



OTTERS!

Tensorflow object detection

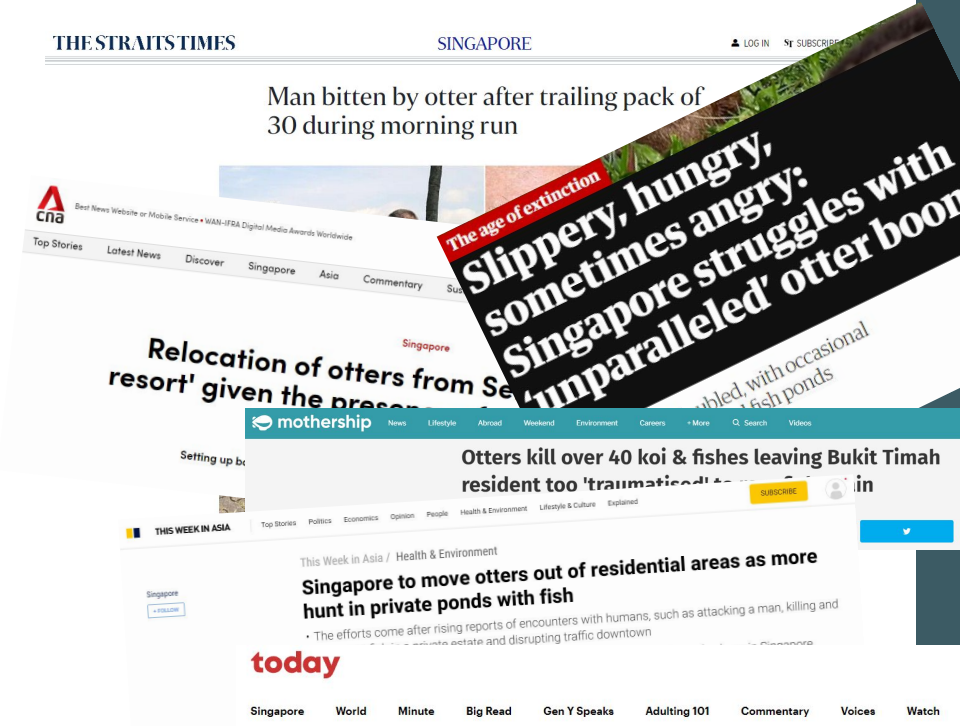
Friend or foe?

By Desmond Yap

BACKGROUND

What is happening in Singapore?

1. Otter population has more than doubled since 2019
2. Number of citizen reports a year about otters have increased from 208 in 2020 to more than 300 in by August in 2022
3. Incidents of otters attacking people - Kallang Riverside Park and Botanic Gardens
4. Incidents of otters killing residents' koi and fishes
5. Sightings were rare up until late 1998 when a pair of otters were spotted at Sungei Buloh Wetlands Reserve
6. Talks on culling or co-existing



Otter population up sharply but still manageable, say experts who urge public to learn to co-exist with them



CONTENT PAGE



01

INTRODUCTION

- Background
- Problem Statement

02

OBJECT DETECTION

- Types of technique
- Mean Average Precision

03

IMAGE DATA

- Flickr scraper API
- Roboflow

04

TENSORFLOW 2

- Training a pretrained model SSD Mobilenet V2
- Model Evaluation

05

STREAMLIT

- Deployment

06

SUMMARY

- Conclusion, limitations and recommendations

01

INTRODUCTION

Background
Problem Statement





PROBLEM STATEMENT



How do we co-exist with the growing otters population

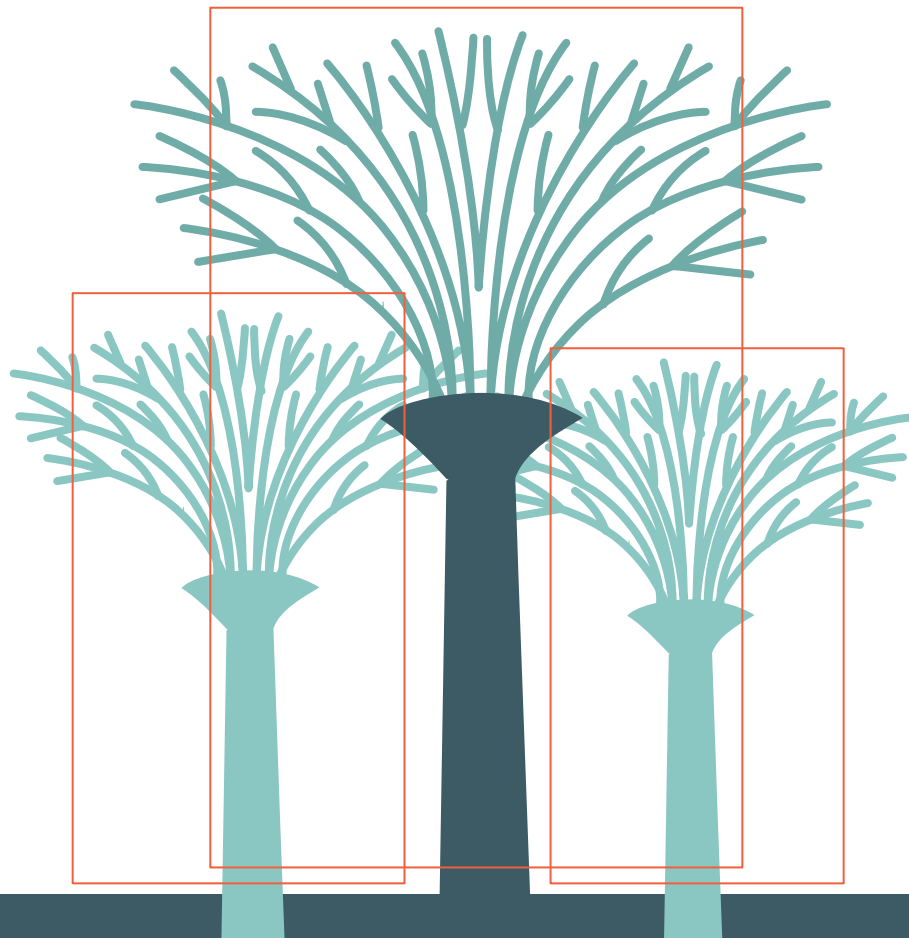
This project aims to detect the number of otters from an image, video or live stream and return the number of counts of otters identified. With that information, it can then be translated into many other uses. For example, security for the home owners and building management - a warning sound could be activate when the number of otters detected is above a threshold number or tracking of the otters by NParks just by counts.



02

OBJECT DETECTION

Types of technique
Mean Average Precision



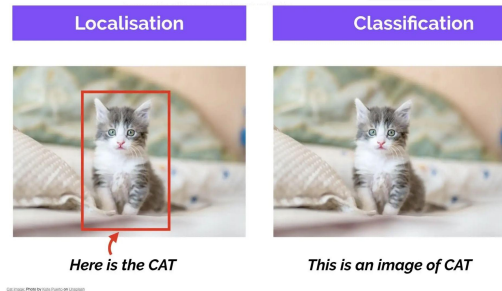
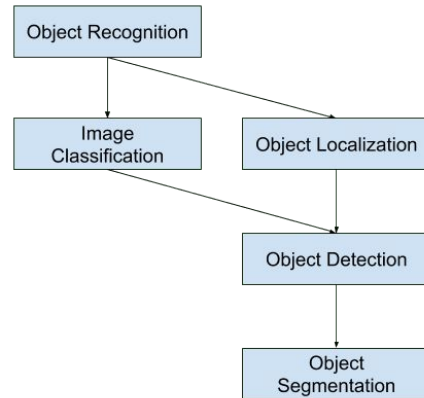
OBJECT RECOGNITION WITH DEEP LEARNING

Object recognition is a general term to describe a collection of related computer vision tasks that involve identifying objects in digital photographs

Image Classification: Predict the type or class of an object in an image

Object Localization: Locate the presence of objects in an image and indicate their location with a bounding box

Object Detection: Locate the presence of objects with a bounding box and types or classes of the located objects in an image



OBJECT RECOGNITION WITH DEEP LEARNING

R-CNN

Region-Based Convolutional Neural Network
R-CNN, Fast R-CNN, Faster R-CNN

YOLO

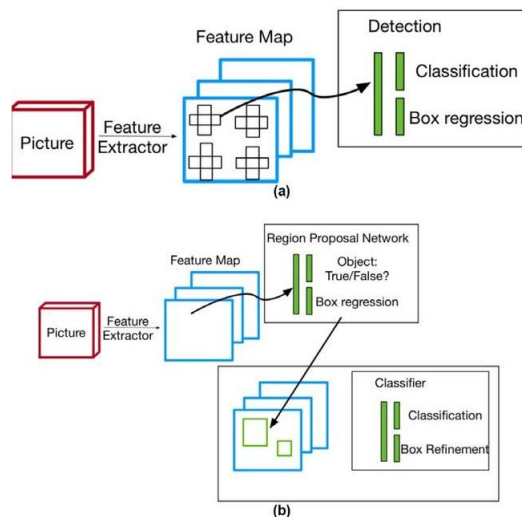
'You Only Look Once'

SSD MobileNet

Single Shot Detectors

Designed to be used in mobile applications and is Tensorflow's first mobile computer vision model

Lightweight deep neural networks



Two-stage (R-CNN)

One-stage (YOLO, SSD)

The major difference between the two is that in the two-stage object detection models, the region of interest is first determined and the detection is then performed only on the region of interest

This implies that the two-stage object detection models are generally more accurate than the one-stage ones but require more computational resources and are slower

TF 2 OBJECT DETECTION MODEL ZOO

Model name	Speed (ms)	COCO mAP	Outputs
SSD MobileNet v2 320x320	19	20.2	Boxes
SSD MobileNet V1 FPN 640x640	48	29.1	Boxes
SSD MobileNet V2 FPNLite 320x320	22	22.2	Boxes
SSD MobileNet V2 FPNLite 640x640	39	28.2	Boxes
Faster R-CNN ResNet50 V1 640x640	53	29.3	Boxes
Faster R-CNN ResNet50 V1 1024x1024	65	31.0	Boxes
Faster R-CNN ResNet50 V1 800x1333	65	31.6	Boxes
Faster R-CNN ResNet101 V1 640x640	55	31.8	Boxes
Faster R-CNN ResNet101 V1 1024x1024	72	37.1	Boxes
Faster R-CNN ResNet101 V1 800x1333	77	36.6	Boxes
Faster R-CNN ResNet152 V1 640x640	64	32.4	Boxes
Faster R-CNN ResNet152 V1 1024x1024	85	37.6	Boxes
Faster R-CNN ResNet152 V1 800x1333	101	37.4	Boxes
Faster R-CNN Inception ResNet V2 640x640	206	37.7	Boxes
Faster R-CNN Inception ResNet V2 1024x1024	236	38.7	Boxes
Mask R-CNN Inception ResNet V2 1024x1024	301	39.0/34.6	Boxes/Masks

What is COCO?



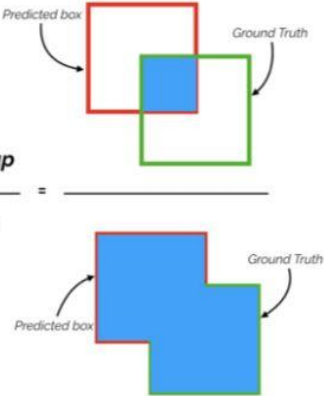
COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

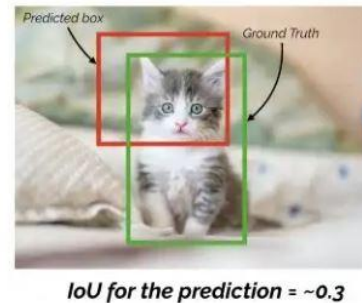
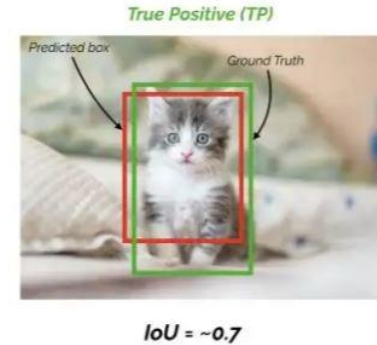
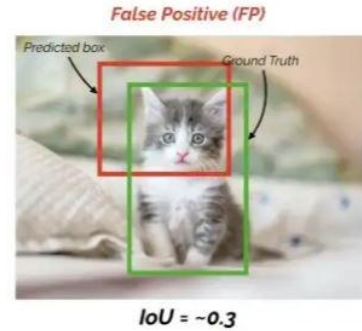
- ✓ Object segmentation
- ✓ Recognition in context
- ✓ Superpixel stuff segmentation
- ✓ 330K images (>200K labeled)
- ✓ 1.5 million object instances
- ✓ 80 object categories
- ✓ 91 stuff categories
- ✓ 5 captions per image
- ✓ 250,000 people with keypoints

Mean Average Precision (mAP)

If IoU threshold = 0.5

Intersection over Union (IoU) = $\frac{\text{Area of Overlap}}{\text{Area of Union}}$





IoU threshold = 0.5 = False Positive (FP)

IoU threshold = 0.2 = True Positive (TP)

Mean Average Precision (mAP)

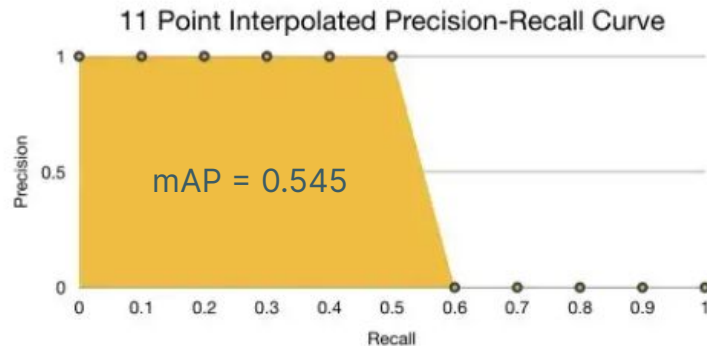
TP = 1, FN = 1, FP = 0



 = Predicted Bounding Box

 = Ground Truth Bounding Box

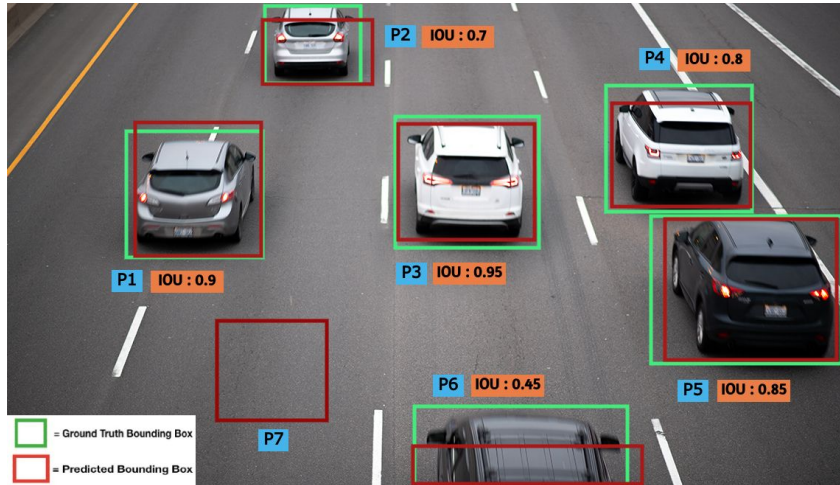
Object detection example for Advanced driver-assistance systems (ADAS)



Rank (confidence)	Correct?	Precision(TP/ TP+FP)	Recall(TP/(TP+FN))
1	True	$1/(1+0) = 1$	$1/(1+1) = 0.5$

Precision and Recall is calculated for each predicted bounding box in the image at a particular IoU threshold ranked according to decreasing confidence of prediction

Mean Average Precision (mAP)



Rank (confidence)	Correct?	Precision(TP/ TP+FP)	Recall(TP/(TP+FN))
P1	True	$1/(1+0) = 1$	$1/(1+5) = 0.167$
P2	True	$2/(2+0) = 1$	$2/(2+4) = 0.33$
P3	True	$3/(3+0) = 1$	$3/(3+3) = 0.5$
P4	True	$4/(4+0) = 1$	$4/(4+2) = 0.67$
P5	True	$5/(5+0) = 1$	$5/(5+1) = 0.83$
P6	True	$6/(6+0) = 1$	$6/(6+0) = 1$
P7	False	$6/(6+1) = 0.857$	$6/(6+0) = 1$

Precision and Recall is calculated for each predicted bounding box in the image at a particular IoU threshold ranked according to decreasing confidence of prediction

03

IMAGE DATA

Flickr Scraper API
Roboflow

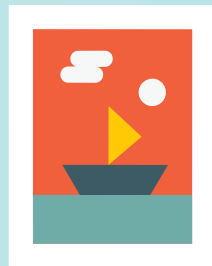
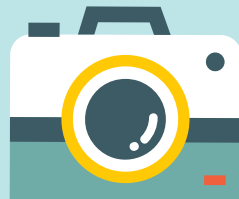


IMAGE DATA PREPARATION



Flickr Scraper API

700 images

The RoboFlow logo, which consists of the word 'roboflow' in a white, lowercase, sans-serif font, centered within a solid purple rectangular box.

roboflow

**500 training
50 validation**

IMAGE DATA PREPARATION



Labels

AI assist labelling

TF Record

For Tensorflow 2

Preprocessing

Resize

Augmentation

Otters3 Image Dataset

[+ Generate New Version](#)

VERSIONS

2022-11-16 12:09am
v2 Nov 16, 2022

2022-11-09 1:35pm
v1 Nov 9, 2022

2022-11-16 12:09am

Version 2 Generated Nov 16, 2022

[Export](#)

[Edit](#)

TRAINING OPTIONS

Use Roboflow Train

Let us train your model and get results within 24 hours along with a hosted API endpoint for making predictions. [Learn More >>](#)

[Start Training](#)

Available Credits: 3

IMAGES



550 images

[View All Images >>](#)

TRAIN / TEST SPLIT

Training Set

91%

500 images

Validation Set

9%

50 images

Testing Set

0%

images

PREPROCESSING

Resize: Stretch to 320x320

AUGMENTATIONS

No augmentations were applied.

DETAILS

Version Name: 2022-11-16 12:09am

Version ID: 2

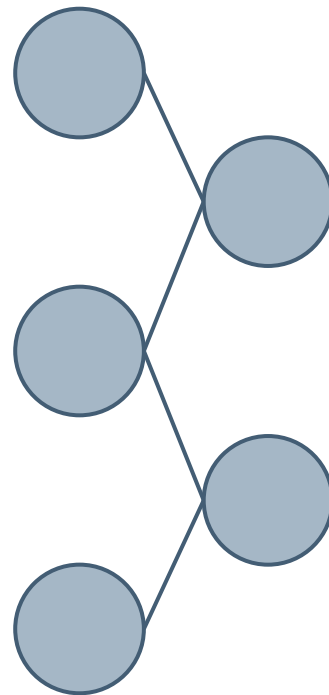
Generated: Nov 16, 2022

Annotation Group: otter

04

TENSORFLOW 2

Training a pretrained model
Model evaluation



TENSORBOARD MODEL EVALUATION



ssd_mobilenet_v2_fpnlite_640×640_coco17_tpu-8 with 2000 training steps, 500 training images, 50 validation images

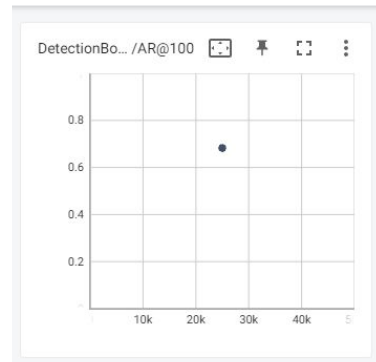
Average Precision (AP) @[IoU=0.50	area=	all	maxDets=100] = 0.661
Average Precision (AP) @[IoU=0.75	area=	all	maxDets=100] = 0.252
Average Recall (AR) @[IoU=0.50:0.95	area=	all	maxDets=100] = 0.576

ssd_mobilenet_v2_fpnlite_320×320_coco17_tpu-8 with 25000 training steps, 500 training images, 50 validation images

Average Precision (AP) @[IoU=0.50	area=	all	maxDets=100] = 0.822
Average Precision (AP) @[IoU=0.75	area=	all	maxDets=100] = 0.617
Average Recall (AR) @[IoU=0.50:0.95	area=	all	maxDets=100] = 0.683

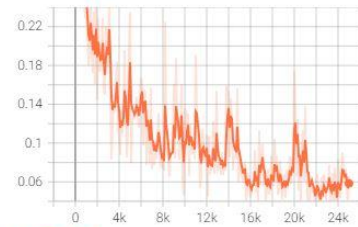
TENSORBOARD MODEL EVALUATION

📌 Pinned 4 cards

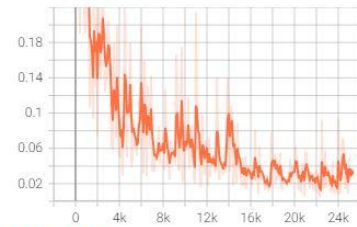


Loss

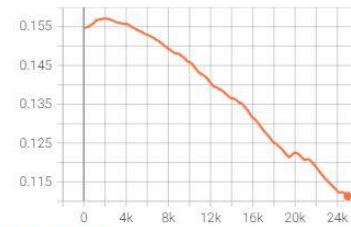
Loss/classification_loss
tag: Loss/classification_loss



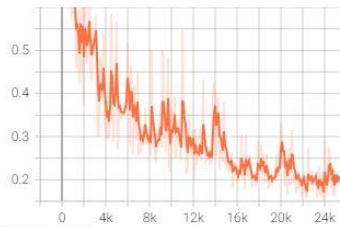
Loss/localization_loss
tag: Loss/localization_loss



Loss/regularization_loss
tag: Loss/regularization_loss



Loss/total_loss
tag: Loss/total_loss



TENSORBOARD MODEL EVALUATION

TensorBoard

TIME SERIES SCALARS IMAGES

INACTIVE

UPLOAD

Filter tags (regular expressions supported)

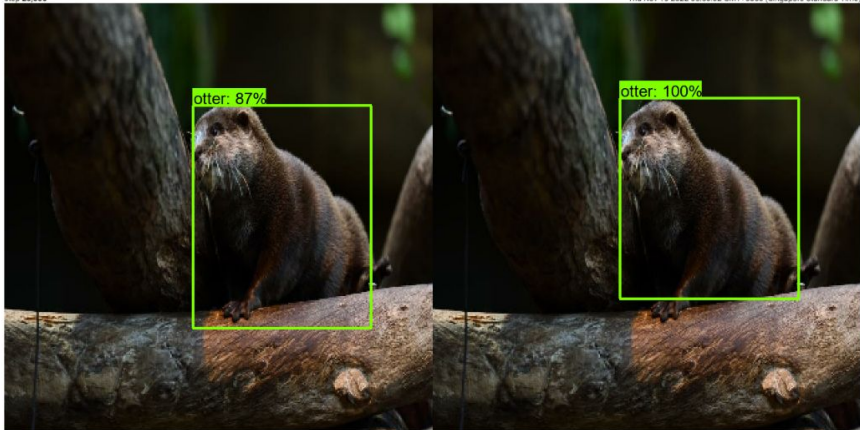
eval_side_by_side_0_0

eval_side_by_side_0_0
tag eval_side_by_side_0_0
step 25,000

Thu Nov 10 2022 05:56:02 GMT+0800 (Singapore Standard Time)

otter: 87%

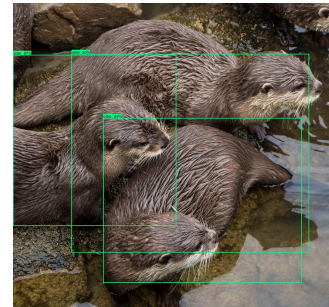
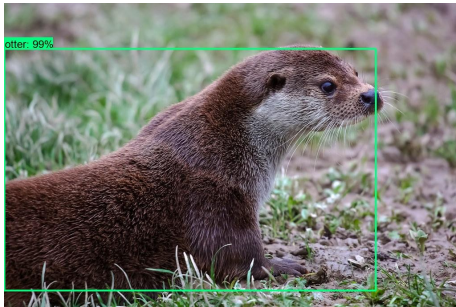
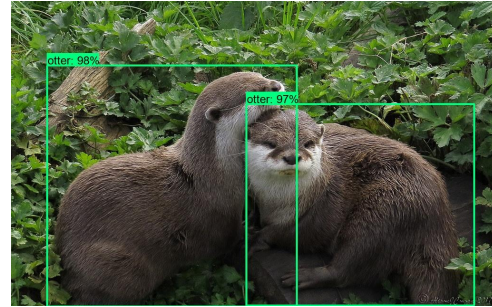
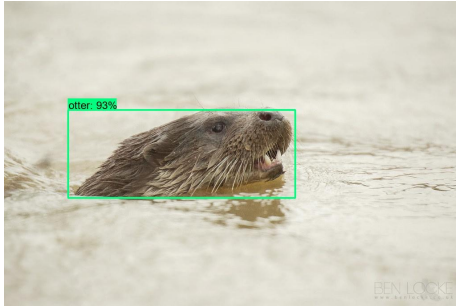
otter: 100%



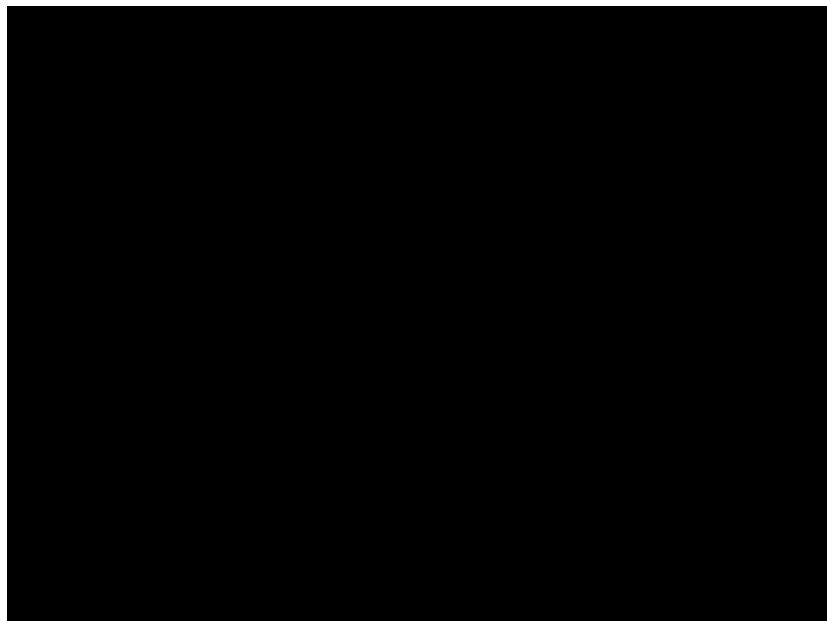
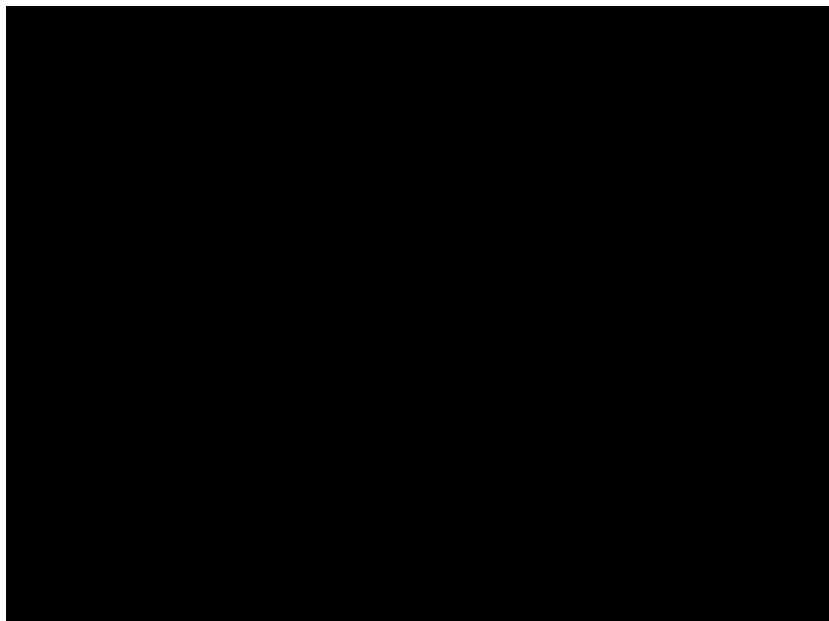
The screenshot displays the TensorBoard web interface for model evaluation. The top navigation bar is orange and contains the 'TensorBoard' logo, tabs for 'TIME SERIES', 'SCALARS', and 'IMAGES', and a status indicator 'INACTIVE' with an 'UPLOAD' button. Below the navigation bar, the left sidebar contains controls for image display: a checkbox for 'Show actual image size', sliders for 'Brightness adjustment' and 'Contrast adjustment' (each with a 'RESET' button), and a 'Runs' section with a text input for filtering and a 'TOGGLE ALL RUNS' button. The main content area shows a search bar for filtering tags. Below the search bar, the selected run 'eval_side_by_side_0_0' is displayed, including its tags and step number. The central part of the interface shows two side-by-side images of an otter on a log. The left image has a green bounding box around the otter with the label 'otter: 87%', and the right image has a green bounding box with the label 'otter: 100%'. A timestamp 'Thu Nov 10 2022 05:56:02 GMT+0800 (Singapore Standard Time)' is visible above the images.



MODEL PREDICTION WITH IMAGES



MODEL PREDICTION WITH VIDEO



MODEL PREDICTION WITH LIVE WEBCAM

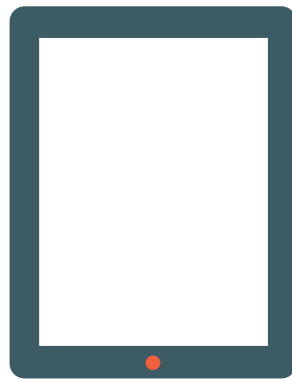




05

STREAMLIT

Deployment





STREAMLIT DEPLOYMENT

Simulation of surveillance camera

USER INPUT

Upload or take photos

PROCESS

Utilizes TF Lite model for processing

OUTPUT

Returns bounding boxes with confidence
Number of otters detected
Warning sound if 3 or more otters are detected

LIMITATIONS

Live streaming possible but excluded in this exercise
TF Lite model is less capable after exporting but lighter

06

SUMMARY

Conclusion, limitations and
recommendations

Otter?





SUMMARY

Conclusion

As the otters' population grows, it would be necessary to find methods to co-exist with them. There are two main concerns to be addressed, from residents and building managements' point of view - a deterrence, and from local authority's point of view - a way to track and identify the otters

SSD Mobilenet V2 returned decent scores

Average Precision (AP) @[IoU=0.50	area=	all	maxDets=100] = 0.822
Average Precision (AP) @[IoU=0.75	area=	all	maxDets=100] = 0.617
Average Recall (AR) @[IoU=0.50:0.95	area=	all	maxDets=100] = 0.683

Computer vision/ object recognition can be very adaptable - the usage should be compounded with other current measures in place to deter otters (deployment of inanimate objects that can be on guard 24/7 is better than risking being attacked by otters if they turn aggressive)

Limitations

Time required for model training, GPU, number of image dataset, preprocessing and augmentation





SUMMARY

Recommendation

The number of models and improvements are constantly being pushed out

Increase the number of images collected, quality of photo, different views and distant in the images play a part in model training and prediction

Does not have to stop at just otters - it is very possible to include any other kind of wildlife, object detection is capable to handle more than one class

It is also possible to retrieve counts from a live stream video - more akin to a real surveillance camera



THANK YOU



Streamlit - Try it!



<https://www.linkedin.com/in/desmond-jjyap/>



<https://github.com/DesmondYapJJ>



desmondyap_jj@hotmail.com

