
# Streamlining camera trap image workflows with MegaDetector©: A test case in Ontario’s managed forests

**Technical Report TR-66**

Science and Research Branch

Ministry of Natural Resources

Ontario

Science and Research Technical Report TR-66

# Streamlining camera trap image workflows with MegaDetector©: A test case in Ontario's managed forests

**C.J.S. Martin[1] and P.D. DeWitt[2]**

[1] Biodiversity and Monitoring Section, Science and Research Branch, Rosslyn, ON

[2] Biodiversity and Monitoring Section, Science and Research Branch, Peterborough, ON

2025

Science and Research Branch

Ministry of Natural Resources

Copies of this publication are available from info.mnrscience@ontario.ca.

Cette publication hautement spécialisée, Streamlining Camera Trap Image Workflows with MegaDetector©: A Test Case in Ontario’s Managed Forests, n’est disponible qu’en anglais conformément au Règlement 671/92, selon lequel il n’est pas obligatoire de la traduire en vertu de la Loi sur les services en français. Pour obtenir des renseignements en français, veuillez communiquer avec le ministère des Richesses naturelles au info.mnrscience@ontario.ca.

Some of the information in this document may not be compatible with assistive technologies. If you need any of the information in an alternate format, please contact info.mnrscience@ontario.ca.

Cite this report as:

Martin, C.J.S. and P.D. DeWitt. 2025. Streamlining camera trap image workflows with MegaDetector©: A test case in Ontario’s managed forests. Ontario Ministry of Natural Resources, Science and Research Branch, Peterborough, ON. Science and Research Technical Report TR-66. 21 p. + appendices.

# Abstract

Camera traps are commonly used to collect information on wildlife, however, manual review of images to extract ecological information can cost substantial time and resources. We tested the ability of MegaDetector© v5a, an artificial intelligence-driven image recognition model, to reliably distinguish between empty and occupied camera trap images collected from Ontario's managed forests. Our aim was to streamline image interpretation workflow by removing many empty images before having staff manually inspect remaining images for wildlife species. We applied MegaDetector to 10 representative image sets generated by Ontario's Multiple Species Inventory and Monitoring effort and used receiver operating characteristic curves to evaluate MegaDetector's discrimination. MegaDetector analyzed 193,226 images at an average rate of ~4.3 images/second, generating 272,333 unique detections within 120,013 of the images. MegaDetector's ability to predict if an image was occupied varied according to the discrimination threshold – the detection confidence level understood to mark the decision boundary between empty and occupied images. With the most risk-averse sensitivity (0.99), MegaDetector initially allowed us to remove 38% of the original data set at a cost of missing 1.4% of all 12,530 occupied images. Integrating predicted events, where one or more images were triggered by the same animal or group of animals, dramatically improved performance. Removing images that fell beneath the 0.67 discrimination threshold and were greater than 15 minutes from predicted events allowed us to remove 83% of the original data set at a cost of missing 0.8% of all occupied images. MegaDetector's performance varied among cameras and variation was likely driven by the density of stumps, foliage, and boulders within the field of view, number of small, non-target wildlife species, and proportion of empty images. We recommend that users of image detection and classification algorithms (1) identify optimal discrimination thresholds for their study system and research aims, (2) leverage the temporal clustering of wildlife images to maximize efficiency, and (3) report the expected mean percentage of occupied images missed in planned workflows.

# Résumé

## Rationalisation des flux de travail liés aux images de pièges photographiques avec MegaDetector© : Un cas d'essai dans les forêts aménagées de l'Ontario

Les pièges photographiques sont couramment utilisés pour recueillir des renseignements sur la faune. Cependant, l'examen manuel des images pour extraire des renseignements écologiques peut s'avérer coûteux en temps et en ressources. Nous avons testé la capacité de MegaDetector© v5a, un modèle de reconnaissance d'images basé sur l'intelligence artificielle, à distinguer de manière fiable les images de pièges photographiques vides et occupées recueillies dans les forêts aménagées de l'Ontario. Notre objectif était de rationaliser le flux de travail lié à l'interprétation des images en supprimant plusieurs images vides avant que le personnel n'inspecte manuellement les images restantes pour détecter la présence d'espèces sauvages. Nous avons appliqué MegaDetector à 10 ensembles d'images représentatifs générés par l'effort d'inventaire et de surveillance des multiples espèces de l'Ontario et utilisé les courbes caractéristiques de fonctionnement du récepteur pour évaluer la discrimination de MegaDetector. MegaDetector a analysé 193 226 images à une vitesse moyenne d'environ

4,3 images par seconde, générant 272 333 détections uniques dans 120 013 images. La capacité de MegaDetector à prédire si une image est occupée varie en fonction du seuil de discrimination – le niveau de confiance de détection considéré comme marquant la limite de décision entre les images vides et les images occupées. Avec la sensibilité la plus prudente face au risque (0,99), MegaDetector nous a initialement permis de supprimer 38 % de l'ensemble de données original au prix de manquer 1,4 % des 12 530 images occupées. L'intégration d'événements prédits, où la prise d'une ou de plusieurs images a été déclenchée par le même animal ou groupe d'animaux, a considérablement amélioré le rendement. La suppression des images inférieures au seuil de discrimination de 0,67 et supérieures à 15 minutes des événements prédits nous a permis de supprimer 83 % de l'ensemble de données d'origine, au prix de manquer 0,8 % de toutes les images occupées. Le rendement de MegaDetector varie selon les caméras. La variation est probablement attribuable à la densité des souches, du feuillage et des rochers dans le champ de vision, au nombre de petites espèces sauvages non ciblées et à la proportion d'images vides. Nous recommandons aux utilisateurs d'algorithmes de détection et de classification d'images (1) de déterminer les seuils de discrimination optimaux pour leur système d'étude et leurs objectifs de recherche, (2) de tirer parti du regroupement temporel des images de la faune pour maximiser l'efficacité, et (3) de rapporter le pourcentage moyen attendu d'images occupées manquées dans les flux de travail planifiés.

# Acknowledgements

We thank Lynn Landriault (Ministry of Natural Resources) for camera trap data management and assistance with study conceptualization. We thank numerous wildlife technicians for camera trap data collection and their manual interpretation of images. We also thank Lynn Landriault, Gillian Holloway, Lani Stinson, Hanna Dorval, Danielle Berube, Emilie Kissler, Lauren McDonald, Erica Newton, Katrina Wisniewski (all Ministry of Natural Resources); Juni Rechter (University of Guelph); and Noemie Jenni, Corey Corbett, and Chloe Bryant (Matrix Solutions Inc.) for their comments on earlier drafts of this paper.

# Contents

# Introduction

Camera traps have become a standard method of monitoring wildlife across time and space (Rovero et al. 2013, Steenweg et al. 2017). Their widespread use generates large numbers of images that require processing and interpretation to provide meaningful ecological information. Manually reviewing and interpreting images is a major time and resource cost for the camera-trapping community (Glover-Kapfer et al. 2019). Scientists have addressed this challenge by developing computer models to process and categorize images (e.g., Vélez et al. 2022). The purpose and architecture of such models varies, and many have shown potential to reduce the time and costs associated with camera traps by accelerating image processing (Greenberg 2020, Fennell et al. 2022) or fully automating image processing workflow (Whytock et al. 2021).

MegaDetector© is an open-source computer model that applies deep machine learning to detect and classify objects in images as animal, person, or vehicle (Beery et al. 2019). MegaDetector version 5a (hereafter MegaDetector) was trained on millions of camera trap images captured from Africa, Colombia, Vietnam, Lao, New Zealand, the United States of America, and other undisclosed locations[1]. MegaDetector is primarily a detector that delineates and assigns confidence levels to objects within an image. This process is a valuable first step because it:

- helps researchers exclude empty images from data sets
- creates bounding boxes that help count multiple individuals in one image
- focuses species classification model efforts to bounding boxes within images

Identifying empty images for exclusion from analyses is crucial given many data sets are dominated by such images (Norouzzadeh et al. 2018, Fennell et al. 2022, Leorna and Brinkman 2022). Assigning bounding box coordinates and a confidence level to each individual detection allows researchers to evaluate MegaDetector's performance at various discrimination thresholds, defined as specified confidence levels marking the decision boundary between empty and occupied images.

Image detection and classification models are sensitive to the size, location, and balance of the training data used to develop them (Schneider et al. 2020). For example, differences in ecosystem structure or light contrast between a model's training data and where it is applied (i.e., domain shift) can decrease model performance. A relatively small number of training images (~10,000) from eastern North America were used to develop MegaDetector. As a result, MegaDetector's performance is less certain in Ontario's forest environments than in ecosystems sharing more characteristics with the training data.

Furthermore, camera traps can generate numerous empty photos when they are repeatedly triggered by moving vegetation and direct sunlight (i.e., false triggers). Integrating MegaDetector into the early stages of camera trap image workflows may help economize staff time by reducing the number of images provided to species classification models, therefore

[1] https://github.com/agentmorris/MegaDetector/blob/main/megadetector.md

shortening the delay between data collection and reporting. We present a test case to evaluate MegaDetector's ability to detect wildlife images from camera trap stations across Ontario's managed forests and identify opportunities to integrate MegaDetector into camera trap workflows.

# Test case in Ontario's managed forests

The Ontario Ministry of Natural Resources (MNR) uses camera traps to monitor medium and large mammals across Ontario's managed forest (Figure 1) which contains temperate and boreal tree species. Camera trapping is used to estimate occupancy, rates of use, and demographic composition of wildlife species across time, space, and environmental conditions (e.g., Brown et al. 2020).

As part of the Multiple Species Inventory and Monitoring (MSIM) plot network, the MNR mounted baited camera traps on trees 50 to 75 cm above the ground to continuously monitor wildlife over a minimum of 100 days during the summers of 2013–2018. Reconyx™ PC900 and PC85 cameras recorded 5 images when triggered by a thermal difference between surfaces of a moving object and background objects (Welbourne et al. 2016). Both camera models illuminate objects using an infrared flash.

During this time, individual cameras (n= 584) captured a median of 1,233 (range: 26 to 54,865) images per summer resulting in a network-wide median of 657,085 (range: 365,343 to 832,227) images per year. Manually reviewing and classifying MSIM images took an average of 8 hours per 100 camera days, or about 750 hours per year, which is equivalent to nine work weeks for two full-time staff working eight hours per day. Ministry staff visually examined every image to identify and document the suite of characteristics related to species identity, age, number, sex, and other measures. We refer to this process as *image interpretation*. An important first step in this process was discretizing images into *events*, which included multiple images triggered by the same animal or group of animals. For each event, we confirmed the presence or absence of animals in individual images. Figure 2 illustrates a method for using MegaDetector to exclude empty images from these initial steps that should be readily transferable to most image interpretation workflows.

Motivated to streamline camera trap image interpretation workflows, we used this test case to:

1) test how effectively MegaDetector discriminated between empty and occupied camera trap images from Ontario's managed forests
2) explore how varying the MegaDetector discrimination threshold affected the number of empty images that could be excluded from image interpretation steps involving animals and humans
3) develop an approach to streamline future camera trap image workflows for boreal and temperate forest systems


**Figure 1.** Map of Ontario's managed forest, including location of 10 Multiple Species Inventory and Monitoring (MSIM) camera traps (yellow triangles) that generated the image sets used in the current study.

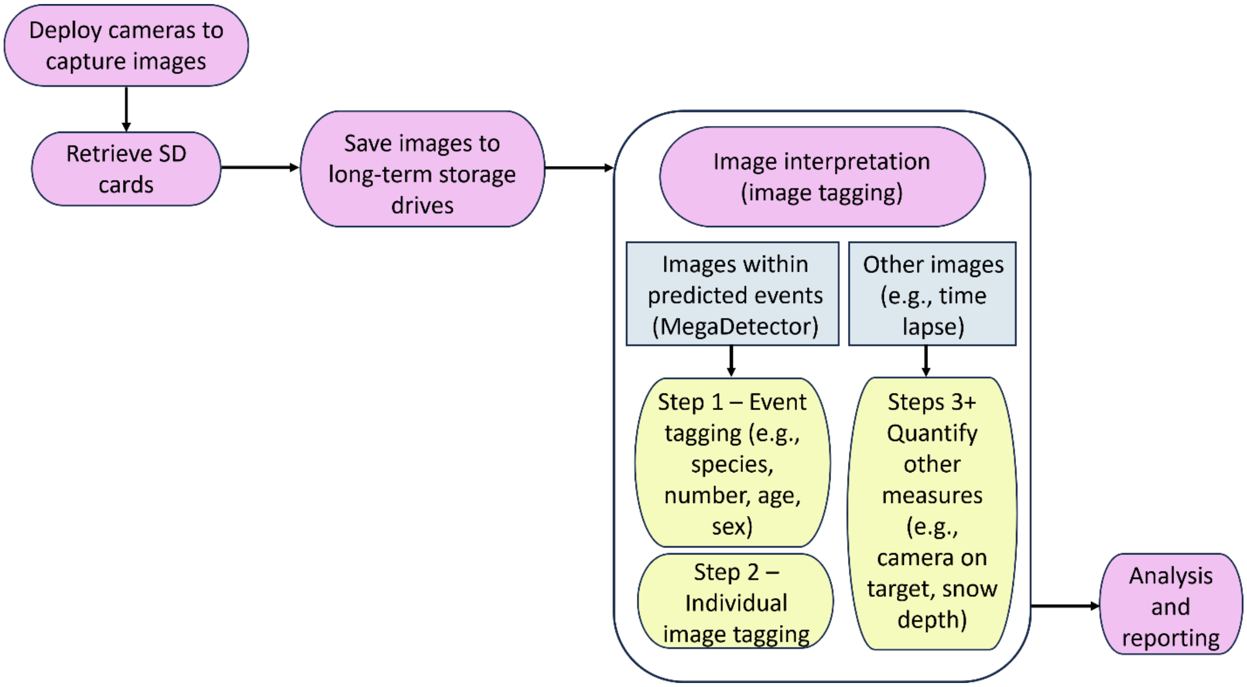


**Figure 2.** A method for incorporating MegaDetector software into camera trap image interpretation workflows. Violet objects represent the main components of the workflow. The image sets provided to interpreters (light blue objects) are identified above the main steps of the image interpretation process (light yellow objects). MegaDetector is used to reduce the number of empty images provided at steps 1–2. Depending on research aims, review of other image sets (e.g., time lapse images) may also be desired to assess whether cameras remain on target, estimate snow depths, and gather other pertinent data (steps 3+).

# Methods and results

## Image set

We used images generated by MSIM efforts from May to October of 2013–2018 to assess MegaDetector's ability to detect wildlife in summer images. Program staff manually tagged images with species presence information, which we assumed were without error and represented truth.

Gains in efficiency from MegaDetector should be proportional to the number of empty images. We therefore selected image sets with a range of empty images. We initially selected the following sets where occupied images included wildlife or humans:

1) "Downer Lk" (Station C1; 2018; 54,865 images; 2% of images occupied)
2) "Cow Rv" (Station C4; 2018; 44,678 images; 14% of images occupied)
3) "Bennett Lk" (Station C4; 2015; 34,815 images; 1% of images occupied)

4) “Tunnel Lk” (Station C4; 2016; 25,886 images; 4% of images occupied)
5) “Scalp Crk Rd” (Station C1; 2016; 25,563 images; 1% of images occupied)
6) “Lawgrave Lk” (Station C1; 2017; 637 images; 90% of images occupied)
7) “Lawagamau Lk” (Station C1; 2018; 1,531 images; 77% of images occupied)

We also assessed whether MegaDetector exhibited any biases in detecting specific wildlife species targeted by camera traps in Ontario’s forests. The initial images included <10 mustelids (weasel family) and no moose (*Alces alces*) so we added three sets with higher numbers of these species:

8) “Barrett” (Station C2; 2013; 1,724 images)
9) “Northshore” (Station C3; 2018; 2,824 images)
10) “Valere Lk” (Station C2; 2016; 827 images)

The combined data set included 193,350 images. Our selection covered the range of variation in occupied image proportions across 1,370 available image sets (Figure 3) and were thus expected to reflect the range of benefits that might be realized by incorporating MegaDetector into our workflow. Before analysis, we removed 124 images where animal presence could not be conclusively determined due to conditions such as extreme fog, blur, or lack of focus (~0.06% of images). The full data set consisted of 193,226 images across the 10 locations.

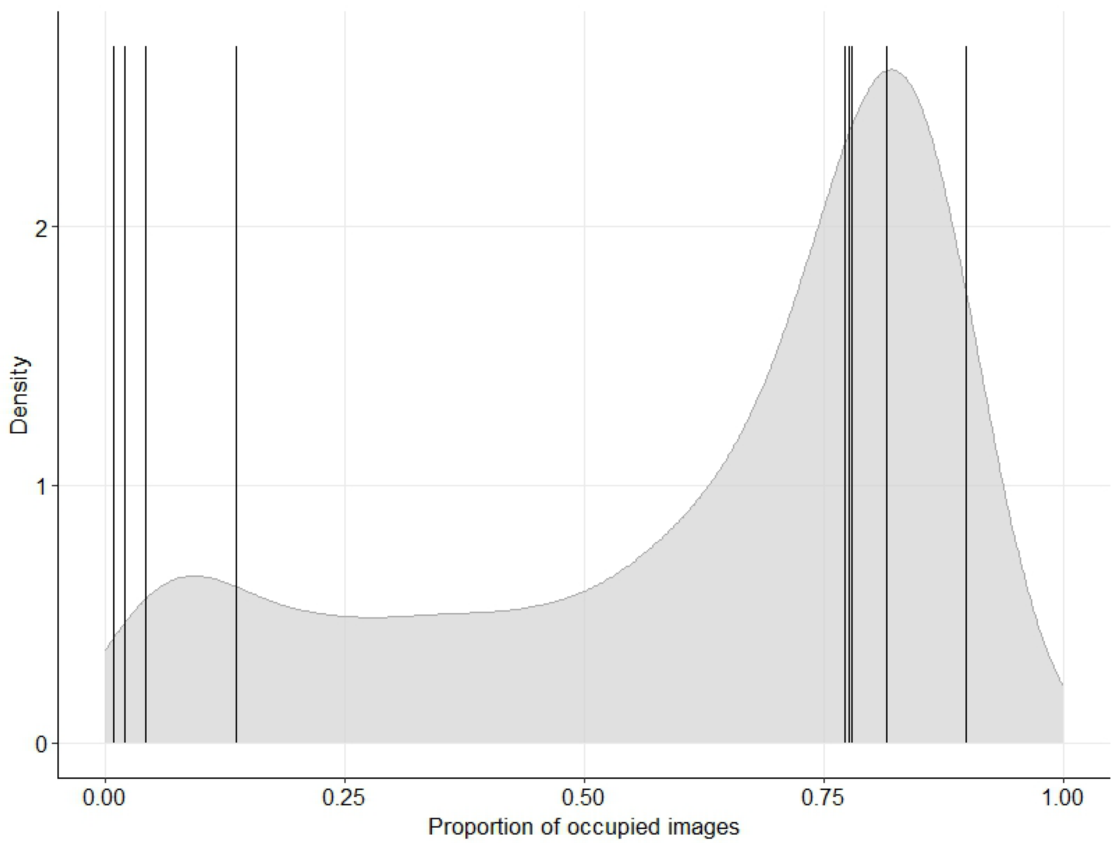


**Figure 3.** Probability density of Multiple Species Inventory and Monitoring camera trap image sets (n = 1,370) as a function of the proportion of images occupied by humans or wildlife. Vertical lines represent the ten image sets used in this study.

# Applying MegaDetector

We applied MegaDetector to our image sets and it analyzed 193,226 images in 12.5 hours, averaging ~4.3 images/second. See Appendix 1, Section 1 for instructions on setting up and running MegaDetector and computer specifications used in this study.

We used the term *image* to describe an entire photograph and *detections* to describe objects within images. A total of 272,333 detections were generated with at least one detection in 120,013 images. The remaining 73,213 images were categorized as empty. MegaDetector assigned confidence levels to detections ranging from 0.005 to 0.987. Detections were classified as animal (98%), person (~2%), or vehicle (<1%), and all three detection categories were retained in our analyses. We uploaded images and detection information from MegaDetector into Timelapse2[2] to create an SQLite database and visualize MegaDetector output (see Appendix 1, Section 2 for details).

Next, we extracted MegaDetector detections and associated confidence levels from the SQLite database and joined them to the manually interpreted data in R (R Core Team 2024) using the tidyverse package (Wickham et al. 2019). Images containing multiple species (65 images) were previously duplicated in the data to capture information for each species separately. Furthermore, MegaDetector generated multiple detections for 70,183 images. Therefore, we specified a 'many-to-many' relationship when joining MegaDetector detections with manually interpreted data to accommodate the data structure. For example, an image manually tagged as containing two different species (two rows of data for the image) for which MegaDetector generated four different detections (four rows of data for the image) produced eight rows of data in the resulting join.

We then created a binary column identifying an image as empty ("0") or occupied by target wildlife, human(s), or vehicle(s) ("1") based on the original image tags. Target species were those of management interest in Ontario and captured effectively by the MSIM camera trap protocol. Target species included all mammals in the families Canidae, Cervidae, Erethizontidae, Felidae, Leporidae, Mephitidae, Mustelidae, Procyonidae, Ursidae; woodchuck (*Marmota monax*); upland game birds; and American woodcock (*Scolopax minor*). Images with non-target species (n = 2,120) were considered empty for the purposes of this study.

All R code used in this project is documented in a script (.R) and available upon request from the authors.

# Evaluating performance

## MegaDetector discrimination ability

A single detection in our investigation had one of four conditions (Figure 4). True positive (TP) indicated agreement between MegaDetector predictions and the presence of target species and true negative (TN) indicated agreement between MegaDetector and the absence of target species. False positive (FP) indicated that MegaDetector incorrectly predicted target species

[2] Timelapse: from camera trap images to data

presence and false negative (FN) indicated MegaDetector failed to detect a target species that was present (a "miss"). Whether or not MegaDetector predicted an image to be occupied depended on the detection confidence level used to mark the decision boundary between empty and occupied images (hereafter the *discrimination threshold*). Different discrimination thresholds can affect the number and type of classification errors in the images. For example, Figure 4a includes two detections with confidence levels of 0.88 and 0.94. Thus, a discrimination threshold <0.88 would lead to two true positives, >0.88 and <0.94 would lead to one true positive and one false negative, and >0.94 would lead to two false negatives.

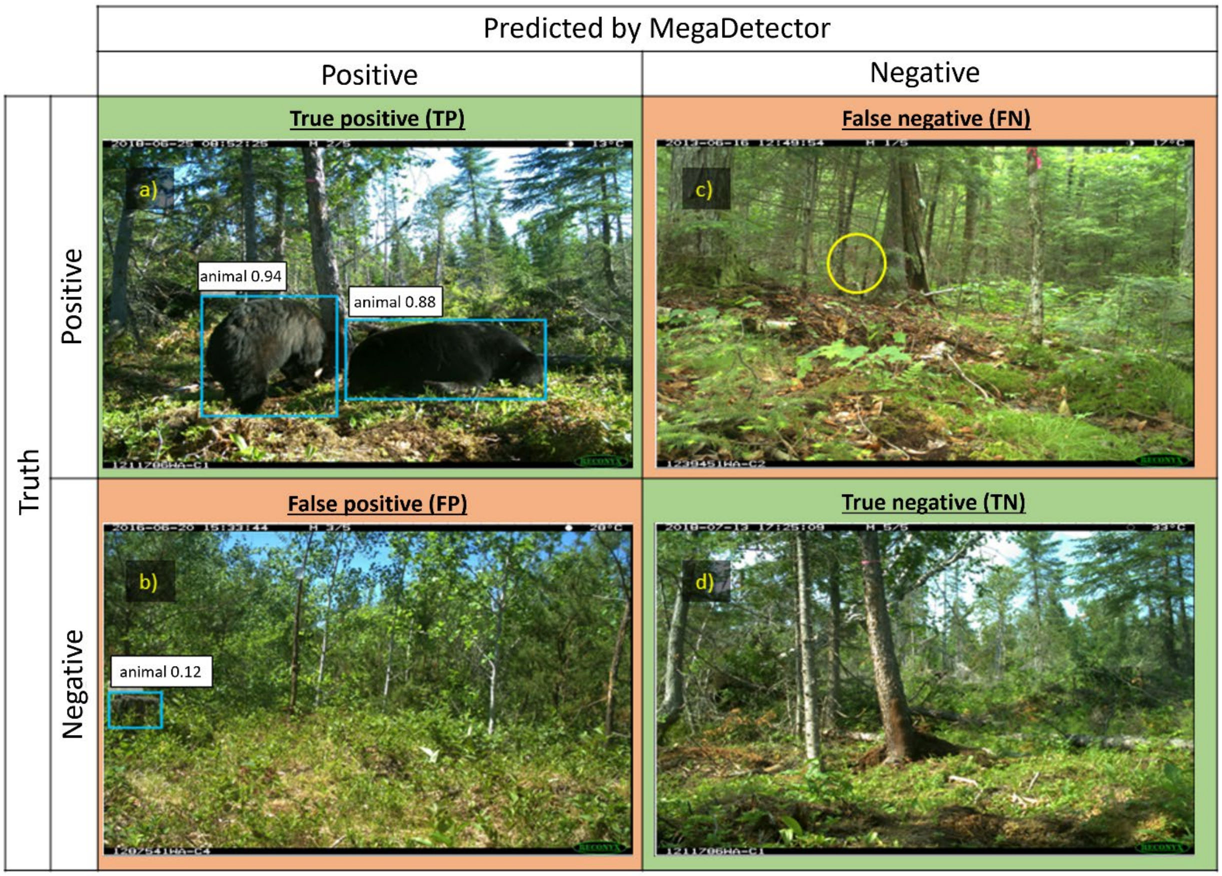


**Figure 4.** Examples of a) true positive (TP), b) false positive (FP), c) false negative (FN), and d) true negative (TN) conditions as defined in this report. Blue squares are bounding boxes generated by MegaDetector, with associated detection confidence levels presented above each box and enlarged for clarity in panels a) and b). In panel c), the yellow circle highlights a partially obscured white-tailed deer (*Odocoileus virginianus*) missed by MegaDetector. A single image may have multiple detections as in panel a).

We used receiver operating characteristic (ROC) curves to assess MegaDetector's discriminatory performance at detecting occupied images (Fawcett 2006). ROC curves illustrate the trade-off between true positives (TP) and false positives (FP) for a given classification model[3] and can be used to compare models and identify optimal discrimination thresholds. ROC curves plot sensitivity (also called true positive rate or TPR) against specificity (also called true negative rate or TNR) of a classifier:

$$sensitivity = TPR = \frac{TP}{(TP+FN)} = 1 - \frac{FN}{(TP+FN)}$$

$$specificity = TNR = \frac{TN}{(TN+FP)} = 1 - \frac{FP}{(TN+FP)}$$

The area under the ROC curve (AUC) provides an aggregate performance measure across a range of potential discrimination thresholds. A perfect classification model has an AUC of 1 whereas a random classification model lacking discriminatory ability has an AUC of 0.5.

We plotted ROC curves for the full range of MegaDetector discrimination thresholds using the full data set and by camera using the pROC package in R (Robin et al. 2011). The full data set had an AUC of 0.82 (Figure 5), a value often considered to represent good discrimination (Hosmer et al. 2013). The AUC of individual cameras ranged from poor (0.58 at Northshore) to excellent (0.91 at Bennett Lk) discrimination. Northshore images included many high confidence detections of non-target wildlife such as rodents and passerines, yielding many false positives of target species. For the full data set, the highest sensitivity (0.99) was attained using the lowest discrimination threshold available in MegaDetector (0.005). As the most risk-averse option, a sensitivity of 0.99 would allow us to remove 73,213 images (TN + FN images), or 38% of the original data set, at a cost of 179 missed wildlife (1.4% of all occupied images) (Table 1).

Using a 0.005 discrimination threshold, the mean number of false positives increased by 57 per 100 images ($F_{1,8} = 71.39$, $p < 0.001$, $R^2 = 0.90$) (Figure 6). Although Tunnel Lk and Scalp Crk Rd plots captured a similar number of total images, a discrimination threshold of 0.005 resulted in over 4,000 more false positive images and over 24,000 more false positive detections at Tunnel Lk (Table 1). A visual scan of MegaDetector's output suggested that more foliage and stumps were misclassified as wildlife at the Tunnel Lake location (*artifacts* per Greenberg 2020) relative to both Scalp Crk Rd and Bennett Lk (e.g., Figure 4b).

[3] Classification models in the ROC context usually refer to a method that attempts to predict the class or category of new observations rather than artificial intelligence-driven image classification models that identify animals within images into species or species groups.

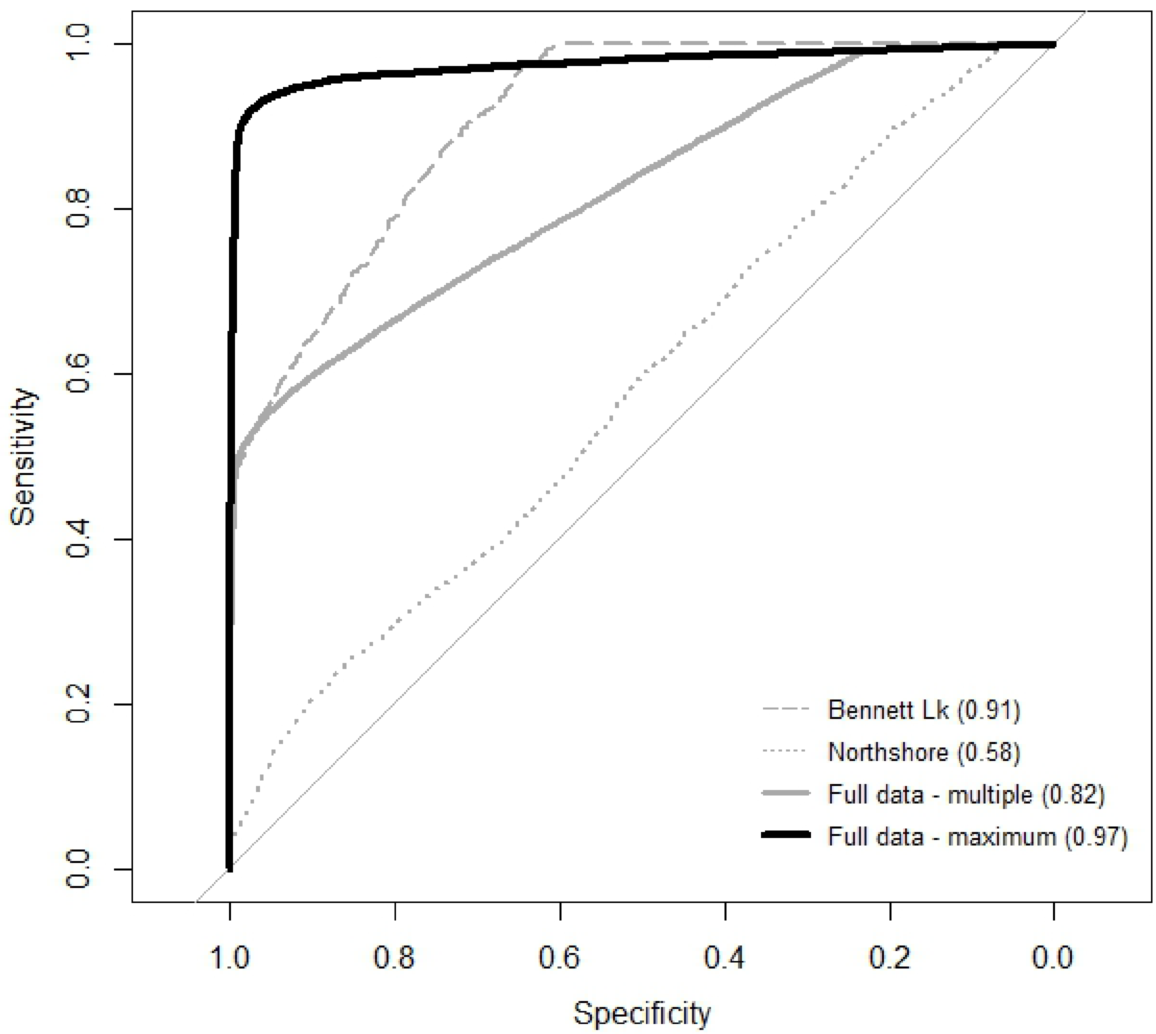


**Figure 5.** Receiver operating characteristic (ROC) curves showing MegaDetector's performance over the entire range of confidence level discrimination thresholds for Bennett Lk and Northshore cameras, the full data set of multiple detections per image (full data – multiple), and full data set of maximum confidence detection per image (full data – maximum). Area under the curve (AUC) for each data set is provided in parenthesis in the legend. The thin, straight grey diagonal line represents the expected performance of a random classifier.

**Table 1**. A summary of the classification results for individual cameras (multiple detections per image), the full data set of multiple detections per image (full data – multiple), and the full data set of maximum confidence detection per image (full data – maximum) using a MegaDetector discrimination confidence level threshold of 0.005. Occupied images contain wildlife or human(s) based on manual classifications, whereas detections are the total number of detections generated by MegaDetector in all images (the sum of true positive and false positive detections). Images tagged with two species (65 cases) were duplicated within the data set. True negative (TN) and false negative (FN) detections represent a MegaDetector assignment of 'empty' and are equal to TN and FN image counts respectively, except in the case of the Lawagamau Lk plot where two duplicated multi-species images within a wildlife event were manually classified as empty.

| Data | Total images | Occupied images | Detections | True positive (TP) | | False positive (FP) | | True negative (TN) | | False negative (FN) | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Detections | Images | Detections | Images | Detections | Images | Detections | Images |
| Downer Lk | 54,865 | 936 | 77,792 | 1,478 | 915 | 76,314 | 31,975 | 21,954 | 21,954 | 21 | 21 |
| Cow Rv | 44,659 | 5,947 | 79,711 | 12,846 | 5,884 | 66,865 | 28,439 | 10,273 | 10,273 | 63 | 63 |
| Bennett Lk | 34,815 | 264 | 17,137 | 739 | 264 | 16,398 | 9,584 | 24,967 | 24,967 | 0 | 0 |
| Tunnel Lk | 25,880 | 883 | 55,713 | 1,992 | 869 | 53,721 | 19,719 | 5,278 | 5,278 | 14 | 14 |
| Scalp Crk Rd | 25,542 | 130 | 29,521 | 181 | 130 | 29,340 | 15,586 | 9,826 | 9,826 | 0 | 0 |
| Lawagamau Lk | 1,527 | 1,170 | 2,125 | 1,751 | 1,156 | 374 | 175 | 184 | 182 | 14 | 14 |
| Lawgrave Lk | 624 | 557 | 981 | 874 | 554 | 107 | 47 | 20 | 20 | 3 | 3 |
| Barrett | 1,674 | 1,231 | 2,522 | 2,178 | 1,176 | 344 | 209 | 234 | 234 | 55 | 55 |
| Northshore | 2,820 | 866 | 5,695 | 2,172 | 860 | 3,523 | 1,724 | 230 | 230 | 6 | 6 |
| Valere Lk | 820 | 546 | 1,136 | 768 | 543 | 368 | 204 | 70 | 70 | 3 | 3 |
| Full data - multiple | 193,226 | 12,530 | 272,333 | 24,979 | 12,351 | 247,354 | 107,662 | 73,036 | 73,034 | 179 | 179 |
| Full data - maximum | 193,226 | 12,530 | 120,076 | 12,351 | 12,351 | 107,725 | 107,662 | 73,036 | 73,034 | 179 | 179 |

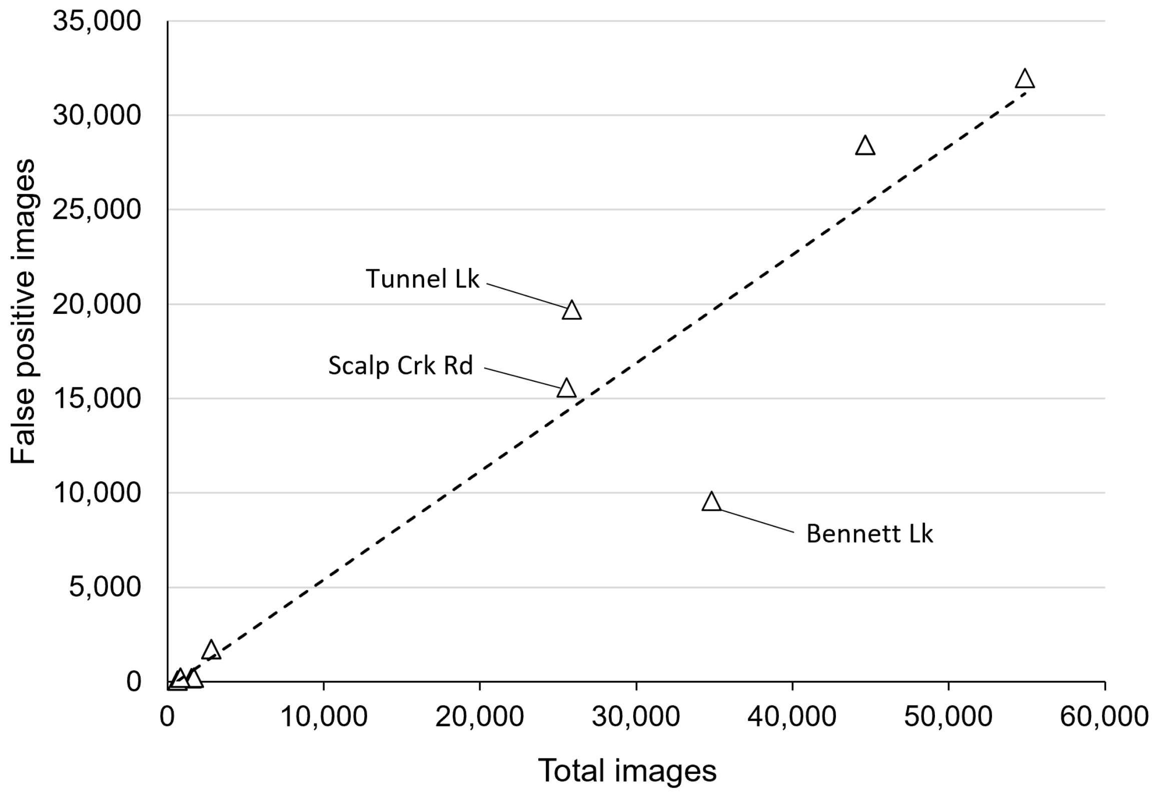


**Figure 6.** Positive relationship between total and false positive images as classified by MegaDetector for individual camera data sets at a discrimination threshold of 0.005.

## Maximum confidence level per image

MegaDetector can generate multiple detections per image. This feature can help locate multiple animals within an image but creates situations where a single image is assigned multiple conditions (e.g., TP and FP). Often, biologists are initially interested in identifying images where at least one animal occurs to reduce the number of images requiring manual examination. We found that retaining images based on the maximum confidence level in an image simplified data processing and improved performance (AUC = 0.97; Figure 5) by eliminating numerous FP and FN cases (Table 1). The remainder of our analyses used the full data set with single maximum confidence level per image.

## Event integration

Many biologists are interested in the occurrence of independent wildlife events captured by a camera trap. In the MSIM image interpretation process, we defined *event* as a group of photographs where the camera was triggered by the same animal or group of animals. A key criterion for distinguishing a new event was a ≥30-minute gap between photographs containing the same individual(s). In our data set, events often spanned many images, some of which showed the animal and some that did not, as the animal moved in and out of the camera's field of view. In practice, the number of true events is unknown until images filtered by MegaDetector are manually reviewed. We therefore developed a conceptual model to remove true and false negative images while maintaining image continuity of predicted events within the image interpretation workflow (Figure 7).

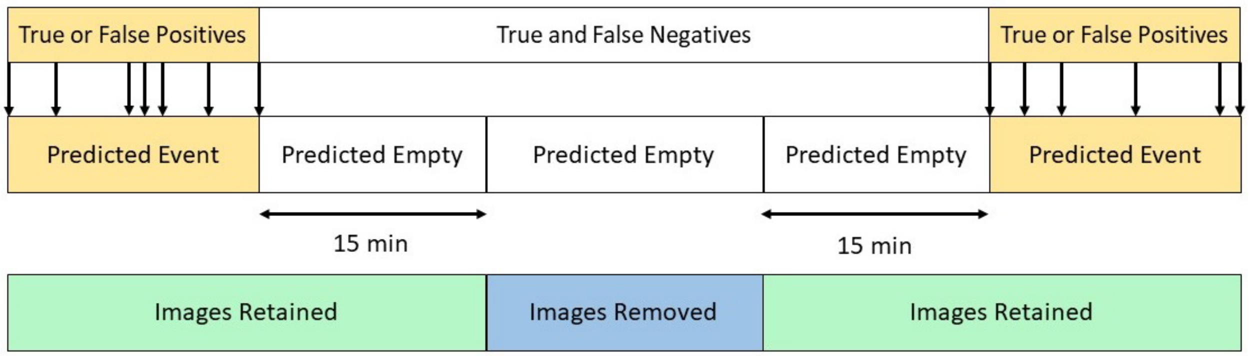


**Figure 7.** Conceptual model ensuring camera trap photographs predicted as empty by MegaDetector were not part of any predicted event (see text for definition of *event*). Downward arrows depict the true or false positive images identified by MegaDetector that define predicted events. True and false negative images situated >15 minutes from a predicted event boundary are identified for removal from Steps 1 and 2 of camera trap image interpretation.

We quantified the benefits and costs of integrating predicted events in future workflows. *Benefits* are the number of predicted empty images (i.e., true and false negatives) identified for removal, and *costs* are the number of false negatives removed. To determine if an image should be removed, we: (1) identified all images where the maximum confidence level per image was less than the discrimination threshold using the ROC curve for the full data set, and (2) removed all identified images that were >15 minutes from predicted positive images (Figure 7).

We calculated benefits and costs across the full range of sensitivities and corresponding discrimination thresholds (Table A2.1). Decreasing sensitivity increased the number of images removed near sigmoidally while the number of wildlife images missed increased exponentially (Figure 8). The resulting curves demonstrated that reducing sensitivity from 0.99 to ~0.86 yielded high benefit while maintaining reasonably low risk of <1% of occupied images missed. A sensitivity of 0.86 (discrimination threshold of 0.67) would have allowed us to remove 160,395 images or 83% of the original data set at a cost of 104 images with missed wildlife (0.8% of occupied images comprising 23 events) (Table A2.1).

The benefit-cost ratio varied at high sensitivities (Figure A2.1) and rapidly declined below a sensitivity of 0.92 (discrimination threshold of 0.43). This decline was caused by diminishing returns on the number of images that could be removed from the image interpretation process with concurrent exponential growth in missed wildlife (Figure 8).

The benefit-cost ratios for a given sensitivity varied among plots. Consistent with results presented in Table 1, the proportion of images removed when considering predicted event boundaries was greatest for plots dominated by empty images (Table 2). At selected sensitivity of 0.86 (discrimination threshold of 0.67), the number of images with missed wildlife ranged from 0 (Lawgrave Lk) to 56 (Barrett). Unknown mammal species were most frequently missed (45 images) followed by snowshoe hare (*Lepus americanus*) (25 images).

Manual review indicated that wildlife in missed images were obscured by other objects, occupying a tiny fraction of the frame, in poor focus, or in poor contrast with the background (e.g., Figure 4c). After controlling for frequency of images by species in our data set, MegaDetector was least proficient at detecting unknown mammal species, followed by white-tailed deer (*Odocoileus virginianus*), fisher (*Pekania pennanti*), and American marten (*Martes americana*) (Table 3). Image interpreters frequently applied the unknown mammal species class to small and obscure mammals within images (e.g., Figure A2.2), so poor discrimination for this class was unsurprising. Human images were relatively common in our data set but never missed by MegaDetector at a discrimination threshold of 0.67.

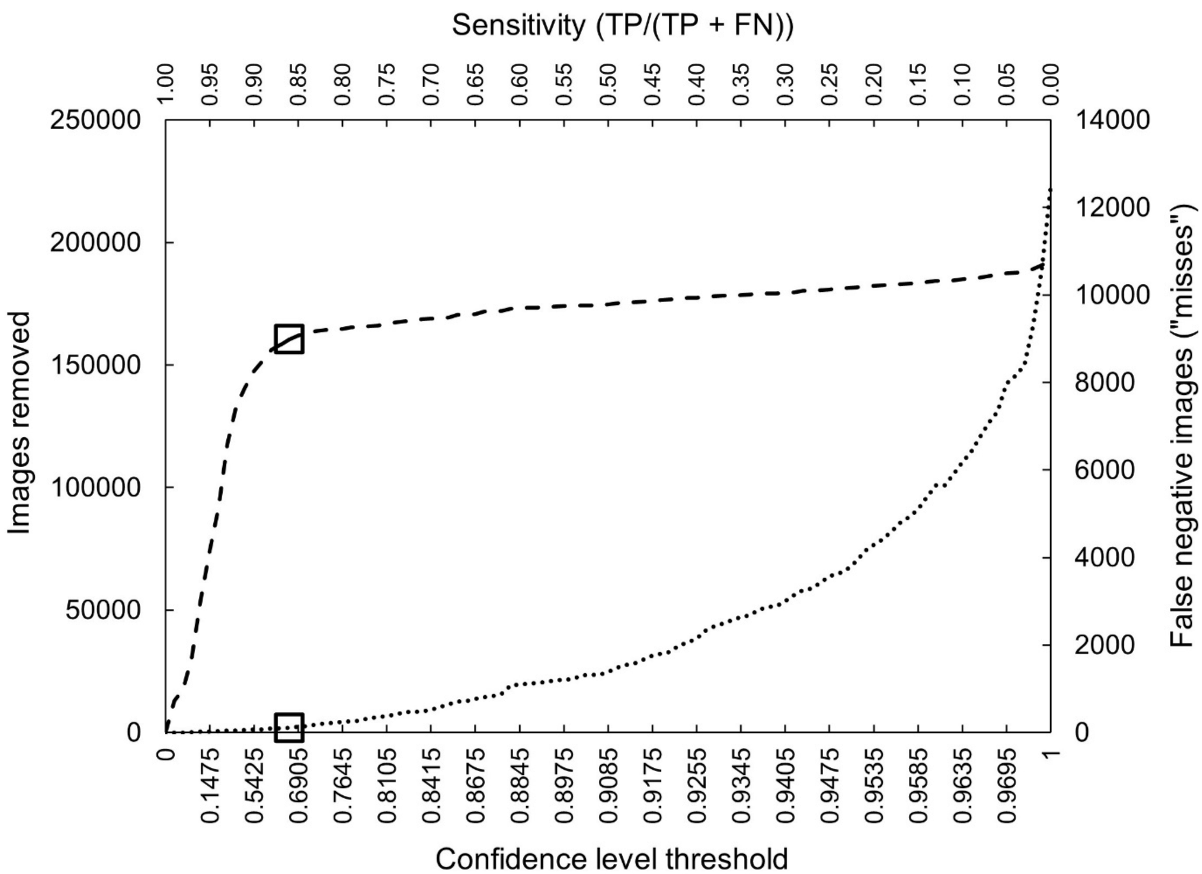


**Figure 8.** Accumulation curves showing the number of photos that could be removed prior to manual review in step 1 of the image interpretation process (dashed line) and number of false negative images (dotted line) across the full range of MegaDetector discrimination thresholds while integrating predicted events. The open squares correspond to a favourable benefit-cost at sensitivity of 0.86 and discrimination threshold of 0.67.

**Table 2.** The camera-specific number (#) and percentage (%) of total images that could be removed from camera trap image workflow, false negative images (misses), and associated benefit-cost ratio when applying MegaDetector at a sensitivity of 0.86 and discrimination threshold of 0.67 while integrating predicted events. Misses are reported separately for six taxa and all species combined, while percentage of occupied images missed ((# misses/# occupied images)*100) is only reported for all species.

| **Plot name** | **Images removed** | **False negative images (misses)** | | | | | | | **Benefit-cost ratio** |
|---|---|---|---|---|---|---|---|---|---|
| | **# (% of total images)** | **Unknown mammal species** | **Snowshoe hare** (*Lepus americanus)* | **American black bear** (*Ursus americanus)* | **Fisher** *(Pekania pennanti)* | **White-tailed deer** *(Odocoileus virginianus)* | **American marten** *(Martes americana)* | **All species # (% of occupied images)** | |
| **Downer Lk** | 45,182 (82) | 0 | 5 | 0 | 0 | 0 | 0 | 5 (0.5) | 9036 |
| **Cow Rv** | 34,332 (77) | 0 | 3 | 0 | 0 | 0 | 0 | 3 (0.1) | 11444 |
| **Bennett Lk** | 33,234 (95) | 5 | 0 | 0 | 0 | 0 | 0 | 5 (1.9) | 6647 |
| **Tunnel Lk** | 22,584 (87) | 0 | 10 | 0 | 0 | 0 | 0 | 10 (1.1) | 2258 |
| **Scalp Crk Rd** | 24,486 (96) | 0 | 0 | 3 | 0 | 0 | 0 | 3 (2.3) | 8162 |
| **Lawagamau Lk** | 175 (11) | 0 | 5 | 0 | 0 | 0 | 0 | 5 (0.4) | 35 |
| **Lawgrave Lk** | 15 (2) | 0 | 0 | 0 | 0 | 0 | 0 | 0 (0) | *N/A* |
| **Barrett** | 147 (9) | 30 | 0 | 0 | 13 | 8 | 5 | 56 (4.5) | 3 |
| **Northshore** | 180 (6) | 0 | 2 | 0 | 0 | 0 | 0 | 2 (0.2) | 90 |
| **Valere Lk** | 60 (7) | 10 | 0 | 0 | 0 | 0 | 5 | 15 (2.7) | 4 |

**Table 3.** Number and percentage of false negative images (misses) by species when applying MegaDetector at a sensitivity of 0.86 and discrimination threshold of 0.67 while integrating predicted events.

| Common name | Scientific name | False negative images (misses) | Total images | % Missed |
|---|---|---|---|---|
| **Snowshoe hare** | *Lepus americanus* | 25 | 6,748 | 0.4 |
| **American black bear** | *Ursus americanus* | 3 | 1,863 | 0.2 |
| **Human** | *Homo sapiens* | 0 | 1,343 | 0.0 |
| **American Marten** | *Martes americana* | 10 | 820 | 1.2 |
| **Fisher** | *Pekania pennanti* | 13 | 479 | 2.7 |
| **Spruce Grouse** | *Falcipennis canadensis* | 0 | 339 | 0.0 |
| **White-tailed Deer** | *Odocoileus virginianus* | 8 | 227 | 3.5 |
| **Northern Raccoon** | *Procyon lotor* | 0 | 216 | 0.0 |
| **Wolf** | *Canis lupus* | 0 | 147 | 0.0 |
| **Unknown Mammal Species** | *N/A* | 45 | 140 | 32.1 |
| **Moose** | *Alces alces* | 0 | 96 | 0.0 |
| **Canada Lynx** | *Lynx canadensis* | 0 | 25 | 0.0 |
| **Unknown Grouse Species** | *N/A* | 0 | 23 | 0.0 |
| **Ruffed Grouse** | *Bonasa umbellus* | 0 | 20 | 0.0 |
| **Coyote or Wolf** | *N/A* | 0 | 12 | 0.0 |
| **Coyote** | *Canis latrans* | 0 | 9 | 0.0 |
| **Red Fox** | *Vulpes vulpes* | 0 | 8 | 0.0 |
| **Woodchuck** | *Marmota monax* | 0 | 5 | 0.0 |
| **Porcupine** | *Erethizon dorsatum* | 0 | 5 | 0.0 |
| **Long-tailed Weasel or Ermine** | *Mustela frenata* or *Mustela erminea* | 0 | 5 | 0.0 |

# Discussion

MegaDetector effectively discriminated between images with and without wildlife in Ontario's managed forests and offers potential to streamline image interpretation workflow. Eighty-three percent of images in our full data set could be removed from our proposed workflow at a low cost of missing <1% of occupied images (Figure 8). For Ontario's managed forests, we suggest that MegaDetector's utility in image interpretation workflow is optimized at a classification sensitivity of 0.86, corresponding to the MegaDetector-assigned confidence level of 0.67. Notionally, MegaDetector could have reduced the time required to manually interpret this data set from 80 to 14 person-hours. Time savings for other image sets will depend on numerous factors including the discrimination threshold selected, proportion of empty images, and the specific image interpretation process.

We focused our analyses primarily on medium to large-bodied mammals. Optimal species-specific discrimination thresholds may vary with animal body size and present a useful area of future research. Because false negatives can bias research and monitoring results, scientists may minimize discrimination thresholds. However, depending on the species and response variable of interest (Whytock et al. 2021) and whether detection error rates can be incorporated (Tabak et al. 2020) in downstream analyses, it may also be reasonable to select discrimination thresholds that miss more than 1% of occupied images.

Our image set's benefit-cost ratio was volatile at high sensitivities (1 to 0.92; Figure A2.1), which likely arose from the grouped nature of motion-triggered wildlife images. For example, a single event can contain many images with consistent MegaDetector-assigned confidence levels. Decreasing sensitivity from 1 to 0.92 therefore resulted in a step like accumulation of false negatives while the number of images removed accumulated gradually. Biologists should be aware that realized benefit-cost ratios can vary widely at high sensitivities when developing protocols that incorporate MegaDetector into workflows.

MegaDetector sometimes produced copious false positives within single images taken within Ontario's managed forests. For example, in an extreme case, an empty image was assigned 19 unique detections with confidence levels ranging from 0.005 to 0.22. While several studies described factors inhibiting target wildlife detection (Villa et al. 2017, Beery et al. 2018, Norouzzadeh et al. 2018, Schneider et al. 2020), less attention has been paid to factors that lead to false positives arising from persistent artifacts (Greenberg 2020). One option is to eliminate detections situated in the same location across many consecutive images[4]. Our approach to use the maximum confidence level assigned to each image dramatically improved classification performance to 0.97 (ΔAUC = 0.15) while simplifying data processing. This approach is consistent with the image filtering process of other software (Greenberg 2023) and our aim to minimize the risk of missing images containing wildlife at the cost of having more images to inspect later in the workflow. Biologists that use either method should clearly

[4] MegaDetector/megadetector/postprocessing/repeat_detection_elimination at main · agentmorris/MegaDetector · GitHub

describe whether errors relate to images or individual detections within images, particularly when assessing aggregate errors via ROC analyses.

We found that MegaDetector discrimination performance varied across space and species. Manual review of bounding boxes revealed that false positives were often associated with stumps (e.g., Figure 4b), foliage, and boulders. Some camera-level variation (Figure 5) was also driven by MegaDetector identifying non-target wildlife species. Although detectability of small-bodied animals is hampered by their small surface area relative to the camera sensor array (Welbourne et al. 2016), these species occasionally triggered the camera traps used in this study. MSIM cameras were deployed 50 to 75 cm above ground level. Adjusting camera height and angle to maximize detection of taller species should reduce the number of images with smaller-bodied species (Tobler et al. 2008, Meek et al. 2016, Jacobs and Ausband 2018, DeWitt and Cocksedge 2023). Nonetheless, we continued to find evidence of variable performance among the target mammal species after integrating predicted events (Table 3). Our sample size for some target species was small; further testing with balanced species representation in images may help elucidate biases among both species and locations.

Our efforts to evaluate MegaDetector in the context of predicted events substantially increased the operational benefits of using the image recognition model. We initially removed 38% of images (n=73,213) at a cost of missing 179 containing wildlife when optimizing confidence for individual images (confidence level = 0.005). In contrast, integrating predicted events allowed us to remove 83% of images (n=160,395) at a cost of missing 104 containing wildlife (confidence level = 0.67). For comparison, 1,742 wildlife images were missed when applying a confidence level threshold of 0.67 without integrating predicted events. Camera trap images containing wildlife tend to cluster through time (Sollmann 2018) and MegaDetector was often successful at detecting at least one wildlife occurrence within an image cluster with relatively high confidence. As a result, integrating predicted events allowed us to leverage the benefits of these high confidence images and decrease the number of images with missed wildlife for a given sensitivity. Exploring temporal clustering of wildlife images within camera trap data could help biologists generate similar rule sets appropriate to their image interpretation process.

Workflow benefits and costs varied by camera (Table 2). As expected, the number and percentage of images identified for removal from the image interpretation process was greatest for plots dominated by empty images. The efficacy of incorporating MegaDetector into camera trap image workflow therefore largely depends on the number of cameras malfunctioning and/or triggered excessively by non-wildlife stimuli. Biologists should also be aware that costs may range considerably across space; at classification sensitivity of 0.86, nearly 5% of wildlife images were missed for one of our study locations (Barrett) (Table 2).

Extreme cold and snow can affect camera traps by reducing camera trigger and recording efficiency, and obscuring the lens, respectively, but few studies have looked at the effects of snow on image recognition models. Snow presumably results in a major domain shift and may affect MegaDetector’s ability to discriminate objects. An earlier version of MegaDetector performed well on camera trap images with snow in Idaho, USA, despite not being trained on such images (Beery et al. 2019). While our study focuses on summer images, we conducted a preliminary test of MegaDetector’s performance on 21,039 snow cover images from 12 cameras deployed since concluding the MSIM effort. Following the methods reported herein, we found that 50% of these images could be removed while only missing 19 of 3,756 images occupied by wildlife (0.5%) (sensitivity = 0.86 and confidence level = 0.16). This preliminary analysis suggests that MegaDetector performs well on images with snow cover although the winter confidence level threshold of 0.16 was notably lower than the summer threshold of 0.67 at the same sensitivity. Further tests could help identify static or seasonally varying confidence level discrimination thresholds that optimize workflows applicable to year-round camera trap studies in boreal and temperate forests.

Introducing automation into camera trap image workflows requires balancing multiple objectives (e.g., Leorna and Brinkman 2022). MegaDetector can help accelerate image processing by detecting wildlife but is not a panacea. For example, the presence of wildlife tracks in time lapse-derived images is sometimes used to assess detection error (Keim et al. 2019) but no tool exists to automate track detection in images. Researchers continue to develop new tools to streamline workflows including automated snow depth measurements (Strickfaden et al. 2023), species classification (e.g., Tabak et al. 2019, Whytock et al. 2021), and distance sampling to support wildlife abundance modeling (Johanns et al. 2022, Henrich et al. 2023). Given the rapid pace of development of such tools, biologists interested in maximizing the information and efficiency of camera trap programs might benefit from modular workflows that can readily incorporate emerging methods.

Others have manually reviewed all predicted empty images (Greenberg 2020, Fennell et al. 2022), however this process may not be feasible for long-term monitoring programs that generate millions of images nor required for analyses where a small number of missed wildlife is acceptable. Where long-term manual review of all empty images is not possible, we recommend assessing and reporting risk based on a representative sample of cameras, as we did herein, and manually inspecting a sample of predicted empty images on an ongoing basis. Biologists may choose to examine a random selection of empty images stratified by discrimination thresholds (e.g., 0.673–0.5, 0.499–0.25, and 0.249–0) and apply greater effort for higher confidence levels to verify error rates more efficiently than a simple random sample.

We showed how biologists can use MegaDetector to streamline camera trap workflows. Based on our analyses, we recommend:

- using the single maximum confidence level detection per image
- identifying optimal discrimination thresholds appropriate to the study system, risk tolerance, and research aim
- leveraging the benefits of temporally clustered wildlife images when possible
- reporting error rates (i.e., the expected mean percentage of occupied images missed in planned workflows) and manually inspecting a subset of predicted empty images

We found that MegaDetector can save time for wildlife management agencies and research teams managing large camera trap data sets. Although the specific costs and benefits vary across image sets, and therefore year and location, others can use our findings as a framework for other camera trap studies in boreal and temperate forests.

# References

Beery, S., G. Van Horn and P. Perona. 2018. Recognition in terra incognita. Pp. 456-473 *in* Ferrari, V., M. Hebert, C. Sminchisescu and Y. Weiss (eds.). Proceedings of the European Conference on Computer Vision. Munich, Germany. 8-14 September 2018. Springer Nature, Berlin, Germany. 871 p.

Beery, S., D. Morris and S. Yang. 2019. Efficient pipeline for camera trap image review. Cornell University, Computer Science, Ithyca, NY. arXiv. < https://arxiv.org/abs/1907.06772> Accessed November 2024.

Brown, G.S., L. Pollock, P.D. DeWitt and N. Dawson. 2020. Responses of terrestrial animals to forest characteristics and climate reveals ecological indicators for sustaining wildlife in managed forests. Forest Ecology and Management 459: 117854.

Fawcett, T. 2006. An introduction to ROC analysis. Pattern Recognition Letters 27(8): 861–874.

Fennell, M., C. Beirne and A.C. Burton. 2022. Use of object detection in camera trap image identification: Assessing a method to rapidly and accurately classify human and animal detections for research and application in recreation ecology. Global Ecology and Conservation 35: e02104.

Glover-Kapfer, P., C.A. Soto-Navarro and O.R. Wearn. 2019. Camera-trapping version 3.0: current constraints and future priorities for development. Remote Sensing in Ecology and Conservation 5(3): 209–223.

Greenberg, S. 2020. Automated image recognition for wildlife camera traps: Making it work for you. University of Calgary, Calgary, AB. Techical report, Prism of Calgary's Digital Repository. 15 p.

Greenberg, S. 2023. Timelapse image recognition guide: A guide to importing and using image recognition data. University of Calgary, Calgary, AB. Part 4 of the Timelapse Manual series. 18 p.

Henrich, M., M. Burgueño, J. Hoyer, T. Haucke, V. Steinhage, H.S. Kühl and M. Heurich. 2023. A semi-automated camera trap distance sampling approach for population density estimation. Remote Sensing in Ecology and Conservation 10(2): 156–171.

Hosmer Jr, D.W., S. Lemeshow and R.X. Sturdivant. 2013. Applied Logistic Regression 3rd Edition. John Wiley & Sons, Hoboken, NJ. 528 p.

Jacobs, C.E. and D.E. Ausband. 2018. An evaluation of camera trap performance – What are we missing and does deployment height matter? Remote Sensing in Ecology and Conservation 4(4): 352–360.

Johanns, P., T. Haucke and V. Steinhage. 2022. Automated distance estimation for wildlife camera trapping. Ecological Informatics 70: 101734.

Keim, J.L., S.R. Lele, P.D. DeWitt, J.J. Fitzpatrick and N.S. Jenni. 2019. Estimating the intensity of use by interacting predators and prey using camera traps. Journal of Animal Ecology 88(5): 690–701.

Leorna, S. and T. Brinkman. 2022. Human vs. machine: Detecting wildlife in camera trap images. Ecological Informatics 72: 101876.

Meek, P.D., G.A. Ballard and G. Falzon. 2016. The higher you go the less you will know: Placing camera traps high to avoid theft will affect detection. Remote Sensing in Ecology and Conservation 2(4): 204–211.

Norouzzadeh, M.S., A. Nguyen, M. Kosmala, A. Swanson, M.S. Palmer, C. Packer and J. Clune. 2018. 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.

R Core Team. 2024. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. Online software. <https://www.R-project.org/>. Accessed January 2024.

Robin, X., N. Turck, A. Hainard, N. Tiberti, F. Lisacek, J.C. Sanchez and M. Müller. 2011. pROC: An open-source package for R and S+ to analyze and compare ROC curves. BMC Bioinformatics 12: 77.

Rovero, F., F. Zimmermann, D. Berzi and P. Meek. 2013. "Which camera trap type and how many do I need?" A review of camera features and study designs for a range of wildlife research applications. Hystrix, the Italian Journal of Mammalogy 24(2): 148–156.

Schneider, S., S. Greenberg, G.W. Taylor and S.C. Kremer. 2020. Three critical factors affecting automated image species recognition performance for camera traps. Ecology and Evolution 10(7): 3503–3517.

Sollmann, R. 2018. A gentle introduction to camera-trap data analysis. African Journal of Ecology 56(4): 740–749.

Steenweg, R., M. Hebblewhite, R. Kays, J. Ahumada, J.T. Fisher, C. Burton, S.E. Townsend, C. Carbone, J.M. Rowcliffe, J. Whittington, J. Brodie, J. Royle, A. Switalski, A. Clevenger, N. Heim and L. Rich. 2017. Scaling-up camera traps: Monitoring the planet's biodiversity with networks of remote sensors. Frontiers in Ecology and the Environment 15(1): 26–34.

Strickfaden, K.M., M.L. Behan, A.M. Marshall, L.K. Svancara, D.E. Ausband and T.E. Link. 2023. Virtual snow stakes: A new method for snow depth measurement at remote camera stations. Wildlife Society Bulletin 47(3): e1481.

Tabak, M. A., M.S. Norouzzadeh, D.W. Wolfson, S.J. Sweeney, K.C. Vercauteren, N.P. Snow, J.M. Halseth, P.A. Di Salvo, J.S. Lewis, M.D. White, B. Teton, J.C. Beasley, P.E. Schlichting, R.K. Boughton, B. Wight, E.S. Newkirk, J.S. Ivan, E.A. Odell, R.K. Brook, P.M. Lukacs, A.K. Moeller, E.G. Mandeville, J. Clune and R.S. Miller. 2019. Machine learning to classify animal species in camera trap images: Applications in ecology. Methods in Ecology and Evolution 10(4): 585–590.

Tabak, M.A., M.S. Norouzzadeh, D.W. Wolfson, E.J. Newton, R.K. Boughton, J.S. Ivan, E.A. Odell, E.S. Newkirk, R.Y. Conrey, J. Stenglein, F. Iannarilli, J. Erb, R.K. Brook, A.J. Davis, J. Lewis, D.P. Walsh, J.C. Beasely, K.C. Vercauteren, J. Clune and R.S. Miller. 2020. Improving the accessibility and transferability of machine learning algorithms for identification of animals in camera trap images: MLWIC2. Ecology and Evolution 10(19): 10374–10383.

Tobler, M.W., S.E. Carrillo-Percastegui, R. Leite Pitman, R. Mares and G. Powell. 2008. An evaluation of camera traps for inventorying large-and medium-sized terrestrial rainforest mammals. Animal Conservation 11(3): 169–178.

Vélez, J., P.J. Castiblanco-Camacho, M.A. Tabak, C. Chalmers, P. Fergus and J. Fieberg. 2022. Choosing an appropriate platform and workflow for processing camera trap data using artificial intelligence. Cornell University, Computer Science, Ithyca, NY. arXiv. < https://arxiv.org/abs/2202.02283> Accessed November 2024.

Villa, A.G., A. Salazar and F. Vargas. 2017. Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks. Ecological Informatics 41: 24–32.

Welbourne, D.J., A.W. Claridge, D.J. Paull and A. Lambert. 2016. How do passive infrared triggered camera traps operate and why does it matter? Breaking down common misconceptions. Remote Sensing in Ecology and Conservation 2(2): 77–83.

Whytock, R.C., J. Świeżewski, J.A. Zwerts, T. Bara-Słupski, A.F. Koumba Pambo, M. Rogala, L. Bahaa-el-din, K. Boekee, S. Brittain, A.W. Cardoso, P. Henschel, D. Lehmann, B. Momboua, C.K. Opepa, C. Orbell, R.T. Pitman, H.S. Robinson and K.A. Abernethy. 2021. Robust ecological analysis of camera trap data labelled by a machine learning model. Methods in Ecology and Evolution 12(6): 1080–1092.

Wickham H., M. Averick, J. Bryan, W. Chang, L.D. McGowan, R. François, G. Grolemund, A. Hayes, L. Henry, J. Hester, M. Kuhn, T.L. Pedersen, E. Miller, S.M. Bache, K. Müller, J. Ooms, D. Robinson, D.P. Seidel, V. Spinu, K. Takahashi, D. Vaughan, C. Wilke, K. Woo and H. Yutani. 2019. Welcome to the tidyverse. Journal of Open Source Software 4(43): 1686.

# Appendix 1 – MegaDetector application

## 1 - Setting up and running MegaDetector©

### Setting Up MegaDetector

We accessed the steps for downloading MegaDetector from Microsoft's MegaDetector User Guide. MegaDetector version 5a (MDv5a.0.0.pt) was downloaded directly from this webpage. Prerequisites for running MegaDetector were also downloaded, including Mambaforge, Git for Windows, and the latest NVIDIA graphics processing unit (GPU) Driver. Relevant specifications of the Lenovo ThinkPad laptop used in this project were as follows:

- Operating System: Windows 10 Professional x64
- RAM: 32 GB
- Processor: 12th Gen Intel(R) Core(TM) i7-12800H 2.40 GHz
- Solid State Drive (SSD) Size: 960 GB
- Graphics Processing Unit (GPU): NVIDIA RTX A2000

Code for cloning three git repositories required for the MegaDetector model was accessed at the now archived MegaDetector setup instructions. In addition to these instructions:

1) the Miniforge Command Prompt was made available through the Mambaforge download.
2) successful download of three required gits (YOLOv5, CameraTraps, and ai4eutils) and running the '.yml' prompt to set up the environment required our laptop's internet connection not be subject to the Ontario Public Service firewall.
3) running the '.yml' prompt took over 30 minutes to complete.

### Running MegaDetector

To run the MegaDetector image detection model, we opened the Miniforge Command Prompt by right clicking on the app and selecting "Run as administrator". The following code was entered and is an example of git repositories saved within a "git" folder on the c:\ drive and the MegaDetector model saved directly to the c:\ drive:

```
cd c:\git\CameraTraps
conda activate cameratraps-detector
set PYTHONPATH=%PYTHONPATH%;c:\git\CameraTraps;c:\git\ai4eutils;c:\git\yolov5
```

Then, the following prompt was used to run a batch of images with the bolded directories customized appropriately:

```
python detection\run_detector_batch.py "c:\megadetector\md_v5a.0.0.pt"
"c:\some_image_folder" "c:\some_image_folder\test_output.json" --
output_relative_filenames --recursive --checkpoint_frequency 10000 --quiet
```

The first bolded portion of the prompt identified the location of images to be analyzed, while the second bolded portion named the .JSON file created by MegaDetector and identified where this file was saved. As an example,

```
python detection\run_detector_batch.py "c:\megadetector\md_v5a.0.0.pt"
"c:\megadetector\TestImages\1328646WA_Bennett Lk_C4"
"c:\megadetector\TestImages\1328646WA_Bennett Lk_C4\1328646WA_Bennett
Lk_C4.json" --output_relative_filenames --recursive --checkpoint_frequency 10000 --
quiet
```

The .JSON file output contained information related to each analyzed image on the presence and/or number of detections, the detection (s) class or category (animal, person, vehicle), the location of each detection as described by bounding box coordinates, and the confidence assigned to each detection (0 or no confidence to 1 or complete confidence).

We repeated this process for all 10 image sets of the test data set. The .JSON file for each set was consistently saved in the same folder as the images analyzed, which facilitated visualization of detection data with Timelapse2, described further below.

# 2 - Visualizing MegaDetector results with Timelapse2©

## Benefits of visualizing detections

This study did not strictly require photographic aids, but visualizing MegaDetector results (i.e., labeled bounding boxes overlaid on images) was informative and beneficial as it:

1) confirmed understanding of the components of a single detection (detection category, confidence level, and bounding box coordinates)
2) allowed verification of photographs with no detections, single detections, or multiple detections at varying detection confidence levels
3) allowed quick verification of R (R Core Team 2024) commands, tables, and other outputs, and/or signaled a need to adjust R code
4) provided visual aids to illustrate points and facilitate knowledge transfer in presentations and reports

Multiple software programs incorporate and display MegaDetector output; MegaDetector's integration with Timelapse2 was claimed to be most mature (MegaDetector setup instructions).

## Installation

Timelapse2 is a tool for viewing, annotating, and sorting images. It incorporates detection information within MegaDetector's .JSON file and other data into SQLite database files (Greenberg 2022).

We downloaded the Timelapse2.ZIP package from the Timelapse website and installed and ran Timelapse2 from our c:\ drive. To function properly, the Timelapse executable (.EXE) files must remain together with other files appearing in the original "Timelapse" folder.

## Creating a template

Prior to viewing any images using Timelapse2.EXE, we created a template using the Timelapse2TemplateEditor application. The template defined the data table's structure and fields and allowed us to customize the information types to be added (tagged) to each image.

In the Timelapse2TemplateEditor, we created a new template by selecting File > New Template. The program immediately asked where to save the template (.TDB) file. To ensure file path recognition by Timelapse2, we located this template file in the same folder that contained all subfolders of images and associated .JSON files used in the project (Figure A1.1). Ultimately, for Timelapse2 and MegaDetector to integrate well, the image file paths must match in both applications (Greenberg 2023a).

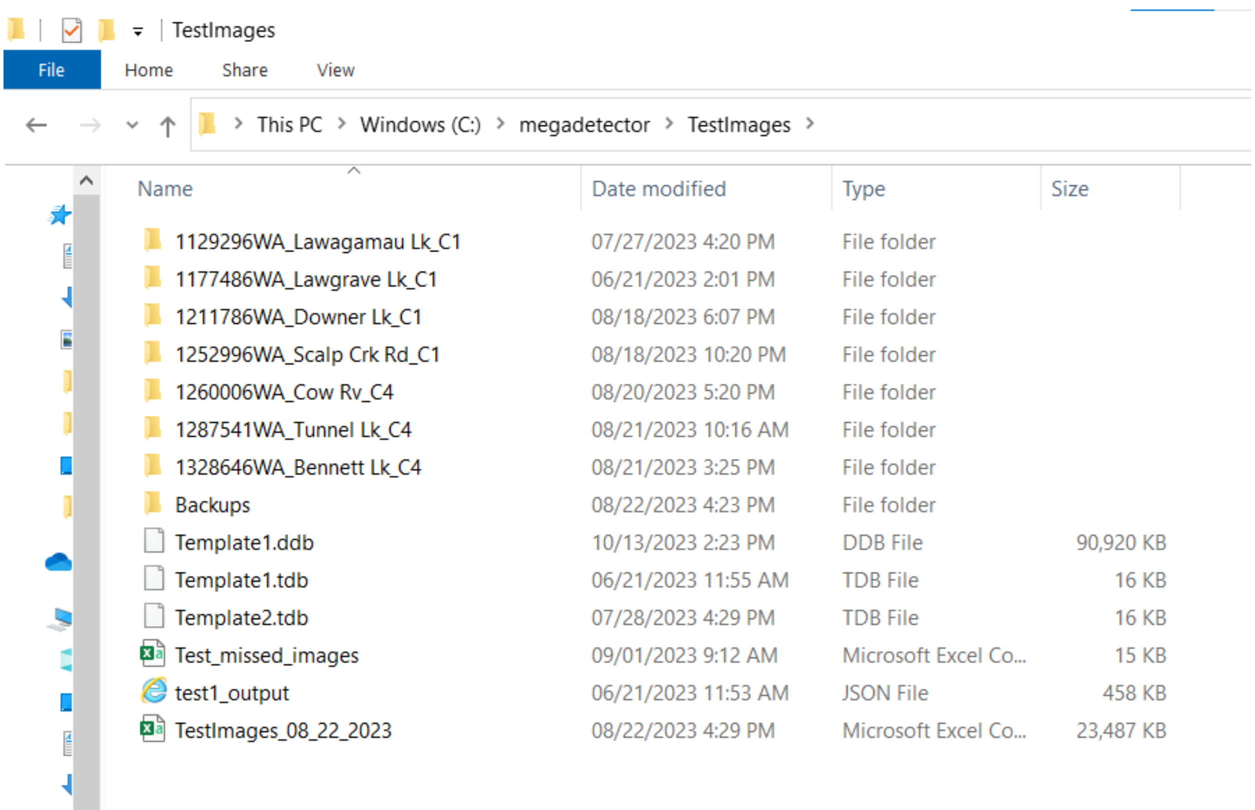


**Figure A1.1.** Folder structure facilitating file path recognition by Timelapse2. Images and associated MegaDetector detection information (.JSON files) are held together within the Multi Species Inventory and Monitoring plot subfolders.

The Timelapse2TemplateEditor allowed us to create and define the interface (image tagging toolbar) and DataTable within Timelapse2. Field types that may be added include Count (positive integers), Choice (user-defined menu of options, such as species names), Note (string), and Flag (true/false). Available image metadata may also be automatically loaded by Timelapse2 into a Note field created by the user. The Timelapse2 developer confirmed it was not possible to incorporate continuous confidence level information into the template, primarily because of the potential for multiple detections per image that complicate the data table (Greenberg 2023, University of Calgary, pers. comm.). While the addition of various fields was explored, we created the simplest (default) template for subsequent image visualization.

A user guide for the Template Editor is available in Greenberg 2023b.

## Viewing images in Timelapse2

Full functionality of Timelapse2 is best described by a series of user guides available online and conveniently linked within the Timelapse2 Help > Guides and manuals menu, particularly the Timelapse Quickstart Guide, Timelapse Reference Guide, and Timelapse Image Recognition Guide.

We performed the following to load, display, sort, and verify images with recognition (detection) information in Timelapse2:

1) After opening Timelapse2, we loaded the template with File > Load Template, Images and Video Files. We used the Windows Explorer pop-up to navigate to our previously created template.
2) We imported images and associated MegaDetector recognition information with Recognitions > Import Images and by navigating to the relevant .JSON file. Image sets for the seven plots of the test data set were analyzed independently by MegaDetector so this step was repeated for each image set and unique .JSON file. If/when a **"**Confirm image path corrections in the recognition file" box appeared, Correct paths was selected to continue. Images were then displayed in the Image Set tab of the Timelapse2 interface (Figure A1.2).
3) Once all images were imported, the total number of image files was displayed in the bottom left corner of the Timelapse interface (193,350; Figure A1.2).
4) We displayed recognitions by selecting Recognitions > Set bounding box options. In the dialogue box, Annotate bounding box was checked and Display bounding boxes at or above this confidence threshold was set to 0.00.
5) We filtered (subset) images using Select > Custom Selection. In the Image Recognition portion of the dialogue box, we checked Use Recognition and the desired Recognized entity (All, Empty, Animal, Person, Vehicle) and Confidence range could then be specified.
6) We frequently used the Data Table tab to locate specific images by name to confirm alignment with R commands, tables, and outputs. Clicking on the data row containing the image of interest immediately displayed the image on the Image Set tab.

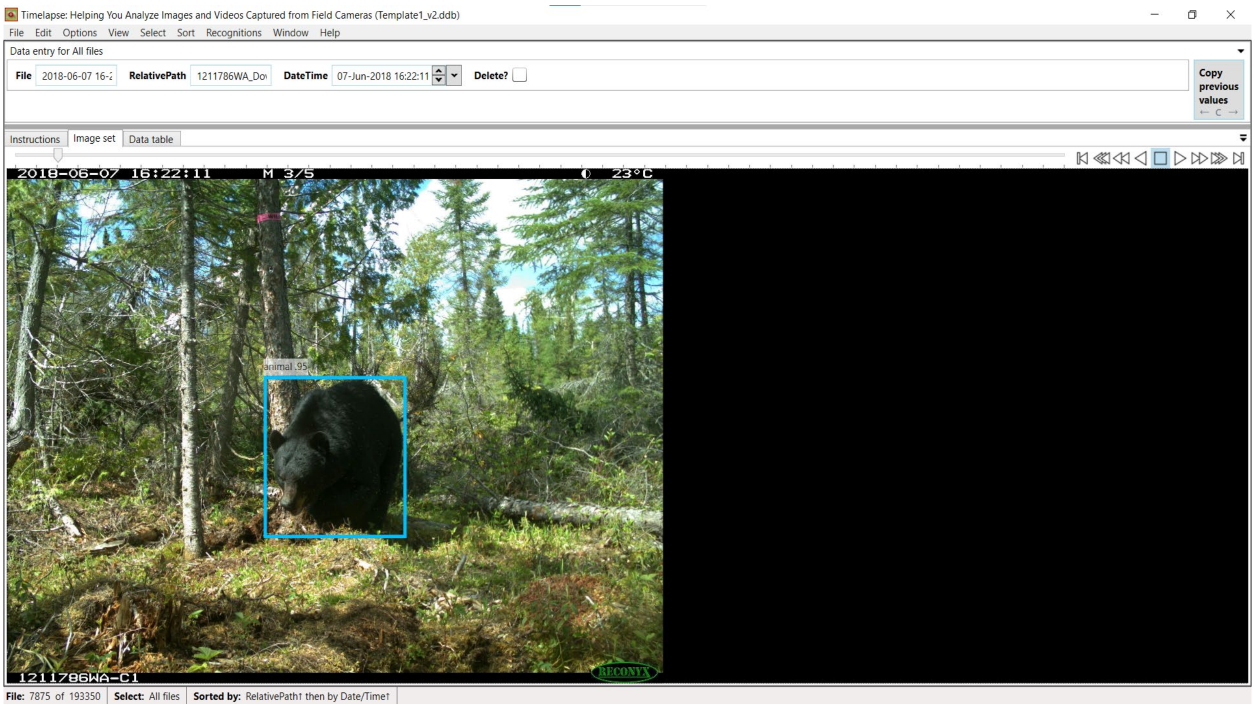


**Figure A1.2.** Timelapse2 interface displaying main menu, some image metadata, image set navigation options, total image (file) count (193,350), and a camera trap image with blue bounding box and detection category (animal) and confidence level (0.95) labels.

## References


Greenberg, S. 2022. Timelapse database guide: A guide to the internal structure of Timelapse database tables. University of Calgary, Calgary, AB. Part 5 of the Timelapse Manual series. 9 p.

Greenberg, S. 2023a. Timelapse image recognition guide: A guide to importing and using image recognition data. University of Calgary, Calgary, AB. Part 4 of the Timelapse Manual series. 18 p.

Greenberg, S. 2023b. Timelapse template guide: A reference guide to Timelapse templates, and how to create them with the Timelapse template editor. University of Calgary, Calgary, AB. Part 3 of the Timelapse Manual series. 9 p.

R Core Team. 2024. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. Online software. <https://www.R-project.org/>. Accessed January 2024.

# Appendix 2 – MegaDetector benefits and costs

**Table A2.1.** Benefits and costs across the full range of MegaDetector classification sensitivities and discrimination thresholds while integrating predicted events. Images removed is the sum of true negatives (truly empty images) and false negatives (occupied images missed by MegaDetector), benefit-cost ratio is the number of images removed divided by false negatives, and occupied images contain humans or target wildlife in the full data set (n = 12,530).

| Sensitivity | Discrimination threshold (confidence level) | Images removed # (% of total images) | False negative images (misses) | Benefit-cost ratio | False negative-occupied images ratio |
|---|---|---|---|---|---|
| 1 | 0 | 0 (0) | 0 | N/A | 0.0000 |
| 0.99 | 0.0025 | 13025 (7) | 0 | N/A | 0.0000 |
| 0.98 | 0.009585 | 17580 (9) | 0 | N/A | 0.0000 |
| 0.97 | 0.02605 | 30779 (16) | 11 | 2798 | 0.0009 |
| 0.96 | 0.07425 | 54014 (28) | 18 | 3001 | 0.0014 |
| 0.95 | 0.1475 | 74455 (39) | 18 | 4136 | 0.0014 |
| 0.94 | 0.2425 | 91799 (48) | 36 | 2550 | 0.0029 |
| 0.93 | 0.3585 | 117640 (61) | 43 | 2736 | 0.0034 |
| 0.92 | 0.4305 | 132919 (69) | 43 | 3091 | 0.0034 |
| 0.91 | 0.4935 | 140680 (73) | 61 | 2306 | 0.0049 |
| 0.9 | 0.5425 | 147425 (76) | 67 | 2200 | 0.0053 |
| 0.89 | 0.5835 | 151866 (79) | 77 | 1972 | 0.0061 |
| 0.88 | 0.6205 | 156239 (81) | 90 | 1736 | 0.0072 |
| 0.87 | 0.6475 | 158300 (82) | 102 | 1552 | 0.0081 |
| 0.86 | 0.6725 | 160395 (83) | 104 | 1542 | 0.0083 |
| 0.85 | 0.6905 | 162079 (84) | 134 | 1210 | 0.0107 |
| 0.84 | 0.7075 | 162828 (84) | 151 | 1078 | 0.0121 |
| 0.83 | 0.7255 | 163725 (85) | 189 | 866 | 0.0151 |
| 0.82 | 0.7415 | 164160 (85) | 208 | 789 | 0.0166 |
| 0.81 | 0.7525 | 164660 (85) | 217 | 759 | 0.0173 |
| 0.8 | 0.7645 | 164800 (85) | 240 | 687 | 0.0192 |
| 0.79 | 0.7765 | 165303 (86) | 255 | 648 | 0.0204 |
| 0.78 | 0.7855 | 165468 (86) | 275 | 602 | 0.0219 |
| 0.77 | 0.7945 | 165794 (86) | 335 | 495 | 0.0267 |
| 0.76 | 0.8025 | 165969 (86) | 344 | 482 | 0.0275 |
| 0.75 | 0.8105 | 166859 (86) | 379 | 440 | 0.0302 |
| 0.74 | 0.8175 | 167424 (87) | 412 | 406 | 0.0329 |
| 0.73 | 0.8245 | 167744 (87) | 467 | 359 | 0.0373 |
| 0.72 | 0.8315 | 168033 (87) | 472 | 356 | 0.0377 |
| 0.71 | 0.8365 | 168799 (87) | 485 | 348 | 0.0387 |

| Sensitivity | Discrimination threshold (confidence level) | Images removed # (% of total images) | False negative images (misses) | Benefit-cost ratio | False negative-occupied images ratio |
|---|---|---|---|---|---|
| 0.7 | 0.8415 | 168854 (87) | 523 | 323 | 0.0417 |
| 0.69 | 0.8475 | 168972 (87) | 581 | 291 | 0.0464 |
| 0.68 | 0.8525 | 169309 (88) | 640 | 265 | 0.0511 |
| 0.67 | 0.8585 | 170378 (88) | 701 | 243 | 0.0559 |
| 0.66 | 0.8625 | 170458 (88) | 721 | 236 | 0.0575 |
| 0.65 | 0.8675 | 170713 (88) | 769 | 222 | 0.0614 |
| 0.64 | 0.8705 | 171756 (89) | 804 | 214 | 0.0642 |
| 0.63 | 0.8745 | 171916 (89) | 836 | 206 | 0.0667 |
| 0.62 | 0.8775 | 172067 (89) | 892 | 193 | 0.0712 |
| 0.61 | 0.8815 | 172908 (89) | 1085 | 159 | 0.0866 |
| 0.6 | 0.8845 | 173138 (90) | 1097 | 158 | 0.0875 |
| 0.59 | 0.8875 | 173388 (90) | 1122 | 155 | 0.0895 |
| 0.58 | 0.8905 | 173483 (90) | 1134 | 153 | 0.0905 |
| 0.57 | 0.8925 | 173554 (90) | 1151 | 151 | 0.0919 |
| 0.56 | 0.8955 | 173829 (90) | 1191 | 146 | 0.0951 |
| 0.55 | 0.8975 | 173939 (90) | 1203 | 145 | 0.0960 |
| 0.54 | 0.8995 | 173989 (90) | 1221 | 142 | 0.0974 |
| 0.53 | 0.9025 | 174209 (90) | 1309 | 133 | 0.1045 |
| 0.52 | 0.9045 | 174304 (90) | 1319 | 132 | 0.1053 |
| 0.51 | 0.9065 | 174379 (90) | 1319 | 132 | 0.1053 |
| 0.5 | 0.9085 | 174659 (90) | 1375 | 127 | 0.1097 |
| 0.49 | 0.9105 | 175175 (91) | 1484 | 118 | 0.1184 |
| 0.48 | 0.9125 | 175510 (91) | 1548 | 113 | 0.1235 |
| 0.47 | 0.9135 | 175560 (91) | 1575 | 111 | 0.1257 |
| 0.46 | 0.9155 | 175770 (91) | 1666 | 106 | 0.1330 |
| 0.45 | 0.9175 | 176093 (91) | 1756 | 100 | 0.1401 |
| 0.44 | 0.9195 | 176563 (91) | 1802 | 98 | 0.1438 |
| 0.43 | 0.9205 | 176673 (91) | 1846 | 96 | 0.1473 |
| 0.42 | 0.9225 | 176908 (92) | 1990 | 89 | 0.1588 |
| 0.41 | 0.9245 | 177347 (92) | 2052 | 86 | 0.1638 |
| 0.4 | 0.9255 | 177482 (92) | 2157 | 82 | 0.1721 |
| 0.39 | 0.9275 | 177873 (92) | 2326 | 76 | 0.1856 |
| 0.38 | 0.9295 | 178084 (92) | 2429 | 73 | 0.1939 |
| 0.37 | 0.9315 | 178241 (92) | 2489 | 72 | 0.1986 |
| 0.36 | 0.9325 | 178389 (92) | 2565 | 70 | 0.2047 |
| 0.35 | 0.9345 | 178540 (92) | 2643 | 68 | 0.2109 |
| 0.34 | 0.9355 | 178675 (92) | 2676 | 67 | 0.2136 |
| 0.33 | 0.9375 | 179080 (93) | 2799 | 64 | 0.2234 |
| 0.32 | 0.9385 | 179181 (93) | 2862 | 63 | 0.2284 |

| Sensitivity | Discrimination threshold (confidence level) | Images removed # (% of total images) | False negative images (misses) | Benefit-cost ratio | False negative-occupied images ratio |
|---|---|---|---|---|---|
| 0.31 | 0.9395 | 179276 (93) | 2896 | 62 | 0.2311 |
| 0.3 | 0.9405 | 179426 (93) | 3006 | 60 | 0.2399 |
| 0.29 | 0.9425 | 179686 (93) | 3137 | 57 | 0.2504 |
| 0.28 | 0.9435 | 180176 (93) | 3253 | 55 | 0.2596 |
| 0.27 | 0.9445 | 180246 (93) | 3284 | 55 | 0.2621 |
| 0.26 | 0.9465 | 180590 (93) | 3424 | 53 | 0.2733 |
| 0.25 | 0.9475 | 180790 (94) | 3571 | 51 | 0.2850 |
| 0.24 | 0.9485 | 181288 (94) | 3627 | 50 | 0.2895 |
| 0.23 | 0.9495 | 181408 (94) | 3726 | 49 | 0.2974 |
| 0.22 | 0.9505 | 181664 (94) | 3912 | 46 | 0.3122 |
| 0.21 | 0.9525 | 182046 (94) | 4157 | 44 | 0.3318 |
| 0.2 | 0.9535 | 182243 (94) | 4293 | 42 | 0.3426 |
| 0.19 | 0.9545 | 182422 (94) | 4410 | 41 | 0.3520 |
| 0.18 | 0.9555 | 182661 (95) | 4599 | 40 | 0.3670 |
| 0.17 | 0.9565 | 182944 (95) | 4795 | 38 | 0.3827 |
| 0.16 | 0.9575 | 183104 (95) | 4930 | 37 | 0.3935 |
| 0.15 | 0.9585 | 183417 (95) | 5101 | 36 | 0.4071 |
| 0.14 | 0.9595 | 183762 (95) | 5371 | 34 | 0.4287 |
| 0.13 | 0.9615 | 184228 (95) | 5653 | 33 | 0.4512 |
| 0.12 | 0.9615 | 184228 (95) | 5653 | 33 | 0.4512 |
| 0.11 | 0.9625 | 184590 (96) | 5892 | 31 | 0.4702 |
| 0.1 | 0.9635 | 184925 (96) | 6171 | 30 | 0.4925 |
| 0.09 | 0.9645 | 185345 (96) | 6424 | 29 | 0.5127 |
| 0.08 | 0.9655 | 185715 (96) | 6754 | 27 | 0.5390 |
| 0.07 | 0.9675 | 186346 (96) | 7089 | 26 | 0.5658 |
| 0.06 | 0.9685 | 186760 (97) | 7338 | 25 | 0.5856 |
| 0.05 | 0.9695 | 187460 (97) | 7969 | 24 | 0.6360 |
| 0.04 | 0.9705 | 187676 (97) | 8151 | 23 | 0.6505 |
| 0.03 | 0.9715 | 187960 (97) | 8387 | 22 | 0.6694 |
| 0.02 | 0.9735 | 189244 (98) | 9295 | 20 | 0.7418 |
| 0.01 | 0.9755 | 190737 (99) | 10590 | 18 | 0.8452 |
| 0 | Inf | 193226 (100) | 12530 | 15 | 1.0000 |

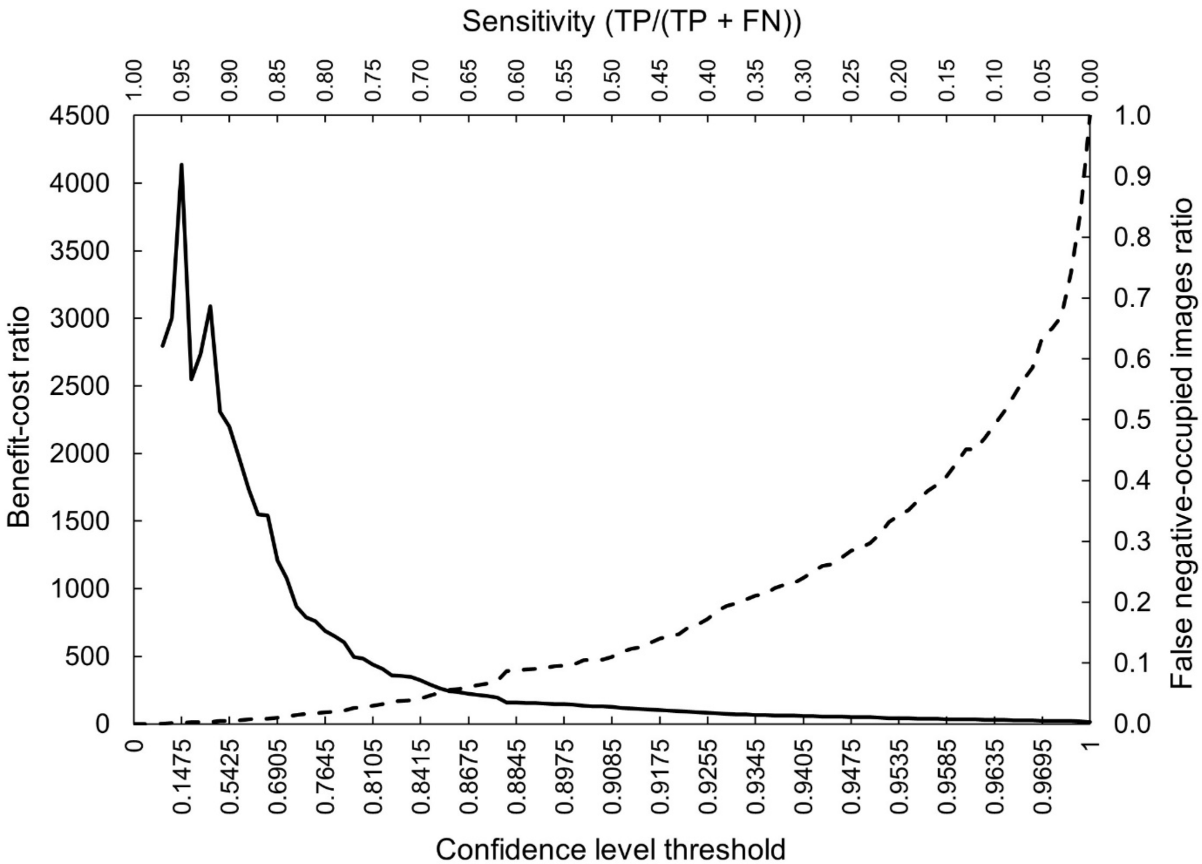


**Figure A2.1.** Benefit-cost ratio (solid line) and false negative-occupied images ratio (dashed line) when applying MegaDetector at a sensitivity of 0.86 and discrimination threshold of 0.67 while maintaining predicted events.

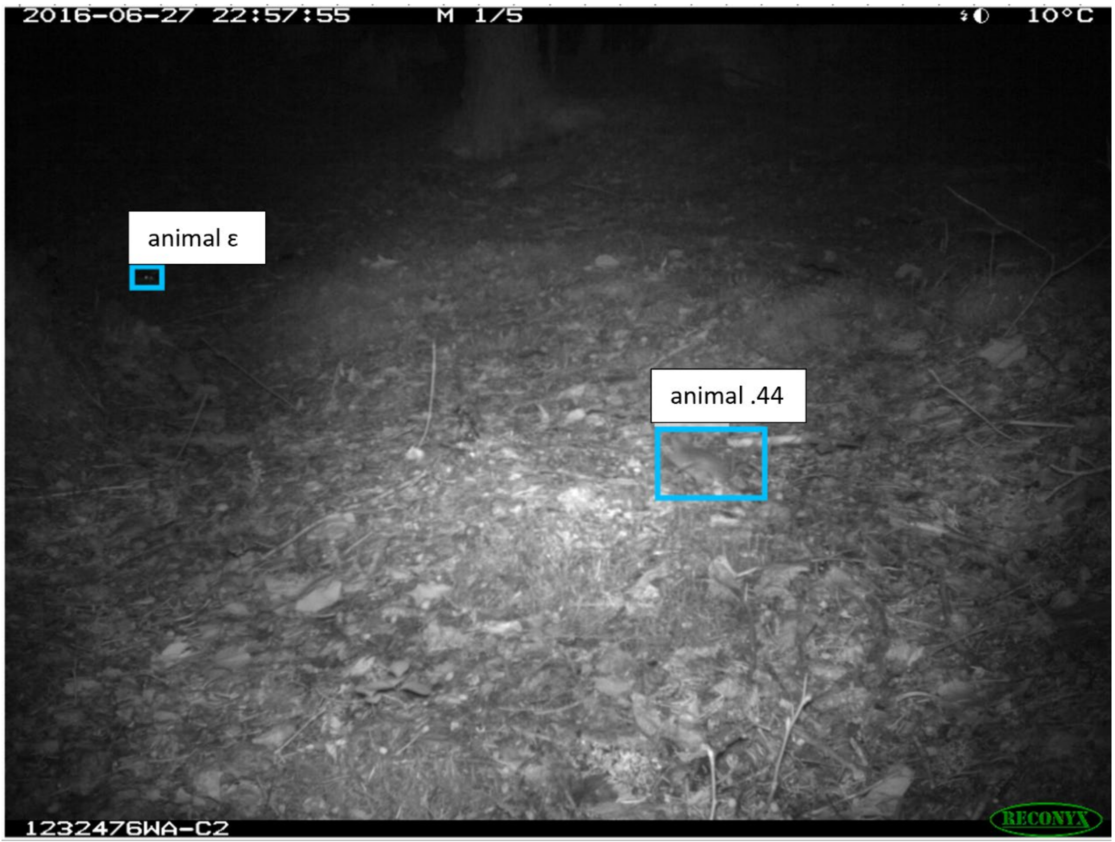


**Figure A2.2.** Example image tagged as “unknown mammal species” and missed by MegaDetector at a confidence level discrimination threshold of 0.67 and classification sensitivity of 0.86. The epsilon (Ɛ) indicates a confidence score of less than 0.1.

(0k P.R. 25 02 14)
ISBN 978-1-4868-8785-9 (pdf)