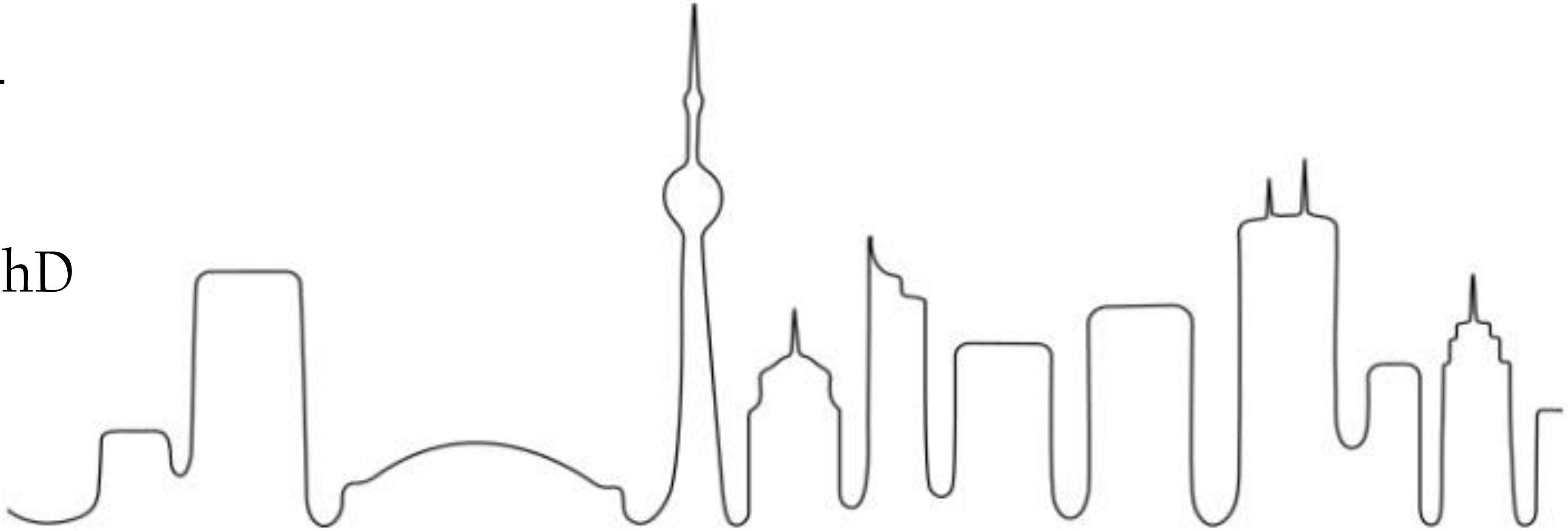
Training an object classifier for our own system

Tiziana A. Gelmi Candusso, PhD
University of Toronto
Focus: Urban mammals



How we started: using machine learning models trained on other systems

We started integrating machine learning into our tagging process by using available models, including Megadetector and a species classifier trained on rural animals from Idaho.

→

<u>Artificial intelligence</u>

MegaDetector

+

Idaho classifier

https://github.com/microsoft/CameraTraps

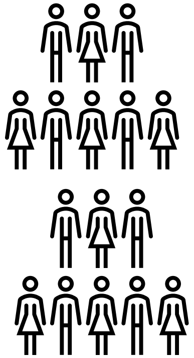
→

Predicted Labels

- People
- Vehicles
- Empty
- Animals
  - Canids
  - Cats
  - Rabbits
  - Small mammals
  - Deer
  - Cow
  - Springhorn

→

<u>Human intelligence</u>



→

True Labels

TIZIANA GELMI CANDUSSO

What we need instead:

The common urban mammals belonged to mainly two categories in the original species classifier we used, therefore the workflow was still too high, considering the large number of dogs and raccoons so we needed to train our own species classifier to streamline the tagging process.

Artificial intelligence

MegaDetector

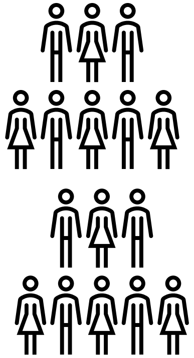
Idaho classifier

https://github.com/microsoft/CameraTraps

Predicted Labels

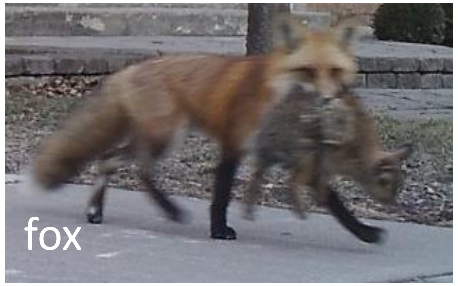
- People
- Vehicles
- Empty
- Animals
  - ~~Canids~~
  - Cats
  - Rabbits
  - ~~Small mammals~~
  - Deer
  - ~~Cow~~
  - ~~Springhorn~~

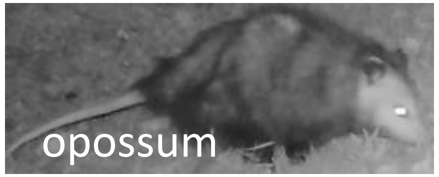
Human intelligence



Volunteers using Timelapse2

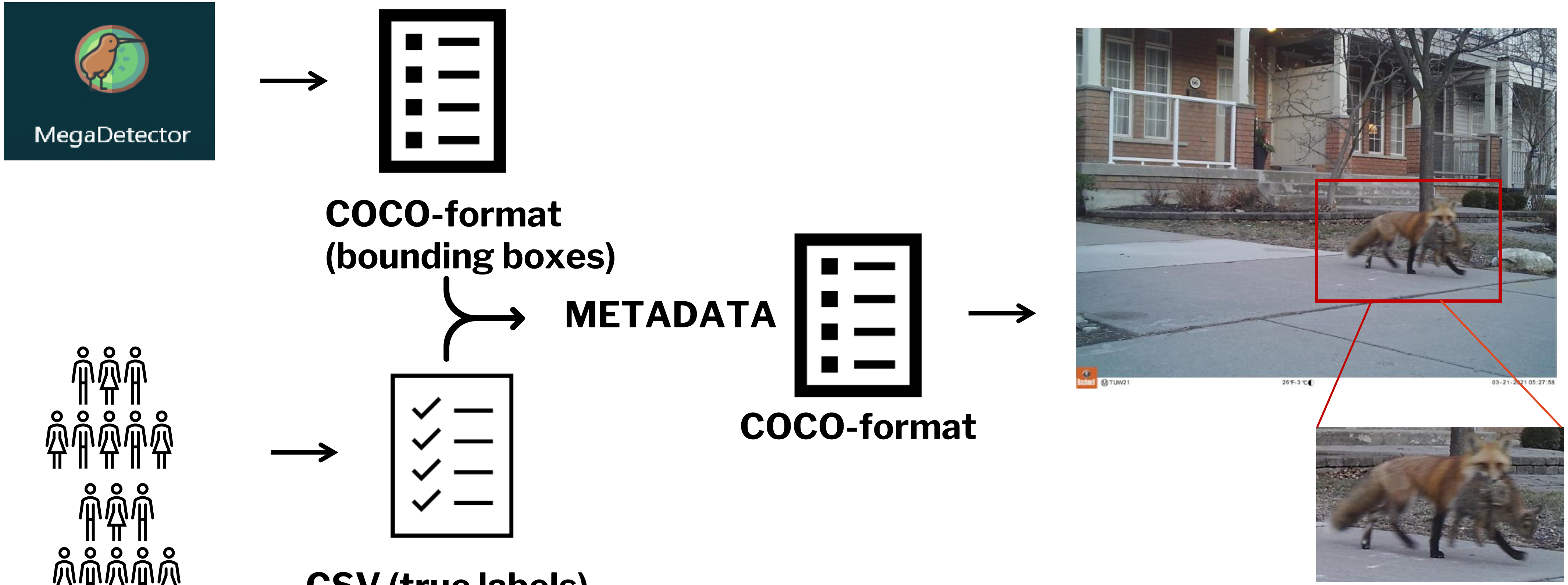
True Labels





First step, prepare data: Compile metadata and crop images

We trained the model with crops. To obtain these crops and keep the labels attributed by the volunteers, since we didn't use a software where volunteers could label bounding boxes, we extracted the best bounding box of each image over 80% confidence level and matched it to the image-level annotation.



CSV (true labels)
- **Datetime**
- **Location**

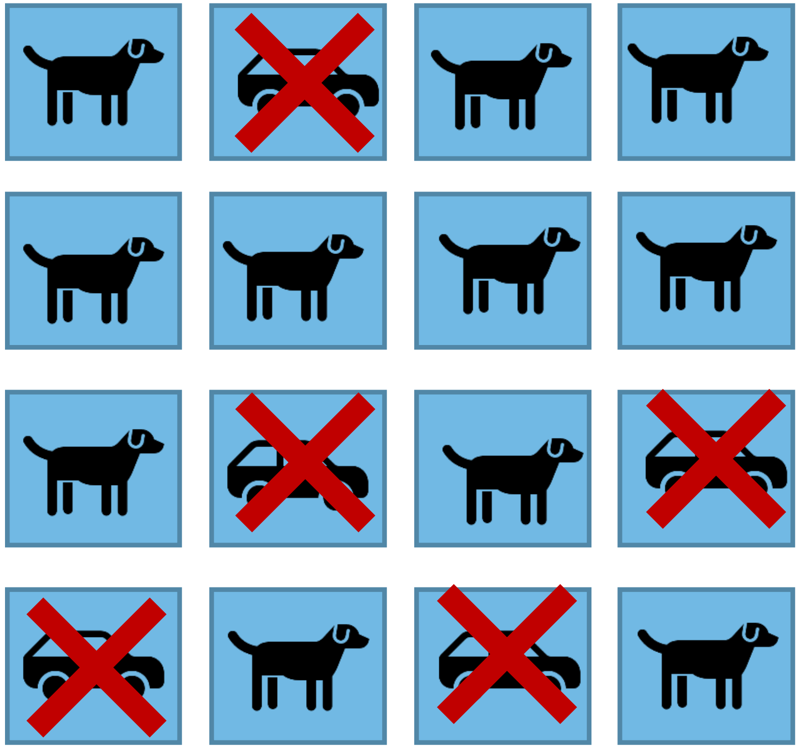
CROPS

TIZIANA GELMI CANDUSSO

First step, prepare data:
Clean crops

Bounding boxes were blindly matched with the image-level annotations, therefore there were still some errors to clean, such as crops with cars or noise objects, redundant images, and objects that were labelled out of context but not recognizable within the crop.

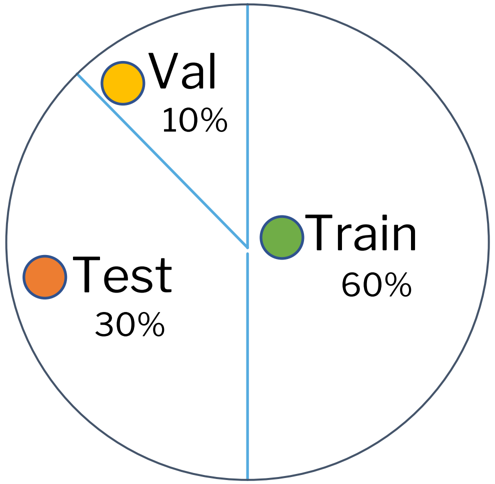
Crops not always contain the animal labelled.
Clean all those errors nd crops you cannot identify yourself without the label.



Second step, split data:

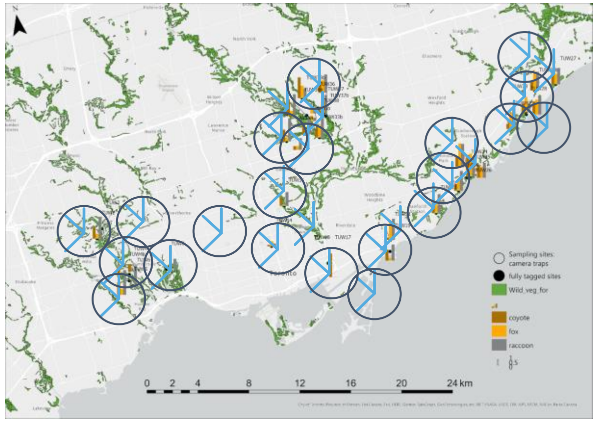
Once the data was cleaned, we selected images for training, and evaluation purposes, we did this with a fixed proportion, but tried three different ways of selecting for the photos.

Select photos

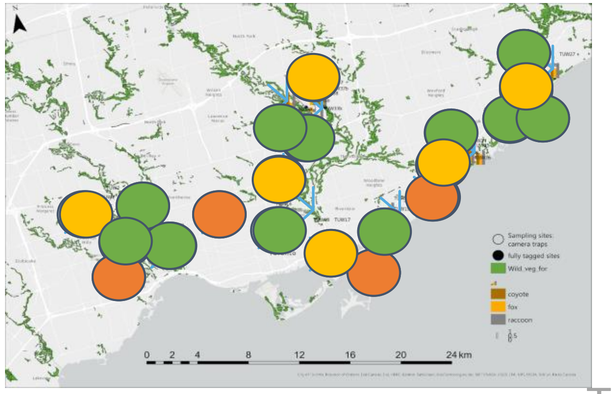


1. RANDOM

2. ACROSS LOCATIONS



3. BY LOCATION



Training the model: Parameters

Epochs:
200 - 300

Batch size:
32 – 128

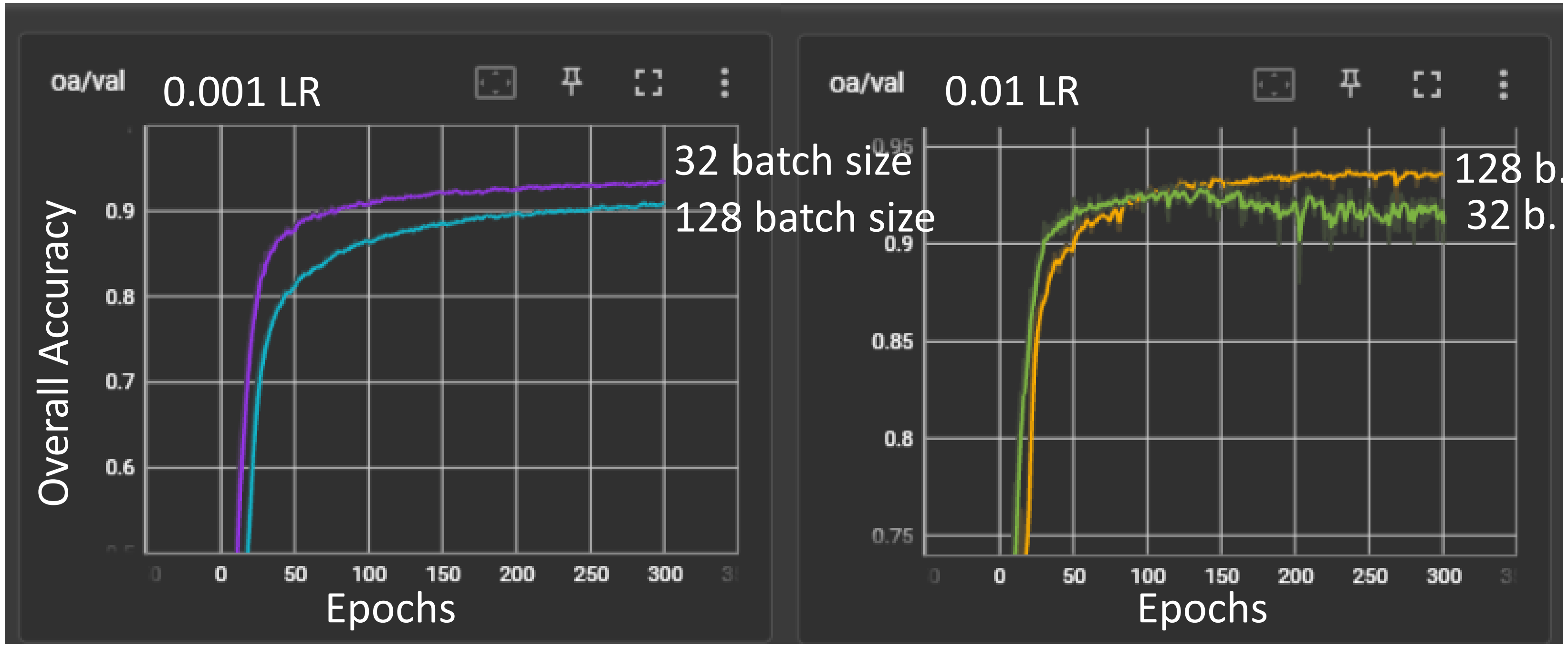
Workers:
6-8

Learning rate
0.001 – 0.01

Weight decay:
0.001

We trained a Resnet18 model and tested a series of parameters, such as increasing number of epochs, changing the learning rate and batch size and visualized the metrics using tensorflow.

In our case: Higher accuracy with 0.01 Learning rate and 128 batch size



TIZIANA GELMI CANDUSSO

Training the model: Fine-tuning to deal with specific problems regarding our system

Our system had strongly imbalanced data, we used curriculum learning to improve the model and adjusted the Loss function by attributing different weights to each class following the number of images within

IMBALANCED DATA: Dogs and squirrels disproportionately more common

Curriculum learning

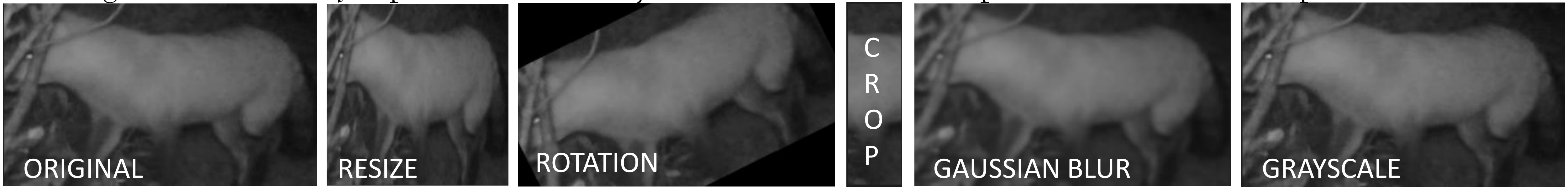
100 x5 epochs,
200 x5 epochs,
500 x5 epochs,
1000 x5 epochs
2000 x :n epochs

Loss weight

Created weights based on frequency of images

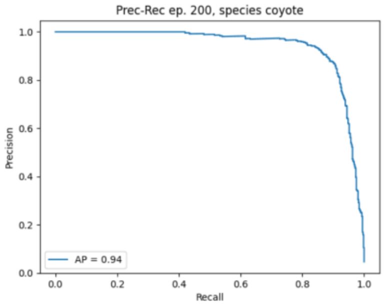
Training the model: Data augmentation

Data augmentation techniques performed differently across classes, so it was important to evaluate class-specific metrics.



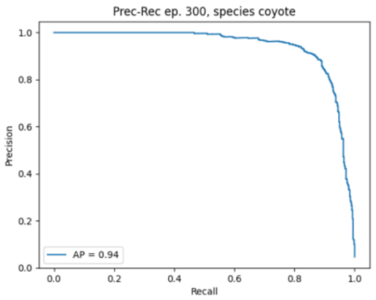
<u>EXAMPLE of how precision recall changed for one class:</u>

Startpoint:
200 epochs, 0.001 LR,
128B, Resize to 224x224



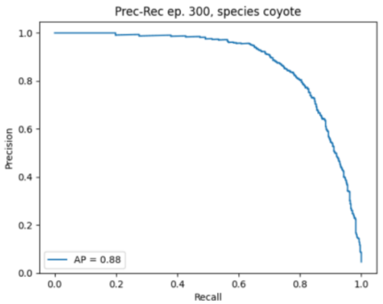
AP = 94
Average
precision

300 epochs



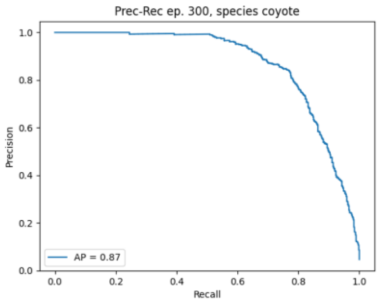
AP = 94

+Norm

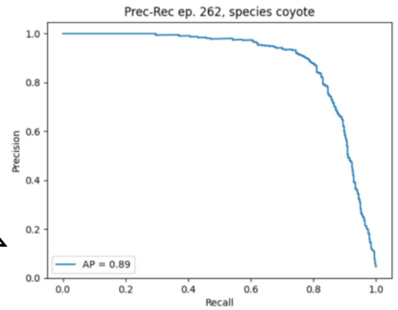


AP = 88

+Norm, Rotation



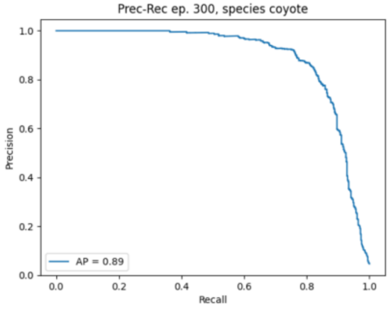
AP = 87



+Norm, Rotat.,
Blur, 0.01LR

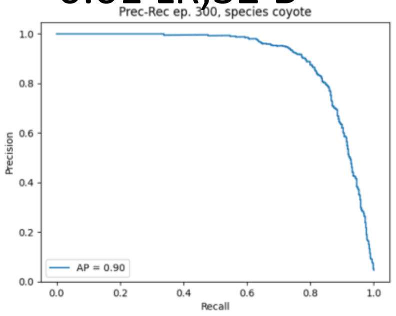
AP = 89

+Norm, Rotation,
0.01 LR



AP = 89

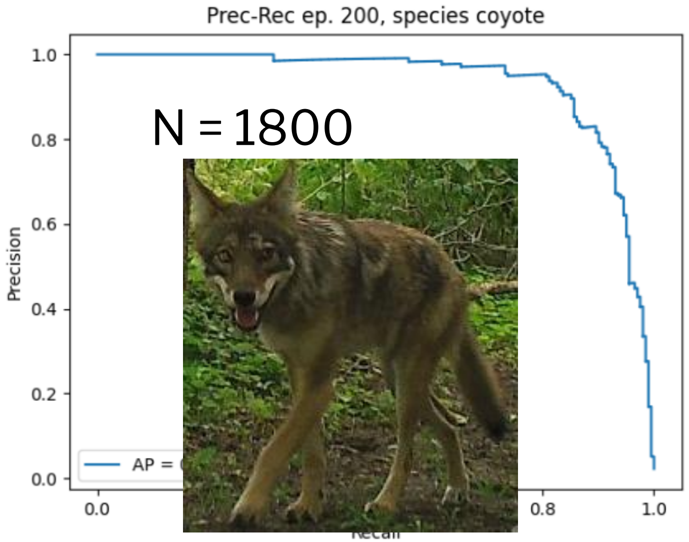
+Norm, Rotation
0.01 LR,32 B

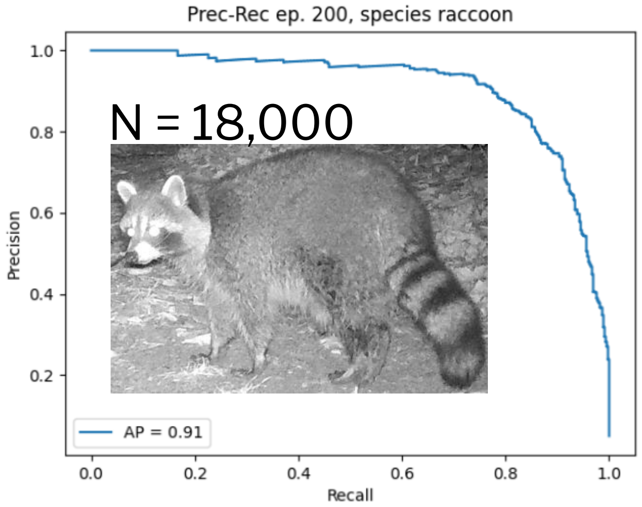


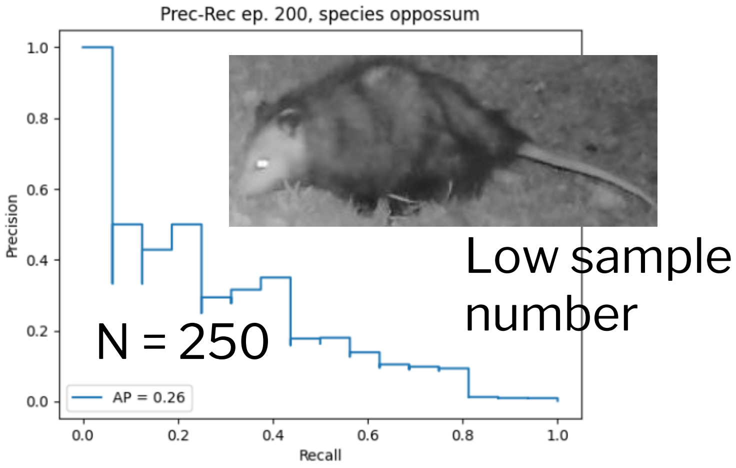
AP = 90

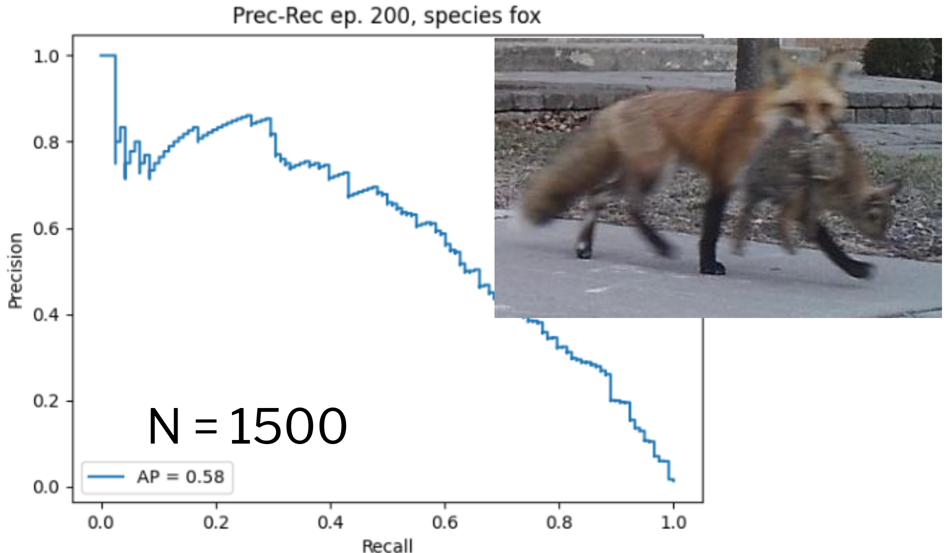
Evaluating the model: class specific metrics

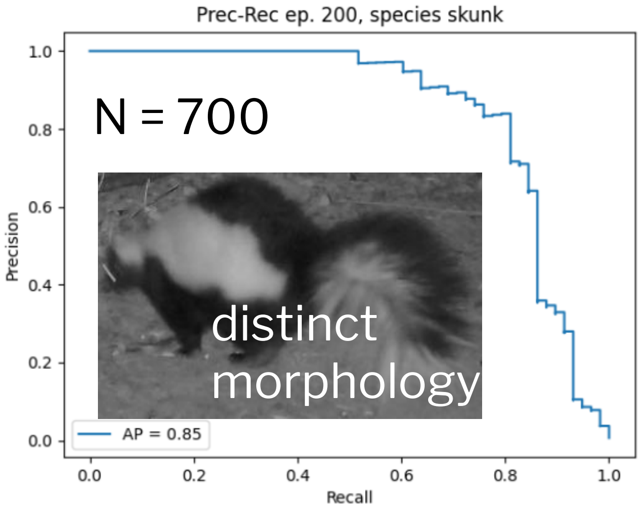
The number of images needed to successfully train a class varied across classes. Animals with more distinct features and consistent behaviour needed less images than less predictable animals like foxes, or with more homogeneous fur like opossums

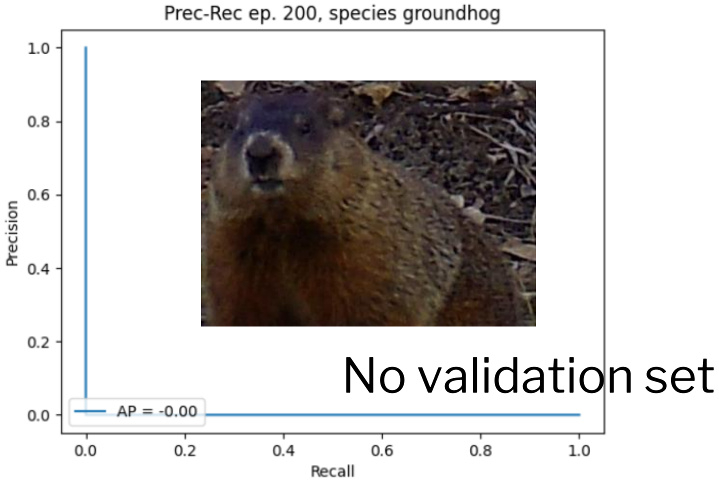










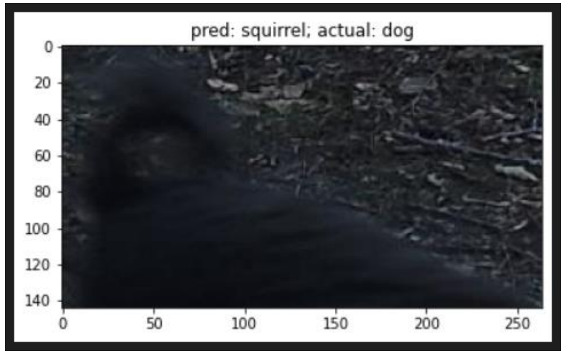


Evaluating the model: visualizing results

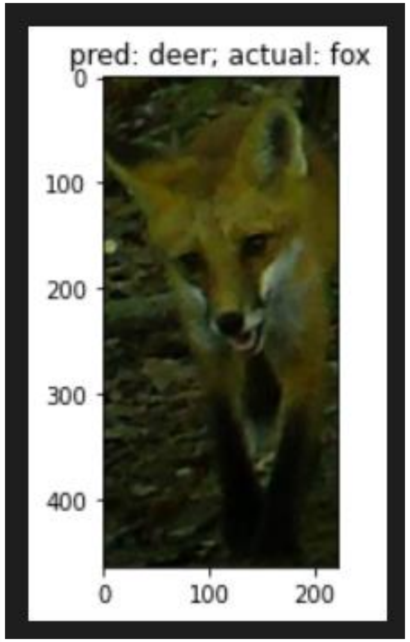
Visualizing the results helped to understand the performance of the model, and the reliability of our data. It was easier to see where the model was failing and find patterns in the errors to decide which steps to take next to improve the model performance

Error patterns based on training data

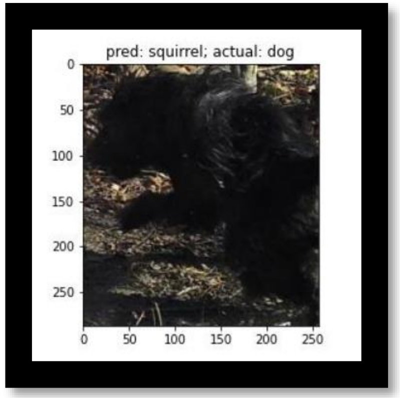
Dog tails predicted as squirrels



Deer usually those staring into the camera

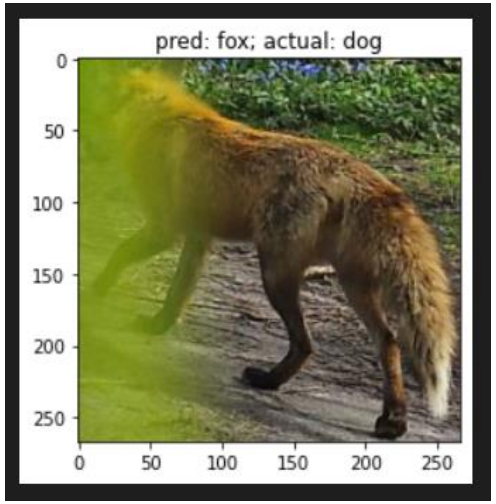


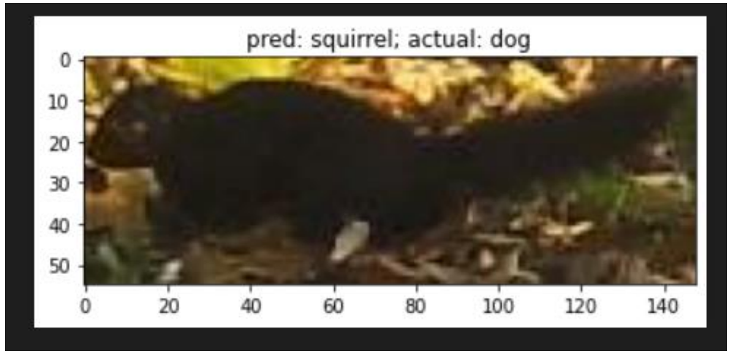
Squirrel or dog?

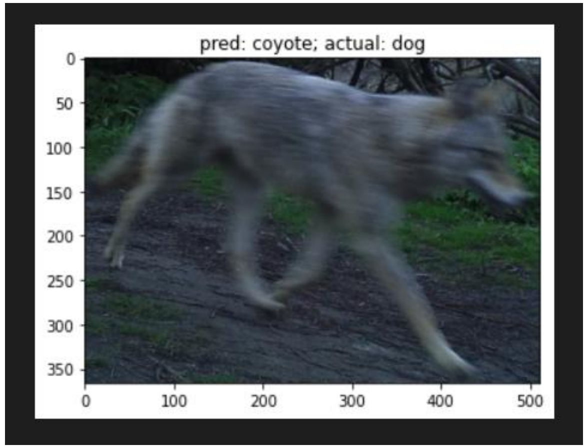


Understandable prediction errors

Classifies correctly the images with wrong labels







TIZIANA GELMI CANDUSSO

Take home message

- Best time investment: **cleaning training data**
- Data augmentation and fine-tuning: **try one at the time**
- **Learn your model evaluation metrics**, and **visualize your predictions** on the images themselves to find patterns and errors in your data

ACKNOWLEDGEMENTS

CV4E

Azure

RESNICKINSTITUTE
science + energy + sustainability

- Benjamin Kellenberger
- Sara Beery
- Justin Kay
- Suzanne Stathatos
- Catherine Breen
- Ethan Shafron
- Casey Youngflesh

UNIVERSITY OF TORONTO

- Fortin lab
- Molnar lab

DFG
Deutsche Forschungsgemeinschaft