Automated Image Recognition for Wildlife Camera Traps: Making it Work for You[1]

Saul Greenberg
University of Calgary / Greenberg Consulting Inc.
saul@ucalgary.ca

Last revised on August 21, 2020

Warning: modest image recognition jargon ahead. *To help, Part 5 below explains a few terms that may be worth reviewing before going on. These terms are also italicized in the text.*

Part 1. Image Recognition Is Ready. Or Is it?

You have likely heard about or read articles that apply automated image recognition to wildlife camera trap images. The basic idea is that the image recognition system will automatically analyze your images to locate and classify the wildlife species within them. But before you jump on the bandwagon, here are a few things you should know about.

Academic papers suggest superb recognition performance. Various academic papers report what appears to be excellent *recognition performance measures*. For example, Norouzzadeh et al. (2018) report classification *accuracy* measures of over 99%, while Schneider et al. report 93% *accuracy* for one recognition algorithm on a particular data set. Tabak et al (2019) claim 97.6% *accuracy* in identifying the correct species. They also provide various performance measures per classified species (e.g., *recall*, *precision*), most with values >90%. Promising? Yes, but as will be described shortly, you should be somewhat skeptical about whether these claims suggest that you too will get the same performance on your data.

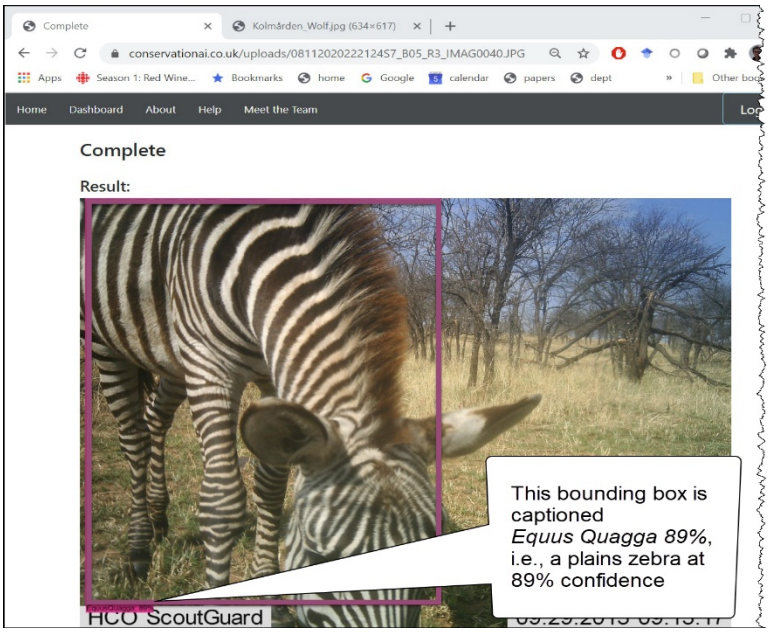
You can easily try out several recognizers on your own sample images. Various image recognition systems – some that work on animals – are available online for you to examine or even try out, with the strong caveat that only some are designed specifically for camera trap images**.** Still, they will give you a rough sense of what is possible and what information these systems can deliver to you. Visit the ones below to see how well they classify your own images, although be aware that each system tries to classify somewhat different things.

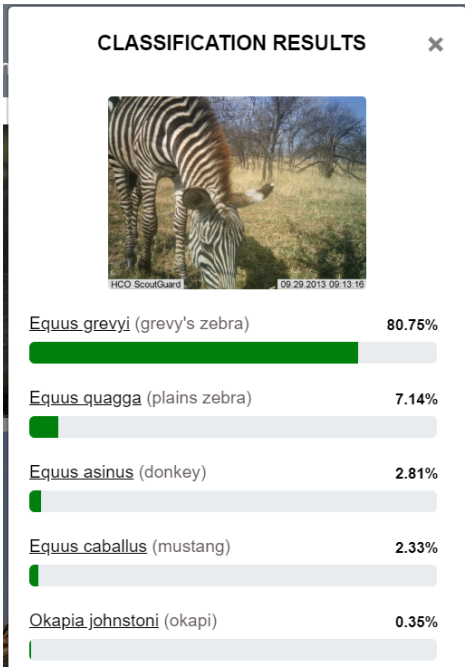
---

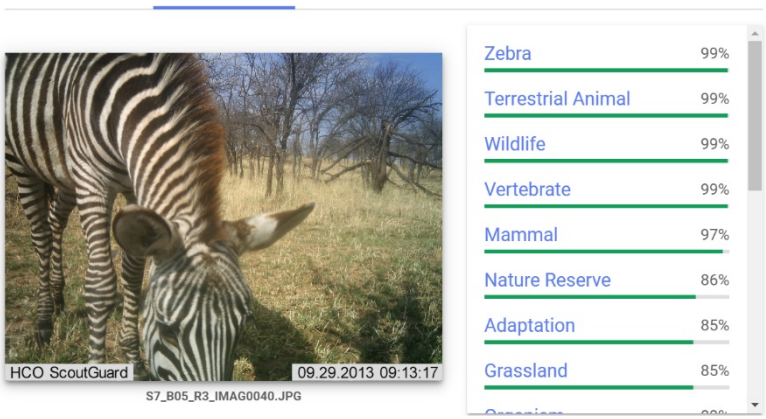
[1] *Cite as*:
Greenberg, S. (2020) *Automated Image Recognition for Wildlife Camera Traps: Making it Work for You.* Technical report, Prism University of Calgary's Digital Repository http://hdl.handle.net/1880/112416, University of Calgary, Calgary, Alberta, Canada. August 21.

- **Conservation AI** https://conservationai.co.uk/. Click the 'Try me' button. Check species availability for different regional *models* via the 'Select a Model' link. Try sub-saharan species, then upload an image of (for example) an elephant or giraffe..
- **Microsoft's AI for Earth Camera Trap** http://cameratrap-demo.eastus.cloudapp.azure.com:5000/ . Choose an appropriate classifier near the upload box, e.g., Snapshot Serengeti currently works best for classifying sub-saharan species, but the Caltech Camera Traps for North American species is also worth a try.
- **Google's Vision AI** https://cloud.google.com/vision is a general classifier that tries to recognize a broad variety of objects. Select 'Try the API' and drop in any image of your choosing. Then click on Objects then Labels.
- **Microsoft's Species Classification API Demo** https://speciesclassification.westus2.cloudapp.azure.com/ is species specific. Select 'Upload' at the top of the page to enter your own image.

For example, I submitted an image of a zebra to several of these systems. These screen snapshots (Conservation AI, Microsoft's Species Classification, Google's Vision API respectively) illustrate what they produce, ranging from an annotated bounding box locating and identifying the wildlife in the first image, to lists of likely classifications ranked by a *confidence* rating, where the top-ranked classification is (in this case) correct. Try to submit both 'easy' and 'difficult' sample images to see how well – or poorly - these recognizers perform.







Does this mean recognition is ready for me to use? The publications and your own attempts above may make you believe that you should now invest in acquiring, learning, and using an image recognition system on your camera trap images. Yet before you do, you should be aware of where image recognition could be a time-saver, where it may be of little value, and/or how you will have to adjust your workflow to make best use of recognition predictions. You should also be somewhat skeptical of any performance claims you read, as they depend on many factors and assumptions that may not align with your own situation.

Importantly, image recognition is fallible. Recognition systems try to *detect* and *classify* objects in an image. When recognition succeeds, it correctly detects and identifies when and where animals are in an image (true positives) along with a classification of each animal (*true classification),* or if the image does not contain an animal (*true negatives*). *Recognition errors* include animals that are present in the image but not detected (*false negatives*), predictions of animals that aren't actually present in the image (*false positives*), and *misclassifications*. Such errors are inevitable. But what does this mean in practice?

Let's say the authors of the image recognition system claim 95% performance on some measure (Part 5: *recognition performance measure*s details various measures and their differences). This implies 5% inaccuracy on that same measure. If you can live with that number of inaccuracies, perhaps because it appears to match (or even exceed) the accuracy and error rate of human classifiers, then it would be tempting to just have the system automatically do all classifications for you, with no need to go through them manually.

Yet accepting recognitions as is - without verifying and correcting them - is a bad idea, at least for the near future. In most cases, you will still need a 'humans in the loop' to review and validate recognition results and correct any errors.

1. You should go through your recognized images to at least get a sense of the real overall performance, as it may vary wildly from the claimed performance.
2. You will need to see how well the recognizer performs on particular species and on particular types of images (e.g., partially occluded animals). While it may accurately detect and correctly classify some species and image types, it could fair considerably worse on others.
3. If recognition performance is less than what you can tolerate, you will have to modify your workflow to check and correct for recognition errors. As part of this, you will also need to consider if there are any real time-savings when using recognition results: correcting many erroneous recognitions may – at some point in your process – be less efficient than manually classifying those images from scratch.

Part 2. The Fallibilities of Image Recognition

The next few sections will dive into the cause of these fallibilities. Understanding them is important, as it will help you determine when and where recognition systems can work for you.

Reported image recognition performance: Be skeptical

As mentioned, the creators of recognition systems may report high *recognition performance* rates. Yet when you submit your own images to the recognizer, the actual performance you achieve may be far lower than what was reported. Here is why.

Those in the image recognition field typically follow a protocol for determining recognition performance. It works something like this. If they have (say) 100,000 previously labelled (i.e., previously classified) images, they will *train* the recognizer on (say) 95,000 images to create a *model*. They will then run the recognizer on the remaining 5,000 images to determine its *accuracy*, where the recognizer's predictions are compared with the previously labelled data.

Yet there are some hidden things at work here. At one extreme, let's say all 100,000 images were taken from a single camera at a single location, all with the same background scene. Of course, the images you submit to the recognizer will be from your own camera. As the background, lighting and other factors in your image scene may be quite different from those previously seen by the recognizer, the recognizer's model will not be as good at discriminating what is in your scenes. Thus the recognizer will usually do much more poorly on your images. And other factors also come into play – as will be explained shortly – that can significantly affect recognition performance.

To give you some numbers, Schneider et. al. (2019) trained several recognizers with ~47,000 images collected from 36 locations representing 55 animal species and human activity. Classification *recall* performance on images taken from the same previously seen locations was ~95%, a promising figure indeed given the relatively small number of training images. Yet when recognition was attempted on images taken from cameras at different locations, *recall* performance dropped to less than 70%. Beery et. al (2018) report a similar performance degradation when trying to recognize images taken from previously unseen locations.

The good news is that the larger and more varied the training set (including the more camera locations used), the more reliable recognition performance will be, and the more generalizable it will be to previously unseen locations. Even so, take that reported performance level with a grain of salt. Check if the authors of whatever paper you are reading explained how training was done, how performance determined, and what assumptions were made. You should also verify how well (or poorly) the recognizer does on your own images.

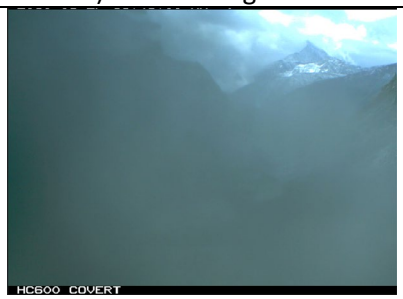
Recognition performance is highly variable per Image

Even if 95% recognition performance is acceptable, you need to know that performance can vary considerably within your images. Somewhat similar to a human, a recognizer can find some images more difficult to analyze correctly than others. Here are a few factors (some illustrated by the following images) that could degrade recognition performance (Norouzzadeh et. al. 2018, and Beery et. al 2018 provide additional details).

- **Persistent recognition of non-wildlife as wildlife.** An artifact, such as a large tree stump or rock, visible in one site's collected images may be incorrectly (and perhaps persistently) recognized as wildlife (a *false positive*).
- **Occlusion.** Some sites have features that partially occlude animals (e.g., trees), which usually makes the animal more difficult to detect in the scene (*false negative*). Even if wildlife is detected, it may be mis-classified as only portions of it are visible.
- **Camouflage.** Some animals are difficult to discern when they blend into the background.
- **Perspective, where only a portion of the animal is visible.** If wildlife passes very close to the camera, or is just entering/leaving the scene, only part of their body will be captured.
- **Weather and lighting** can severely affect the clarity of an image. Lighting effects include sunlight flares, dark shadows, night shots, uneven or overly bright flash or infrared lighting, and others. Weather effects include mist, rain, and fog.

- **Size of animal in the image.** Size does matter. The likely distance of the animal from the camera can affect how well the recognition system works. The recognizer will likely perform better on animals seen in full size directly in front of the camera. It will perform more poorly on small animals or animals that are quite far away.
- **Image quality matters**. Motion blurring can occur by animals moving quickly or during low light conditions. Fidelity is affected by lighting. Of course, some cameras are simply better than others at capturing high quality images.
- **Lens smearing and occlusion**. If the camera lens is not well shielded, camera lens fogging can occur due to environmental conditions: condensation / rain / snow / dirt on the lens. The lens can even be partially or wholly occluded due to vegetation blowing (or growing) in front of the camera, or snow falling on the lens surface.
- **Similar-looking species are harder to disambiguate**. Species classification can include both visually similar and dissimilar wildlife. While a recognizer can find it easy to determine a deer from a crow, it is much more difficult to disambiguate a mule deer from a white-tailed deer, or a wolf from a coyote.

At the very least, you will have to go through some of the images taken at a particular location to get a sense of the performance for that location across and within your images.

<table>
<tr>
<td> 
Lens occlusion by leaves that grew in front of the camera</td>
<td> 
Poor uneven lighting</td>
<td> 
Vegetated stump in scene recognized as an animal</td>
</tr>
<tr>
<td> 
Motion triggering via wind effects of nearby grasses</td>
<td> 
Perspective (animal too close to camera) and blurring</td>
<td> 
Partial occlusion by grass and size of animal in scene</td>
</tr>
<tr>
<td> 
Camouflage (2 animals are in this image)</td>
<td> 
Lens smearing due to snow on lens</td>
<td> 
Weather (fog) obscures the scene</td>
</tr>
</table>

Example images illustrating various factors that could degrade image recognition

Side note. Proper camera placement and lens shielding can mitigate some of the problems above. For example, if you place your camera near a trail where you expect wildlife to pass, optimize the distance of the camera from that trail in order to capture the animal at full size. Is there foliage between the lens and the expected position of wildlife that can introduce occlusion or wind effects? If so, clear that foliage or pick a better spot. Can you shield the lens from snow and rain? How will lighting (sun/shade) affect the image quality over the course of the day?

Image recognition is not balanced across species.

Let's re-examine that '95% performance' figure again a measure of *recall*, as it contains another big caveat: it is the *average* recall performance across all images and classifications. In practice, what you will find is that recognition recall will better on some species, and worse on others. That is, image recognition will produce unbalanced recall rates per species.

As explained in the jargon section, image recognition systems are *trained* to produce a *model*. When you submit previously unseen images to the recognizer, it compares it to the model and spits out its best prediction. The problem is that the training data is often unbalanced. For example, the training data may contain a huge number of images with deer in it, but only very few with wolverines. Thus the recognizer's model may be very good at classifying deer (e.g., 98% recall), but very poor at classifying wolverines (e.g., 20% recall).

To give you some numbers, Schneider et. al. (2019) quantified the species seen in training images that were previously labelled by a Canadian agency doing real ecological work. Of the ~47,000 images and the 55 species seen in them, they noted the following.

- 8,566 had nothing in them.
- The top three tagged species were: white-tailed deer (7,484 images), elk (5,426 images) and wolf (5,269 images). Recall performance was high: about 98 - 99%.
- In the middle of the pack were porcupine (102 images) and marmot (101 images). Recall performance was much poorer: 50% and 75% respectively (the others in the mid-pack produced highly variable results).
- At the bottom were red squirrel (25), small bird (19) and woodrat (8), where recall performance was terrible: 50%, 0% and 0% respectively.

Even these numbers are likely optimistic. As described previously, the train / test method all used images taken from the same camera locations.

Those empty images…

Depending on your setup, many of the image captured by your camera (and perhaps even most of them) may have nothing in them. This often happens for two reasons.

1. **Cameras in timelapse mode.** When images are captured at pre-determined intervals (e.g., once every 5 minutes), most images will have nothing in them as wildlife is simply not that prevalent.
2. **Cameras in motion-triggering mode, part 1**. Motion triggering occurs when something moves through a scene, ideally an animal moving in front of the lens. Triggering usually results in several images automatically taken in a short time interval. However, incorrect motion triggering can easily occur through wind effects, such as moving grasses and tree branches. This can produce a huge number of empty images.
3. **Cameras in motion-triggering mode, part 2**. Depending upon the camera, the motion sensor may have a wider angle of sensitivity than the camera lens. Thus the initial image may be empty

as the animal is not yet within view. Similarly, if motion triggering activates a series of images taken over time, the animal may move out of the scene while those images are still being taken.

To an image recognizer, an empty image is simply one where it could not detect an entity in it, or where it has detected an entity but with very low confidence. As will be discussed shortly, reliable *detection* of empty images is important, as it can help you eliminate a significant number of images from closer review.

Part 3. Human in the Loop - Image Recognition in your Workflow

Even with the above fallibilities, image recognition can still work for you. The key is to consider recognition as an aid to human classification, where it helps a person do their workflow more efficiently, rather than completely automating the classification task.

As described below, the basic strategy is to use image recognition to rapidly filter your images into smaller and smaller subsets. The goal is to:

a) make it easy to quickly eliminate particular images from further detailed review (thus making overall image management faster), and
b) easier to accept correct classifications while still detecting *recognition errors* in each subset.

The filtering strategy assumes that it is easier and far more reliable to review a sequence of like images for anomalies that don't fit *vs.* reviewing a sequence of images containing a somewhat random mix of images that are empty, that contain different animal species, that contain people, and so on.

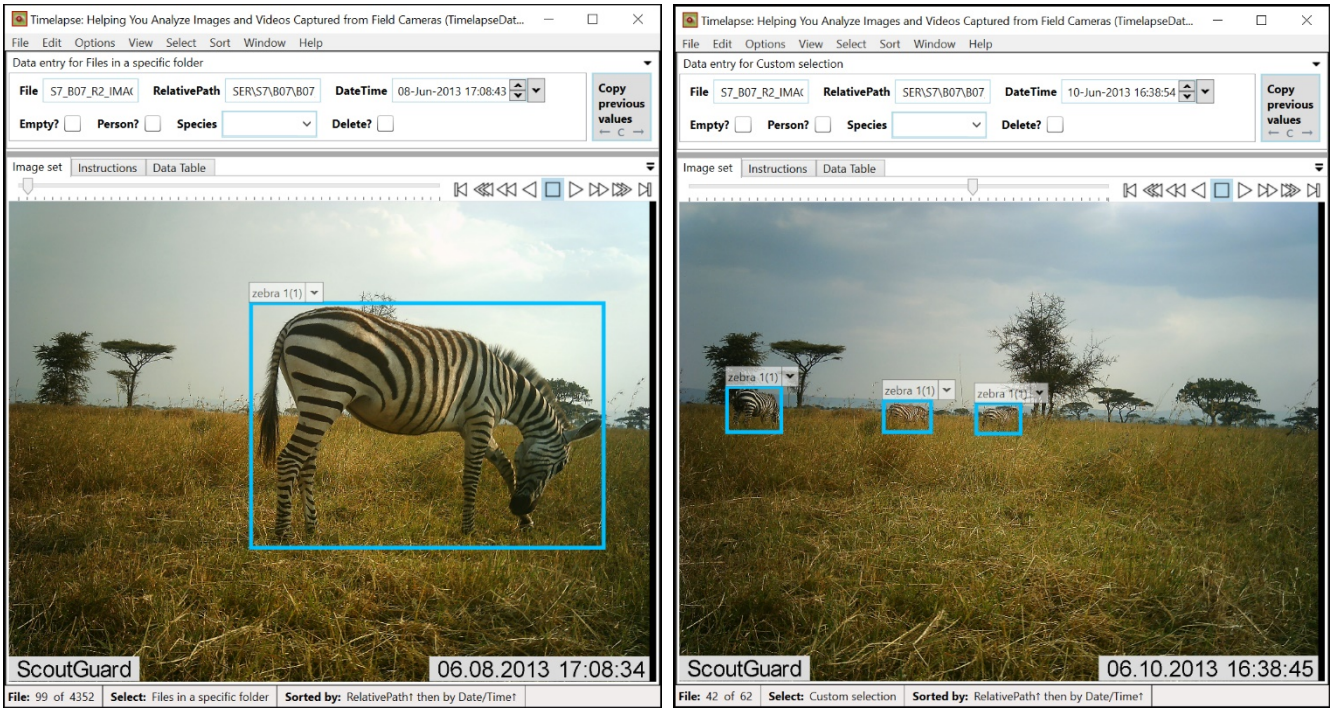
I will illustrate workflow enhancements using the Timelapse Image Analyzer as it is one of the very few systems that seriously considers how image recognition results can be incorporated when analyzing large numbers of camera trap images. Ideally, we expect other systems will eventually have similar capabilities to those shown here. The caveat is that these workflow recommendations are still a work-in-progress, where we are still trying to discover what works best in particular situations.

Side note. Timelapse, including tutorial documentation, is freely available and can be downloaded at http://saul.cpsc.ucalgary.ca/timelapse/ at no cost. Its design is detailed in Greenberg et. al., 2020.

Quickly locating wildlife in your images

Within image recognition, *detection* tries to determine if an entity is in an image, and if so, where it is located. Using that location, a system can then visually highlight that entity to make it more noticeable. This can ease the task of scanning an image to see if it contains wildlife, and if so, where they are.

For example, the two images on the next page illustrate how an analyst sees an image and its recognition data in Timelapse. As common with many classification systems, each detected entity is surrounded by a bounding box labelled by its classification. The box quickly draws the eye to the detected entity. While there may be only modest advantage to this when the wildlife is clearly visible (as in the first image below), seeing bounding boxes effectively draws attention to entities that may otherwise be harder to see (as in the second image below). This eases the work in scanning each image as well as counting. Timelapse also allows closer inspection (and thus validation and error-checking) of small entities within the bounding box via magnification and zooming tools.



Bounding boxes do have two modest disadvantages, although these should not outweigh their utility. First, the analyst may become over-reliant on them: *false negatives* (undetected wildlife that do not display bounding boxes) are inadvertently skipped over by the viewer. Second, *false positives* (e.g., a bounding box around a tree stump) can add noise.

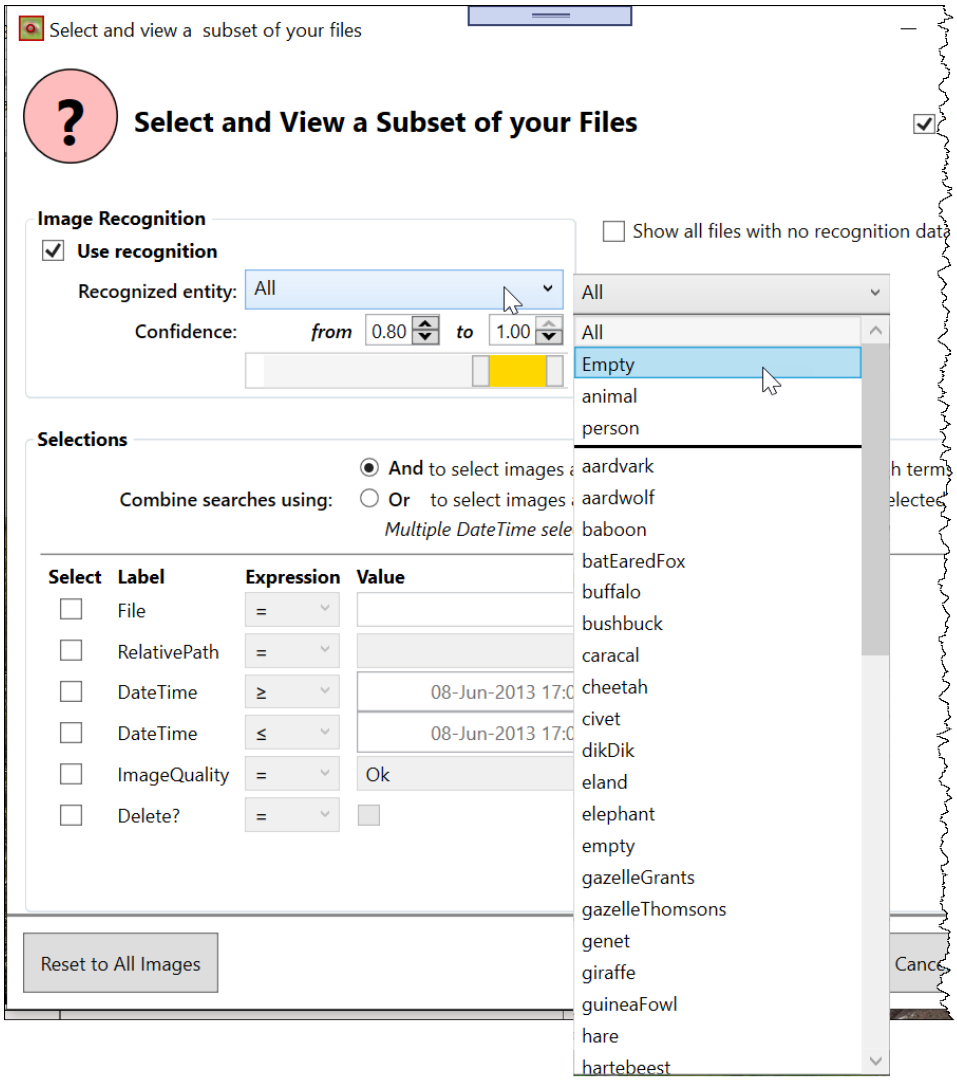
Eliminating empty images

While accurate species identification and counting may be the holy grail in image recognition, a much simpler recognition result can still provide a big win in efficiently analyzing images: differentiating between empty images and images with something of interest in them.

In practice, the number of empty images taken by a camera can be huge. As previously discussed, empty images can occur for various reasons, e.g., the cameras in timelapse mode, because of accidental motion triggering due to wind effects, and so on. Several ecologists we have work with report extreme cases where the vast majority (over 90%) of their images contain nothing of interest. If these empty images can be eliminated quickly from further detailed review, the time savings can be considerable.

Let's see how a workflow filtering out empty images can be done in Timelapse. Timelapse allows the analyst to query (select) images by *detections*, *classifications* and *confidence* values, where only a subset of the images matching that query are then displayed. The screen snapshot on the next page illustrates the query selection dialog. Here, the analyst is asking the system to display all empty images where the recognizer has is a high confidence (between .8 and 1) that nothing is in them.

The analyst then works within this subset of returned images recognized as empty. The analyst initially labels all returned images as empty, which can be done quickly via several Timelapse shortcuts. The analyst then quickly reviews those images to check whether they are indeed empty (in which case nothing needs to be done), or to correct the occasional image that does contain wildlife (by relabelling it). The analyst can repeat the process with progressively lower confidence ranges as long as the error rate remains low. To help make this process more efficient, Timelapse contains a control that automatically 'plays' images one after the other at a user-determined rapid rate, where the analyst just checks for anomalous images (anomalies tend to 'pop out visually). As an option, if – after reviewing a



reasonable number of images – the detector appears to be highly reliable at a given confidence range, the analyst can just accept the remaining 'empty' predictions without further review. Both strategies can quickly reduce the set of images that needs to be closely examined.

Separating out images with people in them

Some image recognizers do broad classifications, including those that simply distinguish images into those containing people *vs*. wildlife. For example, Megadetector [Microsoft, Inc] initially distinguishes non-empty images into several categories: people, animals, and vehicles. As broad classification is a much easier task than fine-grained species classification, it generally produces reasonable performance.

This broad classification simplifies the process of separating out people images from other images. Various agencies have privacy policies, where images with people in them must be given special treatment. Some require those images to be discarded, or that they must be separated and secured away from those that are publicly disseminated. Alternately, it could be that the agency is more concerned with counting human use than in identifying wildlife. If the camera trap is on a popular trail, removing the many 'people' shots from further review would improve efficiency.

Using a similar workflow process as identifying empty images, most 'people' images at a given recognition confidence can be quickly selected (e.g., selecting the 'person' entry from the Timelapse query screen above), labelled, and filtered out from the remaining images. During review, the bounding box also quickly focuses attention on the detected (but perhaps misclassified) person. While some images with people in them will be missed (i.e., *false negatives* where people are not spotted by the recognizer), these will likely be identified during later parts of the workflow process. The advantage is that there should be only relatively few people shots (false negatives) remaining after this step is done.

Focussing in on wildlife images

Recall that, in image recognition, broad classifications – such as animal *vs*. people *vs*. vehicles *vs*. *empty* can produce much more reliable results than identifying wildlife species. By the time empty and people (and perhaps vehicle) images are separated out of the total, the much smaller set of images left will mostly have wildlife in them. This can result in a significantly more manageable subset of images requiring close examination and classification, even if done manually from this point onwards.

Of course, image recognition for species classification can help further, as described next.

Selectively using species classification results

Some ecosystems have large numbers of potential species that could be captured on camera. While the recognition classifier will attempt to predict what that species is, *misclassifications* will become increasingly commonplace. So the question is: can classifications still help to improve workflow? The answer is yes, if used selectively.

One inefficient way to envisage the workflow is to (say) select each species (e.g., by first selecting 'aardvark', then 'aardwolf' then 'baboon' etc. using the Timelapse query above) and examine the resulting images in turn. This could be laborious indeed if many species are possible. And remember that recognition accuracy may be unbalanced, where some species classifications will have far more errors than others.

Another better way to envisage the workflow is to use the classifier to separate and label only those species most likely to be captured by your camera (e.g., deer, elk in some North American regions), and leave the remaining harder images for manual classification. For example, the previous workflow will have separated out the empty and people images, leaving a smaller subset of wildlife images requiring classification. If large numbers of deer are expected, the workflow – using a process similar to those mentioned above – would filter that subset further by deer, and examine/label/correct the deer subset as needed. That process is repeated only for other commonly occurring animal, such as elk. This will leave a remaining much smaller subset of images that may contain animals infrequently captured by the camera: bear, coyotes, wolverines, and so on, that can be classified manually.

Part 4. Image Recognition: Questions You Should Ask.

There are many image recognizers coming on-stream, and you need to know that they can differ considerably from one another, especially in terms of how well they will do on your particular images. Each may use different recognition algorithms. Each will almost certainly be *trained* on a particular set of images, where each generates one or more *models* necessary for recognizing your species. Even if recognizer results appear reliable, each may have obstacles in terms of accessing and efficiently using those results in your workflow.

Here are several questions you should ask of the people offering the image recognition system. They may not have the answers to all of them, but at least you will gain some insight.

1. ***What is the size of the training set?*** *The* larger the size, the better.
2. ***What are the number of tagged images per species?*** This follows from the balance issue – it would be good to gain some insight of how many images of that species were seen during training. For example, Schneider et al. suggests that about 1000 tagged images of a particular species should be included in the training set to produce a reasonably high and stable recall rate.
3. ***How many different scenes (i.e., geographically different camera locations) were included?*** The more different backgrounds and variations, the better.
4. ***What types of scenes were they?*** You can also check to see if the captured scenes are similar to your scenes (e.g., fields vs forest).
5. ***What are the tagged species used to develop the model?*** Ideally, you should be given a list of species that the recognizer has been trained on. This will allow you to compare whether the species in your own region are covered by the model. For example, a recognizer trained on sub-saharan species won't really work on North American species.
6. ***Can we eliminate some species from the model?*** The training data – and thus the model – may have species in it that are not in your region and thus irrelevant (and would add noise if they were included). You should ask if there is a way to eliminate those species from the classification process.
7. ***Does the recognizer come with a system that puts the 'human in the loop' in terms of efficiently examining, verifying and correcting recognition results?*** Analyst need a way to examine the recognition results and correct errors as needed. If the recognition results cannot be incorporated efficiently in the workflow, then the time and effort required to use those recognition results may be far greater than the time and effort required to manually classify the images. Indeed, this was the very reason for the Microsoft AI for Earth project to work with me as the Timelapse creator: Timelapse provides the front end for analysts to incorporate the recognition data in their workflow.
8. ***How do I submit my images to the recognizer?*** You will, of course, have a massive number of images collected from your camera traps, and you need a way to submit them to the recognizer. Methods include mailing a physical hard drive to the people running the recognizer (this is a surprisingly efficient method) , or uploading your images to the cloud or a web site (which can be quite slow depending on your internet link). In some case, you may be able to download and install a recognizer and its model on your local machine. This may require a systems person to do correctly, especially if other software needs to be downloaded to make this all work. Additionally, if there is opportunity to download the recognizer, ask about the computer(s) necessary to do the work: high performance computers are normally required as recognition can be computationally expensive.

Part 5. Some Jargon Explained

Training, learning, models, recognition and confidence.

Image recognition systems must be ***trained*** via machine learning, where it ***learns*** how to distinguish the contents of one image from another. Training and thus learning begin with a large set of previously labelled (i.e., already categorized) images. The system analyzes those images and its labels to create a ***model*** (technically called a 'convolution neural network') that best fits what it sees. To achieve ***recognition,*** the system tries to best match an unlabelled (i.e., previously unseen) image to that model, where its predictions are those classifications that match what is in the model. The model is

sophisticated, where it can associate a ***confidence*** with that prediction (usually a number between 0 and 1). However, confidence should be used only as a very rough indication of likely correctness. Interpret a high confidence value as 'likely correct with occasional errors' and a low confidence value as 'likely incorrect and a large number of errors'.

Of course, image recognition is more complex than that. The key take-away is that training is critical. Good training requires a very large number of correctly labelled images, which in turn require many varied images per location and desired classification.

Detection vs classification.

Recognitions systems use various algorithms for *detecting* entities in an image, and then *classifying* those images. Broadly speaking:

> ***Detections*** determine whether an entity is in an image, or whether the image is 'empty'. If an entity is present, it:
>
> - locates each entity in the image,
> - identifies that location, for example as bounding box coordinates,
> - assigns each detection with a value that very roughly indicates its confidence of correctness,
> - may broadly classify the entity, e.g. as an animal, person or vehicle.
>
> ***Classifications:*** For each detection, classify the detected entity into one or more possible categories along with a confidence value. Classifications may be broad (e.g., animal *vs.* person) or narrow (e.g., species such as deer vs. elk). If multiple classifications are predicted per entity, their confidence value is usually expressed as probabilities summing to 1, such as deer .8, elk .2).

Recognition success vs. errors

Understanding the ways detection and classification can succeed or fail will help you understand how and where you have to examine recognition results and correct recognition errors.

Typical Image recognition successes. Recognition success can be considered as follows.

- ***True positive (detection):*** correctly detects an entity's presence in a scene (e.g., wildlife is present)
- ***True negative (detection):*** correctly determines that an entity is NOT present in a scene (e.g., there is no wildlife present)
- ***Correct classification:*** correctly determines that a detected entity belongs in a particular class (e.g., if looking for deer, it correctly identifies a deer).

Typical image recognition errors. Even the best image recognition system will get it wrong some of the time. The number of errors will depend heavily upon the image recognition algorithm being used, how well the recognition system has been 'trained', and the variations between the actual images submitted to the recognition system. There are three primary error types, each which depends upon whether it is returning a detection *vs* a classification.

- ***False positive (detections):*** incorrectly detects an entity in an image when nothing is there (e.g., a tree is mistakenly recognized as wildlife)
- ***False negative (detection):*** an entity is present, but it is not detected. (e.g., wildlife is in the scene but is not detected)

- ***Misclassification:*** incorrectly classifies an entity as belonging to the desired classification (if looking for deer, an elk is mistakenly recognized as a deer)

The above suggests various strategies you can use for examining images for errors. These are selectively incorporated in the 'filtering' strategies mentioned previously.

1. True vs. false negatives (detections). Examine 'empty' images to correct those that contain an entity within it .
2. True vs. false positives (detections). Examine images that the system claims contain an entity, correcting those that are in fact empty.
3. Correct vs. misclassifications. Examine images that the system claims contain an entity and correct those where the system misclassifies that entity.

Recognition performance measures.

There are myriad ways to measure recognition performance e.g., see Soklolova & Lapalme, 2009. Particular academic papers usually detail how the particular measures are calculated. In general, most report the average effectiveness (correctness) of the recognizer as a ratio. A few common classification measures are described below, but there are many others (e.g., see https://en.wikipedia.org/wiki/Precision_and_recall). Each measure describes (or masks) different aspects of how well the recognizer is doing on average. The best measure depends, of course, on what you are trying to measure. As well, the exact meaning of the terms below depend upon whether detections or classifications are being measured, as they differ somewhat depending on the context. For example, in the context of a detection problem, "precision" means "the fraction of images predicted to contain objects that actually contain objects"; in the context of a classification problem, "precision" is defined for each class, and it might mean "the fraction of images predicted to be deer that actually contain deer".

I illustrate each with an example that assumes that only images containing deer are requested from the image recognizer.

- ***Precision***:
  - What proportion of the classification results to contain an entity are actually correct?
  - Formula: true positives retrieved / all true and false positives retrieved
  - Of the 20 deer classifications returned only 16 of them are actually deer (the remainder are incorrectly classified). The precision is 16/20 or 80%.
- ***Recall***
  - What is the proportion of the positive classifications results returned vs the total true positives available in the set?
  - Formula: true positives retrieved / all true positives and false negatives in the set
  - While 16 correct deer classifications are returned, a total of 24 deer are actually present across the images. The recall is 16/24, or 66%.

- ***Accuracy***
  - What is the proportion of returned results that are correct (either positive or negative)
  - Formula: (True positives + true negatives) / (true positives + true negatives + false positives + false negatives)
- ***F-Score***
  - Combines the precision and recall measures into an approximate average.
  - Formula: 2 * precision * recall / (precision + recall)

- For the examples above, this is 2 * .8 * .66 / (.8 + .66), or 72%.

Acknowledgements

My discussions with Dan Morris (Microsoft) and Stephan Schneider (Guelph University) helped me understand and articulate the prospects and limitations of image recognition. Additionally, comments and feedback by ecologists using the Timelapse/Megadetector system for real work were invaluable: they were the front line for using and tuning recognition workflow strategies. Dan Morris also suggested changes to an earlier version of this document. However, any errors or inaccuracies are my own.

References

1. Beery S, Van Horn G, Perona P. (2018) Recognition in terra incognita. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 456-473). https://arxiv.org/abs/1807.04975
2. Greenberg, S., Godin, T. and Whittington, J. (2020) **User Interface Design Patterns for Wildlife-Related Camera Trap Image Analysis.** Ecology and Evolution, Vol. 9 Issue 24:13706-13730. Wiley, January 10. https://onlinelibrary.wiley.com/doi/full/10.1002/ece3.5767
3. Microsoft (2019). **AI for Earth camera trap image processing API**. Github repository of its MegaDetector recognizer. Retrieved from https://github.com/Microsoft/CameraTraps
4. Norouzzadeha, M.S., Nguyenb, A., Kosmalac, M., Swanson, A. Palmere, M.S., Packere, C. and Clunea, J. (2018) **Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning.** PNAS (Proc. National Academy of Sciences of the USA) , 115 (25), E5716-E5725, https://doi.org/10.1073/pnas.1719367115
5. Schneider, S., Greenberg, S., Taylor, G.W. and Kremer, S.C. (2020) ***Three Critical Factors Affecting Automated Image Species Recognition Performance for Camera Traps***. Ecology and Evolution, 10(7):3503-3517. Wiley, April 8. https://onlinelibrary.wiley.com/doi/full/10.1002/ece3.6147
6. Tabak, M. A., Norouzzadeh, M. S., Wolfson, D. W., Sweeney, S. J.,VerCauteren, K. C., Snow, N. P., et. al., (2019). **Machine learning to classify animal species in camera trap images: Applications in Ecology.** Methods in Ecology and Evolution, 10(4), 585–590. https://doi.org/10.1111/2041-210X.13120

About the Author

Saul Greenberg a Professor (currently Emeritus) in Computer Science, specializing in human computer interaction http://saul.cpsc.ucalgary.ca/. Saul's research is well-recognized and highly cited (h-index = 90). He is an ACM Fellow, and has held the AITF/NSERC/Smart Technologies Industrial Chair in Interactive Technologies. He was elected to the ACM CHI Academy for his overall contributions to the field of Human Computer Interaction, and also received the Canadian Human Computer Communications Society Achievement Award, the ACM UIST Lasting Impact Award, and the ACM ISS 10-Year Impact Award.

More germane to this document, he is also the creator of *Timelapse*, an open-source image analyser for camera traps originally designed to help analysts efficiently inspect and manually enter data describing images http://saul.cpsc.ucalgary.ca/timelapse/ [Greenberg et. al.,

2020]. Timelapse is widely used internationally, where it has helped individuals and agencies efficiently analyze millions of camera trap images.

Recently, Dr. Greenberg has been collaborating with Microsoft, where their AI for Earth team are developing the *MegaDetector* image recognition system [Microsoft, Inc.]. Timelapse does not do image recognition by itself. Rather it can import MegaDetector's recognition data describing a set of images, where analysts can then incorporate that data into their Timelapse workflow. That is, Timelapse was modified to bring the 'human into the loop' for image recognition. Feedback is continuously being gathered from various ecologists who are using the combined systems. We are especially interested in the workflow protocols they are developing, their successes, and bottlenecks where the process can be improved. Greenberg's interest is in further refining Timelapse to make that workflow even more efficient.