WildLive: Near Real-time Visual Wildlife Tracking onboard UAVs

Nguyen Ngoc Dat[1], Tom Richardson[1], Matthew Watson[1], Kilian Meier[1],
Jenna Kline[2], Sid Reid[1], Guy Maalouf[3], Duncan Hine[1], Majid Mirmehdi[1], Tilo Burghardt[1]
[1] University of Bristol, UK [2] Ohio State University, USA [3] University of Southern Denmark, Denmark

Abstract

Live tracking of wildlife via high-resolution video processing directly onboard drones is widely unexplored and most existing solutions rely on streaming video to ground stations to support navigation. Yet, both autonomous animal-reactive flight control beyond visual line of sight and/or mission-specific individual and behaviour recognition tasks rely to some degree on this capability. In response, we introduce WildLive – a near real-time animal detection and tracking framework for high-resolution imagery running directly onboard uncrewed aerial vehicles (UAVs). The system performs multi-animal detection and tracking at 17.81 fps for HD and 7.53 fps on 4K video streams suitable for operation during higher altitude flights to minimise animal disturbance. Our system is optimised for Jetson Orin AGX onboard hardware. It integrates the efficiency of sparse optical flow tracking and mission-specific sampling with device-optimised and proven YOLO-driven object detection and segmentation techniques. Essentially, computational resource is focused onto spatio-temporal regions of high uncertainty to significantly improve UAV processing speeds. Alongside, we introduce our WildLive dataset, which comprises 200K+ annotated animal instances across 19K+ frames from 4K UAV videos collected at the Ol Pejeta Conservancy in Kenya. All frames contain ground truth bounding boxes, segmentation masks, as well as individual tracklets and tracking point trajectories. We compare our system against current object tracking approaches including OC-SORT, ByteTrack, and SORT. Our multi-animal tracking experiments with onboard hardware confirm that near real-time high-resolution wildlife tracking is possible on UAVs whilst maintaining high accuracy levels as needed for future navigational and mission-specific animal-centric operational autonomy. We publish all source code, weights, dataset, and labels for easy utilisation by the community.

1. Introduction and Motivation

Live Tracking of Animals via Drones. Multi-object tracking (MOT) [10, 40] in high-resolution video streams processed live onboard drones [32] poses significant challenges when applied to wildlife monitoring [11, 24, 33, 35] due to environmental demands, small animal resolution [37], platform motion, computational constraints, energy limitations and more. Yet, this capability plays a key role in enabling future animal-reactive and/or Beyond Visual Line of Sight (BVLOS) flight control for mission-specific individual [2, 28] and behaviour recognition [6, 7, 9, 22] – without options to involve ground control due to latency or connectivity constraints. Although first attempts [27] to build such systems exist, operation is currently limited to low resolutions without full benchmarking where datasets/code are so far not fully public[1]. While existing datasets [22, 23, 29]

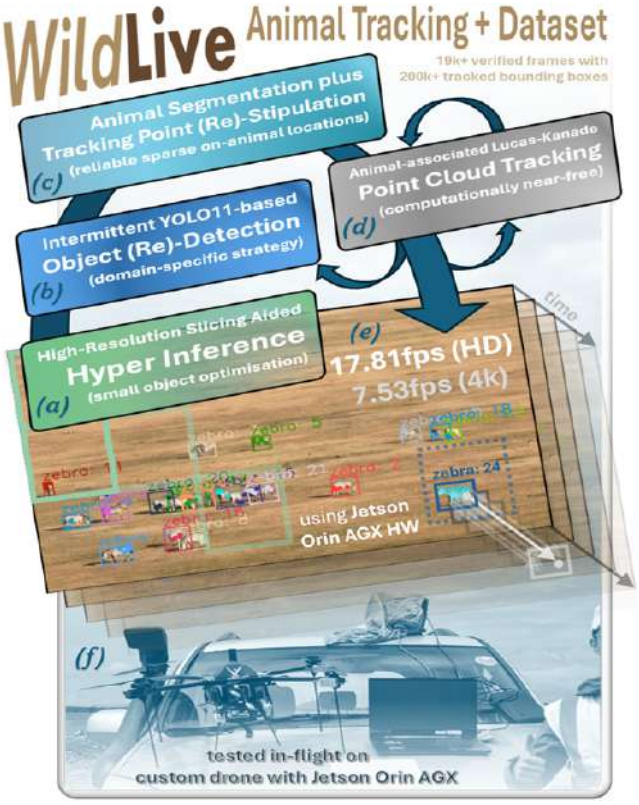

Figure 1. **WildLive System Overview.** Our pioneering approach integrates the efficiency of ***(a)*** Slicing-aided Hyper Inference with proven YOLO-driven ***(b)*** object re-detection and ***(c)*** segmentation techniques. The framework exploits animal-associated ***(d)*** inexpensive Lucas-Kanade point tracks to interpolate intermittent re-detection allowing ***(e)*** high-speed HD/4K tracking directly onboard UAVs utilising ***(f)*** custom drones with Jetson Orin AGX hardware.

[1]Parts of the source code for [27] is available at https://github.com/hardboy12/YOLOv7-DeepSORT.

have addressed multi-object tracking from aerial perspectives or in wildlife-focused contexts, they remain limited in scope or applicability. In contrast, our dataset combines high-resolution 4K wildlife video, a drone perspective and multi-object tracking supporting both the standard MOT task as well as point tracking [13, 14]. It incorporates manually verified animal segmentation masks and individual point tracks in the ground truth. This enables robust MOT benchmarking while also opening up opportunities for Track-Any-Point (TAP) research in challenging wildlife scenarios, making our dataset a valuable resource for both traditional and emerging tracking methodologies.

In response, this paper proposes, benchmarks and shares with the community the near real-time MOT WildLive system (see Fig. 1) suitable for advanced tracking of animals in high-resolution video streams during flight directly onboard UAVs with a Jetson Orin AGX.

Our main contributions in this work are:

1. We introduce and make publicly available[2] our MOT WildLive system (see Sec. 4) for near real-time wildlife tracking optimised for and deployable directly on drones that carry an embedded Jetson Orin AGX computer.
2. We benchmark WildLive against suitable SOTA systems on a domain-specific tracking dataset (see Sec. 3) which we introduce alongside our system. Amongst other rich annotations, it contains 200K+ tracked bounding boxes from representative, UAV-acquired video sequences recorded on site in Kenya under strict ethical oversight (see Ethics Statement).
3. We publish[2] our WildLive Benchmark Dataset and its ground truth information in full for reproducibility and domain-relevant comparability in this evolving field.

2. Paper Concept and Related Work

Detection vs. Tracking. Whilst localising the presence of objects in video, i.e. *detection*, requires matching complex pixel patterns over potentially large object regions, following content across frames, i.e. *tracking*, may either utilise these entire objects [4, 10, 12, 38] or alternatively follow discrete, potentially sparse locations on the objects [13, 14, 20] only. ByteTrack [38], OCSORT [8], and SORT [4] are examples of recent *full object trackers* computationally suitable for edge device deployment. On the other hand, deep trackers such as CoTracker [20] and traditional sparse optical flow trackers such as Lucas-Kanade (LK) [5] implement *point tracking* capabilities. The latter are vastly cheaper computationally, but have shortcomings regarding occlusions, aperture limitations and viewpoint changes. Given that object (re)detection may not be required every frame, this offers performance headroom to combine and balance fast LK tracking with intermittent deep detection

[2]WildLive materials, source code and links can be found at https://dat-nguyenvn.github.io/WildLive/

Figure 2. **WildLive Benchmark Dataset Overview.** 19 representative 4K frames (each sampled from a different video) showcasing the dataset's diversity regarding altitudes, environments, species, approach angles as well as view points. The top right image shows a zoomed-in example patch with ground truth annotations of animal bounding boxes, segmentations, and tracked point trajectories.

to perform light-weight, UAV-suitable wildlife tracking in high-resolution video near real-time.

Slicing Aided Hyper Inference (SAHI) and YOLO. To date, the YOLO [18, 21, 25, 34, 36] detector series have remained amongst the fastest deep object detectors of their time throughout their version history. To process high-resolution 4K images with YOLO without performance-crushing information loss due to downsampling, we apply the SAHI technique [1] in conjunction with YOLOv8 [18] or YOLO11 [21] for optimised processing speeds beyond simple window processing without loss of accuracy. Note that all performance metrics are evaluated under this same regime including tracker benchmarking for a fair comparison and maximal utilisation of video content – particularly given small animal sizes in most UAV-acquired footage.

Evaluation Metrics. We adopt MOTA (Multiple Object Tracking Accuracy) [3, 30] and IDF1 (Identification F1 Score) [3] for performance measurements. These are widely used measures in the domain and allow for broad comparability with other systems. Additionally, we will be benchmarking our system with TETA [26] in future work.

3. Dataset

The WildLive Benchmark Dataset. Our dataset contains 215,800 bounding boxes and animal segmentation masks along its 291 zebra, giraffe, and elephant tracklets, plus 84 point tracks across 22 UAV-acquired 4K video sequences, totaling 19,139 frames recorded on site at the Ol Pejeta Conservancy in Kenya. Acquisition was conducted via DJI Mavic 3 Enterprise and Pro drones plus a custom-built quadcopter for wildlife missions. Figs. 2 and 3 exemplify frames and key metadata. Notably, the SAM2 framework

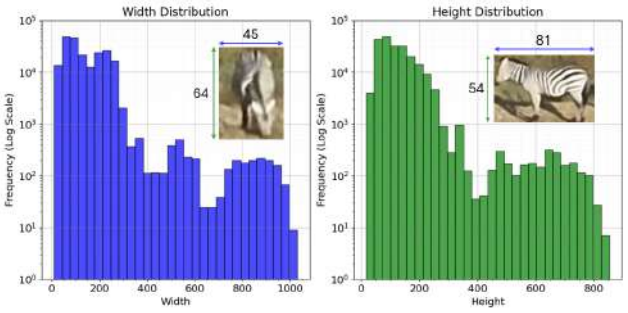

Figure 3. **Distribution of Animal Resolutions.** We show the width and height distributions of ground truth animal bounding boxes in the WildLive Benchmark dataset together with a typical animal patch (sampled from the peak). Distributions peak at about 100 pixels and tail off rapidly (note logarithmic plot scales).

is employed to generate segmentation masks by using each verified bounding box as an input prompt, after which all generated masks are synchronised to reconstruct the full frame segmentation. Overall, the dataset provides verified animal bounding boxes and tracklet IDs, as well as manually corrected, sparse LK pixel trajectories.

4. Method

Framework Overview. As summarised in Fig. 1, our WildLive framework is optimised for near real-time high-resolution video stream tracking onboard drones. It integrates SAHI sampling and YOLO instance segmentation with light-weight sparse optical flow LK tracking and YOLO instance segmentation to localise and follow animals live on UAV platforms. The framework also incorporates TensorRT optimisation to accelerate inference on Jetson Orin AGX hardware. The following sections describe the framework and provide technical details of our system.

4.1. Detection, Segmentation and Point Selection

Initialisation and Strategic Sampling. High-resolution frames are first processed via SAHI [1] using YOLO11 [21] as detection (and during tracking intialisation also segmentation) model to provide fast localisation of low-resolved animals. This full frame scan at $t = 0$ is computationally expensive and cannot operate near real-time at high resolutions given today's UAV hardware. Thus, it is used to initiate tracking, whilst later detection and corroboration only focuses on localised 640×640 windows with highest strategic update need. Biological systems focus computational resource in similar ways via foveal vision [15, 31]. After initiation, two local region categories are prioritised above others for more frequent detection window probing:

- ***Frame Edges:*** to detect new animals entering the frame.
- ***Tracked Instances:*** to validate current tracks and to detect disappearances or occlusions.

Thus, re-detections are applied to all image windows cyclically, however, at higher temporal sampling rate for above frame regions to focus computational resource.

Sparse Location Selection. Selectively processing only point locations on animals rather than performing dense, object-wide, or even frame-wide dense tracking provides order-of-magnitude faster performance to track and thereby interpolate between re-detections. N points $p_i = [(x, y), id_k]$ at locations (x, y) per animal ID id_k are derived as classic Harris corners [17] within the animal segment to be tracked where $N = 5$ provided reliable performance. We note that with $N >= 5$ the system is not sensitive to small changes in N. During re-detection validation of tracked animals, these tracked points are resampled to avoid drift, but are required to widely remain within the predicted YOLO segmentation mask to confirm object persistence (see source code for full details).

4.2. Integration with LK Tracking

Optimised Pyramidal LK Tracker. Expanding on early ideas for detector and LK tracker integration [7] and avoiding frame-by-frame filtered integration for instance via a Kalman Filter [16, 19], we use the pyramidal Lucas-Kanade [5] only as a sparse interpolator between intermittent re-detections to boost speed under a high-resolution regime. Pyramidal processing of the vector of all tracked points $P^L = [p_i]$ from the coarsest level $L = m$ to the finest resolution level $L = 0$ at logarithmic resolution scaling according to

$$\mathbf{u}^L = \frac{\mathbf{u}}{2^L}, \tag{1}$$

can effectively support rapid high-resolution LK processing of 4K+ imagery. Empirically, a value of $m = 5$ yielded fastest overall results without loss of accuracy for our dataset (see Tab. 1).

Linking Point Tracking to Animal Tracklets. After determining LK displacement vectors for tracked points, full animal bounding box shifts are predicted as average displacement of the N points associated with an animal instance. Noting that individual LK tracks may fail due to occlusions, aperture limitations, viewpoint changes and more, model drift is possible. To address this, regular re-detections and point re-initialisations on the animal segments are performed, effectively using LK tracking as a data-driven interpolator with minute computational footprint. Performing re-detection updates requires rapid matching of track IDs to re-detected animal masks.

Tracklet Lifecycle and Confidence. Let a Point-over-Area (PoA) index be defined as the proportion of tracked points p_i with id_k whose locations (x, y) fall within a segmentation mask. Matching re-detections and point clouds associated to animal tracklets based on this index together with Intersection over Union (IoU) considerations to address overlap scenarios provides rapid re-association capability between LK point propagations and YOLO re-detections (see source code for implementation detail). As a result, each YOLO re-detected object is either assigned to an existing animal tracklet or tracked forwards with a new

ID if it does not sufficiently match any existing ID. Following [7], every tracklet also carries a confidence measure, where in our WildLive system, the measure itself accumulates YOLO re-detection confidence values minus a minimal required confidence per detection over time (see [7] for full details). This implements a simple and effective temporal evidence accumulator which, via basic thresholding, controls tracklet termination as well as tracklet validation, that is accepting a tracklet as '1-confident' (otherwise '0-spurious') and labelling it as such to the user. The latter allows the system to track even 'spurious' YOLO detection candidates with low confidence in order to focus computational resources to validate or dismiss those, whilst only 'confident' tracklets are considered for experimentation.

5. Experiments

Experimental Setup. We evaluate WildLive offline on Tesla P100-PCIE-16GB GPU hardware (see Tab. 1) and pinpoint its speed on the Jetson Orin AGX onboard GPU environment (see Tab. 2 and Tab. 3). For the Tesla experiments, we utilise standard YOLO networks combined with SAHI [1] for detection and segmentation, providing baseline performance. In contrast, for Jetson deployment, we perform extensive TensorRT optimisation on these same networks to maximise inference speed and efficiency, publishing optimised network weights[2]. Jetson deployment is further facilitated via Docker containerisation, offering flexibility and scalability for community use. All experiments are conducted with the Jetson Orin AGX running in maximum performance mode. Successful test flights in Kenya (depicted in Fig. 1) constitute a physical proof-of-concept that the full WildLive system can indeed operate on a custom built, flight-tested UAV[2].

6. Results

Comparative System Benchmarks. We compare WildLive against recent SOTA *full object trackers* computationally suitable for edge device deployment, in particular ByteTrack [38], OCSORT [8], and SORT [4]. To support these trackers with detection input, we use the SAHI [1] technique rather than relying solely on a standard YOLO model. We benchmark processing speeds together with both Multi Object Tracking Accuracy (MOTA) and Identification F1 (IDF1) scores. The latter complements MOTA regarding limitations on how long trackers correctly identify objects. Full results are shown in Tab. 1 and confirm order-of-magnitude gains in processing speed compared to other tested techniques resting on only intermittent re-detection bridged by computationally negligible LK point tracking. As shown in Tab. 1 and Tab. 2, WildLive achieves a processing speed approximately 10 times faster than that of the other trackers mentioned above and attains near real-time performance (up to 17.81 fps for HD). Accuracy, maybe surprisingly, slightly improves too, leading to best MOTA at 81.17% and IDF1 at 79.03%: utilising PoA *and* IoU together for ambiguity resolution proves superior compared to the IoU-centred full object tracking methods tested which have no direct access to segmentation masks.

Tracking Method	YOLO	fps 4K	MOTA	IDF1
ByteTrack[38] (2022) + SAHI[1]	8x	0.33	65.34	60.59
	8l	0.41	65.67	63.05
	8m	0.53	63.92	58.24
	8s	0.69	62.55	57.28
	8n	0.74	62.92	57.02
	11x	0.31	72.50	68.44
	11l	0.42	67.36	61.91
	11m	0.50	66.62	60.64
	11s	0.62	66.52	60.70
	11n	0.67	62.43	55.45
OC-SORT[8] (2023) + SAHI[1]	8x	0.33	67.98	65.96
	8l	0.43	67.69	66.35
	8m	0.55	62.11	59.35
	8s	0.74	53.38	54.35
	8n	0.79	47.62	48.15
	11x	0.31	70.82	67.90
	11l	0.44	64.19	63.63
	11m	0.52	62.94	61.00
	11s	0.66	56.90	55.75
	11n	0.71	50.29	49.90
SORT[4] (2016) + SAHI[1]	8x	0.33	74.41	55.69
	8l	0.43	74.77	57.55
	8m	0.56	68.28	51.23
	8s	0.74	61.12	48.89
	8n	0.79	57.93	44.14
	11x	0.31	75.83	60.12
	11l	0.46	71.08	55.60
	11m	0.51	68.47	56.10
	11s	0.66	66.31	50.31
	11n	0.71	59.60	45.91
WildLive (Ours)	8x	4.79	76.65	75.86
	8l	5.29	78.70	**79.03**
	8m	5.60	75.51	74.18
	8s	5.78	77.15	76.46
	8n	5.68	70.29	69.31
	11x	5.72	**81.17**	79.02
	11l	5.12	81.02	78.23
	11m	5.30	77.74	78.71
	11s	5.76	75.52	74.76
	11n	**6.32**	75.07	73.45

Table 1. **Comparative WildLive System Benchmarks (on Tesla GPU).** MOT evaluation of processing speed in frames per second (fps) and MOTA/IDF1 measures (%) for one full run across the WildLive Benchmark Dataset benchmarked at full 4k resolution on a Tesla P100-PCIE-16GB GPU. Note order-of-magnitude speed advantages achieved by utilising LK tracking as temporal interpolator between re-detections. Different YOLO versions are probed, with the YOLO11n, for instance, having a 2.9M parameter resource footprint [18].

Runtime Speed Estimation. The WildLive system employs TensorRT optimisation on Jetson hardware to boost real-time performance, replacing the default PyTorch model

during deployment. We evaluate processing speed on drone hardware through two sets of benchmarks. First, we assess the inference speed of all supported detection models on both HD and 4K video streams; on HD input, for instance, WildLive achieves up to 17.81 fps with a YOLO11n detector (see Tab. 2). Secondly, we measure inference speed across a range of re-detection regimes, from our standard system (single-window re-detection per frame) at 7.53 fps down to the base case of permanent full-frame re-detection at 2.45 fps, using the fastest detection model, YOLO11n (as reported in Tab. 3). These benchmarks show that TensorRT optimisation yields further improvements in inference speed, although it roughly doubles the model size.

Yolo	fps on 4K (TRT)	fps on HD (TRT)	fps on 4K (Pytorch)	fps on HD (Pytorch)
8x	6.09	11.75	5.62	8.56
8l	6.97	14.27	6.02	11.69
8m	7.03	16.44	6.42	13.55
8s	7.21	17.76	6.80	14.66
8n	7.17	17.69	6.88	14.77
11x	5.94	11.02	5.75	8.51
11l	6.21	15.63	5.89	11.32
11m	7.08	16.89	6.31	13.33
11s	7.30	17.57	6.49	15.92
11n	**7.53**	**17.81**	**6.96**	**16.79**

Table 2. **WildLive Speed Across YOLO Versions and Video Resolutions.** Speeds are reported for both 4K and HD input resolutions on Jetson AGX Orin. The system achieves its highest performance with YOLO11n at 17.81 fps on HD data using our single-window re-detection approach and TensorRT optimisation.

Re-Detection Windows (per 4K frame time step)	**fps** (Jetson)
01 (Standard System (Ours))	**7.53**
02	6.94
04	5.87
08	4.59
16	3.28
24 (Full Frame Re-detection)	2.45

Table 3. **WildLive Speed Benchmarks for Different Re-Detection Window Configurations.** System speed in frames per second (fps) on the WildLive Benchmark Dataset, measured across different numbers of re-detection windows per 4K frame. Inference was performed using a TensorRT engine configured based on the number of re-detection windows. WildLive with permanent full frame re-detection performs at 2.45 fps, whilst single window probing (Standard System (Ours)) allows for 7.53 fps.

7. Conclusion and Future Work

We conclude that WildLive's high-resolution, high-speed tracking capabilities are close to those needed to fuel seamless, animal-reactive drone navigation based on video processing directly onboard UAVs, albeit accuracies under a small object regime still leave significant room for detection and tracking improvements. Such availability could revolutionise the way UAV conservation missions are conducted in terms of range, cost, ease of use, and mission type.

Improving Evaluation, Small Object Detection and Segmentation. To address limitations, we plan to develop a fast and efficient instance segmentation model specifically tailored for detecting and segmenting small animals, adapting it to the unique challenges of UAV-based wildlife monitoring. Additionally, we are in the process of benchmarking the system with TETA [26], testing alternative detectors including RT-DETR [39], and producing full speed-accuracy benchmarks that are statistically more robust using multiple runs and sampling widely from the WildLive parameter space, with rigorous field testing of a prototype.

Long Term Goal. Ultimately, our goal is to integrate these computer vision capabilities with UAV navigation, enabling fully autonomous, vision-driven BVLOS wildlife monitoring missions. This may allow to address currently missing conservation capabilities including autonomous monitoring of large wildlife reserves.

Acknowledgements

This work was supported by the WildDrone project (under the Marie Skłodowska-Curie Actions (MSCA) - grant agreement ID: 101071224) and UK Research and Innovation (EPSRC/UKRI - project reference: EP/X029077/1), the Imageomics Institute (NSF HDR Award 2118240), and ICICLE (NSF grant OAC-2112606).

Ethics Statement

When collecting our WildLive Benchmark Dataset and during test flights, we adhered to strict ethical standards to ensure the wellbeing of the animals and the appropriateness and integrity of the research. Our data was collected using drones in the Ol Pejeta Conservancy, Kenya. Drones were operated at safe distances to avoid causing stress or disturbance to animals. Additionally, all recordings were made in a non-invasive manner, with no direct interaction with wildlife. To further protect privacy and adhere to ethical guidelines, we ensured that human faces are not contained in recordings, focusing solely on animals and their behavior. All data collection procedures followed relevant local regulations, such as those provided by the Kenya Civil Aviation Authority (KCAA)(Authorization number KCAA/UAS/OPS/0068/2025), the Kenya Wildlife Research and Training Institute (KWRTI), and other ethical guidelines. Efforts were made to respect the natural behaviours of the animals maximally. By prioritising animal welfare, privacy, and ethical research practices, the dataset can contribute to scientific advancements while minimising any harm to the environment and its inhabitants.

References

[1] Fatih Cagatay Akyon, Sinan Onur Altinuc, and Alptekin Temizel. Slicing aided hyper inference and fine-tuning for small object detection. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 966–970, 2022. 2, 3, 4

[2] William Andrew, Colin Greatwood, and Tilo Burghardt. Aerial animal biometrics: Individual friesian cattle recovery and visual identification via an autonomous uav with onboard deep inference. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 237–243. IEEE, 2019. 1

[3] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008. 2

[4] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016. 2, 4

[5] Jean-Yves Bouguet et al. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel corporation*, 5(1-10):4, 2001. 2, 3

[6] Otto Brookes, Majid Mirmehdi, Colleen Stephens, Samuel Angedakin, Katherine Corogenes, Dervla Dowd, Paula Dieguez, Thurston C Hicks, Sorrel Jones, Kevin Lee, et al. Panaf20k: a large video dataset for wild ape detection and behaviour recognition. *International Journal of Computer Vision*, 132(8):3086–3102, 2024. 1

[7] Tilo Burghardt and Janko Calic. Real-time face detection and tracking of animals. In *2006 8th seminar on neural network applications in electrical engineering*, pages 27–32. IEEE, 2006. 1, 3, 4

[8] Jinkun Cao, Jiangmiao Pang, Xinshuo Weng, Rawal Khirodkar, and Kris Kitani. Observation-centric sort: Rethinking sort for robust multi-object tracking, 2023. 2, 4

[9] Alex Hoi Hang Chan, Prasetia Putra, Harald Schupp, Johanna Köchling, Jana Straßheim, Britta Renner, Julia Schroeder, William D Pearse, Shinichi Nakagawa, Terry Burke, et al. Yolo-behaviour: A simple, flexible framework to automatically quantify animal behaviours from videos. *Methods in Ecology and Evolution*, 2024. 1

[10] Gioele Ciaparrone, Francisco Luque Sánchez, Siham Tabik, Luigi Troiano, Roberto Tagliaferri, and Francisco Herrera. Deep learning in video multi-object tracking: A survey. *Neurocomputing*, 381:61–88, 2020. 1, 2

[11] Evangeline Corcoran, Megan Winsen, Ashlee Sudholz, and Grant Hamilton. Automated detection of wildlife using drones: Synthesis, opportunities and constraints. *Methods in Ecology and Evolution*, 12(6):1103–1114, 2021. 1

[12] Nguyen Ngoc Dat, Valerio Ponzi, Samuele Russo, and Francesco Vincelli. Supporting impaired people with a following robotic assistant by means of end-to-end visual target navigation and reinforcement learning approaches. In *ICYRIME*, 2021. 2

[13] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens, Lucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. *Advances in Neural Information Processing Systems*, 35:13610–13626, 2022. 2

[14] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10061–10072, 2023. 2

[15] Stephen Gould, Joakim Arfvidsson, Adrian Kaehler, Benjamin Sapp, Marius Messner, Gary R Bradski, Paul Baumstarck, Sukwon Chung, Andrew Y Ng, et al. Peripheral-foveal vision for real-time object recognition and tracking in video. In *Ijcai*, pages 2115–2121. Citeseer, 2007. 3

[16] Pramod R Gunjal, Bhagyashri R Gunjal, Haribhau A Shinde, Swapnil M Vanam, and Sachin S Aher. Moving object tracking using kalman filter. In *2018 International Conference On Advances in Communication and Computing Technology (ICACCT)*, pages 544–547. IEEE, 2018. 3

[17] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, pages 10–5244. Citeseer, 1988. 3

[18] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLO, 2023. 2, 4

[19] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960. 3

[20] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. *arXiv preprint arXiv:2307.07635*, 2023. 2

[21] Rahima Khanam and Muhammad Hussain. Yolov11: An overview of the key architectural enhancements, 2024. 2, 3

[22] Maksim Kholiavchenko, Jenna Kline, Maksim Kukushkin, Otto Brookes, Sam Stevens, Isla Duporge, Alec Sheets, Reshma R Babu, Namrata Banerji, Elizabeth Campolongo, et al. Deep dive into kabr: a dataset for understanding ungulate behavior from in-situ drone video. *Multimedia Tools and Applications*, pages 1–20, 2024. 1

[23] Jenna Kline, Samuel Stevens, Guy Maalouf, Camille Rondeau Saint-Jean, Dat Nguyen Ngoc, Majid Mirmehdi, David Guerin, Tilo Burghardt, Elzbieta Pastucha, Blair Costelloe, Matthew Watson, Thomas Richardson, and Ulrik Pagh Schultz Lundquist. Mmla: Multi-environment, multi-species, low-altitude aerial footage dataset, 2025. 1

[24] Jenna Kline, Alison Zhong, Kevyn Irizarry, Charles V Stewart, Christopher Stewart, Daniel I Rubenstein, and Tanya Berger-Wolf. Wildwing: An open-source, autonomous and affordable uas for animal behaviour video monitoring. *Methods in Ecology and Evolution*, 2025. 1

[25] Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, Yiduo Li, Bo Zhang, Yufei Liang, Linyuan Zhou, Xiaoming Xu, Xiangxiang Chu, Xiaoming Wei, and Xiaolin Wei. Yolov6: A single-stage object detection framework for industrial applications, 2022. 2

[26] Siyuan Li, Martin Danelljan, Henghui Ding, Thomas E Huang, and Fisher Yu. Tracking every thing in the wild.

In *European conference on computer vision*, pages 498–515. Springer, 2022. 2, 5

[27] Wei Luo, Guoqing Zhang, Quanqin Shao, Yongxiang Zhao, Dongliang Wang, Xiongyi Zhang, Ke Liu, Xiaoliang Li, Jiandong Liu, Penggang Wang, et al. An efficient visual servo tracker for herd monitoring by uav. *Scientific Reports*, 14(1):10463, 2024. 1

[28] Kilian Meier, Arthur Richards, Matthew Watson, G Maalouf, C Johnson, D Hine, and T Richardson. Wildbridge: Conservation software for animal localisation using commercial drones. In *15th annual International Micro Air Vehicle Conference and Competition*, pages 324–333, 2024. 1

[29] Hemal Naik, Junran Yang, Dipin Das, Margaret Crofoot, Akanksha Rathore, and Vivek Hari Sridhar. Bucktales: A multi-uav dataset for multi-object tracking and re-identification of wild antelopes. *Advances in Neural Information Processing Systems*, 37:81992–82009, 2024. 1

[30] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European conference on computer vision*, pages 17–35. Springer, 2016. 2

[31] Emma EM Stewart, Matteo Valsecchi, and Alexander C Schütz. A review of interactions between peripheral and foveal vision. *Journal of vision*, 20(12):2–2, 2020. 3

[32] Lifan Sun, Xinxiang Li, Zhe Yang, and Dan Gao. Visual object tracking based on the motion prediction and block search in uav videos. *Drones*, 8(6):252, 2024. 1

[33] Devis Tuia, Benjamin Kellenberger, Sara Beery, Blair R Costelloe, Silvia Zuffi, Benjamin Risse, Alexander Mathis, Mackenzie W Mathis, Frank Van Langevelde, Tilo Burghardt, et al. Perspectives in machine learning for wildlife conservation. *Nature communications*, 13(1):1–15, 2022. 1

[34] Ultralytics. ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation. https://github.com/ultralytics/yolov5.com, 2022. Accessed: 7th May, 2023. 2

[35] Jehan-Antoine Vayssade, Rémy Arquet, and Mathieu Bonneau. Automatic activity tracking of goats using drone camera. *Computers and Electronics in Agriculture*, 162:767–772, 2019. 1

[36] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, 2022. 2

[37] Mowen Xue, Theo Greenslade, Majid Mirmehdi, and Tilo Burghardt. Small or far away? exploiting deep super-resolution and altitude data for aerial animal surveillance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 509–519, 2022. 1

[38] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box, 2022. 2, 4

[39] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16965–16974, 2024. 5

[40] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Heng Fan, Qinghua Hu, and Haibin Ling. Detection and tracking meet drones challenge. *IEEE transactions on pattern analysis and machine intelligence*, 44(11):7380–7399, 2021. 1