Multi-Object Tracking in Heterogeneous environments (MOTHe) for animal space-use studies

Akanksha Rathore, Ananth Sharma, Nitika Sharma, Vishwesha Guttal

January 9, 2020

Abstract

1. Videographic observations of animals are important for studying many ecological phenomena such as collective movement, space use patterns of animals and animal census. They provide us with behavioural data at scales and resolutions which are not possible with manual observations. However, extracting data from these high-resolution videos is challenging, specially in the natural settings due to heterogeneity in the habitat.
2. We present an open-source end-to-end pipeline called *Multi-Object Tracking in Heterogenous environments (MOTHe)*, a python based repository that uses convolutional neural network for object detection. MOTHe allows researchers with minimal coding experience to track multiple animals in the natural settings. It identifies animals even when individuals are stationary or camouflaged with the background.
3. MOTHe has a command-line-based interface with one command for each action, for example, finding animals in an image and tracking each individual. Parameters used by the algorithm are well described in a configuration file along with example values for different types of tracking scenario. MOTHe doesn't require any sophisticated infrastructure and can be run on basic desktop computing units.
4. We demonstrate MOTHe on six video clips from two species in their natural habitat - wasp groups on their natural nests and antelope herds in four different type of habitats. Maximum group size in example videos for wasp colonies is 12 and for blackbuck herds is 156; we were able to detect and track all the individuals in these videos. MOTHe's computing time on a personal computer with 4 GB RAM and i5 processor is 5 minutes for a 30 second long ultra-HD (4K resolution) video at 30 FPS.
5. MOTHe is available as an open-source repository with detailed user guide and demonstrations via Github.
   Link: https://github.com/aakanksharathore/MOTHe

1 Introduction

Video-recording of animals is increasingly becoming a norm in behavioural studies of space-use patterns, animal movement and group dynamics [1, 2]. Videography

based methods are getting traction even for animal census [3, 4, 5]. This mode of observation can help us gather high-resolution spatio-temporal data at unprecedented detail and thus aid in answering a novel set of questions that were previously difficult to address. For example, we can obtain movement trajectories of animals to describe space use patterns of animals, to infer fine-scale interactions between individuals within groups and to investigate how these local interactions scale to emergent properties of groups [6, 7, 8, 9, 10, 11, 12]. To address these questions, as a first step, videos need to be converted into data - typically in the form of positions and trajectories of animals. Manually extracting this information from videos can be time consuming, tedious and sometimes practically not feasible at all. Therefore, increasingly, automated tools are being developed to detect and track animals [13, 14, 15, 16, 17, 18]. However, tracking animals from videos recorded in natural settings poses many challenges which remain unresolved.

Some of the challenges that arise in natural settings, for example, are the variability in lighting conditions, vibrations in the camera, disappearance and appearance of animals across video frames, and finally, heterogeneous backgrounds in which animals are moving. Under such conditions, existing tools that rely on traditional computer vision techniques – specifically, image subtraction, colour thresholding, feature mapping, etc. – don't fare well. In the image subtraction method [19], motion of the individuals is tracked based on the differences between pixel values of two consecutive frames; this method is prone to false-detection either when camera shakes or objects other than animals of interest (e.g. grass) move. Color thresholding method [20], on the other hand, segments out animals in images on the basis of how animals differ from their background in terms of their colour. For this method to be efficient, a consistent color/intensity difference between animals and their background is necessary; however, this is seldom true in natural settings with heterogeneous backgrounds because of variability in lighting conditions over both space and time. Likewise, manual feature (e.g. shape, orientation, edges, etc) extraction - which can be considered a generalisation of the colour thresholding - too requires consistent attributes of animals in relation to their background; consequently, this method is also likely to fail in the wild. Therefore, many popular object detection tools in ecology that use the above computer vision algorithms, although efficient for videos taken under controlled conditions, are likely to fail to detect or track animals in natural settings [21, 16].

To resolve this problem, we adopt the so called machine learning techniques. One machine learning technique found to be efficient in solving **detection** problems in heterogeneous backgrounds is the *Convolutional Neural Networks* (CNN) [22, 23, 24]. Neural network based algorithms are designed based on the principles of how neurons in the brain process inputs from the environment and produce an output in terms of a behaviour/brain function. CNNS are used to classify (assign categories) to an image. In the context of object detection, parts of an image are passed to the network and network assigns a category to this image. For classification task (assigning a category), the network uses a training dataset to learn how to classify images (i.e. sets of input pixels) to different types of output categories (e.g. animals, background, other objects of interest). The trained neural network will then be able to classify new images. Despite the promise offered by CNN based

algorithms for object detection in heterogeneous environments, only a few adaptations of them are available in the context of animal tracking [3, 25]. Some of these recent usage of CNN architectures - specifically R-CNN and YOLO - often require high performance computing units such as high-end CPUs and GPUs for such adaptations. Additionally, implementation often requires reasonable proficiency in computer programming together with a great amount of customization. All of these can be limiting factors for many researchers in ecology. Hence, there is a need for an easily customizable end-to-end package that automates the task of object detection and is usable even on simple desktop machines.

Here, we provide an open-source package, *Multi-Object Tracking in Heterogenous environment* (MOTHe), which is easy to customize for different datasets and can run on relatively basic desktop units. MOTHe can detect and track multiple individuals in heterogeneous backgrounds. It uses a small CNN architecture, thus making it fast to train the network even on relatively unsophisticated desktop computing units. The network can then be used on new images to detect animals. The code generates individual tracks using Kalman filter. MOTHe is an end-to-end pipeline which automates each step including the training data generation, detection and tracking. In this paper, we have implemented MOTHe on six video clips from two species (wasps on the nests and antelope herds in four different types of habitats). These videos were recorded in natural and semi-natural settings having background heterogeneity and varying lighting conditions. We also provide an open to use GitHub repository (link) along with a detailed user guide for the implementation.

2 Working principle & features

MOTHe is a python based repository and it uses Convolutional Neural Network (CNN) architecture for the object detection task. In the context of tracking multiple animals in the videos, an object detection task would mean that we need to identify the locations and the category of animals present in an image. MOTHe works for 2-category classification e.g. animal and background. CNNs are specific types of neural network algorithms designed for image classification (assigning a category to an image). It takes a digital image as an input and reads its features to assign a category. These algorithms are learning algorithms which means that they extract features from the images by using huge amounts of labeled training data. Once the CNN models are trained, these models can be used to classify new data (images).

In this section, we present a broad overview of features and principles on which MOTHe works. Details of all the user-inputs and guidelines to run and customize the modules are available in a user manual on the Github repository and also in the supplementary material. MOTHe's network architecture and parameters are fixed to make it user friendly for beginners. However, advanced users can modify these parameters and tweak the architecture in the code files.

MOTHe can automate all the tasks associated with object detection and is divided into four independent modules (see Figure 1):

(i) **Training data-set generation** - The dataset generation is a crucial step towards object detection and tracking. The manual effort required to generate the required amount of training data is huge. The data generation executable file automates the process by allowing the user to crop the regions of interest by simple clicks over a GUI and saves the images in appropriate folders. Users run the command line code to extract images for the two categories i.e. animal and background. On each run, the user can input the category for which the data will be generated and specify the video from which images will be cropped. Program will display the frames from the videos with a click functionality, user can click on the animals to generate samples. Output from this module is saved in two separate folders which contain multiple images of animal (yes) and background (no).

(ii) **Network training** - Network training module is used to create the network and train it using the data-set generated in the previous step. The user runs a command-line script to perform the training. Once the training is complete, the training accuracy is displayed and the trained model (classifier) is saved in the repository. The accuracy of the classifier is dependent on how well the network is trained, which in turn depends on the quality and quantity of training data (see section "How much training data do I need?" in the Supplementary Materials). The various tuning parameters of the network, for e.g; number of nodes, size of nodes, convolutional layers etc., are fixed to render the process easy for the user.

(iii) **Object detection** - This is the most crucial module in the repository. It performs two key tasks - it first identifies the regions in the image which can potentially have animals, this is called localisation; then it performs classification on the cropped regions. Localisation is performed using thresholding approach, color thresholding will output the pixels which are animals along with noise (i.e. many locations in the background). Then the classification at each location is done using classifier generated in the previous module. Output is in the form of *.csv* files which contains the locations of the identified animals in each frame.

(iv) **Track linking** - Animal tracking is the final goal of the MOTHe. This module assigns unique IDs to the detected individuals and generates their trajectories. We have separated detection and tracking modules, so that it can also be used by someone interested only in the count data (eg. surveys). This modularisation also provides flexibility of using more sophisticated tracking algorithms to the experienced users. We use an existing code for the tracking task (from the Github page of Colin Torney - uavTracker/tracking/yolo_tracker.py). This algorithm uses Kalman filter and Hungarian algorithm. This script can be run once the detection are generated in the previous step. Output is a *.csv* file which contains individual IDs and locations in each frame. A video output with the unique IDs on each individual is also generated.

There are two features which make MOTHe fast to train and to run on new videos - localisation using color thresholding approach and a compact CNN architecture. To perform the detection task, we first need to identify the areas in an image where the object can be found, this is called localization or region proposal. Then we classify these regions into different categories (eg whether an animal or background?). In

MOTHe, we use threshold on the color/intensity channels of the image to identify key-points where an animal may be located. This step reduces the computation time compared to a sliding window approach. To further reduce the computation time, we have used a compact architecture with only six convolutional layers. A trade-off of above two approaches is reduced generality of the trained model across different types of data-set. To deal with this drawback, we provide options to change parameters for different data-sets so that the network retains its accuracy for a specific detection task. We demonstrate our software pipeline on two very different video data-sets which are explained in the next section.

3 Implementation on the example videos

We selected two types of videos which were being used for behavioural ecological studies in our labs. These data-sets represent varying complexity in terms of the environment (natural and semi-natural settings), background, animal speed, behaviour and overlaps between individuals (Figure 2). Below, we provide a description of these data-sets.

3.1 Nest space-use by wasps

We video-recorded the social and spatial interactions of nestmates within the colonies of a tropical paper wasp *Ropalidia marginata*. The nests of these wasps are made of paper which offers a low contrast to the dark-bodied social insects on the nest surface. Additionally, the nest comprises of cells wherein various stages of brood are housed and thus add to the heterogeneity in the background of the adults being tracked. Like most social insect nests, nests of *Ropalidia marginata* are the hubs for social interactions between mobile adults as well as between adults and immobile brood. The observation of *Ropalidia marginata* in a semi-natural condition wherein the individuals were allowed to forage freely and were not separated from their natural nests, allowed behavioural studies in a low manipulation context.

Our videos are from different nest colonies which differ in the age and hence there is some variation in wasps appearance across videos. The size of wasps is around 1 cm (in these videos clips it is around 150x150 pixels) and number of individuals ranges from 7-15 individuals. For demonstration of MOTHe performance, we selected two 30 seconds long clips (of two different colonies) from this study system.

3.2 Collective behaviour of blackbuck herds

We recorded blackbuck (*Antilope cervicapra*) group behaviour in their natural habitat. Blackbuck stay in dynamic herds exhibiting frequent merge-split events. These herds consist of adult males & females, sub-adults and juveniles. They are sexually dimorphic and adult males' appearance (colour) also changes with testosterone levels. This variation in their appearance makes it difficult to use color segmentation based techniques to detect them. Another level of complexity in this system arises from heterogeneous environment, comprising of semi-arid grasslands with patches of trees and shrubs. Many individuals don't move much across frames and there is a lot of movement of the background grasses and shrubs. This makes it difficult to

implement image subtraction techniques.

We recorded their movement in different habitat patches - grasslands, shrublands and marshy mudland. These recordings were done using DJI quad-copter equipped with high resolution camera (4K resolution at 30 FPS). The size of blackbuck is 120 cms from head to tail (in these videos it is around 35x35 pixels) and number of individuals ranges from 30-300 individuals. We used four clips (30 seconds long) from different habitats to test MOTHe.

3.3 Parametrization (User input)

For the ease of use, we have kept the parametrization process minimal. Therefore, the various tuning parameters of the network architecture are fixed. However, advanced users may be able to change the parameters in the code itself to customise it for more sophisticated tasks. The only step which requires some amount of parameter scanning by user is *choosing the color thresholds*. As mentioned earlier, to make the MOTHe fast, we use *color thresholding* approach as the localisation step. For this technique, user needs to input the values of minimum and maximum threshold on the pixel values (to be edited in *config.yml* file). The values for blackbuck and wasp data are mentioned in the Github help page and Supplementary Materials. To choose the values for a new data-set, we provide detailed instruction under the section **"Choosing color threshold"** in the Supplementary Materials. All other user inputs are straight forward and described in the instructions, these inputs do not require any parameter scanning.

3.4 Performance

We demonstrate MOTHe on two clips (representing different colonies) of wasp videos and four clips (representing 4 types of habitat) of blackbuck videos (Figure 2). In Figure 1, column (A) shows the results of running object detection on these video clips and column (B) displays the results after implementing track linking on the detection. We also quantify the performance in terms of missed detection and false detection for both the data-sets (Table 1). All video clips are 30 seconds in duration and missed and false detection were noted in each frame, results shown in the table are averaged over the whole clip. Even if some animals were missed in particular frames, they were later detected in the subsequent frames. So, all the wasps and blackbuck present in video clips were detected by MOTHe. The maximum number of individuals present in these test videos are 156 for blackbuck and 12 for wasps. Individual IDs were also intact throughout the videos, even if the IDs were missed for a few frames they got retrieved in the subsequent frames (Supplementary for tracked videos). We also show the time taken to run detection on these video clips (Table 1). On an average MOTHe's computational efficiency is 1 frame per second on an ordinary laptop (4 GB RAM i5 processor). This efficiency can be improved considerably by running MOTHe on servers (or Google colab) and/or GPUs.

Video	**Group size**	**% detected**	**% missed**	**% false positives**	**Run time**	**Habitat**
Blackbuck1	28	89.3	10.7	14.2	00:07:32	Patchy grass
Blackbuck2	78	83.1	16.9	0	00:18:11	Grass
Blackbuck3	156	97.4	2.6	0.64	00:28:56	Grass
Blackbuck4	34	91.4	8.6	0	00:06:08	Shrubs
Wasp_old	15	86.6	13.4	0	00:13:27	Old colony
Wasp_new	16	93.75	6.25	0	00:14:03	New colony

Table 1: Results after running MOTHe on Blackbuck videos in various habitat and Wasps in two colonies. Each video clip is 30 seconds long taken at 30 FPS, these results are averaged over all the frames. % missed shows the number of individuals which were not detected in a frame and % false positives show the background noise identified as animal. Note that although some individuals were missed in particular frames but they were identified in subsequent frames hence giving trajectories for all the individuals.

We quantitatively compared the efficiency of MOTHe architecture with that of two other Computer Vision techniques (Image Subtraction and Color Segmentation) on various data-sets along with video specific recommendations for the users in Rathore A. et. al. (*In preparation*).

4 Discussion

MOTHe is an easy to use software pipeline that allows users to generate data-sets, train a simple neural network , detect and track multiple objects of interest (animals in our videos). We demonstrated MOTHe on two very different type of videos in which not only the species are different but the pixel size of these animals in the videos, animal-background contrast and background heterogeneity also differs, it shows that MOTHe is capable of solving a wide variety of detection problems. In our examples, maximum number of individuals presented to the detection algorithm was 156 and MOTHe was able to detect all 156 individuals. We surmise that it should be able to detect even larger number of individuals as long as the distance between individuals is at least one body length.

In table 3, we compare MOTHe's qualitative performance with some popular tracking solutions, a detailed quantitative comparison of MOTHe with other computer-vision based object detection techniques on various datasets is in preparation (Rathore A. et. al). The use of machine learning for classification enables MOTHe to detect stationary objects and bypasses the necessity of relying on motion as a detection indicator. MOTHe has various built-in functions and is developed to be user-friendly and highly customizable. Pre-configuration by the user helps to keep the inputs to a minimum during the tracking process. MOTHe is modular, organized and well-automated which helps the user to achieve object tracking with minimum efforts. MOTHe can be used to track objects on a desktop computer. Training and tracking with complex algorithms such as YOLO or faster RCNN may take several days and needs specialized infrastructure such as the NVIDIA DGX

series which uses NVIDIA GPU Cloud Deep Learning Stack with optimized versions of today's most popular frameworks and may cost upwards of 50,000 USD. Using machine learning algorithms makes MOTHe highly versatile and training the CNN with sufficient examples of variations results in high accuracy for detection in complex ecosystems.

MOTHe excels in scenarios with poor object contrast with background, bad lighting, background noise, variation of scale and viewpoint. Use of CNN in this package accounts for morphological variations and scaling issues. However, like most tracking algorithms [13, 14, 15, 16, 17, 18], MOTHe is incapable of resolving tracks of individuals in close proximity (less than one body length). As a trade-off to its computational efficiency, MOTHe also cannot deal with camera shake. Nonetheless, it can be used in combination with image stabilizing algorithms to solve camera shake issues.

In summary, MOTHe allows researchers with minimal coding experience to track stationary as well as moving animals in their natural habitat. For each steps of the detection and tracking process users need to run a single command. MOTHe is available as an open-source repository with detailed user guide and demonstrations via Github. We believe that this end-to-end package will encourage more researchers to use video observations for studying the animal group behaviour in their natural habitat and would be of use to a larger research community.

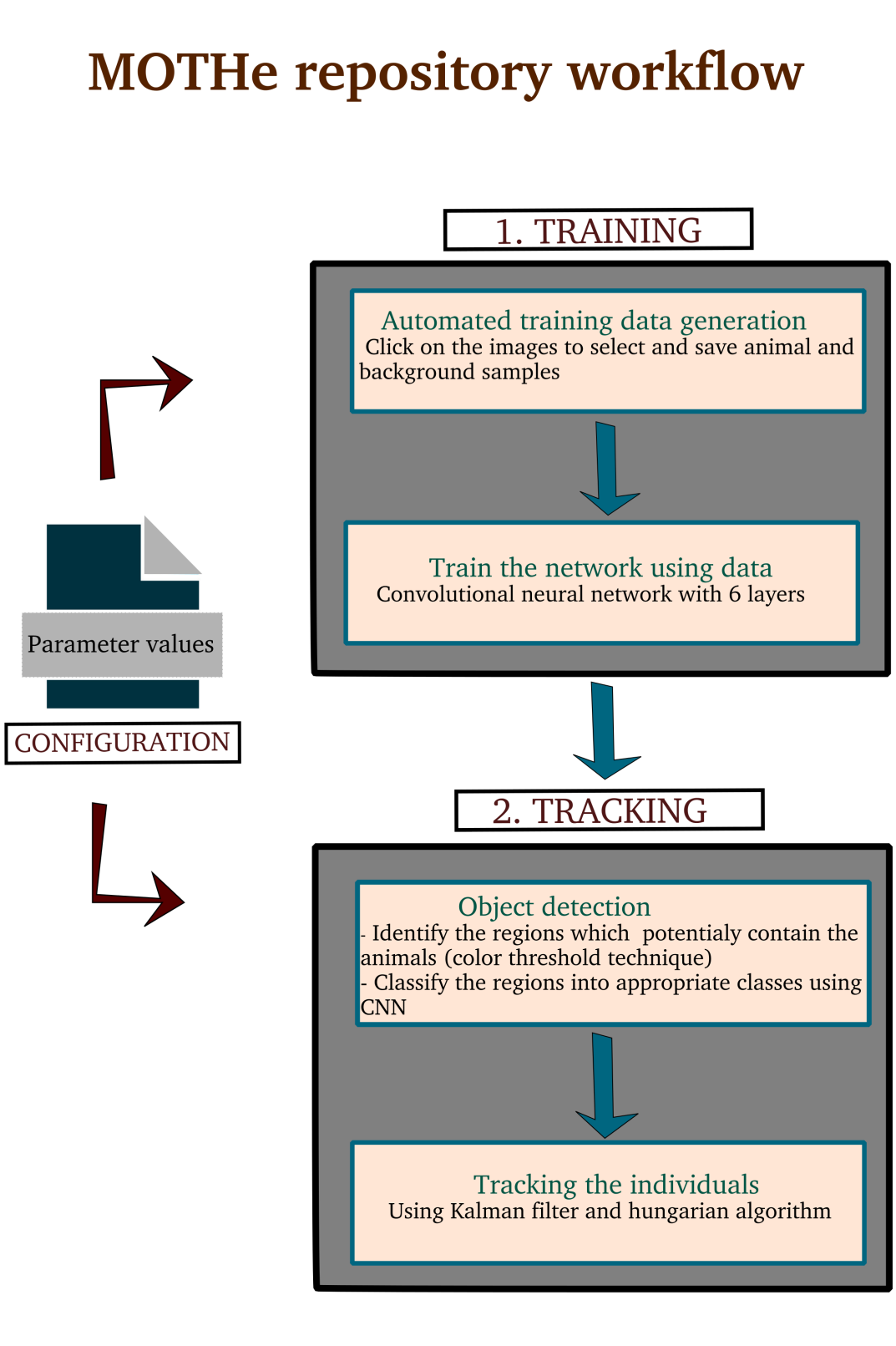

Figure 1: The layout of our Github repository. A configuration file is generated in the first step, it maintains directory paths and parameter values which are used by subsequent modules. Tracking happens in two steps, first we need to train the network on positive and negative examples from the data-set, then object detection is done indirectly using object classification on the points of interest. Each step here is a separate module which can be run by users. Blue arrows represent the directional flow of the executable files.

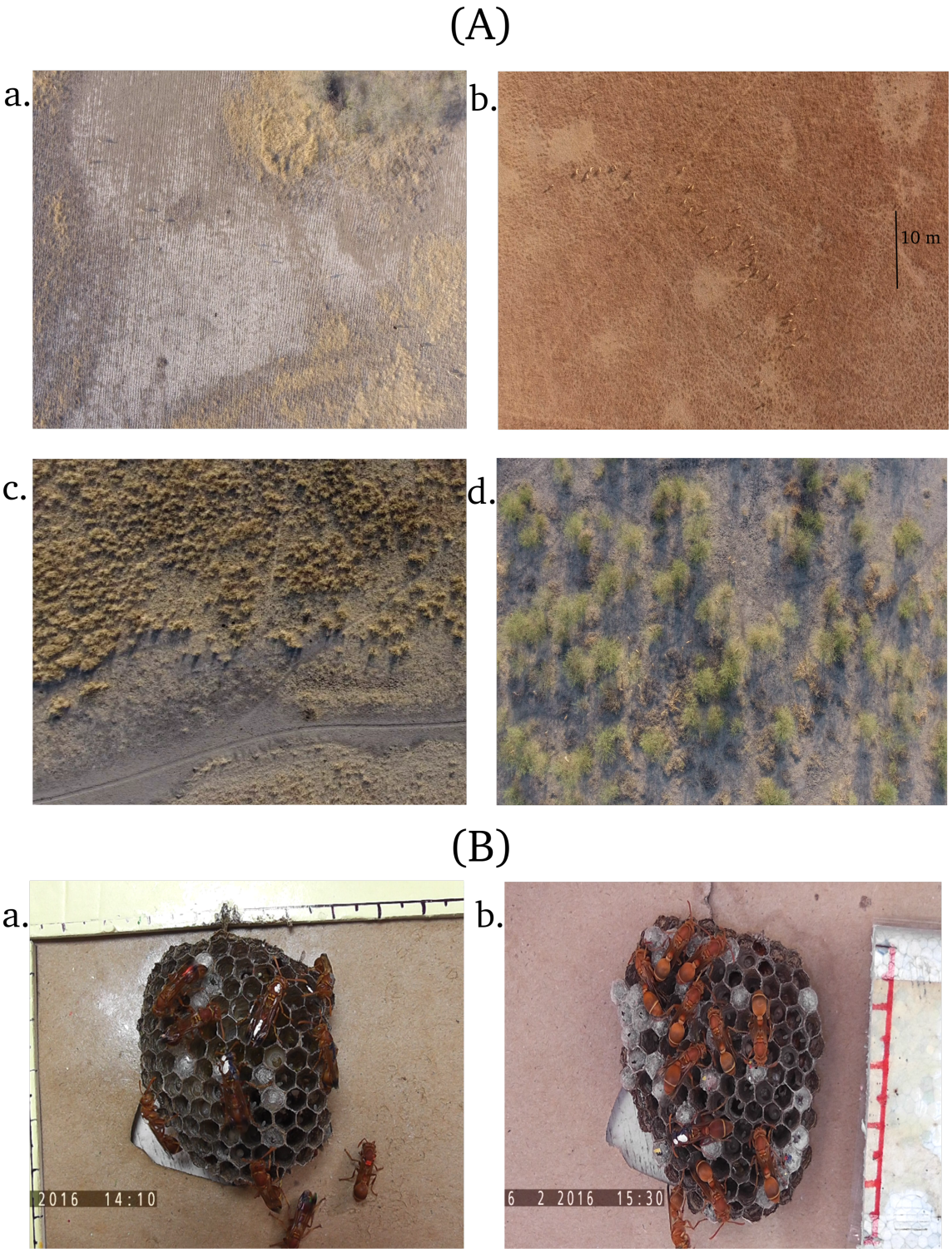

Figure 2: (A) Variation in Blackbuck habitat. (a) Blackbuck herd in the salty mudland part of the National Park; (b) Blackbuck camouflage very well in a homogeneous grassland; (c) Herd grazing in a patchy grassland, shape of grass patches is similar to blackbuck body shape; (d) Blackbuck herd in a shrubby area, it presents a lot of clutter and similar shaped objects as a challenge to the detection algorithm. (B) (a) & (b) Example frames from the wasp videos of different colonies. This experiment is done semi-natural settings. Wasps are brought into lab along with their nests. The color of the nest is very similar to wasp color and they don't move around much.

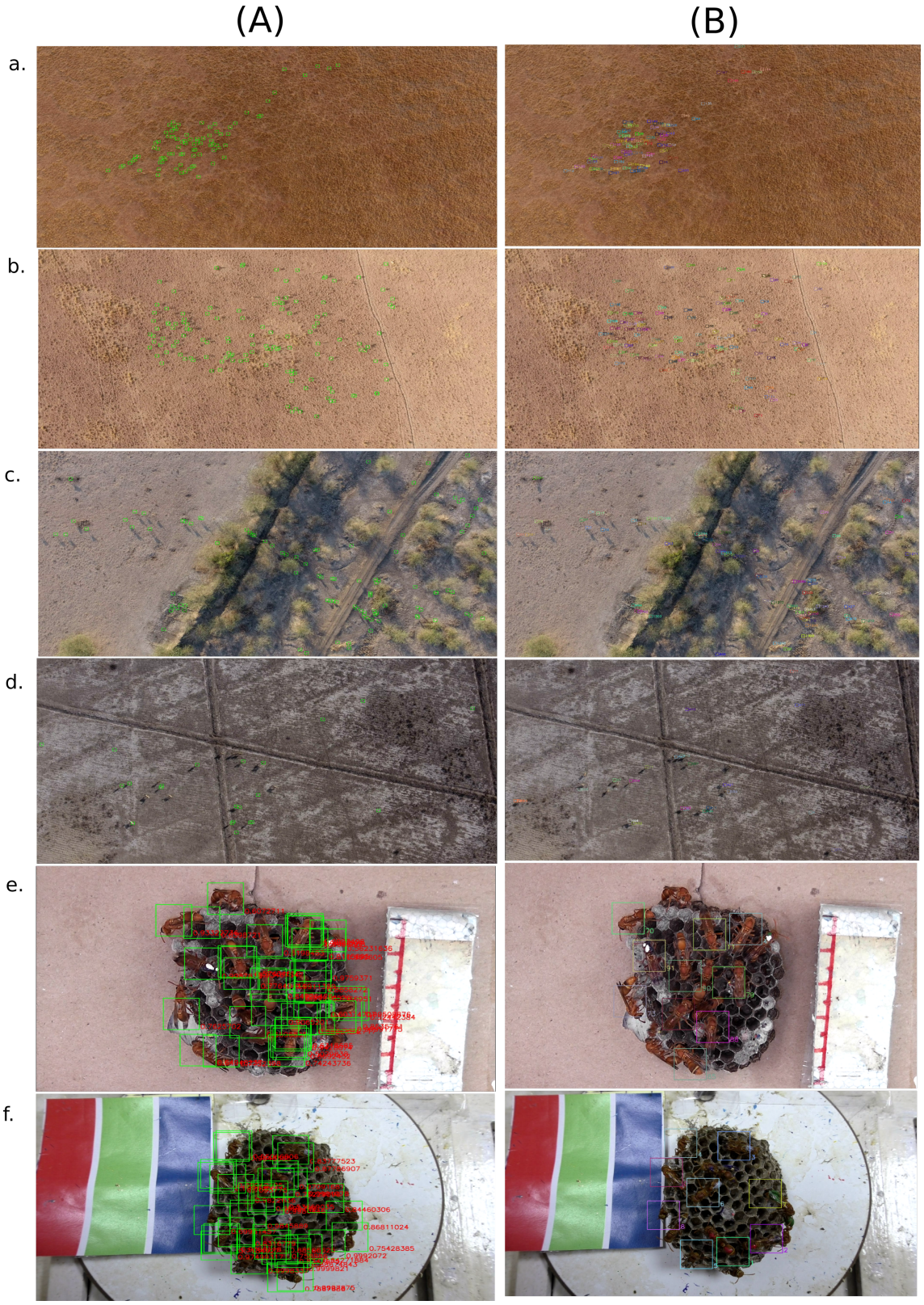

Figure 3: (A) Detection and (B) Tracking results on six example videos. a. Moderate size blackbuck herd in a grassland; b. A big herd (blackbuck - 158 individuals) in the grassland; c. blackbuck herd in a shrubby area; d. blackbuck herd in a salty marshland; e. an old wasp colony and f.relatively new wasp colony

	Features	MOTHe	Tracktor	idTracker	Yolo v3	BioTracker	ToxTrac
Video Complexity	Detection against heterogeneous background?	yes	for single individual	no	yes	no	no
	Multiple individual tracking?	yes	in homogeneous background	yes	yes	yes	yes
	Identifies stationary animals as well?	yes	no	yes	yes	no	NA
Ease of use	Requires sophisticated infrastructure? (GPUs)	no	no	minimum 8GB RAM	yes	no	no
	Interface and installation	Command based	Command based	GUI	Command based	GUI	GUI
	Click and drag functionality for training-data generation	yes	NA	NA	no	NA	NA
Performance	Maximum number of individuals tracked in test run	156	8	35	yes	11 in the example figure	20
	Computational efficiency	180 frames/minute for 4K resolution video	9 minutes 43 seconds for 33 MB video (fish schooling)	2s per frame for a video with 20 medaka fish	5 frames/minute for 4K resolution video	NA	25 frames per second in HD videos using modern computers
	Species on which testing was done	Antelope, Wasp	Fish, spider, termite, mice, tadpole	Fish, ant, mice, flies	Wildebeest, Zebra	Fish	Fish, mice, cockroach, ant
	Tested in conditions	Natural and semi-natural	Controlled environment for multiple individuals	Common lab conditions and manipulations	Natural	Controlled	Controlled

Table 2: Comparison of MOTHe with other popular tracking solutions in terms of three qualitative features: Video complexity, ease of use and performance.

Acknowledgment

We thank Department of Science and Technology, India and Ministry of Human Resource Development, India for funding this research.

References

[1] Luis Gonzalez, Glen Montes, Eduard Puig, Sandra Johnson, Kerrie Mengersen, and Kevin Gaston. Unmanned aerial vehicles (uavs) and artificial intelligence revolutionizing wildlife monitoring and conservation. *Sensors*, 16(1):97, 2016.

[2] Danielle P Mersch, Alessandro Crespi, and Laurent Keller. Tracking individuals shows spatial fidelity is a key regulator of ant social organization. *Science*, 340(6136):1090–1093, 2013.

[3] Colin J Torney, David J Lloyd-Jones, Mark Chevallier, David C Moyer, Honori T Maliti, Machoke Mwita, Edward M Kohi, and Grant C Hopcraft. A comparison of deep learning and citizen science techniques for counting wildlife in aerial survey images. *Methods in Ecology and Evolution*, 2019.

[4] Jarrod C Hodgson, Rowan Mott, Shane M Baylis, Trung T Pham, Simon Wotherspoon, Adam D Kilpatrick, Ramesh Raja Segaran, Ian Reid, Aleks Terauds, and Lian Pin Koh. Drones count wildlife more accurately and precisely than humans. *Methods in Ecology and Evolution*, 9(5):1160–1167, 2018.

[5] Dominique Chabot, Shawn R Craik, and David M Bird. Population census of a large common tern colony with a small unmanned aircraft. *PloS one*, 10(4):e0122588, 2015.

[6] Roland Kays, Margaret C Crofoot, Walter Jetz, and Martin Wikelski. Terrestrial animal tracking as an eye on life and planet. *Science*, 348(6240):aaa2478, 2015.

[7] Colin J Torney, Myles Lamont, Leon Debell, Ryan J Angohiatok, Lisa-Marie Leclerc, and Andrew M Berdahl. Inferring the rules of social interaction in migrating caribou. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 373(1746):20170385, 2018.

[8] Michele Ballerini, Nicola Cabibbo, Raphael Candelier, Andrea Cavagna, Evaristo Cisbani, Irene Giardina, Vivien Lecomte, Alberto Orlandi, Giorgio Parisi, Andrea Procaccini, et al. Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *Proceedings of the national academy of sciences*, 105(4):1232–1237, 2008.

[9] Yael Katz, Kolbjørn Tunstrøm, Christos C Ioannou, Cristián Huepe, and Iain D Couzin. Inferring the structure and dynamics of interactions in schooling fish. *Proceedings of the National Academy of Sciences*, 108(46):18720–18725, 2011.

[10] Damien R Farine, Ariana Strandburg-Peshkin, Tanya Berger-Wolf, Brian Ziebart, Ivan Brugere, Jia Li, and Margaret C Crofoot. Both nearest neighbours and long-term affiliates predict individual locations during collective movement in wild baboons. *Scientific reports*, 6:27704, 2016.

[11] Andrea Flack, Máté Nagy, Wolfgang Fiedler, Iain D Couzin, and Martin Wikelski. From local collective behavior to global migratory patterns in white storks. *Science*, 360(6391):911–914, 2018.

[12] Julia K Parrish and Leah Edelstein-Keshet. Complexity, pattern, and evolutionary trade-offs in animal aggregation. *Science*, 284(5411):99–101, 1999.

[13] Alfonso Pérez-Escudero, Julián Vicente-Page, Robert C Hinz, Sara Arganda, and Gonzalo G De Polavieja. idtracker: tracking individuals in a group by automatic identification of unmarked animals. *Nature methods*, 11(7):743, 2014.

[14] Benjamin Risse, Dimitri Berh, Nils Otto, Christian Klämbt, and Xiaoyi Jiang. Fimtrack: An open source tracking and locomotion analysis software for small animals. *PLoS computational biology*, 13(5):e1005530, 2017.

[15] Hauke Jürgen Mönck, Andreas Jörg, Tobias von Falkenhausen, Julian Tanke, Benjamin Wild, David Dormagen, Jonas Piotrowski, Claudia Winklmayr, David Bierbach, and Tim Landgraf. Biotracker: An open-source computer vision framework for visual animal tracking. *arXiv preprint arXiv:1803.07985*, 2018.

[16] Vivek Hari Sridhar, Dominique G Roche, and Simon Gingins. Tracktor: image-based automated tracking of animal movement and behaviour. *Methods in Ecology and Evolution*, 2018.

[17] Alvaro Rodriguez, Hanqing Zhang, Jonatan Klaminder, Tomas Brodin, Patrik L Andersson, and Magnus Andersson. Toxtrac: a fast and robust software for tracking organisms. *Methods in Ecology and Evolution*, 9(3):460–464, 2018.

[18] Osamu Yamanaka and Rito Takeuchi. Umatracker: an intuitive image-based tracking platform. *Journal of Experimental Biology*, 221(16):jeb182469, 2018.

[19] Massimo Piccardi. Background subtraction techniques: a review. 4:3099–3104, 2004.

[20] Mahdi Bagheri, Mehdi Madani, Ramin Sahba, and Amin Sahba. Real time object detection using a novel adaptive color thresholding method. In *Proceedings of the 2011 international ACM workshop on Ubiquitous meta user interfaces*, pages 13–16. ACM, 2011.

[21] Anthony I Dell, John A Bender, Kristin Branson, Iain D Couzin, Gonzalo G de Polavieja, Lucas PJJ Noldus, Alfonso Pérez-Escudero, Pietro Perona, Andrew D Straw, Martin Wikelski, et al. Automated image-based tracking and its application in ecology. *Trends in ecology & evolution*, 29(7):417–428, 2014.

[22] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. In *Advances in neural information processing systems*, pages 2553–2561, 2013.

[23] Connor Bowley, Alicia Andes, Susan Ellis-Felege, and Travis Desell. Detecting wildlife in uncontrolled outdoor video using convolutional neural networks. In *2016 IEEE 12th International Conference on e-Science (e-Science)*, pages 251–259. IEEE, 2016.

[24] Mohammad Sadegh Norouzzadeh, Anh Nguyen, Margaret Kosmala, Alexandra Swanson, Meredith S Palmer, Craig Packer, and Jeff Clune. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proceedings of the National Academy of Sciences*, 115(25):E5716–E5725, 2018.

[25] Jacob M Graving, Daniel Chae, Hemal Naik, Liang Li, Benjamin Koger, Blair R Costelloe, and Iain D Couzin. Fast and robust animal pose estimation. *bioRxiv*, page 620245, 2019.