AUTH-Sheep: An Annotated Video Dataset for Detection and Tracking of Sheep in UAV Imagery

Oliver Doll[1], Alexander Loos[2]

Audio-Visual Systems, Fraunhofer IDMT, Ehrenbergstr. 31, 98693 Ilmenau, Germany

Abstract

Object detection and tracking in drone imagery is still an open research field, especially for livestock monitoring and when detection is carried out on the drone itself. In this paper, we present the first annotated aerial video dataset of sheep, which we will make publicly available to the research community to foster further research in this field. Our AUTH-Sheep dataset consists of 4 videos with frame-accurate annotations of oriented bounding boxes and consistent track IDs per object and video. Furthermore, we developed a full detection and tracking pipeline as a baseline implementation to give other researchers a reference approach to compare their algorithms against. For this, we compared horizontal and oriented bounding box detection for the task at hand. Therefor, the YOLOv8 nano detector is utilized, which was pre-trained on a different dataset. To be able to train this detector of oriented bounding boxes, we semi-automatically created new oriented annotations for an existing dataset of sheep images.

Keywords

dataset, OBB, sheep detection, MOT

1. Introduction

Recently, unmanned aerial vehicles (UAVs) equipped with camera systems and edge computing devices have become an alternative to camera traps located on the ground as a promising tool for monitoring wild as well as livestock animals. Due to the technical possibilities of UAVs, new perspectives on monitoring are opened up. Typically, UAVs can only fly and record for several minutes, but at the same time they can cover a larger area than camera traps. In this paper, we focus on the livestock farming use-case. In particular, we consider free ranging sheep living unattended at the island of Lesvos, Greece. The goal is to develop a system for autonomous detection and tracking of sheep to enable a reliable counting and monitoring of the flock. Usually, flocks of sheep are supervised by a shepherd who is continuously present to keep track of their numbers, health and position. In the case of free-range sheep, there is no such authority and those responsible must carry out checks at regular intervals. These inspections can be difficult to carry out on terrain that is difficult to access and where visibility is limited. UAVs are suitable for overcoming these difficulties, as they are not restricted by the terrain on the ground. However, for drones to be a practical solution for the task at hand, the information obtained from UAVs must be accurate and reliable. Instead of manual inspection of the obtained video footage, recent developments in deep-learning based computer vision methods for

4th International Workshop on Camera Traps, AI, and Ecology, September 5–6, 2024, Hagenberg, Austria

oliver.doll@idmt.fraunhofer.de (O. Doll); alexander.loos@idmt.fraunhofer.de (A. Loos)

0009-0006-5968-5042 (O. Doll); 0000-0003-1920-8189 (A. Loos)

object detection and tracking paved the way for fast and accurate automatic analysis. One possible way to realize this is to stream videos from the drone to the ground and use dedicated hardware as well as large and cutting-edge deep learning models for sheep detection and tracking. Unfortunately, streaming high-quality videos in real-time from a drone to the ground is often not trivial and hardly feasible, especially in areas without suitable infrastructure. An arguably more practical approach is to integrate the necessary computer vision algorithms on the UAV itself, and only stream the resulting metadata to the ground, which requires drastically less bandwidth than streaming the video directly. However, this means that the complexity of the algorithms must be kept to a minimum, as the available computing power is limited.

In this paper, we present the first publicly available annotated dataset of aerial videos of sheep recorded at the University Farm of the Aristotle University of Thessaloniki (A.U.Th.). Our AUTH-Sheep dataset consists of 4 videos with frame-accurate ground truth annotations of oriented bounding boxes and consistent track ID per object and video. By providing such a dataset together with a baseline implementation of a full detection and tracking pipeline, we hope to stipulate further research in this field. As object detector we build upon the YOLOv8 nano model which we found to be most suitable in our previous work [1]. In our experiments, we compare the utilization of horizontal bounding boxes (HBB) and oriented bounding boxes (OBB) for detection directly on the drone. Sheep are often clustered in flocks and bounding boxes are heavily overlapping, which often introduces ambiguity during tracking. We argue that when using OBB instead of HBB the ambiguity is greatly reduced and thus more accurate results can be expected. On top of that, a state-of-the-art tracking algorithm is tested based on the obtained detections in order to be able to assign unique object IDs to the detected sheep. This allows for more accurate counting and possibly even additional traits such as animal welfare assessment.

To enable comparison of horizontal and oriented object detection, we semi-automatically have created new annotations based on the available rectangular ground truth regions for a publicly available UAV image dataset of sheep named SheepCounter [2].

The dataset and scripts will be made publicly available at https://github.com/idmt-odoll/AUTH-Sheep/.

2. Related Work

2.1. Animal Detection in Aerial Images

Despite advancements in object detection, detecting animals in UAV imagery is still a challenging task and requires accurate detection models. At the same time, energy efficient models are desired to enable implementation on edge-devices. Thus, recent trends in computer vision investigate possibilities for smaller and more efficient models which do not suffer from a significant drop in accuracy.

In [3], YOLOv4 and YOLOv5 models were compared for counting cattle at various altitudes from 20 to 100 m, with YOLOv5 being better than YOLOv4 and all models exceeding a precision of 92 %. Interestingly, the simpler YOLOv5-s model outperformed the more complex YOLOv5-m model. Wang et al. [4] enhanced the YOLOX nano model for small object detection, a common weakness of YOLO detectors, enabling detection of cattle, sheep and horse at an altitude of 300 m. They found that for increasing scale differences from training data, the detection performance decreased, but differently for all classes. For common cranes, [5] showed that automatic counting with the YOLOv3 model (99.91 % precision, 94.59 % recall) was more accurate than manual counting for RGB images at daylight. In [6], YOLOv4 outperformed YOLOv3 and SSD in detecting deer, achieving 86 % precision and 75 % recall. A different approach in [7] used a segmentation algorithm based on species-specific sRGB color profiles, achieving 100 % precision and 98.87 % recall for Arabian Oryx.

In our own previous work, we presented initial findings by comparing the performance of different state-of-the-art object detectors on publicly available UAV images of sheep [1] in order to be able to better pre-select potential object detectors for the task at hand. In this paper, we will build on our previous work, where we showed that the nano version of the YOLOv8 model series is best suited for sheep detection in aerial imagery on edge devices. It will thus be utilized throughout the experiments in this paper as well.

2.2. Multiple Object Tracking

Multiple Object detection (MOT) is the task to detect and associate objects from a specific class across a video. One approach to accomplish this is to use heuristic information such as spatial-based and appearance-based information. In our work, we focus on tracker that strongly use those spatial-based information. For short time intervals between frames, the movement of an object is likely to be small and can usually be treated as linear. Most of those works, pioneered by SORT (Simple online and realtime tracking) [8], utilize Kalman filter [9] to predict the location of the object in the new frame based on previous movement of that object. The association then is performed using the *Intersection over Union* (IoU) metric. ByteTrack improves this approach by introducing a two-stage association step [10]. In the first step, the high confidence detections are matched. A new feature is the matching of low confidence detections in the second step, which can include partially occluded and motion blurred objects. BoT-SORT builds on ByteTrack and introduces an improved Kalman filter and camera motion compensation, resulting in better predictions of the object positions in new frames [11]. OC-SORT on the other hand improves the prediction of new object positions during occlusion and non-linear movement [12]. They compute a virtual trajectory using measurements of the object detector and allow the matching with lost tracks.

3. Datasets

Two different datasets were used in this work. The SheepCounter dataset was used for training and validation of the YOLO detectors. For testing, the AUTH-Sheep dataset was used, which will be discussed in detail in section 3.2. The images of both datasets have a resolution of 3840 x 2160 pixels, while there are some images with a resolution of 4096 x 2160 pixels in SheepCounter.

3.1. SheepCounter

The SheepCounter dataset is available at roboflow and consists of 1727 images. They have green meadows as backgrounds with different lighting conditions, saturation and shadow lengths. The images are from several flights, but only selected frames were kept. Most of the sheep are white. Besides sheep, a few cows appear, but they are not annotated. The original annotations contain 55 435 instances of sheep.

We used these rectangular annotations as basis and transformed them into oriented bounding boxes to be able to train and evaluate OBB detectors. First, we utilized Microsoft's Segment Anything Model (SAM) to generate one segmentation mask per bounding box [13]. If multiple masks were generated, only the largest was kept. The image moment of the object is calculated, which allows the determination of a major and minor axis and the orientation of the objects with respect to their major axis [14]. In the next step, the found orientation is used to align the major axis with the x-axis. This allows the smallest box around the region contour and parallel to the major axis to be determined using a simple horizontal box. By reversing the alignment, the oriented bounding box was obtained.

In the next step, the new bounding boxes were manually verified using CVAT, a publicly available tool commonly used by researchers for ground truth annotation of images and videos [15]. A common problem was the existence of multiple bounding boxes for a sheep, while the original bounding boxes of other sheep were erased. Other problems were multiple animals per bounding box or shadows included as part of the animal, segmented by SAM. A few images had no annotations at all or not all animals were annotated. The new annotations include 56 681 oriented bounding boxes and also include partial sheep that appear at the edge of the image. Horizontal bounding boxes for the comparison of OBB vs. HBB algorithms were created by taking the minimum and maximum pixel positions of the oriented box in each direction. These new horizontal annotations have been created to ensure that the annotations from SheepCounter and the new AUTH-Sheep dataset have similar label quality. For AUTH-Sheep, there are no such best-fitting horizontal annotations to work with, they would have to be created from scratch. This would have been a lot of extra work on top of the oriented bounding boxes, which we wanted to avoid.

The size of objects is directly related to the altitude of the UAV. To better estimate the altitude at which the detectors can reliably detect sheep, objects are classified by their bounding box size in each frame. Inspired by the COCO dataset, five scale groups were defined and evaluated separately [16]. We found it necessary to define new groups because the area sizes for COCO were introduced for an image size of 640 x 480 pixels. The imagery used in this work has a minimum resolution of 3840 x 2160 pixels, which results in a completely different scale of objects. The five new scale groups are named *nano*, *small*, *medium*, *large* and *extended*. Objects in the nano group contain less than 64^2 pixels. The thresholds for small, medium and large objects are 96^2, 128^2 and 160^2 respectively while all objects larger than 160^2 are considered as extended. Since young animals are usually smaller than older ones, they are also categorized as correspondingly smaller objects for most of the recording altitudes, which can lead to a kind of bias. For our work, we ignore this because we only evaluate object size without paying attention to the age of animals.

	train		valid		all	
images	1203		350		1727	
instances	43 730		12 951		56 681	
	OBB	HBB	OBB	HBB	OBB	HBB
nano	2759	1772	649	538	3408	2310
small	24 416	6576	5278	1320	29 694	7896
medium	12 471	18 075	6219	4515	18 690	22 590
large	1302	11 696	356	4638	1658	16 334
extended	2782	5611	449	1940	3231	7551

Table 1
New annotations for the restructured SheepCounter dataset, broken down for the 5 new scale groups. Data are provided for oriented bounding boxes (OBB) and horizontal bounding boxes (HBB).

SheepCounter was used for training and validation, but not for testing. Also, the frames of the original videos seem to be evenly distributed among the predefined training, validation, and test sets. This leads to similar frames in all three subsets, which is not beneficial for testing the generalization capability of the model. In an attempt to correct this, the SheepCounter dataset was restructured. The restructured dataset consists of a training and validation split only. All frames were sorted into their original source videos based on their naming and content, resulting in five source videos. These videos were then manually split up into two parts. This reduces the amount of very similar samples in both subsets.

The new dataset annotations have been broken down in more detail in Table 1. As expected, the horizontal bounding boxes are more often categorized into larger groups than the oriented boxes. While for oriented boxes the most common objects are categorized as small or medium, most horizontal boxes are categorized as medium or large. Oriented boxes have an average area of 10 021 pixels, while horizontal boxes have an average area of 18 618 pixels.

3.2. AUTH-Sheep

The new dataset we present in this paper consists of four videos recorded at the University Farm of the Aristotle University of Thessaloniki (A.U.Th.). Figure 1 shows the first, middle, and last frame of each video, which gives some idea of the movement of the drone and objects. The drone was moving in all the videos, constantly changing its position and altitude, but with different patterns. Videos 1 and 2 were recorded at the same location but at different times, with goats also present in video 2. Video 3 was recorded at a different location and also the animals and the camera movement are the least dynamic of all the videos. Video 4 seems to be the most challenging recording, with the highest altitude and most clustered sheep. The combined length of all the videos in the dataset is 2:58 minutes, or 5328 frames, and contains a total of 152 837 annotated instances. The annotations consist of oriented bounding boxes and unique object IDs, which allow the evaluation of tracking algorithms. For each video, the

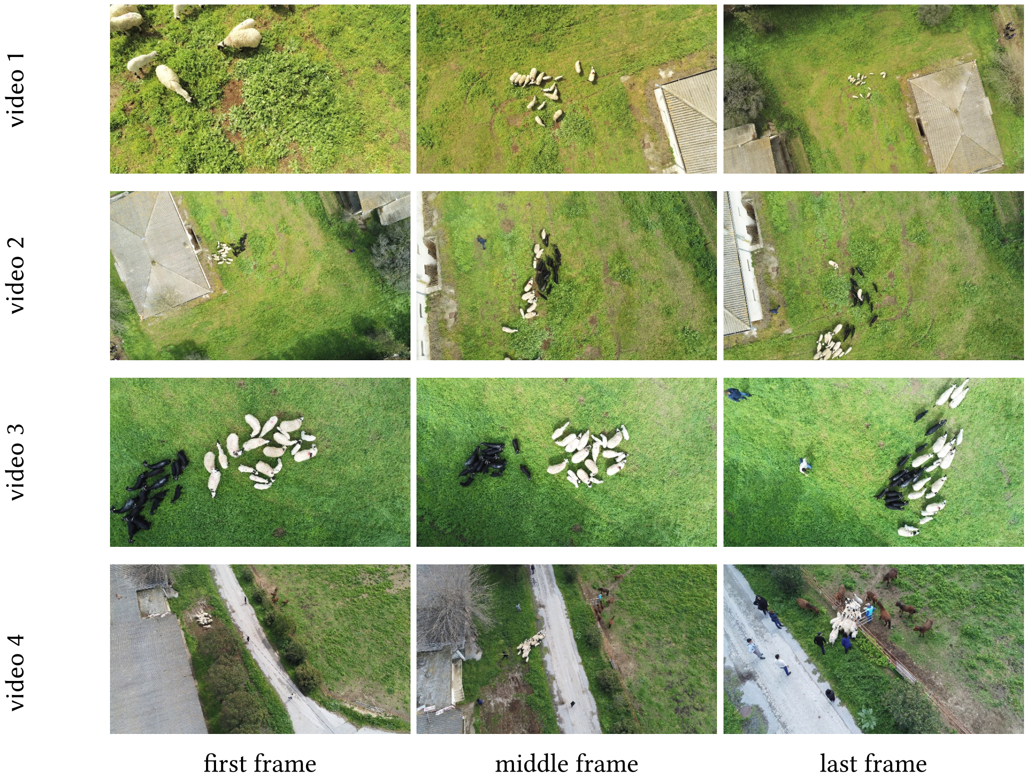

Figure 1: Sample frames from the 4 videos of AUTH-Sheep, including the first, middle and last frame of each video.

	frames	instances	sheep	human	goat	horse
video 1	1198	21 336	20 814	522	-	-
video 2	929	31 408	16 509	1638	13 261	-
video 3	1406	46 644	26 598	62	19 984	-
video 4	1795	53 449	31 612	10 402	-	11 435

Table 2
Overview of the annotations per video and class of AUTH-Sheep.

ID of an object remains the same, even if the object leaves the frame or is occluded for some time. Four different classes are annotated, namely goats, horses, humans and sheep. One thing missing is metadata for accurate information about the drone's altitude, speed, and orientation. A more detailed overview of the dataset is presented in Table 2, including the length of the videos and the number of instances for each class.

For our experiments, we focus only on the sheep class with a total of 95 533 object instances. In Table 3 the sheep instances are analyzed by size in the same way as for the SheepCounter dataset. Based on the amount of instances per scale group, it can be seen that the four videos were recorded for different flight altitudes and patterns. Video 4 has the highest amount of nano and small bounding boxes and lowest amount of medium to extended instances. It can be

	sheep per scale group				
	extended	large	medium	small	nano
video 1	6109	1925	2889	5208	4683
video 2	2	1719	5863	5579	3346
video 3	17 448	6005	2833	312	-
video 4	1	407	1171	13 545	16 488

Table 3
Distribution of the sheep annotations from AUTH-Sheep per video, with respect to the scale group.

said with a high degree of certainty that this video was recorded at the highest average flight altitude. Video 2 also has only 2 extended instances, but is more balanced in the remaining four groups than video 4. The most balanced video seems to be video 1. The lowest average altitude can be expected in video 3, where almost all instances are medium to extended.

4. Object Detection

For the detection task, two different variants of the YOLOv8-nano model were compared. The first was pre-trained on the COCO dataset [16] and predicts horizontal bounding boxes, hence this version will be referred to as the HBB model. A second variant was pre-trained on the DOTAv1 dataset [17] and predicts oriented bounding boxes, hence this variant is called OBB model. All pre-trained models used were provided by Ultralytics, whose environment was also used for the transfer learning for the task at hand.

Both models were transfer learned and validated on the restructured SheepCounter dataset described in section 3.1. The loss was monitored on the validation set until convergence. If no improvement in the mAP50-95 score was observed for the last 50 epochs, the training was stopped. The model layers were not frozen and all weights could be adjusted. For augmentations during transfer learning, the standard Ultralytics hyperparameter optimized for the COCO challenge were used. These augmentations include translation, scaling, left-right flipping, altering of the HSV color space, and erasing random portions of the image. Only the mosaic augmentation was disabled, as previous experiments showed that this improves the learning process for our use case. The batch size was set to 16 and the AdamW optimizer was used with an initial learning rate of 0.002 and a momentum of 0.9. As a post-processing step, only predictions with a confidence of 0.25 or higher were kept and non-maximum suppression was performed with an IoU threshold of 0.6.

4.1. Metrics

The main metric used was the COCO variant of the mean average precision (mAP). The mAP is the mean value of the average precision (AP) over all classes averaged over ten IoU thresholds $IoU = 0.5, 0.55, ..., 0.95$. In accordance with the Ultralytics framework [18] used for the experiments, this metric is called *mAP50-95* in the following. In addition, the mean average recall (mAR) is also used in the same version, resulting in the *mAR50-95* score. To evaluate the oriented bounding boxes, they were treated as segmentation masks. For better insight, the

model input size	pixels	usable size	percentage used	MACs (B)
640 x 640	409 600	640 x 360	56.25 %	4.53
832 x 480	399 360	832 x 468	97.50 %	4.42

Table 4
Comparison of two model input sizes with a similar amount of pixels in terms of used area when the input image has a 16:9 aspect ratio. It is assumed that the image is padded to the full model input size. MACs (in billions), as measure of computational effort, have been calculated for ONNX models.

mAP50-95 is also calculated for the five scale groups defined in section 3.1. The evaluation was performed using pycocotools, an API for the evaluation methods used for COCO.

4.2. Model Input Size

When applying the object detector on the edge, power is limited and hence should be used optimally. Typically, deep learning models expect square images as input, while the actual images are often non-square. These images are then typically padded to fit the input size of the deep learning model, which introduces unnecessary data and thus avoidable overhead.

Since we already knew that most of the training images and all the test videos had a 16:9 aspect ratio, a fixed new input size with a similar aspect ratio was calculated. There were two boundary conditions that were taken into account. First, the used YOLO model has five downsampling layers, which requires the input size to be a multiple of $2^5 = 32$. Second, the new input should not contain more pixels than the original input size of 640 x 640.

The new model input size was set to 832 x 480 pixels. Table 4 shows the theoretical comparison with the standard input size of 640 x 640. While the amount of pixels for the new input size is 2.5 % lower, the percentage of the input area used increases by 41.25 % to a total of 97.5 %. Therefore, it can be expected that there will be only minimal additional padding at the edge of the image. As expected, the computational effort, expressed in MACs (Multiply-Accumulate Operations), decreases by 2.5 %, proportional to the amount of pixels.

Comparing the actual results on the validation set, it's clear that the new model input size improves performance for both model types and for all metrics used. For the HBB model, all mAP metrics were improved by about 0.06 for all scale groups, except the nano objects. The mAR50-95 score also increased by 0.053. For the OBB model, however, the improvement is not as significant. mAR50-95 improved by 0.043 and mAP50-95 by 0.042. The largest gain was seen for small objects (0.056) and the smallest gain for extended objects (0.012). A notable result is that although the OBB model is better than the HBB model in all scale groups except medium objects, the value for mAP50-95 (all) is lower. This can be attributed to the fact that there are more small and nano objects for OBB, and the models generally perform worse on these compared to medium to large objects.

		mAP50-95 (per scale group)					
	mAR50-95	all	extended	large	medium	small	nano
HBB (640 x 640)	0.706	0.665	0.759	0.717	0.649	0.463	0.044
HBB (832 x 480)	0.759	0.724	0.817	0.781	0.711	0.527	0.055
OBB (640 x 640)	0.644	0.604	0.818	0.752	0.650	0.573	0.093
OBB (832 x 480)	0.687	0.646	0.830	0.787	0.687	0.629	0.122

Table 5
Comparison of model performance when the model input size is adjusted to match the aspect ratio of the input images, while maintaining similar computational complexity. Results are for the validation set of the restructured SheepCounter dataset.

HBB model		mAP50-95 (per scale group)					
	mAR50-95	all	extended	large	medium	small	nano
video 1	0.403	0.298	0.076	0.336	0.462	0.497	0.434
video 2	0.302	0.220	-	0.190	0.325	0.281	0.154
video 3	0.371	0.301	0.307	0.295	0.352	0.268	-
video 4	0.093	0.057	-	0.015	0.034	0.109	0.082

Table 6
Detection results of the HBB model on the AUTH-Sheep dataset. For videos 2 and 4, there are no mAP50-95 results for extended objects because there were only 1 and 2 ground truth instances, respectively. There were no nano objects in video 3.

4.3. Results on AUTH-Sheep

The AUTH-Sheep dataset is used for the final evaluation. In three cases there are no mAP50-95 results for a particular scale group because there were not enough or no ground truth instances. These cases are extended objects in videos 2 and 4 and nano objects in video 3. Table 6 shows the detection results of the HBB model for each video. Compared to the results on the validation set of SheepCounter, the model performs worse. The only exception is that nano objects are detected much more reliably in all videos than on the validation set, with an increase of 0.379 for video 1. In the same video, nano to medium objects are better detected than large and extended objects, which is completely different from the training results. Similar observations can be made for video 2, but without the nano objects. For video 3, the mAP50-95 score is the most balanced across all scale groups. For video 4, the model seems to fail completely.

The trend of results for the OBB model, shown in Table 7, is comparable to that of the HBB model. In general, the OBB model performed worse than the HBB model for all metrics on all videos, with the only exceptions being extended objects in videos 1 and 3, and also large objects in video 2. While the OBB model performed better on nano objects in training than the HBB model, the opposite is true for the test set.

There are several possible reasons why the detection performance is worse on the test set. One reason could be overfitting of the models. The YOLOv8 nano models used are quite small and the diversity of training data was limited. In addition, the AUTH-Sheep dataset is quite different

OBB model		mAP50-95 (per scale group)					
	mAR50-95	all	extended	large	medium	small	nano
video 1	0.353	0.259	0.121	0.294	0.394	0.355	0.254
video 2	0.261	0.193	-	0.277	0.294	0.179	0.103
video 3	0.327	0.272	0.318	0.225	0.210	0.139	-
video 4	0.041	0.025	-	0.010	0.015	0.031	0.030

Table 7
Detection results of the OBB model on the AUTH-Sheep dataset. For videos 2 and 4, there are no mAP50-95 results for extended objects because there were only 1 and 2 ground truth instances respectively. There were no nano objects in video 3.

from the SheepCounter dataset used fpr training and is more challenging. AUTH-Sheep is more dynamic, including new perspectives, backgrounds, classes, and scaling of objects. Sheep are more often occluded with only small parts visible, making them more difficult to detect. Another factor is that the training images almost exclusively included sheep, so the model didn't learn to discriminate sheep from other classes. One aspect to consider is that the annotations for the horizontal bounding boxes were generated from the rotated boxes. This resulted in boxes that were coarser, including more background and parts of other objects. It can be assumed that this affected the adaptability of the HBB model to the new dataset.

5. Object Tracking

For the tracking task, we used the BoT-SORT algorithm without the re-identification module. As for the detection task, the implementation of the Ultralytics framework was used, since it includes the tracking of oriented bounding boxes. While the Kalman filter was not changed, the matching algorithm and the tracklet include the rotation of the boxes. The Kalman filter uses a constant-velocity model to predict the bounding box in the next frame. Camera motion can interfere with these predictions, resulting in an incorrect location of the predicted box. The BoT-SORT includes a camera motion compensation model to counteract this problem. An optional re-identification module was not used because such a pre-trained module was not available and would have a high impact on the computational complexity anyway. A main objective of our work is an application on the edge, which demands more lightweight algorithms.

5.1. Metrics

Tracking performance is evaluated using three metrics, namely CLEAR metrics [19], IDF1 [20] and Higher-Order Tracking Accuracy (HOTA) [21]. For testing, all frames were used consecutively without skipping any frames. The evaluation tool used was the TrackEval framework [22] and all tracking results were transformed into the *mots* format [23]. The most important score of the CLEAR metrics is MOTA (multiple object tracking accuracy), which focuses more on detection performance than identity association. IDF1 focuses more on the identity association performance of the tracker, while HOTA is a metric that considers both detection and identification almost equally. In addition to these specific metrics, the number of detections, ground truth objects, associated IDs, and ground truth IDs were considered.

5.2. Results on AUTH-Sheep

The results for the tracking task are less one-sided than those for detection. Overall, the HBB model (Table 8) outperformed the OBB model (Table 9) in HOTA, MOTA, and IDF1 scores. For both models, the performance is best on video 3, followed by videos 1 and 2, and worst on video 4, which is similar to the mAP50-95 score for detection. Looking at individual videos, the OBB model performed better on video 3 and slightly better on video 4. Especially for video 3 with mostly extended and large objects, the OBB model scored high for IDF1 (92.23) and MOTA (93.40). The performance in video 4 is very low for both models on all scores, which leads to the conclusion that tracking failed completely in this case.

HBB model	HOTA	MOTA	IDF1	Dets	GT-Dets	IDs	GT-IDs
video 1	49.53	60.44	72.84	17 801	20 814	63	19
video 2	40.75	63.74	53.08	14 601	16 509	84	19
video 3	63.37	85.21	90.18	28 504	26 598	70	19
video 4	6.91	6.17	12.19	5348	31 612	98	19
combined	45.64	49.95	61.09	66 254	95 533	315	76

Table 8
Tracking results for the HBB model on the AUTH-Sheep dataset.

OBB model	HOTA	MOTA	IDF1	Dets	GT-Dets	IDs	GT-IDs
video 1	39.62	55.64	67.43	12 788	20 814	32	19
video 2	35.14	51.68	54.72	11 746	16 509	77	19
video 3	66.68	93.40	92.23	27 574	26 598	51	19
video 4	7.40	6.36	11.91	8287	31 612	134	19
combined	42.65	49.16	59.53	60 395	95 533	294	76

Table 9
Tracking results for the OBB model on the AUTH-Sheep dataset.

Comparing the performance with the distribution of sheep in different scale groups in Table 3, the results correspond to the sum of medium to extended objects per video. Video 3 has almost only medium to extended objects (98.8 %) and shows the best tracking results. At the same time, only a fraction of objects are medium to extended (5 %) in video 4, for which both models show equally poor performance. This suggests that the models are able to track sheep in UAV images when the sheep are large enough. The seemingly increased detection ability, in the form of the MOTA score, compared to the pure detection results from section 4 can be explained by a lower confidence threshold during tracking. BoT-SORT includes detections with a confidence of 0.1 or higher, while for detection the threshold was set to 0.25. Also, the MOTA and IDF1 scores were only calculated for an IoU threshold of 0.5, so the localization performance wasn't taken into account.

6. Conclusion

In this study, we presented AUTH-Sheep, the first UAV video dataset of sheep with frame-accurate annotations of oriented bounding boxes and track IDs, which we will make publicly available to the scientific community. Furthermore, we also investigated two methods for object detection and tracking of sheep on the drone itself. The primary focus was on evaluating the performance of detection and tracking when using horizontal and oriented bounding boxes. For this purpose, the YOLOv8-nano model was used and tuned for specific input sizes. Surprisingly, and against our expectations, the HBB model outperformed the OBB model for detection and tracking. While the detection performance clearly favors the HBB model, the tracking results are less clear and vary depending on the video and metric. This behavior definitely needs further investigation in future work.

The restructured SheepCounter dataset, with its new annotations for horizontal and oriented bounding boxes, significantly contributed to the training process. The manual verification step ensured the accuracy of bounding boxes and ID tracks for both datasets used. The BoT-SORT algorithm, without the re-identification module, was effective for tracking. However, the tracking performance varied significantly between videos, indicating the influence of factors such as flight altitude and flight patterns.

The limited amount of data and the inherent variations in flight altitude, lighting conditions, and object size posed significant challenges. This was evident in the performance drop when models were tested on the AUTH-Sheep dataset, which differed from the training dataset in several aspects.

To further improve the robustness and accuracy of object detection and tracking in UAV videos, we propose to increase the dataset size and diversity by including more varied environmental conditions and flight parameters to improve model generalization. Despite the computational overhead, incorporating re-identification modules could improve tracking performance, especially in scenarios with frequent occlusions and object reappearances.

In conclusion, although the study demonstrates promising results in object detection and tracking of sheep in UAV videos, there is room for improvement. Addressing the identified challenges and following the recommended future work will pave the way for more reliable and efficient systems, with broader applications in wildlife monitoring and agricultural management.

Acknowledgments

Funded by HORIZON Europe HE-2022: SPADE – 101060778 ©2023 IEEE. We thank our student worker Touseef Ashraf, who heavily contributed to the annotation of AUTH-Sheep. We also wish to extend our appreciation to Professor Bossis of Aristotle University of Thessaloniki (https://www.auth.gr/, http://www.agroctima.auth.gr/en/) and his team for organizing the first SPADE Livestock Trial and recording the videos of the presented AUTH-Sheep dataset.

References

[1] O. Doll, A. Loos, Comparison of Object Detection Algorithms for Livestock Monitoring of Sheep in UAV images, in: Camera traps, AI, and Ecology - 3rd International Workshop, Jena, 2023. doi:10.24406/publica-2164.

[2] G. Nolan, SheepCounter Dataset, 2023. URL: https://universe.roboflow.com/riisprivate/sheepcounter, visited on 2024-08-29.

[3] F. de Lima Weber, V. A. de Moraes Weber, P. H. de Moraes, E. T. Matsubara, D. M. B. Paiva, M. d. N. B. Gomes, L. O. F. de Oliveira, S. R. de Medeiros, M. I. Cagnin, Counting cattle in UAV images using convolutional neural network, Remote Sensing Applications: Society and Environment 29 (2023) 100900. doi:10.1016/j.rsase.2022.100900.

[4] Y. Wang, L. Ma, Q. Wang, N. Wang, D. Wang, X. Wang, Q. Zheng, X. Hou, G. Ouyang, A Lightweight and High Accuracy Deep Learning Method for Grassland Grazing Livestock Detection Using UAV Imagery, Remote Sensing 15 (2023) 1593. doi:10.3390/rs15061593.

[5] A. Chen, M. Jacob, G. Shoshani, M. Charter, Using Computer Vision, Image Analysis and UAVs for the Automatic Recognition and Counting of Common Cranes (Grus grus), Journal of Environmental Management 328 (2023) 116948. doi:10.1016/j.jenvman.2022.116948.

[6] K. Rančić, B. Blagojević, A. Bezdan, B. Ivošević, B. Tubić, M. Vranešević, B. Pejak, V. Crnojević, O. Marko, Animal Detection and Counting from UAV Images Using Convolutional Neural Networks, Drones 7 (2023) 179. doi:10.3390/drones7030179.

[7] M. E. De Kock, V. Pohůnek, P. Hejcmanová, Semi-automated detection of ungulates using UAV imagery and reflective spectrometry, Journal of Environmental Management 320 (2022) 115807. doi:10.1016/j.jenvman.2022.115807.

[8] A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft, Simple online and realtime tracking, in: Proceedings of the 23rd IEEE International Conference on Image Processing (ICIP), 2016, pp. 3464–3468. doi:10.1109/ICIP.2016.7533003.

[9] R. E. Kalman, et al., Contributions to the theory of optimal control, Boletin Sociedad Matematica Mexicana 5 (1960) 102–119.

[10] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, X. Wang, ByteTrack: Multi-object Tracking by Associating Every Detection Box, in: Proceedings of the European Conference on Computer Vision (ECCV), 2022, pp. 1–21. doi:10.1007/978-3-031-20047-2_1.

[11] N. Aharon, R. Orfaig, B.-Z. Bobrovsky, BoT-SORT: Robust Associations Multi-Pedestrian Tracking, arXiv preprint arXiv:2206.14651 (2022).

[12] J. Cao, J. Pang, X. Weng, R. Khirodkar, K. Kitani, Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 9686–9696.

[13] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollar, R. Girshick, Segment anything, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2023, pp. 3992–4003. doi:10.1109/ICCV51070.2023.00371.

[14] B. Jähne, Digitale Bildverarbeitung, 5. ed., Springer Berlin, Heidelberg, 2013. doi:10.1007/978-3-662-06731-4.

[15] CVAT.ai Corporation, Computer Vision Annotation Tool (CVAT), 2023. URL: https://github.com/cvat-ai/cvat.

[16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common Objects in Context, in: Proceedings of the European Conference on Computer Vision (ECCV), 2014, pp. 740–755. doi:10.1007/978-3-319-10602-1_48.

[17] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, L. Zhang, DOTA: A Large-Scale Dataset for Object Detection in Aerial Images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 3974–3983. doi:10.1109/CVPR.2018.00418.

[18] G. Jocher, A. Chaurasia, J. Qiu, YOLO by Ultralytics, 2024. URL: https://github.com/ultralytics/ultralytics.

[19] K. Bernardin, R. Stiefelhagen, Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics, EURASIP Journal on Image and Video Processing 2008 (2008) 1–10. doi:10.1155/2008/246309.

[20] E. Ristani, F. Solera, R. Zou, R. Cucchiara, C. Tomasi, Performance Measures and a Data Set for Multi-target, Multi-camera Tracking, in: Proceedings of the European Conference on Computer Vision (ECCV), 2016, pp. 17–35. doi:10.1007/978-3-319-48881-3_2.

[21] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, B. Leibe, HOTA: A Higher Order Metric for Evaluating Multi-object Tracking, International Journal of Computer Vision 129 (2021) 548–578. doi:10.1007/s11263-020-01375-2.

[22] J. Luiten, A. Hoffhues, TrackEval, https://github.com/JonathonLuiten/TrackEval, 2020.

[23] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, B. Leibe, MOTS: Multi-Object Tracking and Segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 7942–7951. doi:10.1109/CVPR.2019.00813.