APPENDIX S2: User interface and functional features of the Trapper Citizen Science platform

Manuscript Title: Trapper Citizen Science: an open-source camera trap platform for citizen science in wildlife research and management

Authors: Magali Frauendorf, Jakub W. Bubnicki, Filip Ånöstam, Piotr Tynecki, Łukasz Wałejko, Joris P. G. M. Cromsigt, Fredrik Widemo, Tim R. Hofmeester

Contents

Text

Text S1: Functional modules and user interaction design

1. Data upload

The Trapper CS upload interface enables users to upload images (up to 10k files in one request) directly to specific camera trap deployments and locations, allowing immediate classification (Appendix S2, Fig. S1). Users can choose existing deployments or create new ones on-the-fly by entering coordinates (or using an interactive map), deployment name (ID) and observation period along with optional metadata as camera model, camera height, bait type or habitat type. Images can be uploaded via drag-and-drop or manual selection, with options to keep them private or make them public. All deployment attributes are fully compatible with Camtrap DP standard.

Upon upload, Trapper automatically creates background objects such as Location, Deployment, Collection and Sequence (images are automatically aggregated into sequences i.e. independent ecological events), as well as database links to associated research and classification projects and generates thumbnails and preview files. Backend processes then automatically trigger pre-configured AI-pipeline to detect objects (defaulting to the Megadetector model), anonymize humans and vehicles as well as classify species. Project coordinators (admins) can choose AI models for detection and species classification from those available in the Trapper AI Manager module. This automated pipeline ensures that images are ready for efficient classification while maintaining data privacy.

2. Dashboard with basic insights for data exploration

The Trapper CS dashboard provides users with an overview of accessible data and a summary of the current Classification Project they are working on (Appendix S2, Fig. S2). The dashboard is divided into two sections: *Project Statistics* and *My Statistics*. *Project Statistics* displays aggregate data for active users, locations, deployments, images and recordings in the project, with visuals including a bar graph of species counts, a pie chart of classification metrics and a summary of total camera trap days. *My Statistics* mirrors these metrics but highlights the user's personal contributions or access within the project. Additional features include notifications for unread messages, easy navigation through the left sidebar and user options for language selection and profile management. This layout facilitates efficient tracking of project-wide and individual progress, enhancing user engagement with large-scale wildlife monitoring data.

3. Deployment and images

3.1 Image view

The Trapper *CS Images view* enables users to browse, filter and manage all accessible images within a (*Classification) Project*. Users can filter images based on various criteria, such as species, location, deployment, observation type, date, owner and classification status, with quick toggles to view images they own, those shared by team members (see section on Teams below) or favourites. Filters can also include custom attributes defined at the classificator object level (e.g. a 'Human activity' field with a predefined list of options). The search bar allows users to locate images by filename, and a table displays thumbnails (with the opportunity of getting a bigger view by clicking on it), location details, observation type, species, classification statuses (e.g., AI-processed, user-verified, approved), and ownership icons, alongside options to view, edit or mark images as favourites (Appendix S2, Fig. S3a&b).

The 'view image' option provides a focused look at a single image along with its full metadata, including camera trap location, deployment details and classification results from both AI and human review. Users can see an interactive map pinpointing the camera trap's location (Appendix S2, Fig. S3c).

3.2 Deployment view

The Deployments view provides users with a centralized overview of camera trap deployments, accessible by all project members (except deployments marked as private by the owners) and filtered by location, owner and date. A search bar and quick filters allow users to search by deployment and view only their own deployments or those shared within teams or set as public, respectively. Each deployment entry includes metadata such as location coordinates, species detected, date range and total number of sequences (events). Users can switch from list to map view for spatial insights. Key actions available include editing and deleting (for own deployments) and accessing images captured at each deployment for viewing or classification, enabling efficient management of large datasets (Appendix S2, Fig. S4a).

When viewing the deployment, users can review metadata and related location data for individual deployments, visualized on an interactive map (Appendix S2, Fig. S4b). Here, users with appropriate permissions can update metadata, delete deployments and automatically correct wrong image timestamps with specified time offsets (expressed as +/- days : minutes : seconds) for the entire deployment with one click. This view also includes nested access to the deployment's images, showing statistics on image classification by AI, users and experts,

supporting streamlined data validation and temporal accuracy in collaborative wildlife monitoring projects.

4. Spatial visualization

The Trapper CS platform integrates web GIS functionality to facilitate spatial visualization of camera trap deployments, enabling researchers to track wildlife monitoring and environmental patterns through an interactive, map-based interface (Appendix S2, Fig. S5). This interface, located on the deployments page, displays camera trap locations as markers on a map powered by Leaflet and OpenStreetMap, allowing users to visualize data concentration across landscapes. Users can filter deployments by criteria such as location, owner, date range and incomplete status, with options to toggle between map and list views, and share selected deployments with team members for collaborative analysis. The interactive map supports zooming, clustering and a customizable basemap (open street map and satellite), enhancing spatial exploration by offering a visual overview of deployment locations and direct access to preview images and videos for each site.

5. Classification interface

The *Classification View* in Trapper CS is a comprehensive and visually appealing interface designed to facilitate the object-based and AI-assisted classification of camera trap images by users. Initially, images are classified by the object detection algorithm (e.g. MegaDetector v6), filtering out those that contain humans, vehicles or blank images, thereby allowing users to focus on classifying animal images. Additionally, when specified in the project's configuration, the species-level model is automatically run on all images with at least one animal detected. To validate the images, users begin by selecting a target deployment and can opt to include only AI-classified animal images or blank images as well (e.g., for extra validation of missed detections of animals or humans).

Images are presented in a dynamically ordered horizontal list, enabling easy navigation (both image- and sequence-based) through potentially thousands of images. This list clearly differentiates between various classifications, including those approved, those classified by the user and those flagged for feedback. Users can select an image for detailed classification, where they can view the AI-generated bounding boxes and species labels and use a dynamic classification form tailored to their project context (Appendix S2, Fig. S6a).

The interface supports object-level annotation with features for creating, editing and managing bounding boxes. Users can classify one or more detected objects on a single image (bounding boxes) by selecting species, age, sex and other attributes (including custom ones defined in the project's configuration). They can also mark images as favourites or flag them as empty. Once classified, the next image automatically appears, reducing the amount of necessary manual mouse clicks for the user. Advanced functionalities allow for bulk classification of images within sequences (or even group of sequences), streamlining the process for images that share common characteristics. The inclusion of custom attributes, such as, for example, health status or distance to the animal, further enriches the data captured. Furthermore, the user can apply image filters to adjust contrast, brightness and saturation of the image, improving the visibility and detection of animals on the images (Appendix S2, Fig S6b&c).

Additionally, a robust feedback mechanism enables users to provide input on AI performance, enhancing the model's accuracy over time. Users designated for their expertise as error-proof can approve user classifications. Users with an error-proof role will automatically approve all user classifications. Project admins and error-proof users can also modify (correct) existing approved classifications.

Overall, this interface not only streamlines data validation and enhances the classification process but also integrates human expertise with automated systems, making it a vital tool for ecological research and wildlife monitoring.

6. Teams and data sharing

The *Teams* function in Trapper CS is a collaborative tool designed to facilitate data sharing among users with shared geographic, institutional or research interests. Teams enable members to share their private media and deployments within a specific classification project, enhancing data accessibility and accelerating classification tasks by pooling resources. In other words, the Teams module is a tool that citizen scientists can use to self-organize around shared tasks and goals.

To create a team, an active user initiates a three-step form, entering a team name and description, defining a geographical area using an interactive map or GPX file and selecting team members from the platform, with justification for each invite (Appendix S2, Fig. S7). Team leaders, who have administrator status within the team, can add members, manage deployments, and classify all shared images, while regular team members can classify shared images that do not contain humans or vehicles. Once invited, users receive notifications and can accept or decline the invitation; ignoring the invite keeps the membership status pending.

Team members can search and filter teams by name and description. Teams can have geographic boundaries, restricting shared data to camera trap locations within these areas, which are stored as GeoDjango PolygonFields for spatial queries. Additionally, users maintain control over which classification projects they wish to share within a team, with the flexibility to leave or be removed by the team leader without losing their own classifications. This function supports efficient, self-organized collaboration, allowing researchers to enhance the depth and breadth of their data within a structured, permission-based framework.

7. Export

From the Trapper backend, classification results can be exported as a single CSV or a complete data package with both CSV tables and accompanying metadata file formatted according to Camtrap DP standard specification (Bubnicki et al. 2024). This is important as preparing metadata files in one of the available standards is often a very time-consuming task, which usually has the lowest priority in scientific workflow, but enhances collaboration as publishing metadata files in global metadata repositories makes local camera trapping data visible to the world-wide research community.

Figures

Figure S1: Data upload view of Trapper CS.

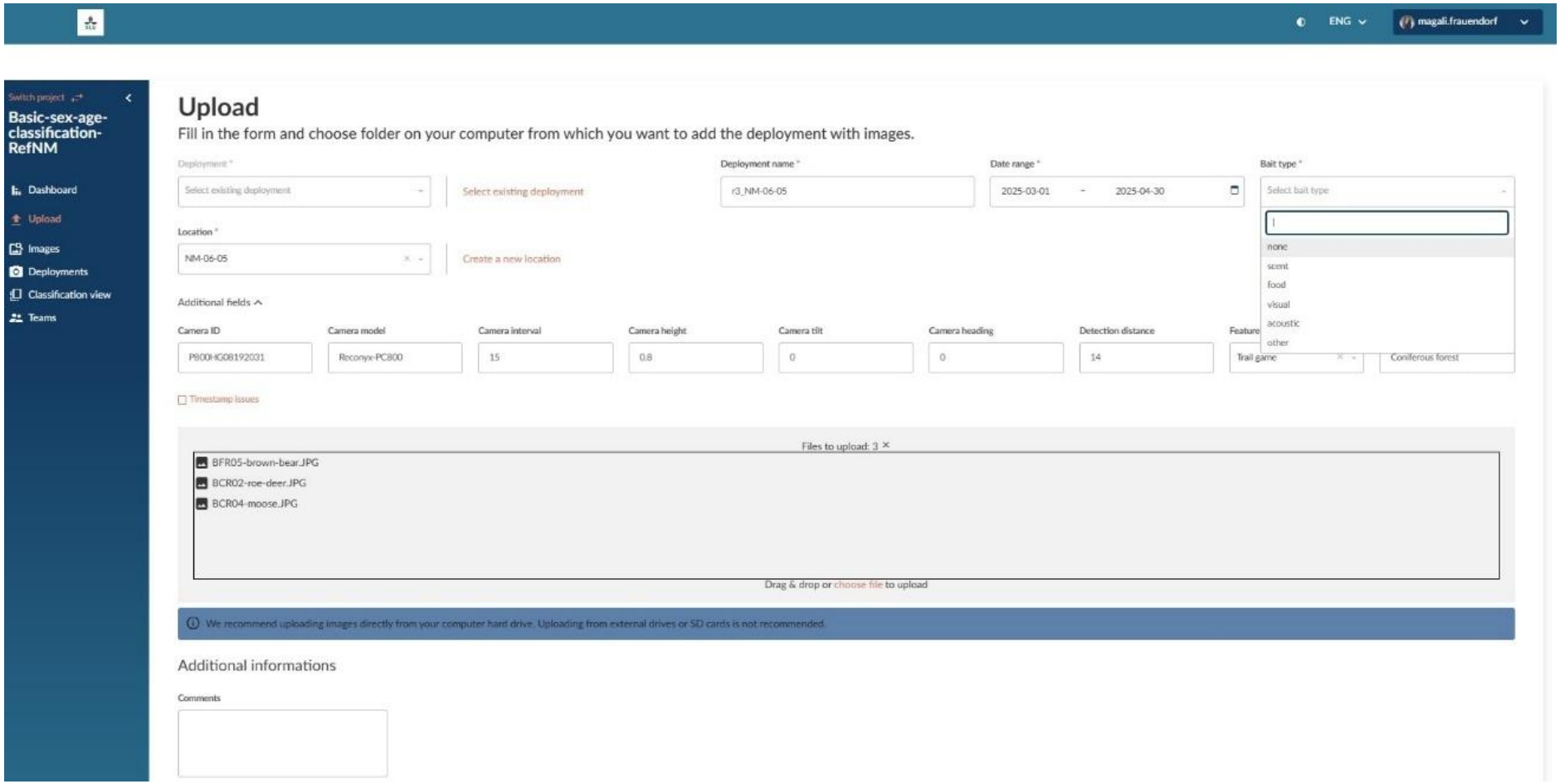

Figure S2: Overview of the personal dashboard for each user, showing a summary of the collected data in (a) the project statistics and (b) the personal statistics (within the project).

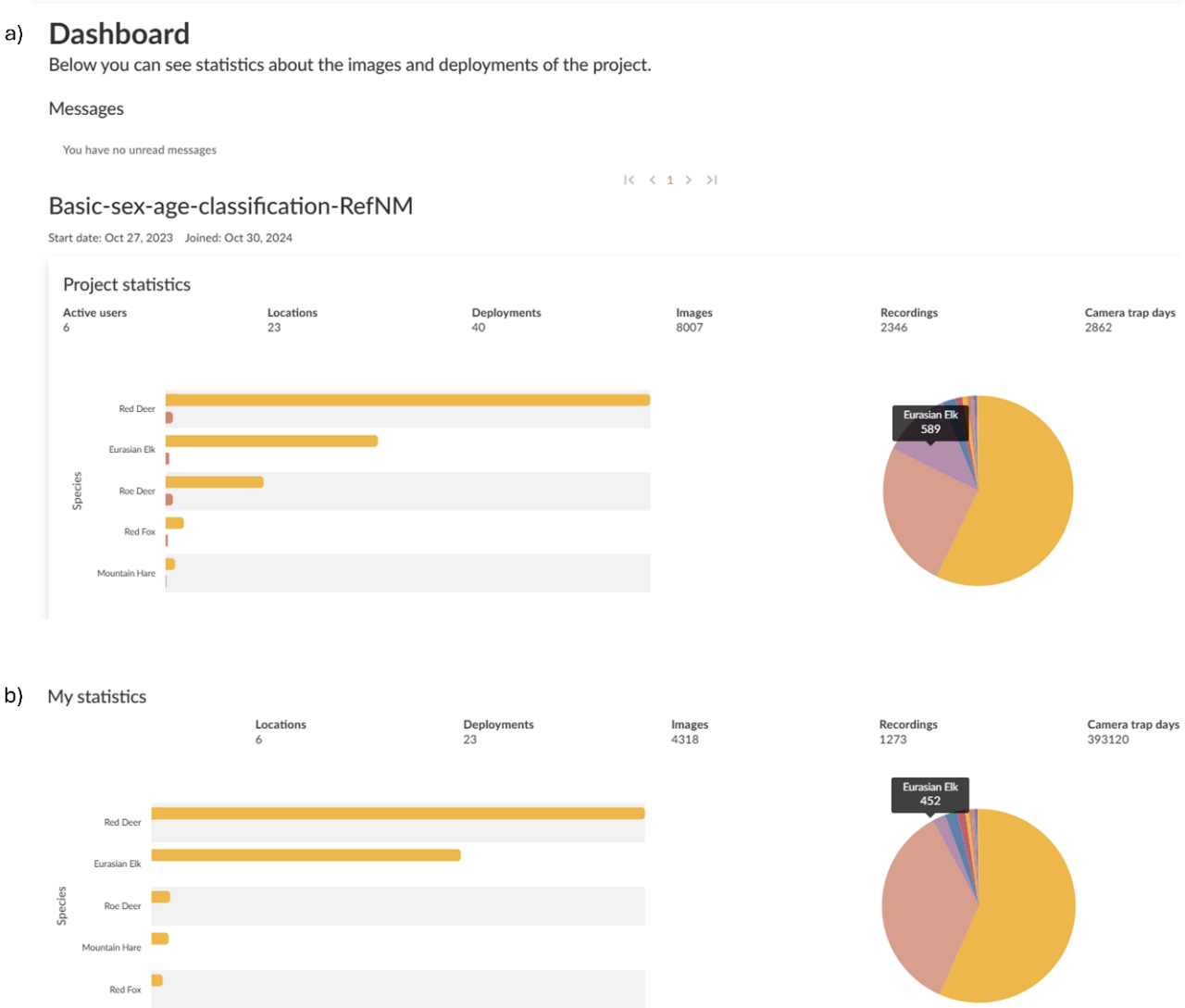

Figure S3: (a) Trapper CS image listing view, (b) detailed view of a specific image and (c) interactive map visualizing the spatial component of the data.

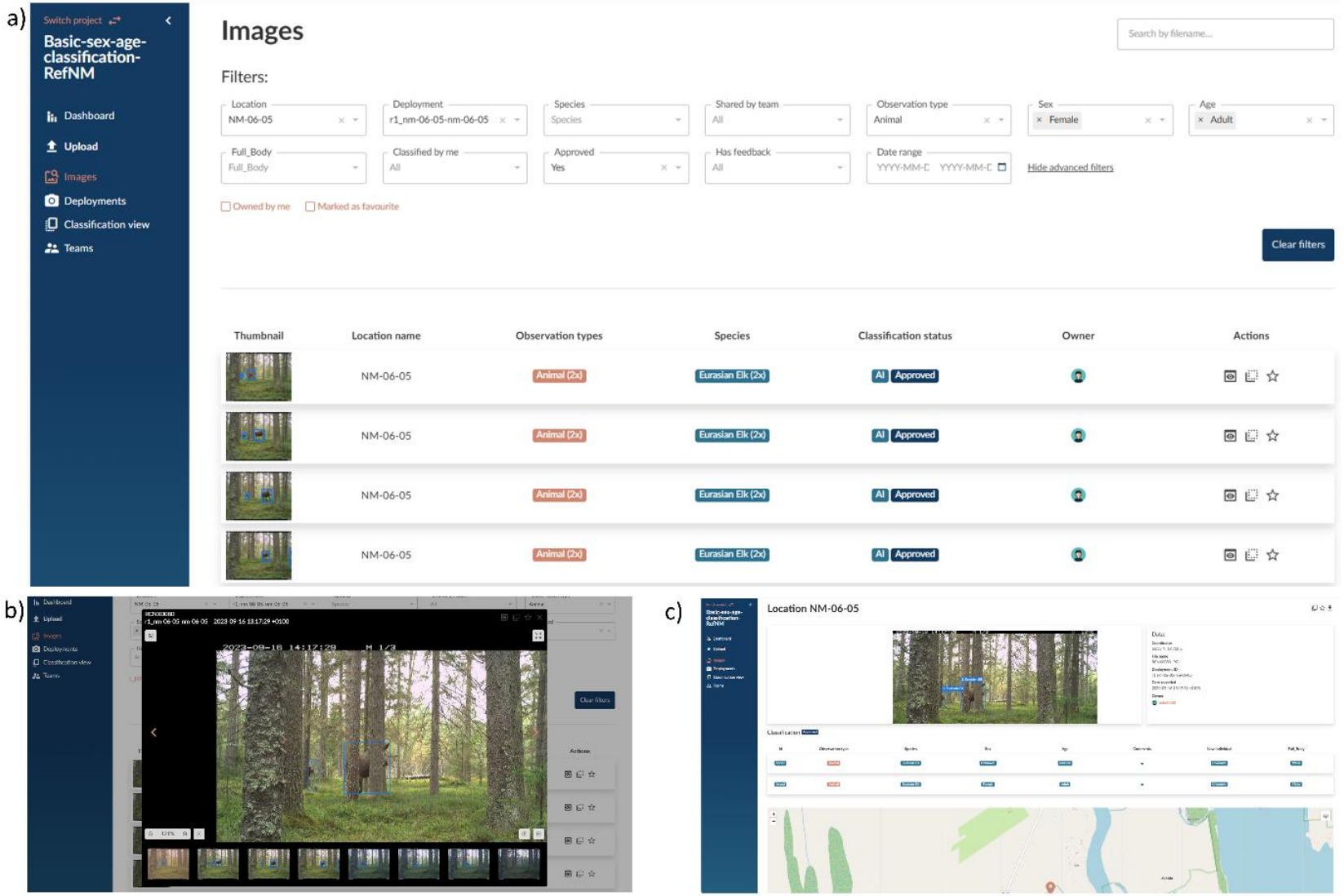

Figure S4: (a) Deployment listing view and (b) detailed information about a specific deployment.

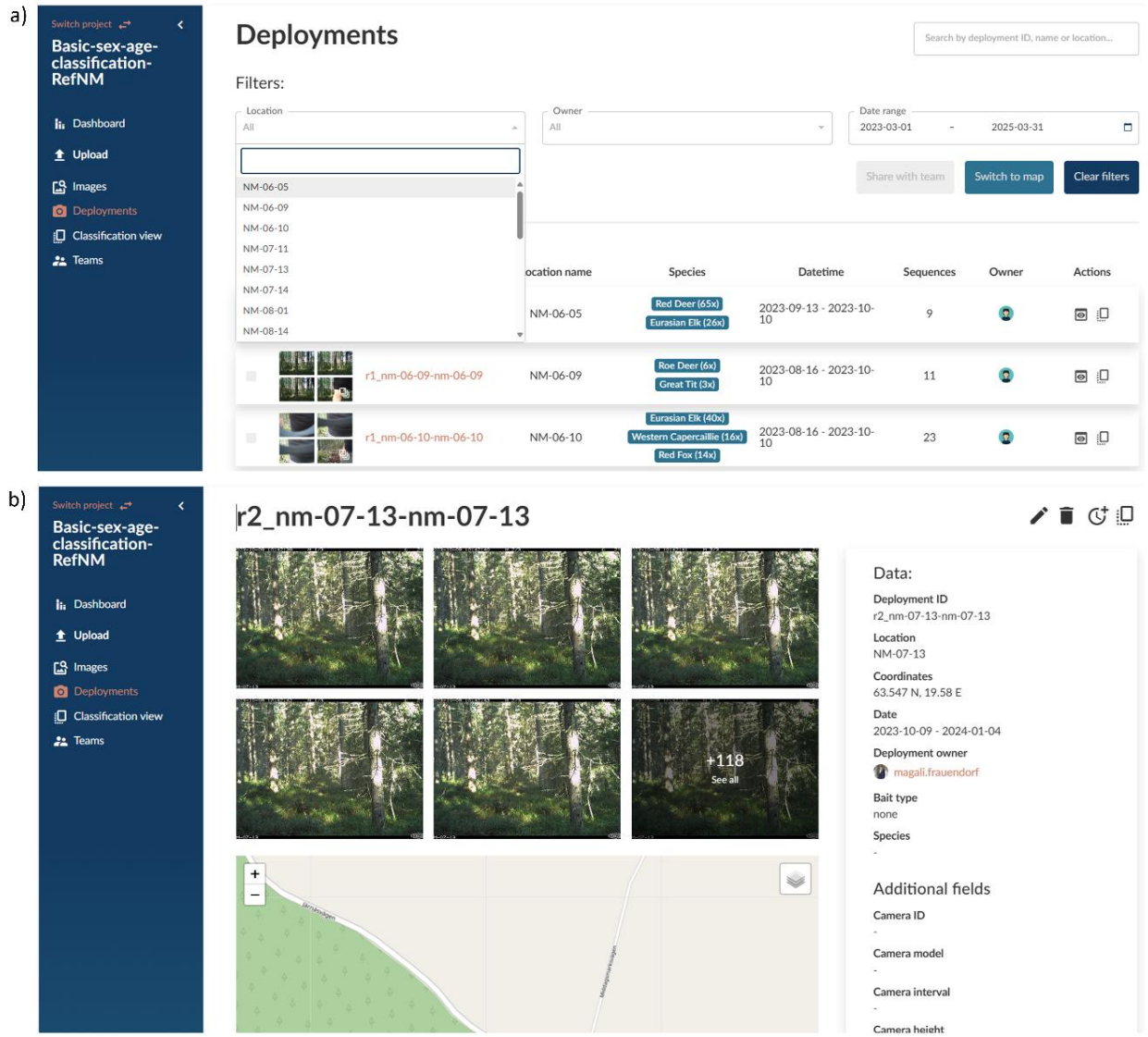

Figure S5: Spatial visualization of the collected data.

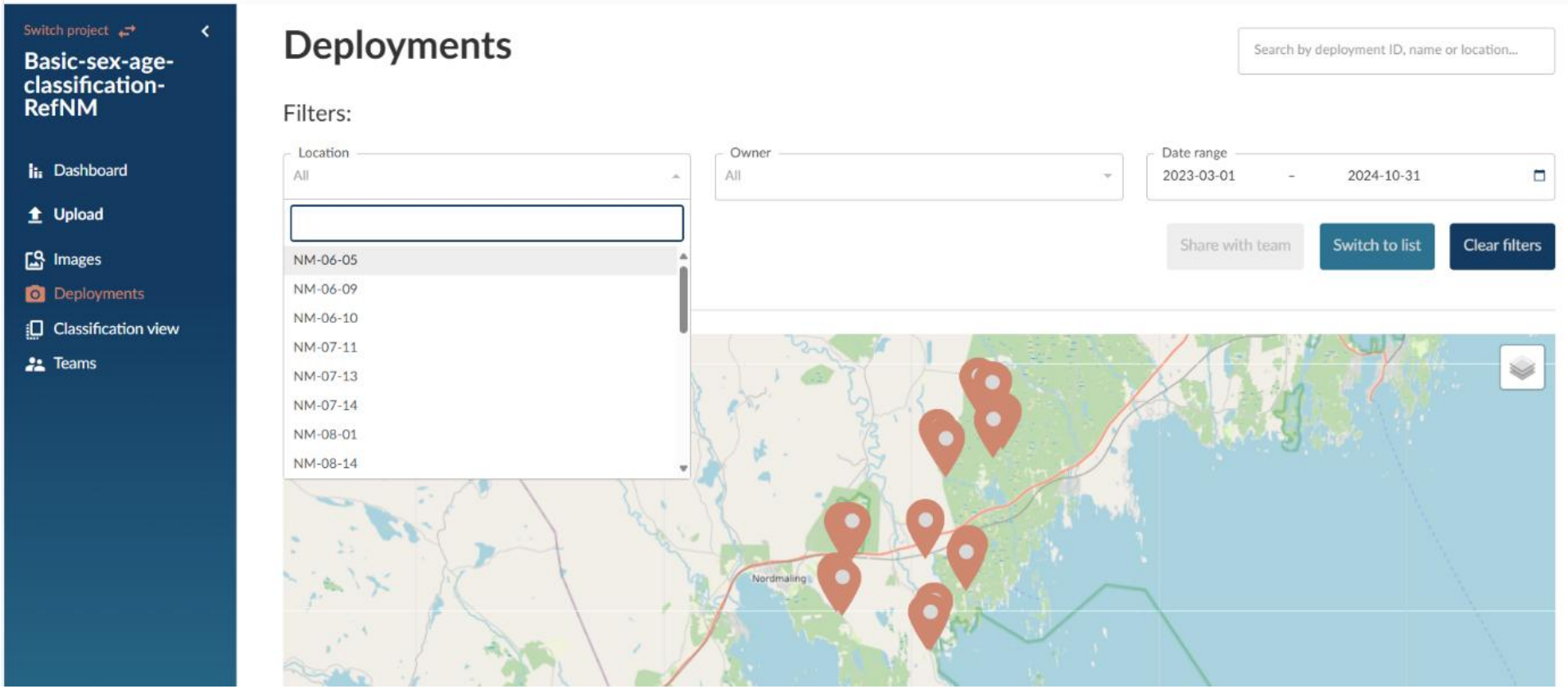

Figure S6: (a) Classification view with bounding boxes in orange around the object (animal), (b) interface for user classification and (c) with adjusted and applied image filters for easier recognition of the object.

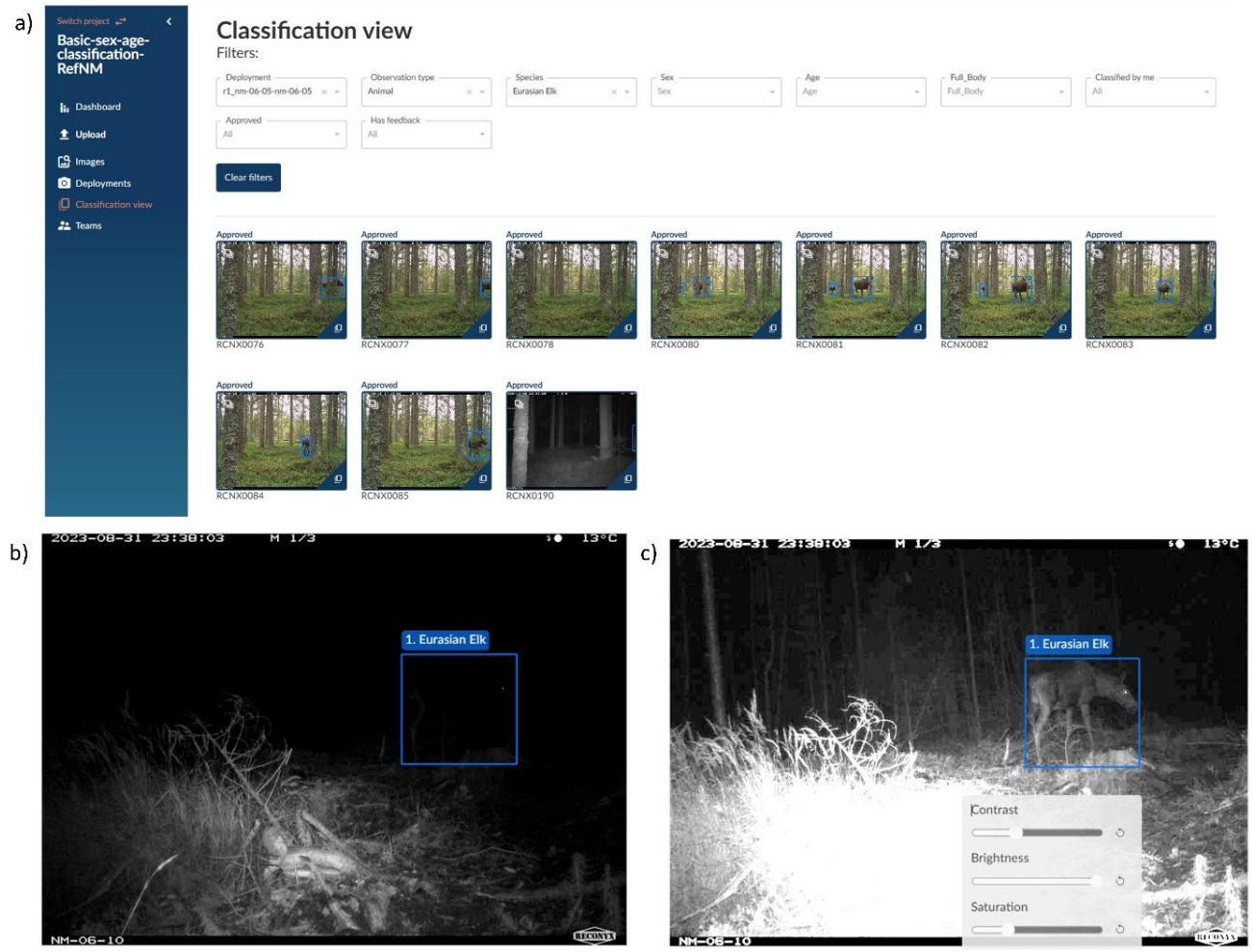

Figure S7: (a) you can create your team by writing the name with a description. Next you draw a polygon to indicate the boundaries of the teams area. (b) shows the overview of the created team and (c) shows how you can invite users to your team.