Final Year Project Demo

Edge Compute Camera Trap for Object Detection

This presentation

- Covers the problem addressed and my solution
- Outlines the approach taken to create the solution
- Demonstrates the final prototype

Camera traps

- Used in various ecological studies
- Track movement
- Observe behaviour and patterns
- Estimate populations

Disadvantages

- Time to analyse each manually
- False trigger events

Application of AI

- Can be used to go through images and detect animals and species in each
- Reduces time taken to analyse pictures
- Saves man hours

Disadvantages of AI

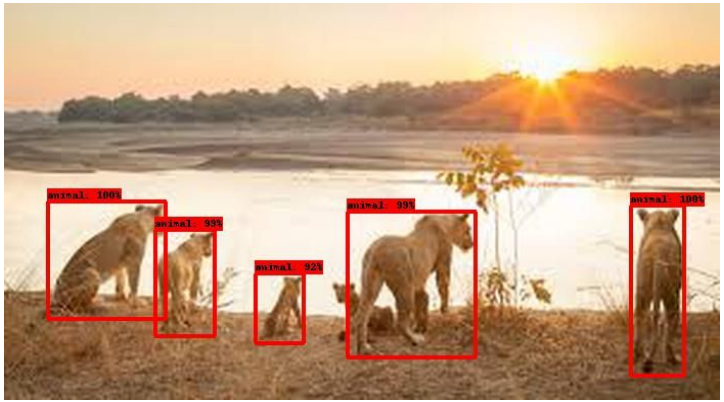
- Often done at offsite data centers
- Large amount of data takes a long time to transfer back and forth
- Sanctuaries often don’t have good internet connection for this

My Solution

- A camera trap that can run AI locally
- Saves man hours for analyzing images
- Saves time spent transferring to data centers
- Uses just one Animal class to help identify wider variety of species

Approach

- First tested MegaDetector by MicroSoft AI for Earth team
- Was not compatible with Intel Neural Compute Stick 2 (NCS2)



Neural Network Version 1

- Trained all versions using dataset from LILA BC
- Tensorboard results looked promising
- This version not compatible with NCS2
- Erroneous detection and subject not detected


TensorBoard
SCALARS
IMAGES
TIME SERIES
INACTIVE
UPLOAD
Show data download links
Ignore outliers in chart scaling
Tooltip sorting method: default
Smoothing
0.6
Horizontal Axis
STEP
RELATIVE
WALL
Runs
Write a regex to filter runs
train
TOGGLE ALL RUNS
./
Loss
Loss/classification_loss
tag: Loss/classification_loss
Loss/localization_loss
tag: Loss/localization_loss
Loss/normalized_total_loss
tag: Loss/normalized_total_loss
Loss/regularization_loss
tag: Loss/regularization_loss
Loss/total_loss
tag: Loss/total_loss
learning_rate
learning_rate
tag: learning_rate


American Black Bear: 32%
Virginia Opossum: 34%
American Crow: 30%
Striped Skunk: 36%
Domestic Cat: 38%
Virginia Opossum: 31%
Northern Raccoo
1:23
0072

Version 2

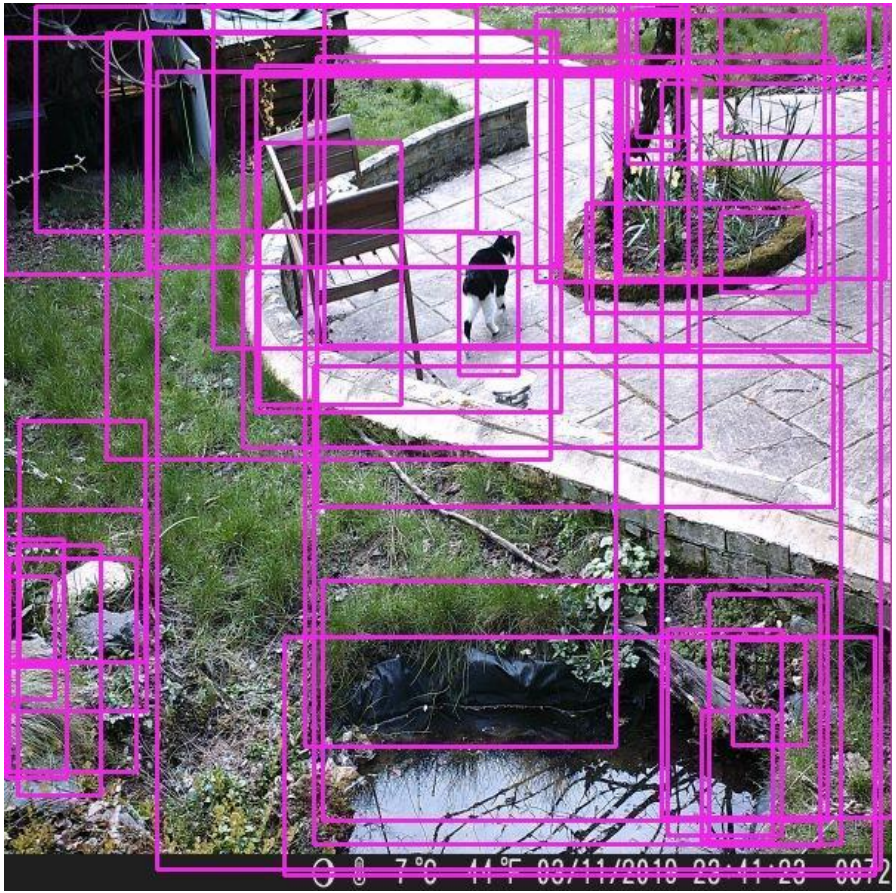
- Better for correct detections
- False positives as well
- Faster R-CNN not compatible with code that was written for SSD


Latency: 340.6 ms
FPS: 6.9
Label
Label #16.0 984
5 °C 41 °F 03/11/2019 22:22:37 0055

Version 3

- Discovered Google Colab and got access to an Nvidia Tesla T4
- Roughly followed a guide for training Tensorflow 1 models
- Manually drew bounding boxes for over 250 images
- After training there were lots of false positive results
- The same occurred for training twice as long


7°C 44°F 03/11/2019 23:4


7°C 44°F 03/11/2010 23:11:23 0072

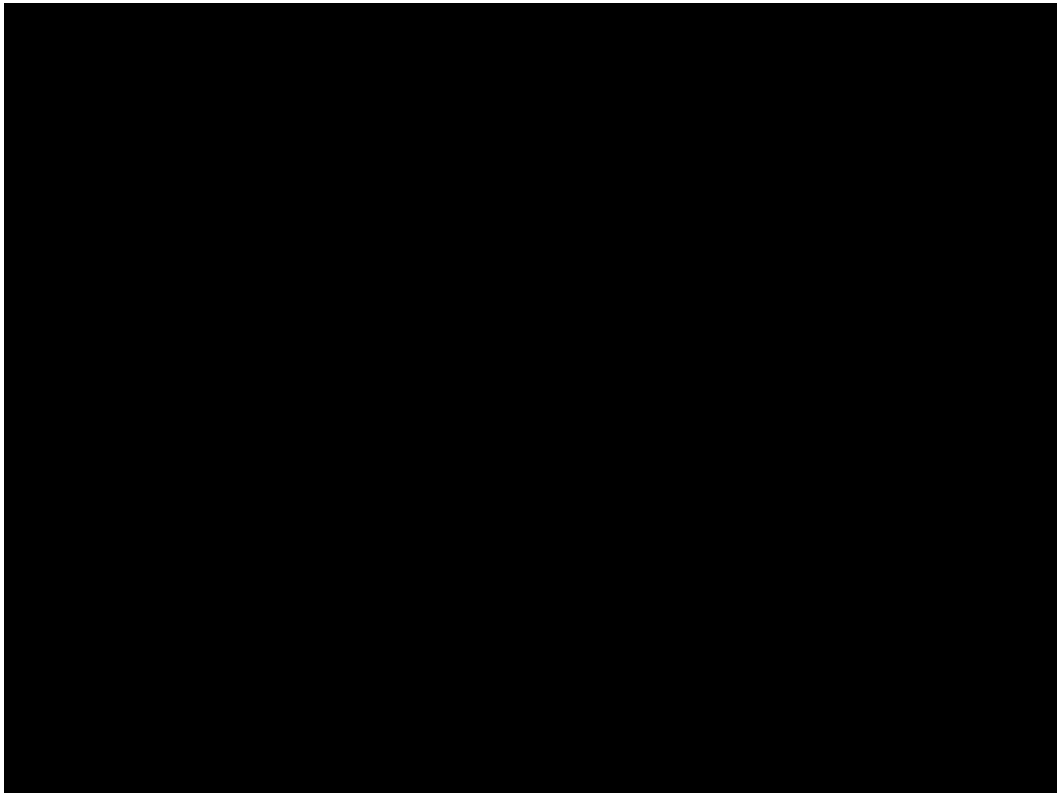
Version 4

- Changed parameters to make SSD Inception V2 model similar to Faster R-CNN Inception V2 model
- Score converter changes to SOFTMAX
- Learning rate changed from momentum based to manual step with much lower values
- Model training and testing was a success


2016-06-12 4:42:38 AM M 8/10 22°C
WC11
RECONYX


2016-06-13 6:45:45 PM M 2/5
28°C
Non-Potable
Use Only
NRC226
RECONYX


2016-01-30 9:17:28 AM M 9/10 1°C
NRC39
RECONYX



To Conclude

- A working Neural Network was created that can detect a variety of species
- This was successfully incorporated with software and hardware to create a working system
- The camera trap can capture images, run inference on them and sort them based on detection