Ecological Informatics 87 (2025) 103106

Contents lists available at ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf

BEHAVE - facilitating behaviour coding from videos with AI-detected animals

Reinoud Elhorst [a,*], Martyna Syposz [b], Katarzyna Wojczulanis-Jakubas [b]

[a] *Independent Researcher, Haarlem, Netherlands*
[b] *University of Gdańsk, Faculty of Biology, Department of Vertebrate Ecology and Zoology, Wita Stwosza 59, 80-308 Gdańsk, Poland*

ARTICLE INFO

Keywords:
Animal behaviour
Behavioural analysis
Artificial intelligence
Deep learning
Image recognition
Continuous video surveillance

ABSTRACT

Applying video recording to investigate behaviour of wild animals reduces field workload, enhances data accuracy, and minimises disturbance to animals. However, extracting information from collected video data remains a cumbersome and time-consuming task if not, at least partly, automated. Recent advancements in artificial intelligence (AI) offer automatic detection of target animals in video streams, however integrating these detections with software to annotate behaviours is missing. In addition, programs that are able to do these AI detections are often not easy to install or require specialised hardware to run. To address this gap, we introduce BEHAVE, a user-friendly, open-source, free, zero-install tool for coding animal behaviour in video recordings. BEHAVE can use the results of AI detections to skip sections of the video, can extract timestamps from video data, and supports programmable ethograms. The results are saved in a .csv file for further processing. BEHAVE includes a component that allows doing AI detections, on non-specialised hardware, also in a zero-install, user-friendly way. Due to these advantages, the behaviour coding process can be significantly accelerated, resulting in well-organised and readily exportable/importable data.

1. Introduction

Observing and examining animal behaviour provides critical insights into understanding various evolutionary processes, ecological interactions and effects of conservation actions (Dawkins, 2007). However, traditional direct observations of animal behaviour come with inherent limitations. Being dependent on a human observer they are severely limited in time and range, and thus may not detect all the desired behaviours and may be also inaccurate. In response to these challenges, researchers increasingly turn to bio-logging techniques, such as Global Positioning System (GPS), Global Location Sensing (GLS), accelerometers (Blumstein et al., 2011; Caravaggi et al., 2020; Chung et al., 2021), etc. These innovations are breaking-through solutions, but they do not capture exact animal behaviour and thus passive video recordings are still the best alternative for the human observer. However, the video material, either obtained with camera traps or continuously recording cameras, is often a large dataset that requires subsequent post-processing and analyses. One of such analyses, which is the focus of our study, is to extract records of behaviour from the video (behaviour coding) and associate that with the real time.

Recent developments in artificial intelligence (AI) with its versatile applications present a promising avenue for simplifying and accelerating data extraction (Valletta et al., 2017). Specifically, in the field of image recognition, a process of inferring the content of the image from pixel values (LeCun et al., 2015) greatly helps with identifying presence/absence and numbers of animals on image, as well as recognising species, individuals and behaviours (Carl et al., 2020; Ferreira et al., 2020; Hayes et al., 2021; Kholiavchenko et al., 2024; Norouzzadeh et al., 2018; Schofield et al., 2019). These advancements have spawned easy-to-use platforms summarised in Vélez et al. (2023), designed to facilitate ecologists' integration of AI techniques, particularly those reliant on camera-trap photos and short videos.

Despite the efficacy of camera-trap images, there are scenarios where continuous video footage is indispensable. Certain animals fail to trigger

* Corresponding author.
E-mail address: behave-app@claude.nl (R. Elhorst).

https://doi.org/10.1016/j.ecoinf.2025.103106
Received 6 September 2024; Received in revised form 13 March 2025; Accepted 14 March 2025
Available online 16 March 2025

camera-traps, and quick behaviours may be overlooked by automatically triggered video (Chan et al., 2024; Grissot et al., 2019; Hobbs and Brehme, 2017). This is particularly the case for cold-blooded animals, but for other species it is also an issue (Chan et al., 2024; Grissot et al., 2019; Hobbs and Brehme, 2017). For example, in a context of avian parental care, to establish the rate of feeding it is important to record all the nest visits of the parents while many visits, often being quite rapid, may be undetected under camera-traps. Tagging individuals with some loggers such as GLS tags or Passive Integrated Transponders (PIT-tags) does not solve the problem as these devices fail to capture additional information that may be crucial in the study context, e.g. individual interactions, quality of provisioned food, purpose of the visit (with/without food) (e.g., Ferreira et al., 2020; Grissot et al., 2023). Furthermore, for some experiments both in the wild and in the laboratory, it is necessary to collect detailed information on the situation before and/or after a focal event, to properly evaluate the focal behaviour (Kabra et al., 2013; Muheim et al., 2014; Syposz et al., 2021). Consequently, researchers adopt continuous video recording. This results in long videos with parts that could be discarded, however identifying these parts is a time-consuming task if performed manually. Employing a hybrid method, where first automatic algorithms are used to identify potentially interesting sections of the video, after which an expert does manual annotation, can save a considerable amount of time (Chan et al., 2024).

Several programs have been developed in the last two decades, to help researchers processing the video recordings with animal behaviour. Of those for simple, manual coding of behaviour (presence/absence, duration and frequency) and free of charge, the most popular are: BORIS (Friard and Gamba, 2016), Solomon Coder (Peter, 2017), JWatcher (Blumstein and Daniel, 2007) and CowLog (Hänninen and Pastell, 2009). These softwares vary in their functionality but all greatly facilitate the coding of animal behaviours. Nevertheless, they all lack the seamless integration with AI animal detection, which is particularly useful when there are a lot of time intervals on the video without animals on, so these video parts could be easily skipped, saving a lot of time during analysis. In response to such a need, some other software has been developed. A great example is DeepMeerkat (Weinstein, 2018), which is specifically designed to remove parts of the video with only background. This, however, does not provide features for behavioural annotation.

To address this gap, we introduce a comprehensive, free and open-source tool for behaviour coding, BEHAVE, which can use the output of AI detection to speed up the coding process. In addition, BEHAVE contains a component allowing users without previous knowledge in image recognition to run the AI detection process, without the need for additional hardware. This component can be used either with AI models developed for the specific domain, or with general animal detection AI models. BEHAVE greatly speeds up the process of behavioural coding, especially on long video recordings where most frames do not contain subjects of interest, and provides consistent and well organised data.

2. BEHAVE

BEHAVE (Behaviour Extraction with the Help of AI from Videos) is a web application (webapp) that can be accessed by going to https://behave.claude-apps.com/. The application can be used without registration, and runs fully on the user's computer, just like a normal (downloaded and installed) application, but then inside the browser. No internet connection is needed except to initially download the BEHAVE webapp. The source-code to the BEHAVE app is MIT-licensed and can be accessed at https://github.com/behave-app/behave. The BEHAVE website contains an elaborate manual, as well as demo files and a quick-start guide, meaning that a user can start using BEHAVE on their own videos in a number of minutes.

In order to start coding in BEHAVE, one opens a video file in BEHAVE UI. Adding a file with AI detections (created by BEHAVE Infer or a third party tool) will denote detected individuals. The detection file is optional (without it one may use BEHAVE as any other software without AI detections, Fig. 1), but having the animals detected allows one to skip parts of the video where nothing is happening, and informs the observer in advance about the presence and number of animals on the recording. BEHAVE allows an ethogram to be programmed and shared between users (in order to consistently annotate observed behaviour). BEHAVE can be fully controlled by keyboard, mouse, or a combination of the two.

BEHAVE is designed to work on common video files, such as produced by commercial cameras (.MTS and .MP4). The BEHAVE UI, though, works only on MP4 video files (this is a limitation of the browser that the BEHAVE UI runs in). Other video formats (e.g. MTS) can be quickly converted to MP4 using the convert tool inside BEHAVE. More information on supported formats can be found at https://behave.claude-apps.com/help/formats-faq.html.

Both the BEHAVE UI and the accompanying tools run 100 % on the user's computer, and no data is sent to or stored on the cloud (with the exception of anonymous statistics).

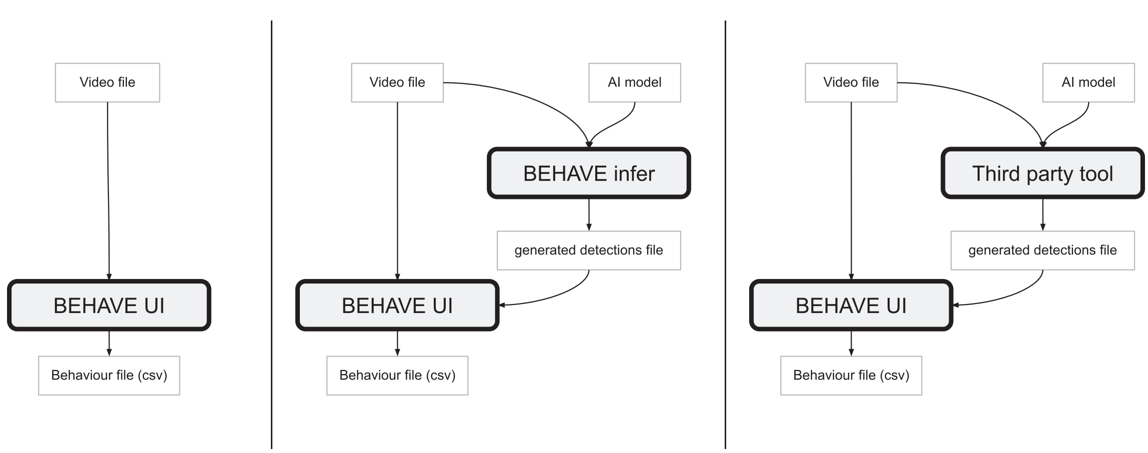


Fig. 1. BEHAVE can be run without a detection file (left), with AI detections generated through the BEHAVE infer tool (center) or with AI detections generated by some third party tool (right). Note that both BEHAVE infer as third party tools need an AI model in order to do AI detections (although the third party tool may have the AI model built in).

2.1. BEHAVE UI (behaviour coding)

The BEHAVE UI allows a user to open a video file and (optionally) a detection file. Once the files are loaded, an interface is shown with three vertical sections: video player, detections overview (if a detection file was loaded) and behaviour coding. By default the behaviour coding section starts empty (with the option to create a new behaviour file), however it's also possible to open a previously created behaviour file, easily jump through the recorded lines, and update them if needed. On the left a toolbar is shown with the major settings and configuration options.

The video player supports single-frame stepping forwards and backwards, as well as jumping to the next/previous detection. If a detection file is loaded, detected objects are colour-framed (Fig. 2). Playback speed can be adjusted with hotkeys. Under the video player is the detection bar which indicates all the frames with detections on the whole video timeline (if a detection file is loaded). The top half of this section is a zoomed area of the timeline that the user is currently viewing. If the used AI model can differentiate between various classes of detected objects (e.g. a model may support three classes: animal, human, vehicle), different colours will be shown for the specific classes (Fig. 2). Finally at the bottom, there is the behaviour annotation section, where a user can define a subject (multiple subjects may be coded simultaneously) and its behaviour using a programmable ethogram with connected hotkeys. The corresponding frame number (and timestamp, if available) will be extracted from the video and inserted automatically. Comments to the given record can be also added. The coded behaviour is saved locally on the user's computer as a simple-structured table (as shown in Fig. 3) in the form of a csv file.

The BEHAVE UI can be completely controlled with the keyboard, and each key can be associated with a subject and a point event according to the specific ethogram (programmed by the user). This configuration is exportable and importable, making it easy to share the ethogram between computers and users. Behaviour recording can also be done by mouse, or by a combination of keyboard and mouse.

2.2. BEHAVE inference (AI detection)

Especially in cases where the recorded videos contain large periods when nothing happens, behaviour coding can be sped up greatly by using AI to determine which sections of video can be skipped. BEHAVE UI allows the user to open a detection file specifying which frames contain interesting data, and this data will be integrated into the BEHAVE UI behaviour coding process. To initiate the inference, one needs an AI model (see below) and a video file. The output is a single .json that describes for each frame of the video what objects were detected. The exact format of this file is described on the BEHAVE website (https://behave.claude-apps.com/help/infer-faq.html), and output of many third-party AI detection pipelines can easily be transformed into this format.

Although many detection pipelines (which allow one to run an AI model) are available as open source, sometimes a certain amount of computer knowledge is required to install and run them, or they require specific hardware to run. In order to make AI detections available for everyone, BEHAVE includes a no-install tool (Fig. 4) to run Ultralytics YOLO (Jocher et al., 2023) AI models on video files (inference), and output a file that can be used directly in the BEHAVE UI. We have successfully used this method to create detection files using standard,

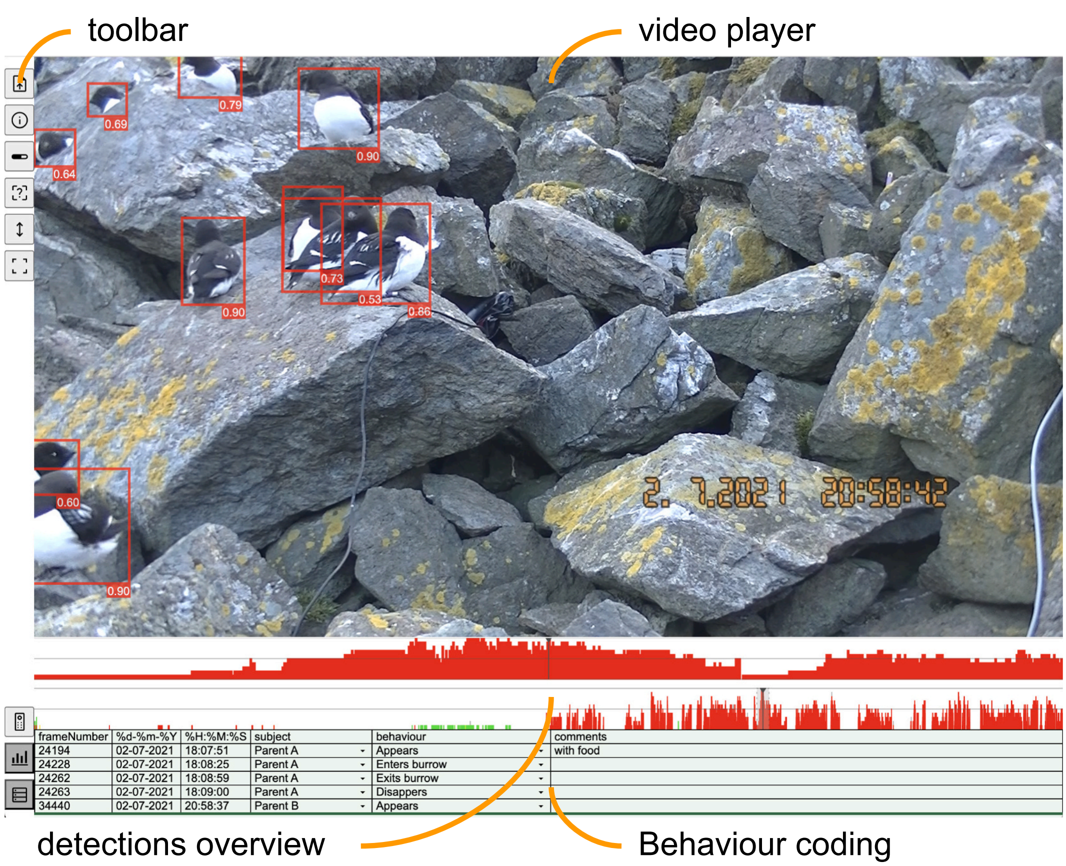
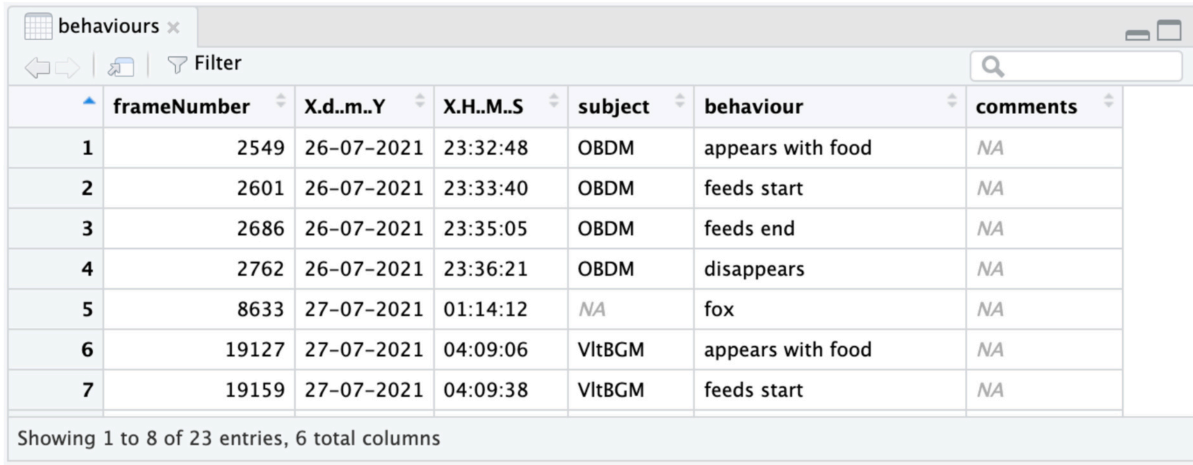


Fig. 2. BEHAVE UI showing a video frame with ten animals (little auks) with boxes and confidence of the detection as defined in the detection file. The detection overview shows in red when and how many animals were detected (higher bar indicates more subjects), and in green those moments that humans were detected. Multiple behaviours have been recorded in the behaviour recording part. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



behaviours

	frameNumber	X.d..m..Y	X.H..M..S	subject	behaviour	comments
1	2549	26-07-2021	23:32:48	OBDM	appears with food	NA
2	2601	26-07-2021	23:33:40	OBDM	feeds start	NA
3	2686	26-07-2021	23:35:05	OBDM	feeds end	NA
4	2762	26-07-2021	23:36:21	OBDM	disappears	NA
5	8633	27-07-2021	01:14:12	NA	fox	NA
6	19127	27-07-2021	04:09:06	VltBGM	appears with food	NA
7	19159	27-07-2021	04:09:38	VltBGM	feeds start	NA

Showing 1 to 8 of 23 entries, 6 total columns

Fig. 3. BEHAVE UI saves the coded behaviour in CSV format, which can be easily imported for further processing in programs such as R Studio or Excel.

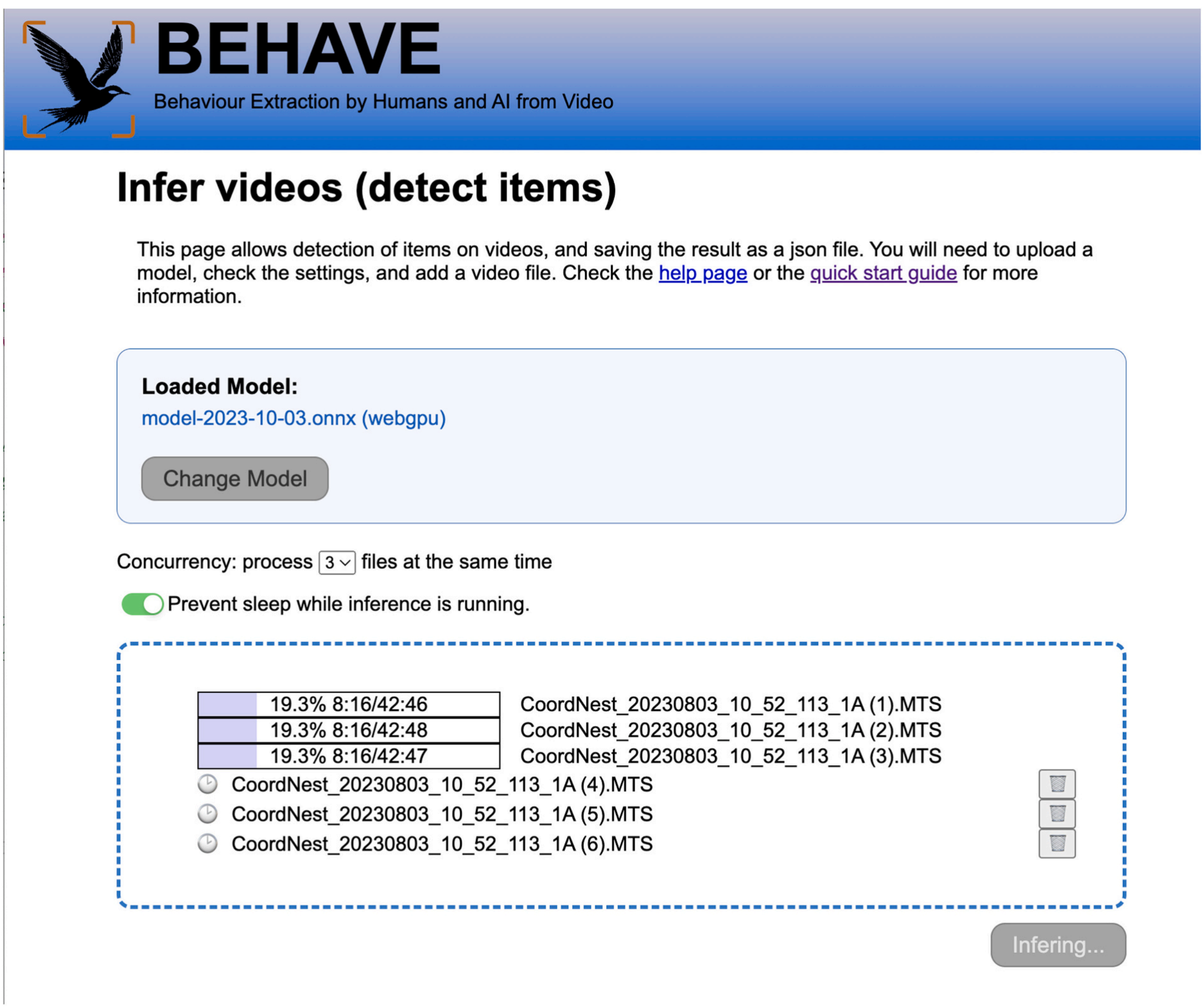


Fig. 4. BEHAVE includes a tool that can run AI models on video files, in order to detect which frames contain items of interest. This tool requires no installation, and will run directly from the browser. Multiple files can be queued so that inference can happen in the background or while the computer is not being used.

couple-of-years old, mid-range laptops.

When the infer tool is opened for the first time, it asks to load a model. The infer tool supports Ultralytics YOLO Object Detection models; although BEHAVE does not provide its own models, we have gotten good results with the freely available MegaDetectorV6 suite (Hernandez et al., 2024). Going into the performance and trade-offs between models is outside the scope of this paper, however we provide extra information on https://behave.claude-apps.com/help/model-faq.html.

Running inference on a video can take some time, depending on the video length, the chosen object detector and the hardware used. For example, half an hour of video processed (typically 45.000 frames at 25 frames/s) with a YOLOv8 nano size object detector, may take up to 6 h of inference time on older/simpler laptops, while some newer laptop models take less than 10 min. During inference, a progress bar with estimated time is shown (Fig. 4). On https://behave.claude-apps.com/help/infer-faq.html we provide more information and tips on how to speed up inference.

The inference step may be run in the background or during off-hours (e.g., overnight, over the weekend), which may be particularly useful if using an older/simpler hardware system. For this reason, this step is set up in a way that multiple files can be queued, with inference happening without user interaction (Fig. 4). There is an option to prevent the laptop going to sleep while inference runs. Inference runs fully on a laptop itself, in a browser. It makes use of modern web technologies (such as WebGPU if available), to speed up inference.

2.3. Video conversion

The BEHAVE UI needs the videos to be in MP4 format. Even though there are open source tools to convert video from different formats into MP4, we found that installing this software and finding the correct settings is non-trivial. Therefore BEHAVE includes a conversion tool that runs inside the browser, based on a version of ffmpeg (Tomar, 2006).

3. Example of use

A large part of the research of the Polar Ecology Group (University of Gdańsk, Poland) is focused on parental care in seabirds, mostly the little auk (*Alle alle*). For many research questions video footage is indispensable, providing detailed information on birds' behaviour (nest visits with/without food, interactions of focal subjects with the partner, offspring or experimental objects, etc.; e.g. Grissot et al., 2019; Kidawa et al., 2023; Syposz et al., 2024; Wojczulanis-Jakubas et al., 2022). Empirical field tests have proved camera traps to not be an appropriate solution in the study system, failing to detect rapid nest visits of focal individuals or quickly filling up with recordings. Therefore continuous time-lapsed (1 frame per second) video recordings are used, of at least 48 h each. The 48 h allow recording multiple foraging trips (one trip can last for up to 24 h (Jakubas et al., 2020)). A great part of this continuous video covers the time when there are no birds on the screen.

Over a typical field season this leads to 767 h of time-lapsed video (average over 2019–2024 seasons). Until 2023 behaviour coding on these videos was done fully by hand: someone would watch the video and whenever something happened would pause to note the exact behaviour. Since 2023 researchers have been experimenting with (early versions of) BEHAVE; no data has been collected on how long behaviour coding a single video took with and without BEHAVE, but it's obvious that if large parts of the video do not have to be watched, this results in a major speedup.

By now 991 h of video have been behaviour-coded by the team with the help of BEHAVE. In the beginning the publicly available MegaDetector v5 model (Beery et al., 2025) was used to do the AI detections, later the team switched to a custom trained model based on "YOLO v8 nano" (on https://behave.claude-apps.com/help/model-faq.html we go into detail on how to choose a model and whether one should be training their own model). It is important for the team to use a single tool for watching the video and skipping sections, as well as behaviour coding, in order to get consistent output results, which greatly facilitates later data handling and improves overall research reproducibility.

4. Conclusions and future directions

We present a valuable addition to the toolkit of animal behaviour researchers that greatly reduces time needed for data extraction from videos. The animals are detected with AI and their behaviours may be annotated by an observer using a programmable ethogram in a user-friendly viewer. This advancement contributes to the ongoing transformation of animal research into "big data" science. Based on our own research experience, we believe that wherever a user faces a need to automatically distinguish an animal from the background and further code its behaviour, BEHAVE becomes a great help. It can be thus utilised in diverse context, including analysing experimental videos, remote monitoring in inaccessible areas, and routine recordings of bird nests, coral reefs or captive animals (Fisher et al., 2016; Grissot et al., 2019; Zuerl et al., 2022).

Advances in AI development will allow ever more powerful features to be used within BEHAVE. Especially interesting will be models that use multiple video frames to detect moving animals (Jenkins et al., 2024). These models can potentially be more accurate than the current single-frame AI models. Equally interesting are AI models that can distinguish between individuals, record their characteristics and behaviours allowing for more precise selection of interesting frames or full automatisation of the behaviour recording (Bain et al., 2021; Chimento et al., 2024; Clapham et al., 2020; Ferreira et al., 2020; Hentati-Sundberg et al., 2023; Kabra et al., 2013; Norouzzadeh et al., 2018; Schindler and Steinhage, 2021).

In addition to AI improvements, BEHAVE can also be used to collect feedback on the current AI, for instance by adding buttons to indicate frames where AI detection failed, and allowing these to be collected automatically. Such data can be used to further improve the AI models.

Funding information

The study was supported by grants funded by the Polish National Science Centre (SONATINA: No 2021/40/C/NZ8/00043 to MS, OPUS: No 2023/49/B/NZ8/00370 to KWJ).

CRediT authorship contribution statement

Reinoud Elhorst: Writing – review & editing, Writing – original draft, Software, Project administration, Methodology, Conceptualization. **Martyna Syposz:** Writing – review & editing, Writing – original draft, Conceptualization. **Katarzyna Wojczulanis-Jakubas:** Writing – review & editing, Data curation, Conceptualization.

Declaration of competing interest

Authors declare no competing interest.

Acknowledgements

We would like to especially thank all the people that gave feedback on the early versions of BEHAVE: Pauline Bodson, Guillaume Coppa, Maureen Hocquet, Kristine Piening, Guillaume Verchère, and Feliksa Żurawska.

Data availability

The application is available from an online website.

References

Bain, M., Nagrani, A., Schofield, D., Berdugo, S., Bessa, J., Owen, J., Hockings, K.J., Matsuzawa, T., Hayashi, M., Biro, D., et al., 2021. Automated audiovisual behavior recognition in wild primates. Sci. Adv. 7 (46), eabi4883.

Beery, S., Morris, D., Yang, S., 2025. Efficient Pipeline for Camera Trap Image Review [Computer software]. [Computer software]. http://github.com/agentmorris/MegaDetector.

Blumstein, D.T., Daniel, J.C., 2007. Quantifying Behavior the JWatcher Way. Sinauer Associates.

Blumstein, D.T., Mennill, D.J., Clemins, P., Girod, L., Yao, K., Patricelli, G., Deppe, J.L., Krakauer, A.H., Clark, C., Cortopassi, K.A., Hanser, S.F., McCowan, B., Ali, A.M., Kirschel, A.N.G., 2011. Acoustic monitoring in terrestrial environments using microphone arrays: applications, technological considerations and prospectus. J. Appl. Ecol. 48 (3), 758–767. https://doi.org/10.1111/j.1365-2664.2011.01993.x.

Caravaggi, A., Burton, A.C., Clark, D.A., Fisher, J.T., Grass, A., Green, S., Hobaiter, C., Hofmeester, T.R., Kalan, A.K., Rabaiotti, D., Rivet, D., 2020. A review of factors to consider when using camera traps to study animal behavior to inform wildlife ecology and conservation. Conserv. Sci. Pract. 2 (8), e239. https://doi.org/10.1111/csp2.239.

Carl, C., Schönfeld, F., Profft, I., Klamm, A., Landgraf, D., 2020. Automated detection of European wild mammal species in camera trap images with an existing and pre-trained computer vision model. Eur. J. Wildl. Res. 66 (4), 62. https://doi.org/10.1007/s10344-020-01404-y.

Chan, A.H.H., Liu, J., Burke, T., Pearse, W.D., Schroeder, J., 2024. Comparison of manual, machine learning, and hybrid methods for video annotation to extract parental care data. J. Avian Biol. 2024 (3–4), e03167. https://doi.org/10.1111/jav.03167.

Chimento, M., Chan, A.H.H., Aplin, L.M., Kano, F., 2024. Peering into the world of wild passerines with 3D-SOCS: synchronized video capture and posture estimation. bioRxiv, 2024.06.30.601375.

Chung, H., Lee, J., Lee, W.Y., 2021. A review: marine bio-logging of animal behaviour and ocean environments. Ocean Sci. J. 56 (2), 117–131. https://doi.org/10.1007/s12601-021-00015-1.

Clapham, M., Miller, E., Nguyen, M., Darimont, C.T., 2020. Automated facial recognition for wildlife that lack unique markings: a deep learning approach for brown bears. Ecol. Evol. 10 (23), 12883–12892.

Dawkins, M.S., 2007. 1The power of observation. In: Stamp Dawkins, M. (Ed.), Observing Animal Behaviour: Design and Analysis of Quantitative Data. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780198569350.003.0001.

Ferreira, A.C., Silva, L.R., Renna, F., Brandl, H.B., Renoult, J.P., Farine, D.R., Covas, R., Doutrelant, C., 2020. Deep learning-based methods for individual recognition in small birds. Methods Ecol. Evol. 11 (9), 1072–1085. https://doi.org/10.1111/2041-210X.13436.

Fisher, R.B., Shao, K.-T., Chen-Burger, Y.-H., 2016. Overview of the Fish4Knowledge project. In: Fisher, R.B., Chen-Burger, Y.-H., Giordano, D., Hardman, L., Lin, F.-P. (Eds.), Fish4Knowledge: Collecting and Analyzing Massive Coral Reef Fish Video Data. Springer International Publishing, pp. 1–17. https://doi.org/10.1007/978-3-319-30208-9_1.

Friard, O., Gamba, M., 2016. BORIS: a free, versatile open-source event-logging software for video/audio coding and live observations. Methods Ecol. Evol. 7 (11), 1325–1330. https://doi.org/10.1111/2041-210X.12584.

Grissot, A., Araya-Salas, M., Jakubas, D., Kidawa, D., Boehnke, R., Błachowiak-Samołyk, K., Wojczulanis-Jakubas, K., 2019. Parental coordination of Chick provisioning in a Planktivorous Arctic seabird under divergent conditions on foraging grounds. Front. Ecol. Evol. 7. https://doi.org/10.3389/fevo.2019.00349.

Grissot, A., Borrel, C., Devogel, M., Altmeyer, L., Johansen, M.K., Strøm, H., Wojczulanis-Jakubas, K., 2023. Use of geolocators for investigating breeding ecology of a rock crevice-nesting seabird: method validation and impact assessment. Ecol. Evol. 13 (3), e9846. https://doi.org/10.1002/ece3.9846.

Hänninen, L., Pastell, M., 2009. CowLog: open-source software for coding behaviors from digital video. Behav. Res. Methods 41 (2), 472–476. https://doi.org/10.3758/BRM.41.2.472.

Hayes, M.C., Gray, P.C., Harris, G., Sedgwick, W.C., Crawford, V.D., Chazal, N., Crofts, S., Johnston, D.W., 2021. Drones and deep learning produce accurate and efficient monitoring of large-scale seabird colonies. Ornithol. Appl. 123 (3), duab022. https://doi.org/10.1093/ornithapp/duab022.

Hentati-Sundberg, J., Olin, A.B., Reddy, S., Berglund, P.-A., Svensson, E., Reddy, M., Kasarareni, S., Carlsen, A.A., Hanes, M., Kad, S., Olsson, O., 2023. Seabird surveillance: combining CCTV and artificial intelligence for monitoring and research. Remote Sens. Ecol. Conserv. 9 (4), 568–581. https://doi.org/10.1002/rse2.329.

Hernandez, A., Miao, Z., Vargas, L., Beery, S., Dodhia, R., Lavista, J., 2024. Pytorch-Wildlife: A Collaborative Deep Learning Framework for Conservation arXiv: 2405.12930.

Hobbs, M., Brehme, C., 2017. An improved camera trap for amphibians, reptiles, small mammals, and large invertebrates. PLoS One 12, e0185026. https://doi.org/10.1371/journal.pone.0185026.

Jakubas, D., Wojczulanis-Jakubas, K., Iliszko, L.M., Kidawa, D., Boehnke, R., Blachowiak-Samolyk, K., Stempniewicz, L., 2020. Flexibility of little auks foraging in various oceanographic features in a changing Arctic. Sci. Rep. 10 (1). https://doi.org/10.1038/s41598-020-65210-x.

Jenkins, M., Franklin, K.A., Nicoll, M.A.C., Cole, N.C., Ruhomaun, K., Tatayah, V., Mackiewicz, M., 2024. Improving object detection for time-lapse imagery using temporal features in wildlife monitoring. Sensors 24 (24). https://doi.org/10.3390/s24248002.

Jocher, G., Qiu, J., Chaurasia, A., 2023. Ultralytics YOLO (Version 8.0.0) [Computer software] [Computer software]. https://github.com/ultralytics/ultralytics.

Kabra, M., Robie, A.A., Rivera-Alba, M., Branson, S., Branson, K., 2013. JAABA: interactive machine learning for automatic annotation of animal behavior. Nat. Methods 10 (1), 64–67. https://doi.org/10.1038/nmeth.2281.

Kholiavchenko, M., Kline, J., Ramirez, M., Stevens, S., Sheets, A., Babu, R., Banerji, N., Campolongo, E., Thompson, M., Van Tiel, N., 2024. KABR: In-situ Dataset for Kenyan Animal Behavior Recognition from Drone Videos, pp. 31–40.

Kidawa, D., Wojczulanis-Jakubas, K., Jakubas, D., Palme, R., Barcikowski, M., 2023. Mine or my neighbours' offspring: an experimental study on parental discrimination of offspring in a colonial seabird, the little auk *Alle alle*. Sci. Rep. 13 (1), 15088.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444. https://doi.org/10.1038/nature14539.

Muheim, R., Henshaw, I., Sjöberg, S., Deutschlander, M.E., 2014. BirdOriTrack: a new video-tracking program for orientation research with migratory birds. J. Field Ornithol. 85 (1), 91–105. https://doi.org/10.1111/jofo.12053.

Norouzzadeh, M.S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M.S., Packer, C., Clune, J., 2018. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. Proc. Natl. Acad. Sci. 115 (25). https://doi.org/10.1073/pnas.1719367115.

Peter, A., 2017. Solomon Coder (Version Beta: 17.03.22): A Simple Solution for Behaviour Coding [Computer software]. solomoncoder.com.

Schindler, F., Steinhage, V., 2021. Identification of animals and recognition of their actions in wildlife videos using deep learning techniques. Eco. Inform. 61, 101215.

Schofield, D., Nagrani, A., Zisserman, A., Hayashi, M., Matsuzawa, T., Biro, D., Carvalho, S., 2019. Chimpanzee face recognition from videos in the wild using deep learning. Sci. Adv. 5 (9), eaaw0736.

Syposz, M., Padget, O., Willis, J., Van Doren, B.M., Gillies, N., Fayet, A.L., Wood, M.J., Alejo, A., Guilford, T., 2021. Avoidance of different durations, colours and intensities of artificial light by adult seabirds. Sci. Rep. 11 (1), 18941. https://doi.org/10.1038/s41598-021-97986-x.

Syposz, M., Devogel, M., Grissot, A., Jakubas, D., Wojczulanis-Jakubas, K., 2024. The first evidence of alloparental feeding in a crevice-nesting seabird, the little auk. Ecol. Evol. 14 (4), e11188.

Tomar, S., 2006. Converting video formats with FFmpeg. Linux J. 2006 (146), 10.

Valletta, J.J., Torney, C., Kings, M., Thornton, A., Madden, J., 2017. Applications of machine learning in animal behaviour studies. Anim. Behav. 124, 203–220. https://doi.org/10.1016/j.anbehav.2016.12.005.

Vélez, J., McShea, W., Shamon, H., Castiblanco-Camacho, P.J., Tabak, M.A., Chalmers, C., Fergus, P., Fieberg, J., 2023. An evaluation of platforms for processing camera-trap data using artificial intelligence. Methods Ecol. Evol. 14 (2), 459–477. https://doi.org/10.1111/2041-210X.14044.

Weinstein, B.G., 2018. Scene-specific convolutional neural networks for video-based biodiversity detection. Methods Ecol. Evol. 9 (6), 1435–1441. https://doi.org/10.1111/2041-210X.13011.

Wojczulanis-Jakubas, K., Grissot, A., Devogel, M., Altmeyer, L., Fujisaki, T., Jakubas, D., Kidawa, D., Karnovsky, N., 2022. Post-foraging in-colony behaviour of a central-place foraging seabird. Sci. Rep. 12 (1), 12981.

Zuerl, M., Stoll, P., Brehm, I., Raab, R., Zanca, D., Kabri, S., Happold, J., Nille, H., Prechtel, K., Wuensch, S., Krause, M., Seegerer, S., von Fersen, L., Eskofier, B., 2022. Automated video-based analysis framework for behavior monitoring of individual animals in zoos using deep learning—a study on polar bears. Animals 12 (6). https://doi.org/10.3390/ani12060692.