It should be noted that the creation of sequences is to aid such an application with already giving associated detections, however, might work better if given the raw detections (giving the application the ability to track subjects itself). The application could use the following details of a sequence to compute probabilities of associations: subjects position, velocity and features within the sequences, and the time bounds of the sequence.

Several data association algorithms have been used through-out SOTA applications. Markov Chain Monte Carlo Data [64], cost or association matrices [24], Lifted edges [29] and Tracklet-plane matching [42] association was be used to solve the problem by giving multiple hypothesis [65] from which the most likely solution can be obtained.

Chapter 9

Conclusion

The goal of this thesis was to create a pipeline that converts a collection of raw heterogeneous videos into an informative database about the occurrence of each subject in the video, whilst being unsupervised (excluding the object detection training). The direct usage of this pipeline would be the ability to retrieve reference data which can be used when using active learning queries on sensor data. In addition, this pipeline would have the ability to aid in other projects like unsupervised unique animal counting, subject following over a multi-camera setup and as a part of a project that extracts video streams of a unique subject.

This thesis defined 3 tasks; Object Detection, Subject Tracking, and Subject Re-identification, which the pipeline should complete to be able to extract the desired information. The three tasks are evaluated and discussed separately in the chapters 4, 5 and 6 respectively. Using existing applications, object detection and subject tracking was achieved with an accurate result where FN in detections and IDT in tracking were minimized. Existing applications also showed promising re-ID results with short-term features, however they did not fulfil the requirements of what the re-ID application should do. The usage of the current re-ID can be seen as the extension of the tracker, aiding in solving occlusions. The output of the current pipeline are sequences of subjects that still need to be associated with an additional re-ID part as illustrated in Figure 9.1.

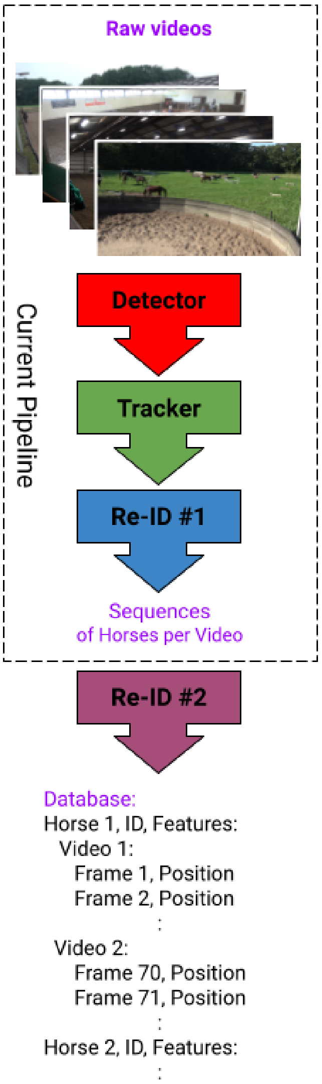


Figure 9.1: An extension of Figure 1.1 on page 2. The current output of the pipeline are sequences of horses that need to be associated into finalized tracks. 'Re-ID #2' refers to the discussed additional part to re-ID that needs to be created (as discussed in section 8.3 on page 36).

To conclude, with the SOTA applications YoloV4+DeepSORT and FairMOT the original 3 tasks were completed. The pipeline has the ability to detect 93% of the horses with a detector and only has 5 IDTs in the evaluation fragments totalling 2 minutes of video. The current pipeline gave a decent result on fragment 1 where the video conditions were the best. From this footage, the pipeline was able to extract all the desired information: For each subject; the video path, the frame in the video and the x and y coordinates of the horses BB.

Glossary

BB Bounding Box. 1, 3, 10–17, 19, 20, 23–27, 33, 38

CNN Convolutional Neural Network. 1, 3, 4, 7, 18, 24, 35

FN False Negative. 18, 22, 38, 42
FP False Positive. 18, 20, 22, 26, 27, 33, 35, 37, 40, 43

GT Ground-Truth. 6, 20, 27, 31

IDSW Identity Switch. 24–27, 29–32, 42
IDT Identity Transfer. 4, 24, 27, 34, 36, 38, 43
IoU Intersection over Union. 20–22, 25, 26

mAP mean Average Precision. 18, 20–22
ML Machine Learning. 3, 5, 6

NN Neural Network. 2–5, 7, 10, 15–19, 36

OotB Out-of-the-Box. 17, 23

re-ID Re-Identification. 1, 11, 25, 28, 30–34, 36, 38

SOTA State of the Art. 1, 2, 17, 32, 34, 36–38

V2V Video to Video. 1, 28, 32–34

Appendix

Training setup The University of Twente provides a GPU-assisted computing platform with high amounts of allocatable VRAM, which allows for higher training speeds. These virtual machines have a restriction for up- and downgrading packages, which resulted in complications for running applications that were created for older versions of packages. The version of OpenCV was not compatible with Darknet, not allowing the backbones' augmentations. For that reason the augmentations were made manually through the method explained in 3.4 on page 13. DNCv2 is an commonly used module that needs to be complied in GCC and G++ versions 7. The virtual machines have versions 9. Due to these complications, most other atempted applications could not be trained through these virtual machines. The other machine used to test en train (FairMOT) was a P51 Thinkpad, i7-core with 4GB VRAM, which limited the amount of training that could be performed due to the low GPU power.

The main programming language thesis thesis uses is Python. This due to the Python support most backbones have and Python is the language AiSensus was written in. No other languages were considered. The training datasets used for the models are the images explained in 3.4 on page 13 and the evaluation dataset is the dataset explained in 3.2 on page 11.

Reference images The images below are referred to throughout the report.

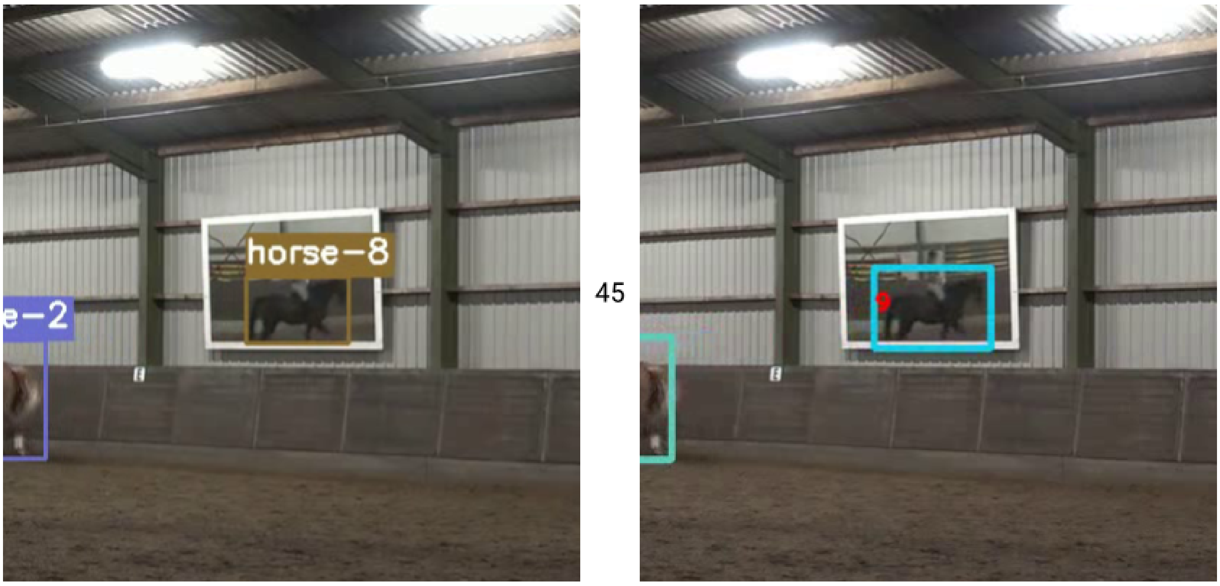


Figure A.1: Example of where a mirror creates FP detections. The left image is DeepSORT, the right is FairMOT, frame 45 taken from fragment 3.

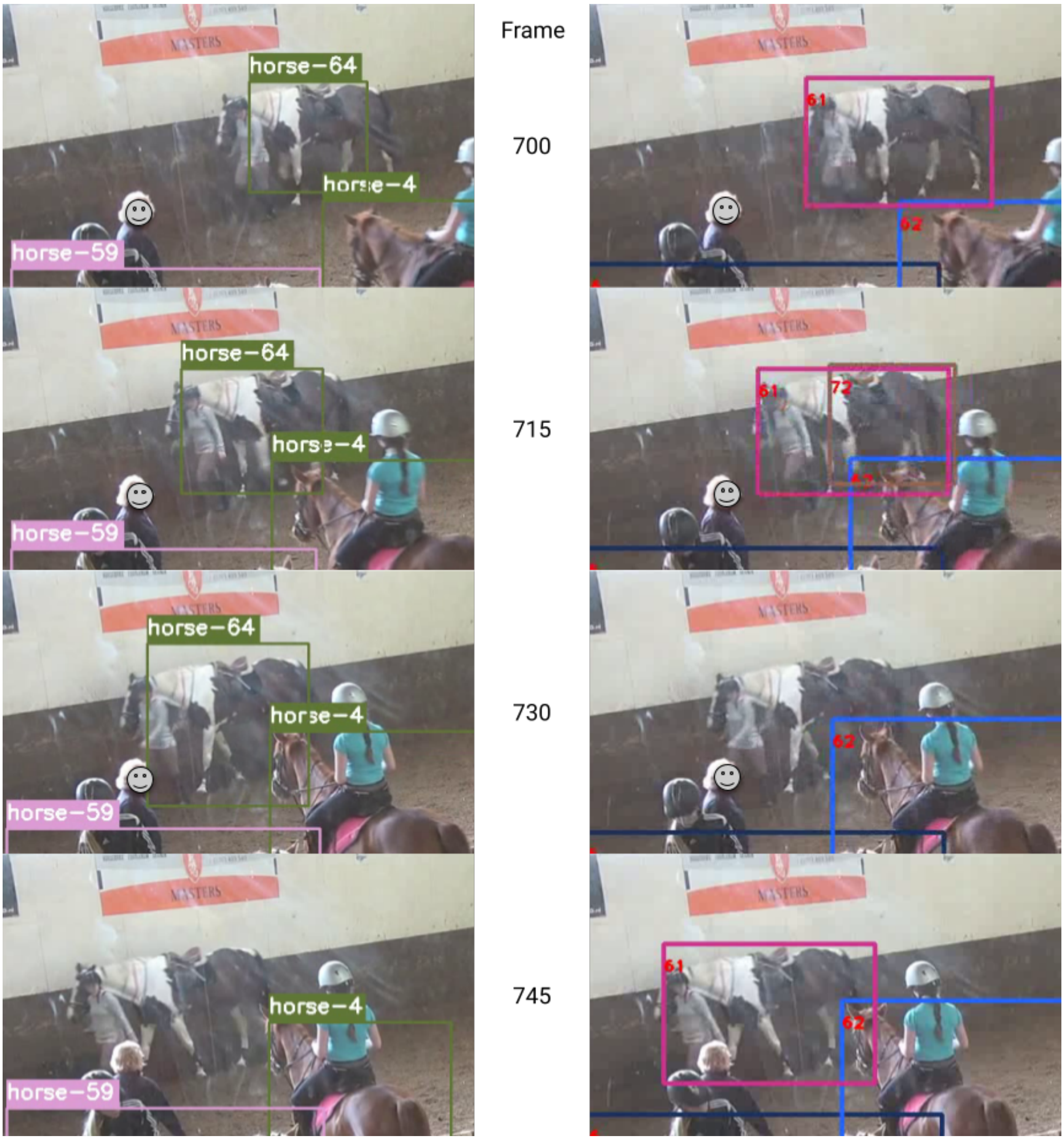


Figure A.2: Example of where the flare of the sun and a dirty window can influence the detectors ability to localize the horses correctly. This is most like due to the features of a horse (patterns, colors)being disrupted. The left column is DeepSORT, the right is FairMOT. The frames are numbered and taken from fragment 4.

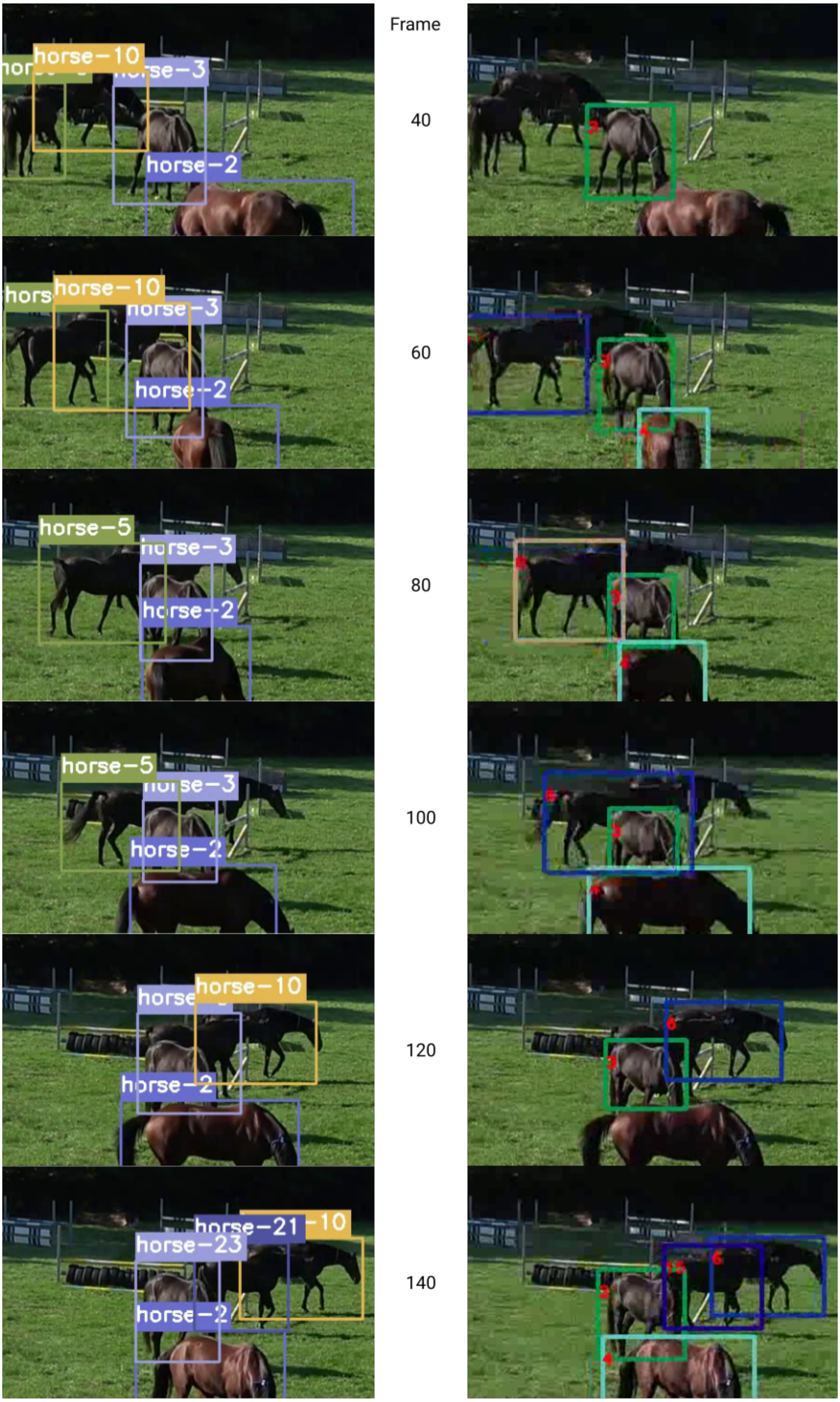


Figure A.3: Example of a difficult occlusion resulting in FNs and IDSWs. The left column is DeepSORT, the right is FairMOT. The frames are numbered and taken from fragment 2.

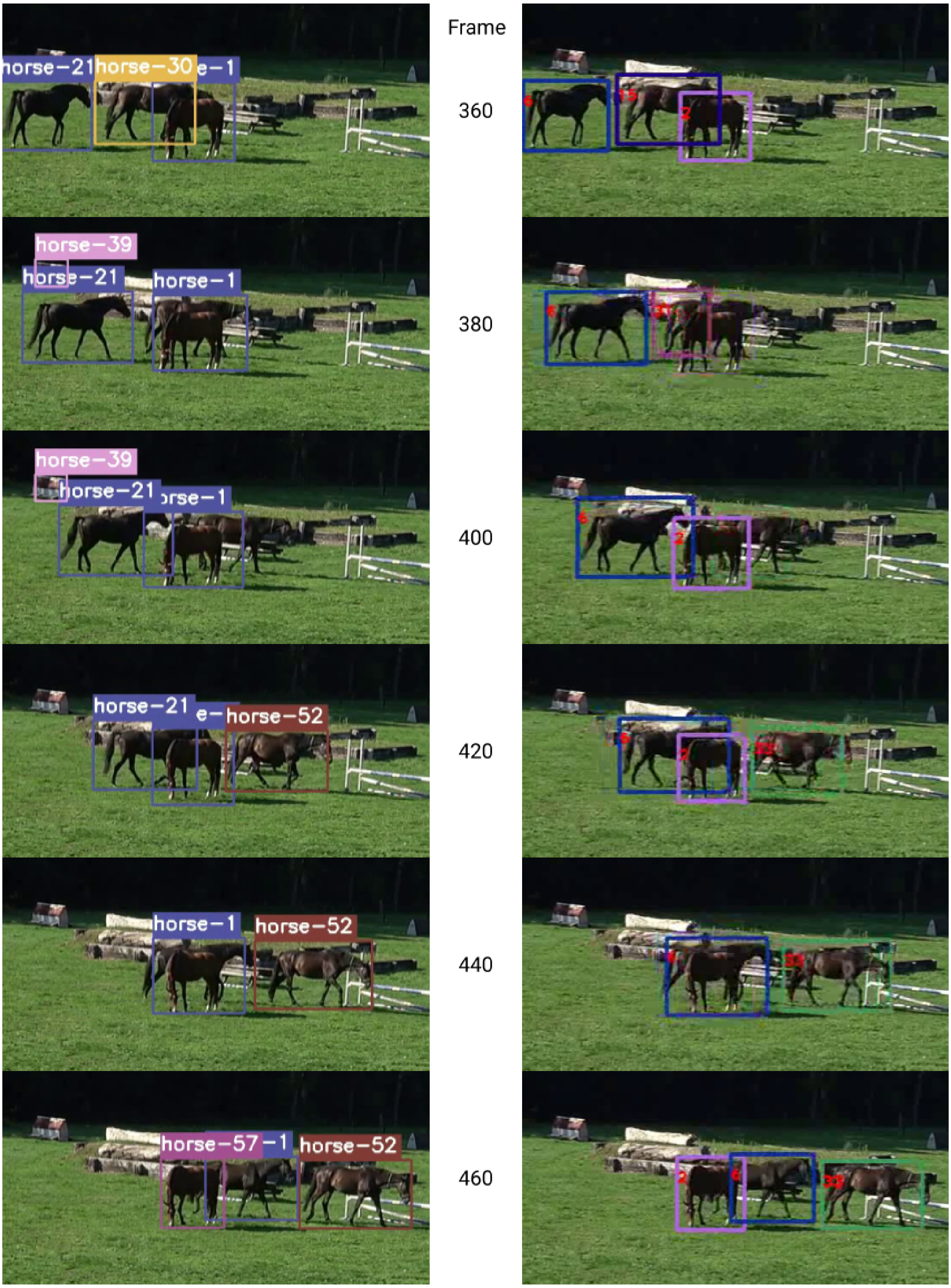


Figure A.4: Example of a FP (Horse ID 39 by DeepSORT) and a difficult occlusion resulting in IDT by DeepSORT. Horse ID 1 gets transfered to ID 21, creating a new ID 57. The left column is DeepSORT, the right is FairMOT. The frames are numbered and taken from fragment 2.

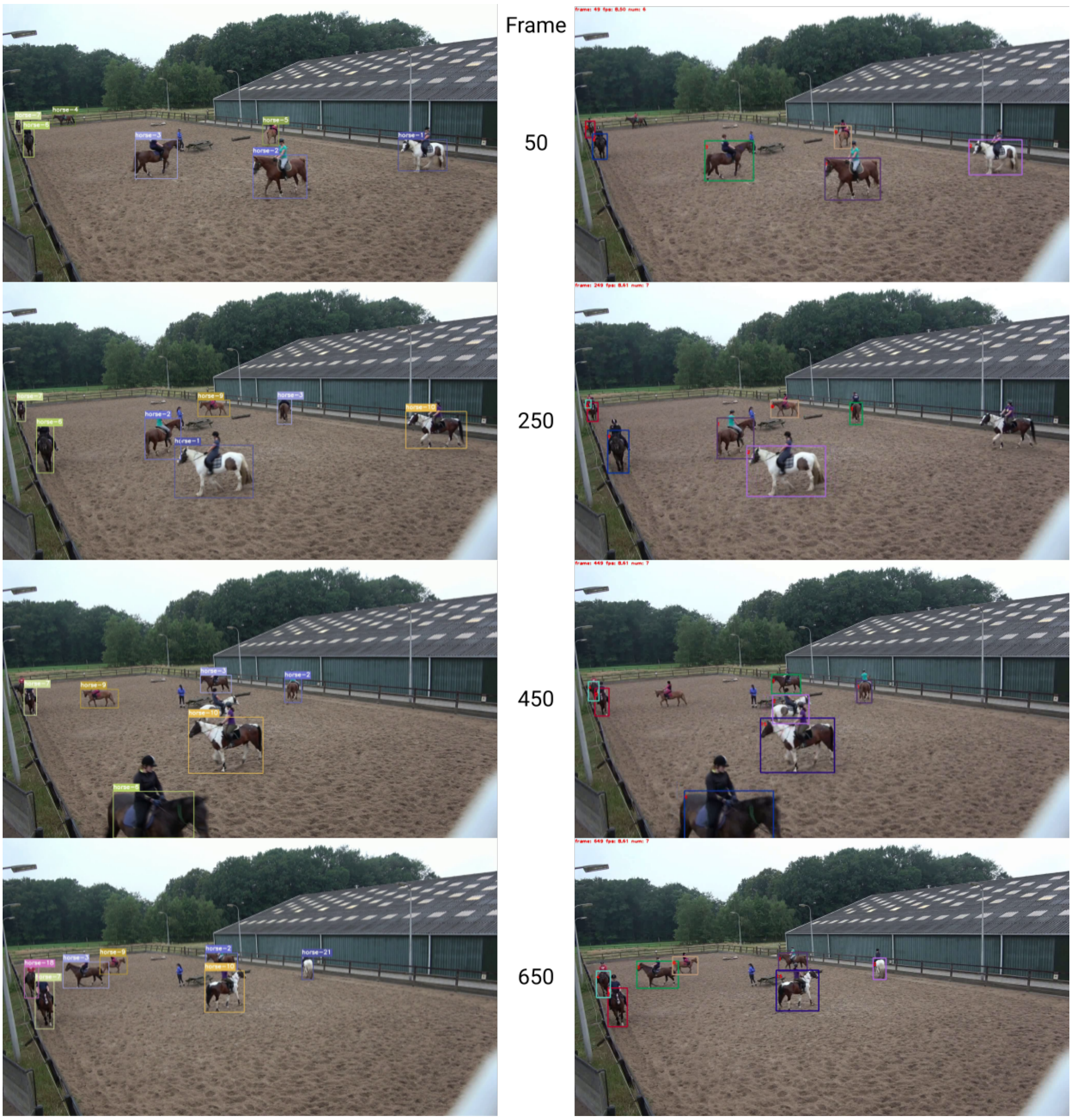


Figure A.5: Illustration of how the applications performs on fragment 1. Both applications have trouble with the far away horse and when that same horse gets occluded behind in the row. The left column is DeepSORT, the right is FairMOT.

References

[1] Alvaro Rodriguez et al. “ToxId: An efficient algorithm to solve occlusions when tracking multiple animals”. In: *Scientific Reports* 7.1 (2017). DOI: 10.1038/s41598-017-15104-2 (cited on page 1).

[2] Francisco Romero-Ferrero et al. “Idtracker.ai: Tracking All Individuals in Small or Large Collectives of Unmarked Animals”. In: *Nature Methods* 16.2 (2019), pp. 179–182. DOI: 10.1038/s41592-018-0295-5 (cited on pages 1, 3, 36).

[3] Mubarak Shah, Omar Javed, and Khurram Shafique. “Automated visual surveillance in realistic scenarios”. In: *IEEE Multimedia* 14.1 (2007), pp. 30–39. DOI: 10.1109/MMUL.2007.3 (cited on pages 1, 7).

[4] Srikrishna Karanam, Yang Li, and Richard J. Radke. “Person re-identification with discriminatively trained viewpoint invariant dictionaries”. In: *Proceedings of the IEEE International Conference on Computer Vision* 2015 Inter (2015), pp. 4516–4524. DOI: 10.1109/ICCV.2015.513 (cited on pages 1, 4, 7).

[5] Zhizheng Zhang et al. “Multi-granularity reference-aided attentive feature aggregation for video-based person re-identification”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2020, pp. 10404–10413. DOI: 10.1109/CVPR42600.2020.01042 (cited on pages 1, 4, 36).

[6] Wei Shen et al. “DeepSkeleton: Learning Multi-Task Scale-Associated Deep Side Outputs for Object Skeleton Extraction in Natural Images”. In: *IEEE Transactions on Image Processing* 26.11 (2017), pp. 5298–5311. DOI: 10.1109/TIP.2017.2735182 (cited on pages 1, 4, 5, 36).

[7] Michalis Vrigkas, Christophoros Nikou, and Ioannis A. Kakadiaris. *A review of human activity recognition methods*. 2015. DOI: 10.3389/frobt.2015.00028 (cited on page 1).

[8] Jacob W. Kamminga. *Hiding in the Deep Online Animal Activity Recognition Using Motion Sensors Andmachine Learning*. Tech. rep. 9. University of Twente, Oct. 2020, pp. 1689–1699. DOI: 10.3990/1.9789036550550 (cited on pages 1, 6).

[9] Jonathan P Crall et al. “HotSpotter - Patterned Species Instance Recognition Siva R . Sundaresan”. In: (2012), pp. 230–237 (cited on pages 1, 5).

[10] Stephen G. Dunbar et al. “HotSpotter: Using a computer-driven photo-id application to identify sea turtles”. In: *Journal of Experimental Marine Biology and Ecology* 535.April 2020 (2021), p. 151490. DOI: 10.1016/j.jembe.2020.151490 (cited on pages 1, 5).

[11] Mohammad Sadegh Norouzzadeh et al. “Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning”. In: *Proceedings of the National Academy of Sciences of the United States of America* 115.25 (2018), E5716–E5725. DOI: 10.1073/pnas.1719367115 (cited on pages 1, 7).

[12] Trish Franklin et al. “Photo-identification of individual humpback whales (Megaptera novaeangliae) using all available natural marks: Implications for misidentification and automated algorithm matching technology”. In: *Journal of Cetacean Research and Management* 21.1 (2020), pp. 71–83. DOI: 10.47536/JCRM.V21I1.186 (cited on pages 1, 5).

[13] Wei Liu et al. “SSD: Single shot multibox detector”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 9905 LNCS. 2016, pp. 21–37. DOI: 10.1007/978-3-319-46448-0_2 (cited on pages 1, 3).

[14] Shaoqing Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (2017), pp. 1137–1149. DOI: 10.1109/TPAMI.2016.2577031 (cited on pages 1, 3).

[15] Joseph Redmon et al. “You only look once: Unified, real-time object detection”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2016-Decem. 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.91 (cited on pages 1, 15, 18).

[16] Joseph Redmon and Ali Farhadi. “YOLO v.3: An Incremental Improvement”. In: *Tech report* (2018), pp. 1–6 (cited on page 1).

[17] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. “YOLOv4: Optimal Speed and Accuracy of Object Detection”. In: *arXiv* (2020) (cited on pages 1, 3, 15, 18).

[18] Ross Girshick et al. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 1 (2014), pp. 580–587. DOI: 10.1109/CVPR.2014.81 (cited on page 1).

[19] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. “Tracking Objects as Points”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12349 LNCS.Figure 1 (2020), pp. 474–490. DOI: 10.1007/978-3-030-58548-8_28 (cited on pages 1, 10, 26).

[20] Erik Bochinski, Volker Eiselein, and Thomas Sikora. “High-Speed tracking-by-detection without using image information”. In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2017* August (2017). DOI: 10.1109/AVSS.2017.8078516 (cited on pages 1, 3).

[21] Alex Bewley et al. “Simple online and realtime tracking”. In: *Proceedings - International Conference on Image Processing, ICIP* 2016-Augus (2016), pp. 3464–3468. DOI: 10.1109/ICIP.2016.7533003 (cited on page 1).

[22] Nicolai Wojke and Alex Bewley. “Deep cosine metric learning for person re-identification”. In: *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018* 2018-Janua (2018), pp. 748–756. DOI: 10.1109/WACV.2018.00087 (cited on pages 1, 24, 29).

[23] Mathilde Caron et al. “Deep clustering for unsupervised learning of visual features”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11218 LNCS (2018), pp. 139–156. DOI: 10.1007/978-3-030-01264-9_9 (cited on pages 1, 4).

[24] Shijie Sun et al. “Deep Affinity Network for Multiple Object Tracking”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.1 (2021), pp. 104–119. DOI: 10.1109/TPAMI.2019.2929520 (cited on pages 1, 4, 37).

[25] Liang Zheng et al. “Scalable person re-identification: A benchmark”. In: *Proceedings of the IEEE International Conference on Computer Vision* 2015 Inter (2015), pp. 1116–1124. DOI: 10.1109/ICCV.2015.133 (cited on pages 1, 36, 37).

[26] Siyu Tang et al. “Multiple people tracking by lifted multicut and person re-identification”. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 2017-Janua (2017), pp. 3701–3710. DOI: 10.1109/CVPR.2017.394 (cited on pages 1, 5, 36, 37).

[27] Li Zhang, Yuan Li, and Ramakant Nevatia. "Global data association for multi-object tracking using network flows". In: *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR* (2008). DOI: 10.1109/CVPR.2008.4587584 (cited on page 1).

[28] Fengwei Yu et al. "POI: Multiple object tracking with high performance detection and appearance feature". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9914 LNCS (2016), pp. 36–42. DOI: 10.1007/978-3-319-48881-3_3 (cited on page 1).

[29] Andrea Hornakova et al. "Lifted disjoint paths with application in multiple object tracking". In: *37th International Conference on Machine Learning, ICML 2020* PartF16814 (2020), pp. 4314–4325 (cited on pages 1, 37).

[30] Yifu Zhang et al. "FairMOT: On the Fairness of Detection and Re-Identification in Multiple Object Tracking". In: (2020), pp. 1–13 (cited on pages 1, 10, 19).

[31] Han Shen et al. "Tracklet Association Tracker: An End-to-End Learning-based Association Approach for Multi-Object Tracking". In: *arXiv* (2018) (cited on pages 1, 4, 5).

[32] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: (2009), pp. 248–255. DOI: 10.1109/cvprw.2009.5206848 (cited on pages 3, 14).

[33] Alina Kuznetsova et al. "The Open Images Dataset V4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale". In: *International Journal of Computer Vision* 128.7 (2020), pp. 1956–1981. DOI: 10.1007/s11263-020-01316-z (cited on pages 3, 13).

[34] Mark Everingham et al. "The pascal visual object classes (VOC) challenge". In: *International Journal of Computer Vision* 88.2 (2010), pp. 303–338. DOI: 10.1007/s11263-009-0275-4 (cited on page 3).

[35] Tsung Yi Lin et al. "Microsoft COCO: Common objects in context". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8693 LNCS.PART 5 (2014), pp. 740–755. DOI: 10.1007/978-3-319-10602-1_48 (cited on page 3).

[36] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. "Objects as Points". In: *arXiv* (2019) (cited on pages 3, 10, 19).

[37] Mingxing Tan and Quoc V. Le. "EfficientNet: Rethinking model scaling for convolutional neural networks". In: *36th International Conference on Machine Learning, ICML 2019* 2019-June (2019), pp. 10691–10700 (cited on pages 3, 20).

[38] Trabajo Fin de Máster, A Delgado, and H Kusetogullari. "Detecting and Tracking Horses Using Deep Neural Networks". In: (2018) (cited on page 3).

[39] Zahra Soleimanitaleb, Mohammad Ali Keyvanrad, and Ali Jafari. "Object tracking methods: A review". In: *2019 9th International Conference on Computer and Knowledge Engineering, ICCKE 2019* Iccke (2019), pp. 282–288. DOI: 10.1109/ICCKE48569.2019.8964761 (cited on page 3).

[40] Vijaya D. Gayki et al. "A Review of Object Detection and Tracking Methods". In: *International Journal of Advance Engineering and Research Development* 4.10 (2017), p. 1513. DOI: 10.21090/ijaerd.45913 (cited on page 3).

[41] Tristan Walter and Iain D. Couzin. *TRex, a fast multi-animal tracking system with markerless identi cation, and 2d estimation of posture and visual fields reconstruction*. 2020. DOI: 10.7554/eLife.64000 (cited on page 4).

[42] Jinlong Peng et al. "TPM: Multiple object tracking with tracklet-plane matching". In: *Pattern Recognition* 107 (2020), p. 107480. DOI: 10.1016/j.patcog.2020.107480 (cited on pages 4, 5, 37).

[43] Nieves Crasto et al. "MARS : Motion-Augmented RGB Stream for Action Recognition To cite this version : HAL Id : hal-02140558 MARS : Motion-Augmented RGB Stream for Action Recognition". In: (2019) (cited on pages 4, 29, 36).

[44] Ji Zhu et al. "Online Multi-Object Tracking with Dual Matching Attention Networks". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11209 LNCS (2018), pp. 379–396. DOI: 10.1007/978-3-030-01228-1_23 (cited on pages 4, 36).

[45] Niall McLaughlin, Jesus Martinez Del Rincon, and Paul Miller. "Video Person Re-Identification for Wide Area Tracking Based on Recurrent Neural Networks". In: *IEEE Transactions on Circuits and Systems for Video Technology* 29.9 (2019), pp. 2613–2626. DOI: 10.1109/TCSVT.2017.2736599 (cited on pages 4, 7, 36).

[46] Xinqian Gu et al. "Temporal knowledge propagation for image-to-video person re-identification". In: *Proceedings of the IEEE International Conference on Computer Vision* 2019-Octob (2019), pp. 9646–9655. DOI: 10.1109/ICCV.2019.00974 (cited on pages 4, 36).

[47] Yutian Lin et al. "Improving person re-identification by attribute and identity learning". In: *Pattern Recognition* 95 (2019), pp. 151–161. DOI: 10.1016/j.patcog.2019.06.006 (cited on pages 4, 36).

[48] Wongun Choi. "Near-online multi-target tracking with aggregated local flow descriptor". In: *Proceedings of the IEEE International Conference on Computer Vision* 2015 Inter (2015), pp. 3029–3037. DOI: 10.1109/ICCV.2015.347 (cited on page 5).

[49] Yumi Iwashita et al. "First-person animal activity recognition from egocentric videos". In: *Proceedings - International Conference on Pattern Recognition*. 2014, pp. 4310–4315. DOI: 10.1109/ICPR.2014.739 (cited on page 5).

[50] Tiphaine Jeanniard-du-Dot et al. "Accelerometers can measure total and activity-specific energy expenditures in free-ranging marine mammals only if linked to time-activity budgets". In: *Functional Ecology* 31.2 (2017), pp. 377–386. DOI: 10.1111/1365-2435.12729 (cited on page 5).

[51] Daniel Smith et al. "Behavior classification of cows fitted with motion collars: Decomposing multi-class classification into a set of binary problems". In: *Computers and Electronics in Agriculture* 131 (2016), pp. 40–50. DOI: 10.1016/j.compag.2016.10.006 (cited on page 5).

[52] Jacob W. Kamminga et al. "Synchronization between sensors and cameras in movement data labeling frameworks". In: *DATA 2019 - Proceedings of the 2nd ACM Workshop on Data Acquisition To Analysis, Part of SenSys 2019* (2019), pp. 37–39. DOI: 10.1145/3359427.3361920 (cited on page 6).

[53] Burr Settles. "From theories to queries active learning in practice". In: *Proceedings of the Workshop on Active Learning and Experimental Design* 16 (2011) (cited on page 6).

[54] Jacob W. Kamminga et al. "Horsing around—A dataset comprising horse movement". In: *Data* 4.4 (2019), pp. 1–13. DOI: 10.3390/data4040131 (cited on page 8).

[55] Jacob W. Kamminga, Nirvana Meratnia, and Paul J.M. Havinga. "Dataset: Horse movement data and analysis of its potential for activity recognition". In: *DATA 2019 - Proceedings of the 2nd ACM Workshop on Data Acquisition To Analysis, Part of SenSys 2019* (2019), pp. 22–25. DOI: 10.1145/3359427.3361908 (cited on page 8).

[56] Fisher Yu et al. "Deep Layer Aggregation". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2018), pp. 2403–2412. DOI: 10.1109/CVPR.2018.00255 (cited on pages 10, 19).

[57] Angelo Vittorio. *Toolkit to download and visualize single or multiple classes from the huge Open Images v4 dataset*. 2018. URL: https://github.com/EscVM/OIDv4_ToolKit (cited on page 13).

[58] AlexeyAB. *Yolo mark*. 2017. URL: https://github.com/AlexeyAB/Yolo_mark (cited on page 14).

[59] Qizhe Xie et al. "Self-training with noisy student improves imagenet classification". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2020), pp. 10684–10695. DOI: 10.1109/CVPR42600.2020.01070 (cited on pages 20, 35).

[60] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. "Simple online and realtime tracking with a deep association metric". In: *Proceedings - International Conference on Image Processing, ICIP* 2017-Septe (2018), pp. 3645–3649. DOI: 10.1109/ICIP.2017.8296962 (cited on pages 24, 25).

[61] Jonathon Luiten et al. "HOTA: A Higher Order Metric for Evaluating Multi-object Tracking". In: *International Journal of Computer Vision* 129.2 (2020), pp. 548–578. DOI: 10.1007/s11263-020-01375-2 (cited on page 31).

[62] Keni Bernardin and Rainer Stiefelhagen. "Evaluating multiple object tracking performance: The CLEAR MOT metrics". In: *Eurasip Journal on Image and Video Processing* 2008 (2008). DOI: 10.1155/2008/246309 (cited on page 31).

[63] Omar Elharrouss et al. "Gait recognition for person re-identification". In: *Journal of Supercomputing* 77.4 (2021), pp. 3653–3672. DOI: 10.1007/s11227-020-03409-5 (cited on page 36).

[64] Songhwai Oh, Stuart Russell, and Shankar Sastry. "Markov chain Monte Carlo data association for multi-target tracking". In: *IEEE Transactions on Automatic Control* 54.3 (2009), pp. 481–497. DOI: 10.1109/TAC.2009.2012975 (cited on page 37).

[65] Zheng Wu et al. "Coupling detection and data association for multiple object tracking". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2012), pp. 1948–1955. DOI: 10.1109/CVPR.2012.6247896 (cited on page 37).