Automatic Detection of Endangered Species in Aerial Images Using Deep Learning

by Daniel Torres Cirina

Bachelor's Thesis in Informatics Engineering
Specialization in Computing
Director: Enrique Romero Merino
Co-Director: Ludwig Houegnigan
April 2021

UPC-FIB

Contents

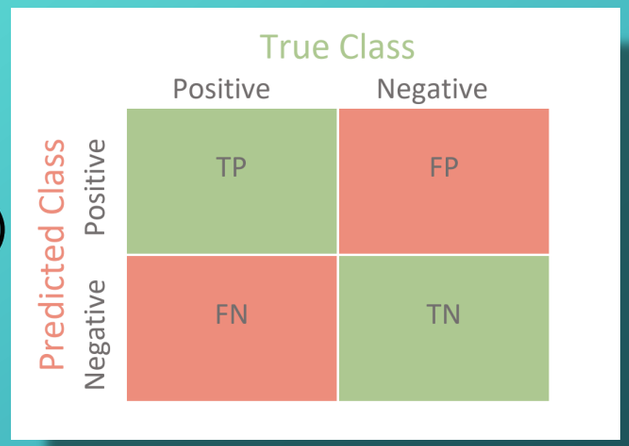
1. Context and scope

- Learn about the different kinds of machine learning architectures.
- See an existing project previously developed by Rubén Martín Pinardel for his Master's Thesis.
- Build a complete dataset of whales from scratch.
- Conduct experiments to see how the models perform in different scenarios.

2. Theoretical background

- True Positive (TP) is a valid detection.
- False Positive (FP) is an invalid detection.
- False Negative (FN) is a ground-truth box missed by the model.

(1)



(2)

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{TP}{\text{all detections}}$$

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{TP}{\text{all ground-truths}}$$

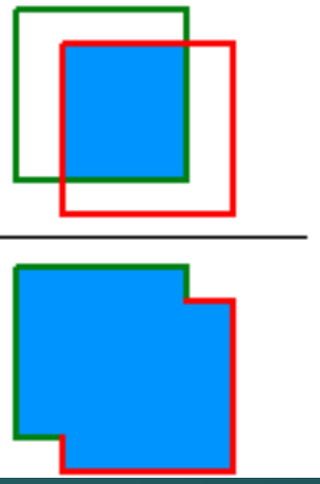
(3)

(4)

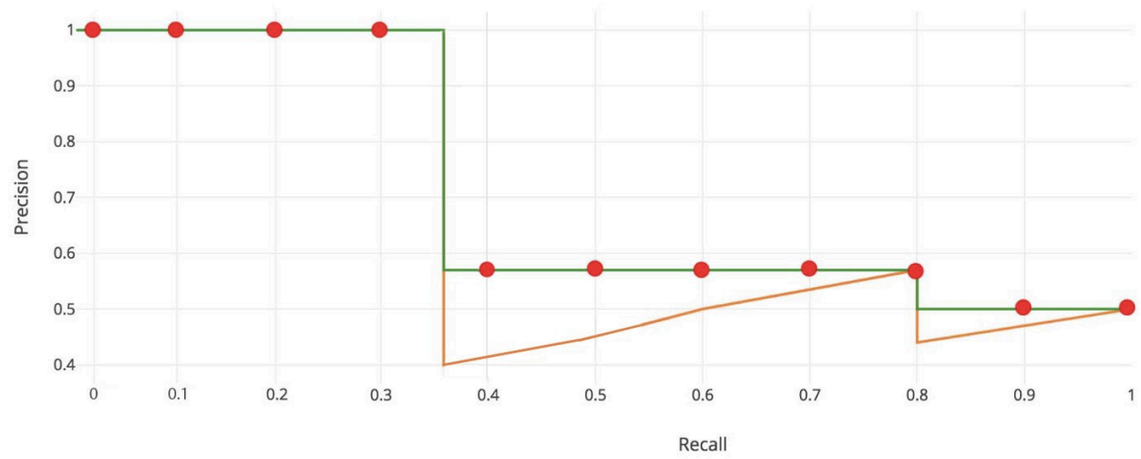
2. Theoretical background

- Intersection over union (IoU) measures the overlap area between the predicted bounding box and the ground-truth box.

(5) $$IOU = \frac{\text{area of overlap}}{\text{area of union}} =$$



(6)



- Average precision (AP) is the average of the areas under the 101-point smoothened interpolated precision/recall curves for every IoU thresholds in the range [0.50:0.05:0.95].

- Average recall (AR) is the recall averaged of all predictions for every IoU thresholds in the range [0.50:0.05:0.95].

- The possible values for IoU, AP and AR all vary between 0 and 1.

3. Understanding the architecture

- Faster R-CNN with Inception-ResNet V2 feature extractor was the architecture that obtained the best results.

- Convolutional neural network (CNN). It has a large number of hidden layers and uses convolution.

- Faster R-CNN is one of the most popular two-stage object detectors.

- Two-stage detectors use a Region Proposal Network (RPN) to generate regions of interest and feed a second stage with them in order to classify the object.

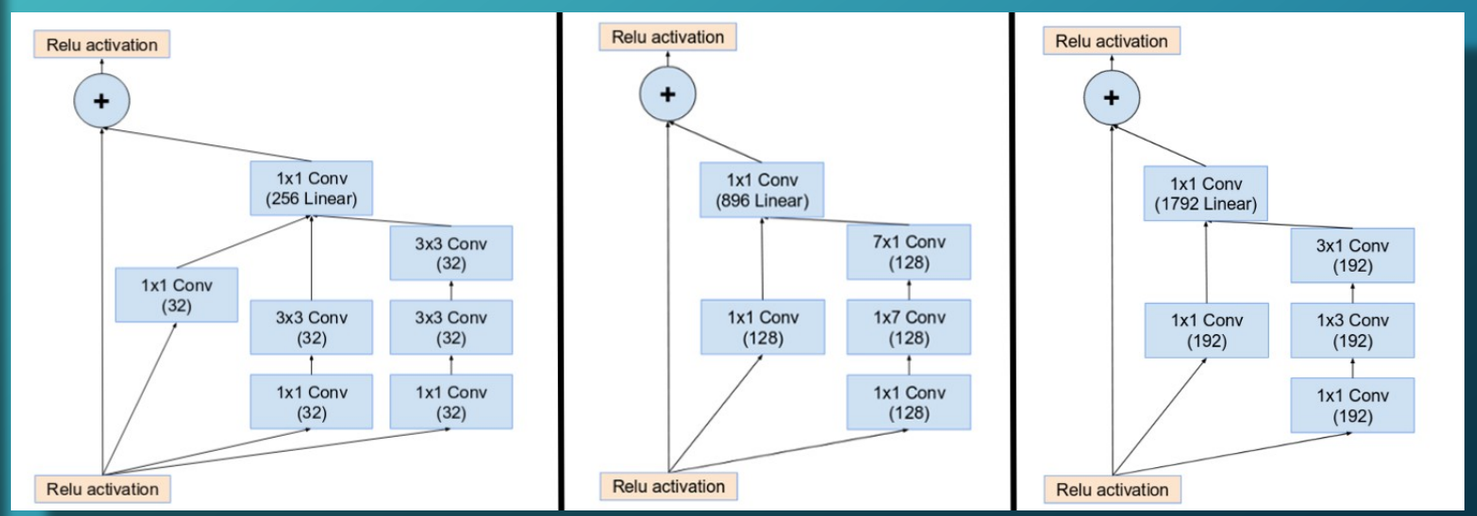
3. Understanding the architecture

3.1. Feature extractor

Scheme for Inception-ResNet V2

(7)

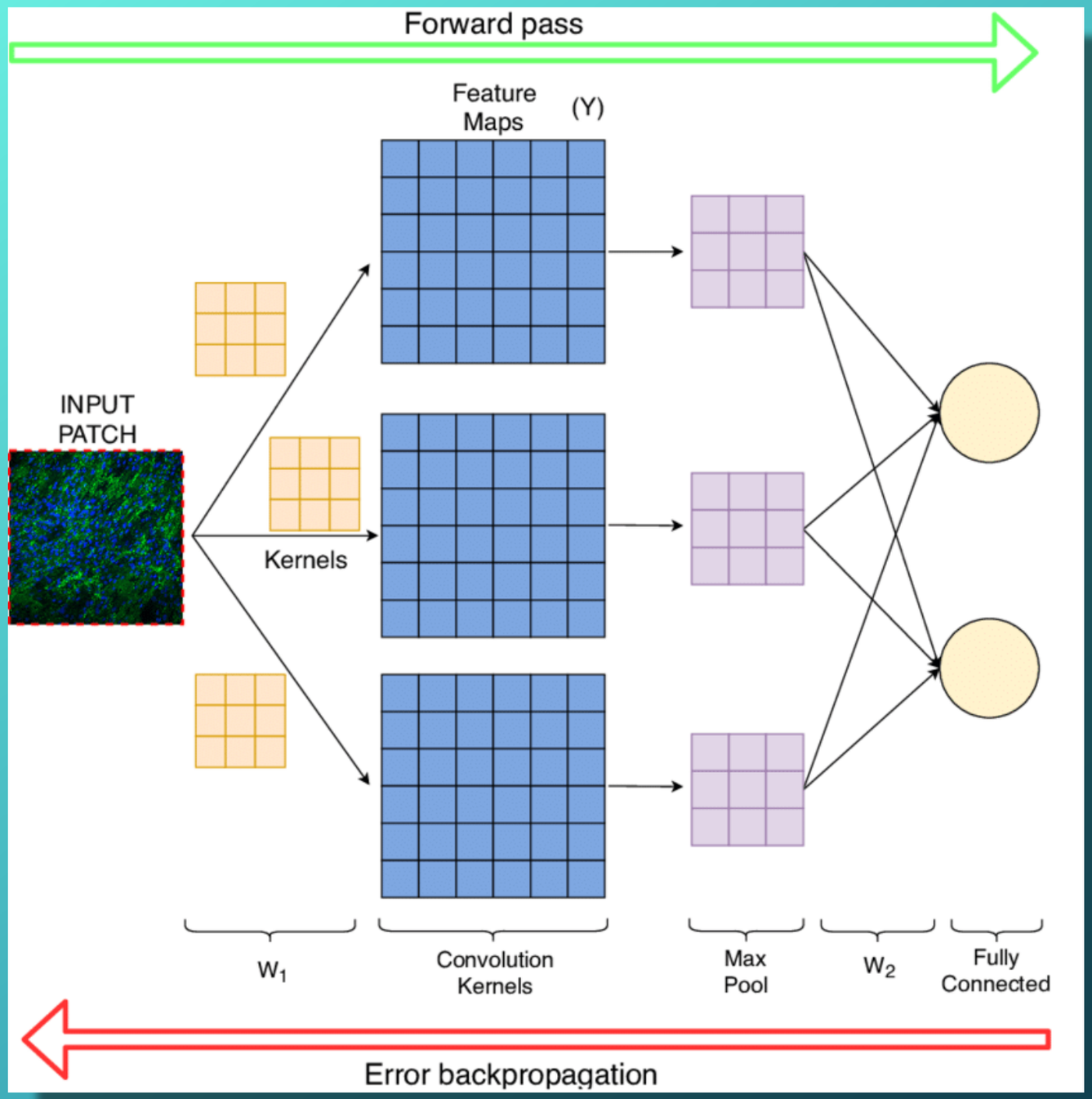
Inception modules A, B & C



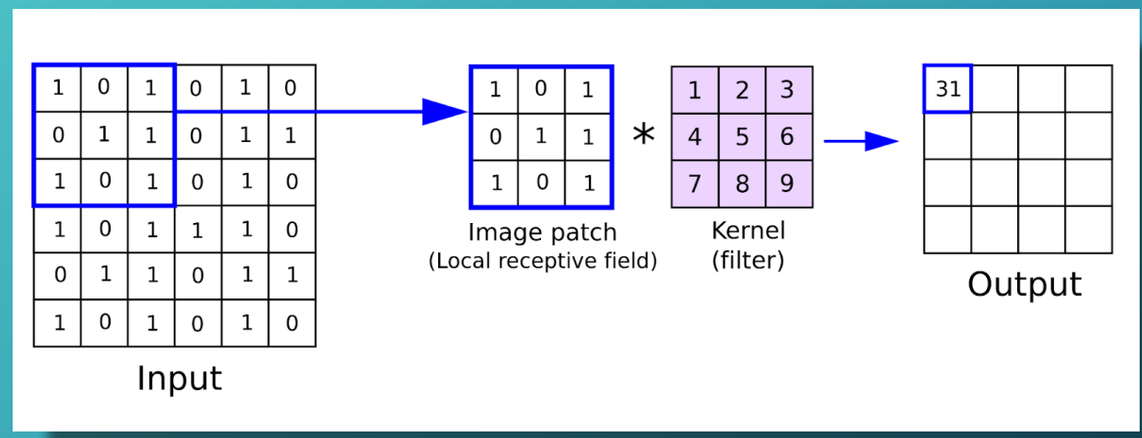
(8)

3. Understanding the architecture

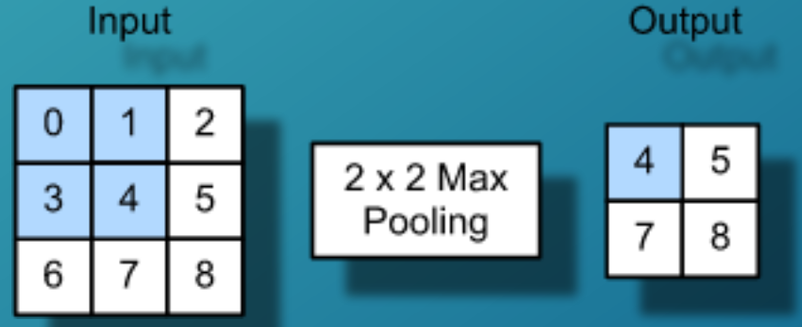
3.2. Backpropagation in the CNN



(9) Backpropagation process divided into the forwards pass and the error backpropagation.



(10) Convolutional layer where the filter is convoluted through the image patch to produce the output.



(11) Applying maximum pooling to a subsection of size 2x2.

Episode IV – A New ~~*Hope*~~ Dataset

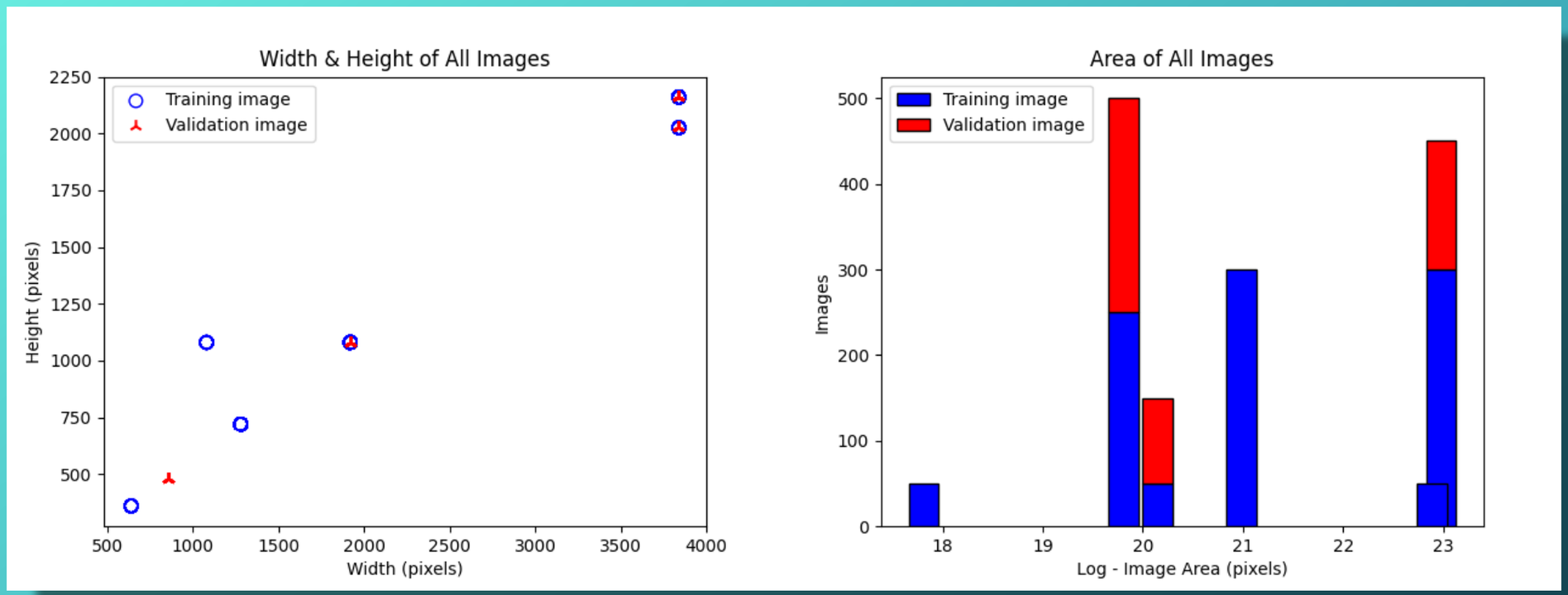
- Existing project dataset - 4,212 images (6,760 ground-truth boxes).
- + 1,000 images for training (1,438 ground-truth boxes).
- + 500 images for validation (745 ground-truth boxes).
- 30 subsets of 50 images each. 20 subsets for training, 10 subsets for validation.

(12)

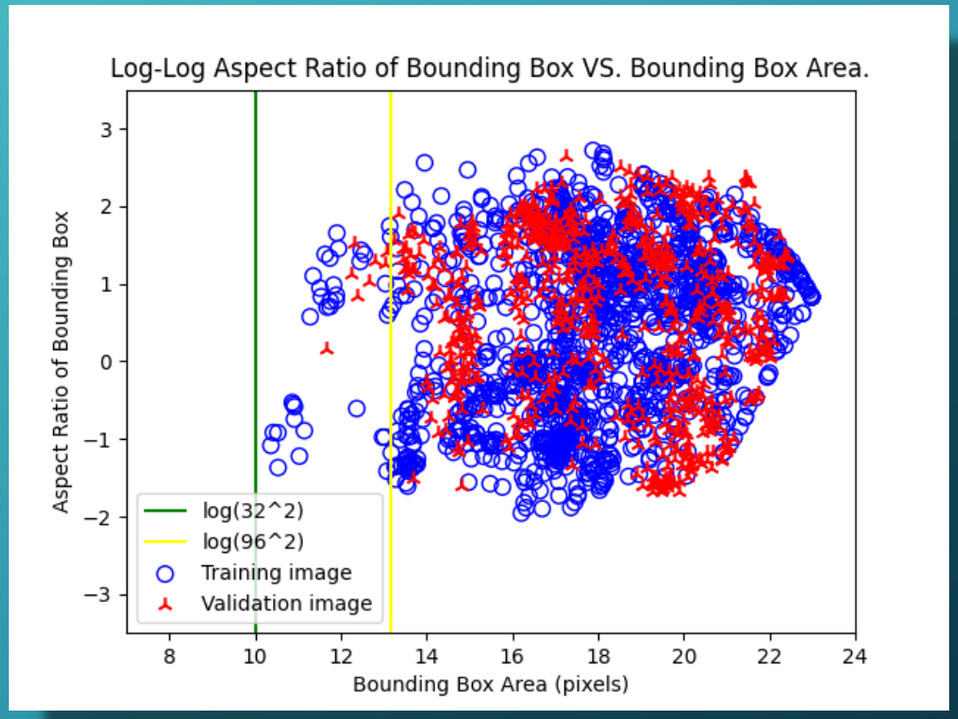
Whale species	Dataset	Number of ground-truth boxes
Blue whale	Training	158
Blue whale	Validation	135
Fin whale	Training	309
Fin whale	Validation	192
Grey whale	Training	368
Grey whale	Validation	130
Humpback whale	Training	253
Humpback whale	Validation	117
Right whale	Training	350
Right whale	Validation	171

Episode IV – A New ~~*Hope*~~ Dataset

(13)



(14)



(15)

filename	width	height	class	xmin	ymin	xmax	ymax
/home/danieltorres/dataset/TrainFasterV2/Blue Whale/Validation/1dbyKDw27h8/subset/frame116_4.jpg	1327	392	Whales	1361	1564	2688	1956

5. Experiments and results

5.1. Choosing a pre-trained model

- We need a pre-trained model to use as a starting point.
- We will recreate the already existing project so the result will be our starting point.
- To recreate it we also need another pre-trained model.

5. Experiments and results

5.1. Choosing a pre-trained model

Options:

1. No pre-trained model.
2. Model pre-trained on the COCO dataset. TensorFlow 1 Detection Model Zoo.
3. Model already stored in the server.

Training:

- Existing project - 4,212 images (6,760 ground-truth boxes).
- 65,000 steps.

Validation:

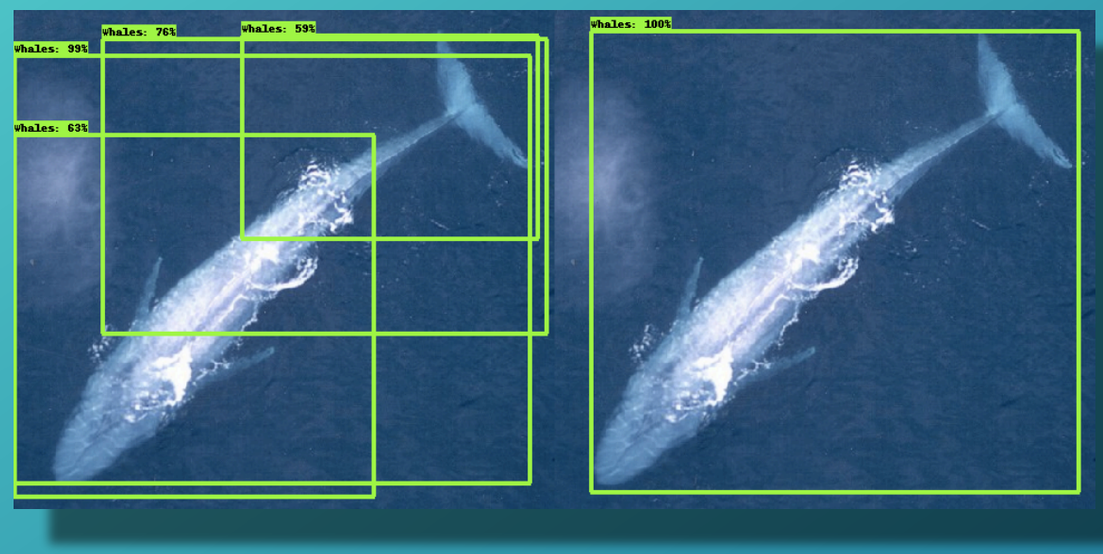
- We’ll use the same dataset used for training.

5. Experiments and results

5.1. Choosing a pre-trained model

No pre-trained model	
Average Precision (AP) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.439
Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.566

(16)



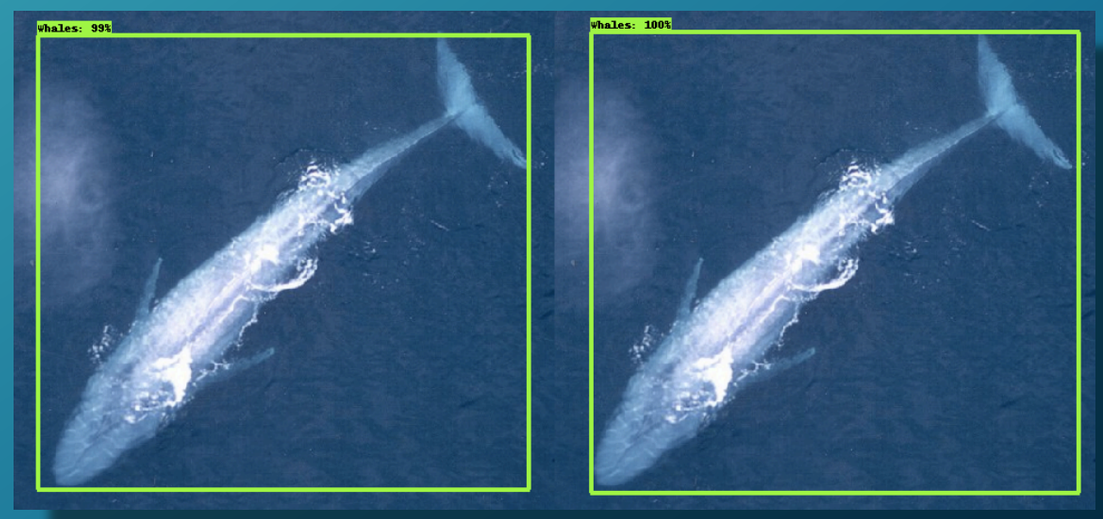
(19)

TensorFlow 1 Detection Model Zoo	
Average Precision (AP) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.863
Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.886

(17)

Model stored in the server used for Ruben's thesis.	
Average Precision (AP) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.852
Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.879

(18)



(20)

5. Experiments and results

5.1. Choosing a pre-trained model

(16)

No pre-trained model	
Average Precision (AP) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.439
Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.566

(17)

TensorFlow 1 Detection Model Zoo	
Average Precision (AP) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.863
Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.886

(18)

Model stored in the server used for Ruben's thesis.	
Average Precision (AP) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.852
Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.879

We will use the obtained model to validate our data and have the resulting COCO metrics as a reference (iteration 0).

5. Experiments and results

5.2. Types of trainings

Using the existing project training data.

Adding 2 training subsets at a time (per iteration).

Non-cumulative training (Type A):

- For every iteration, start training from pre-trained model.

Cumulative training (Type B):

- For every iteration, start training from the last checkpoint.

Number of training steps: 3K, 10K & 30K.

Validating eleven times. The whole validation dataset + one per subset.

5. Experiments and results

Non-cumulative training (Type A) - 3K training steps

(21)

MS COCO Metrics / Iteration	0	1	2	3	4	5	6	7	8	9	10
Average Precision (AP) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.445	0.462	0.480	0.477	0.475	0.483	0.455	0.508	0.512	0.510	0.521
Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.567	0.573	0.580	0.579	0.585	0.606	0.574	0.607	0.617	0.608	0.619

Non-cumulative training (Type A) - 10K training steps

(22)

MS COCO Metrics / Iteration	0	1	2	3	4	5	6	7	8	9	10
Average Precision (AP) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.445	0.454	0.468	0.483	0.473	0.498	0.507	0.506	0.522	0.530	0.538
Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.567	0.568	0.576	0.582	0.591	0.609	0.612	0.612	0.610	0.631	0.631

Non-cumulative training (Type A) - 30K training steps

(23)

MS COCO Metrics / Iteration	0	1	2	3	4	5	6	7	8	9	10
Average Precision (AP) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.445	0.456	0.469	0.487	0.478	0.458	0.494	0.489	0.496	0.524	0.521
Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.567	0.567	0.586	0.581	0.591	0.571	0.598	0.595	0.597	0.626	0.615

Cumulative training (Type B) - 3K training steps

(24)

MS COCO Metrics / Iteration	0	1	2	3	4	5	6	7	8	9	10
Average Precision (AP) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.445	0.468	0.476	0.488	0.475	0.510	0.521	0.509	0.525	0.529	0.525
Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.567	0.582	0.586	0.588	0.580	0.612	0.622	0.603	0.620	0.629	0.620

Cumulative training (Type B) - 10K training steps

(25)

MS COCO Metrics / Iteration	0	1	2	3	4	5	6	7	8	9	10
Average Precision (AP) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.445	0.480	0.465	0.477	0.471	0.479	0.493	0.482	0.484	0.510	0.496
Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.567	0.589	0.573	0.581	0.583	0.604	0.612	0.611	0.614	0.626	0.616

Cumulative training (Type B) - 30K training steps

(26)

MS COCO Metrics / Iteration	0	1	2	3	4	5	6	7	8	9	10
Average Precision (AP) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.445	0.474	0.451	0.479	0.430	0.441	0.462	0.462	0.470	0.487	0.484
Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.567	0.581	0.573	0.601	0.573	0.599	0.604	0.603	0.612	0.620	0.617

5. Experiments and results

5.3. Confusion matrices

Non-cumulative training (Type A) - 10K training steps

(27)

MS COCO Metrics / Iteration	0	1	2	3	4	5	6	7	8	9	10
Average Precision (AP) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.298	0.358	0.390	0.408	0.364	0.460	0.424	0.414	0.435	0.456	0.491
Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.442	0.488	0.506	0.531	0.482	0.558	0.528	0.521	0.532	0.553	0.573

Cumulative training (Type B) - 3K training steps

(28)

MS COCO Metrics / Iteration	0	1	2	3	4	5	6	7	8	9	10
Average Precision (AP) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.298	0.398	0.413	0.393	0.396	0.413	0.449	0.457	0.474	0.445	0.418
Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.442	0.503	0.519	0.494	0.503	0.520	0.555	0.561	0.560	0.536	0.516

(29)

		True condition	
		T	F
Predicted condition	T	95	14
	F	21	0

(30)

		True condition	
		T	F
Predicted condition	T	86	8
	F	30	0

(31) Non-cumulative training (Type A) - 10K training steps

(32) Cumulative training (Type B) - 3K training steps

5. Experiments and results

5.3. Confusion matrices

Non-cumulative training (Type A) - 10K training steps

(33)

MS COCO Metrics / Iteration	0	1	2	3	4	5	6	7	8	9	10
Average Precision (AP) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.521	0.536	0.534	0.588	0.511	0.551	0.519	0.574	0.548	0.546	0.580
Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.624	0.634	0.640	0.664	0.638	0.638	0.624	0.650	0.622	0.638	0.658

Cumulative training (Type B) - 3K training steps

(34)

MS COCO Metrics / Iteration	0	1	2	3	4	5	6	7	8	9	10
Average Precision (AP) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.521	0.518	0.481	0.538	0.482	0.552	0.528	0.580	0.564	0.555	0.621
Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.624	0.628	0.616	0.634	0.624	0.660	0.626	0.638	0.652	0.648	0.688

(35)

		True condition	
		T	F
Predicted condition	T	50	34
	F	0	0

(36)

		True condition	
		T	F
Predicted condition	T	50	17
	F	0	0

(37) Non-cumulative training (Type A) - 10K training steps

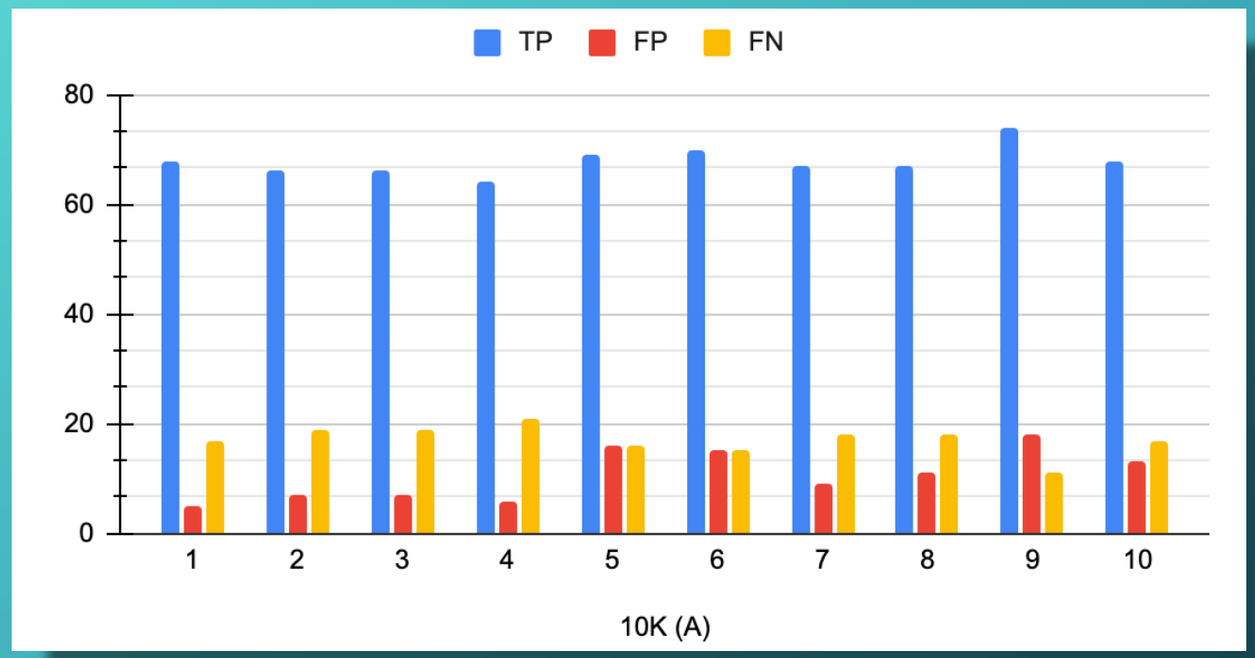
(38) Cumulative training (Type B) - 3K training steps

5. Experiments and results

5.4. Data augmentation

- The number of false positives is higher than normal and the number of false negatives doesn't decrease.

(39)

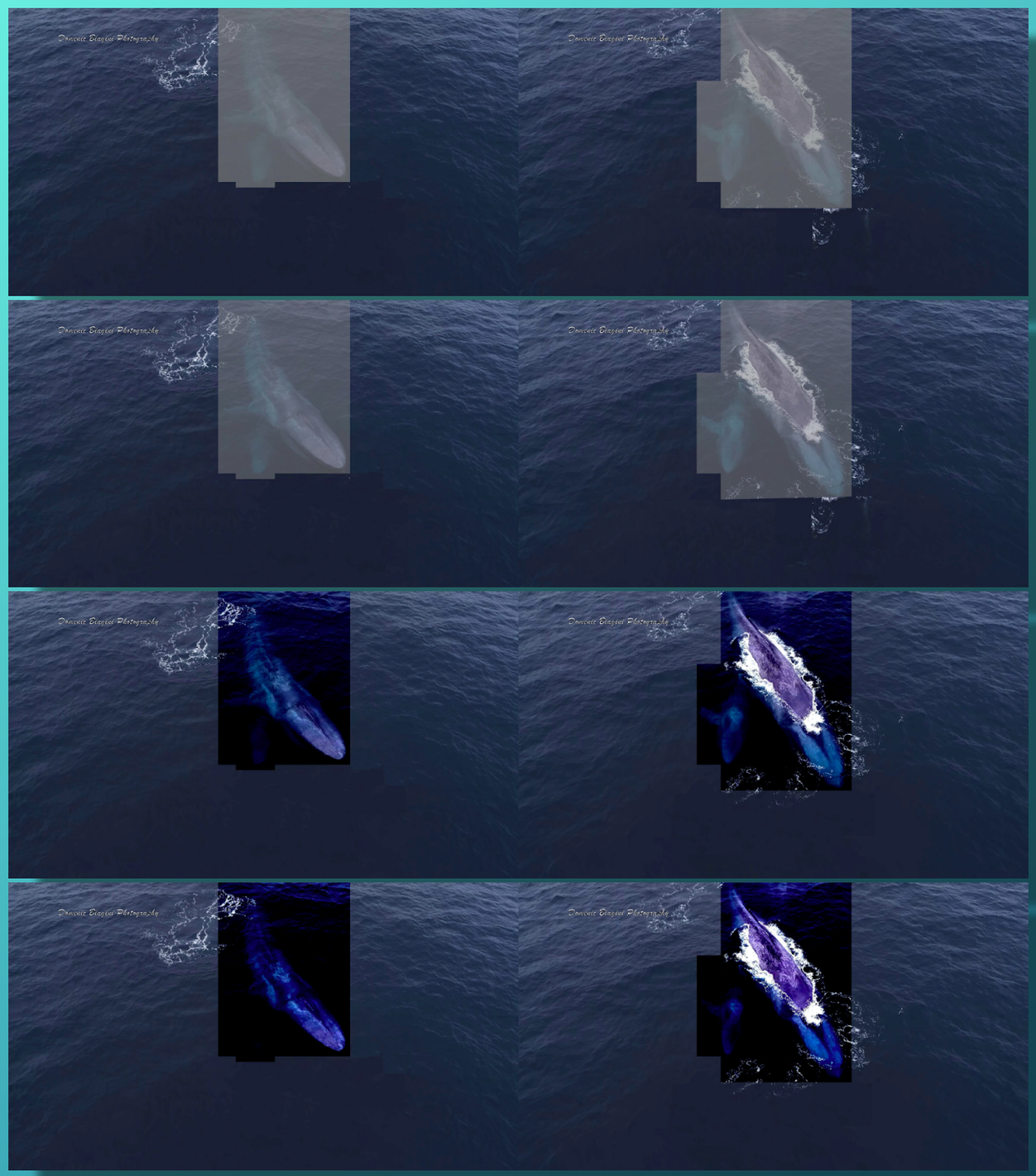


(40)

- Cut the dolphins out and apply contrast variations inside the bounding boxes.

5. Experiments and results

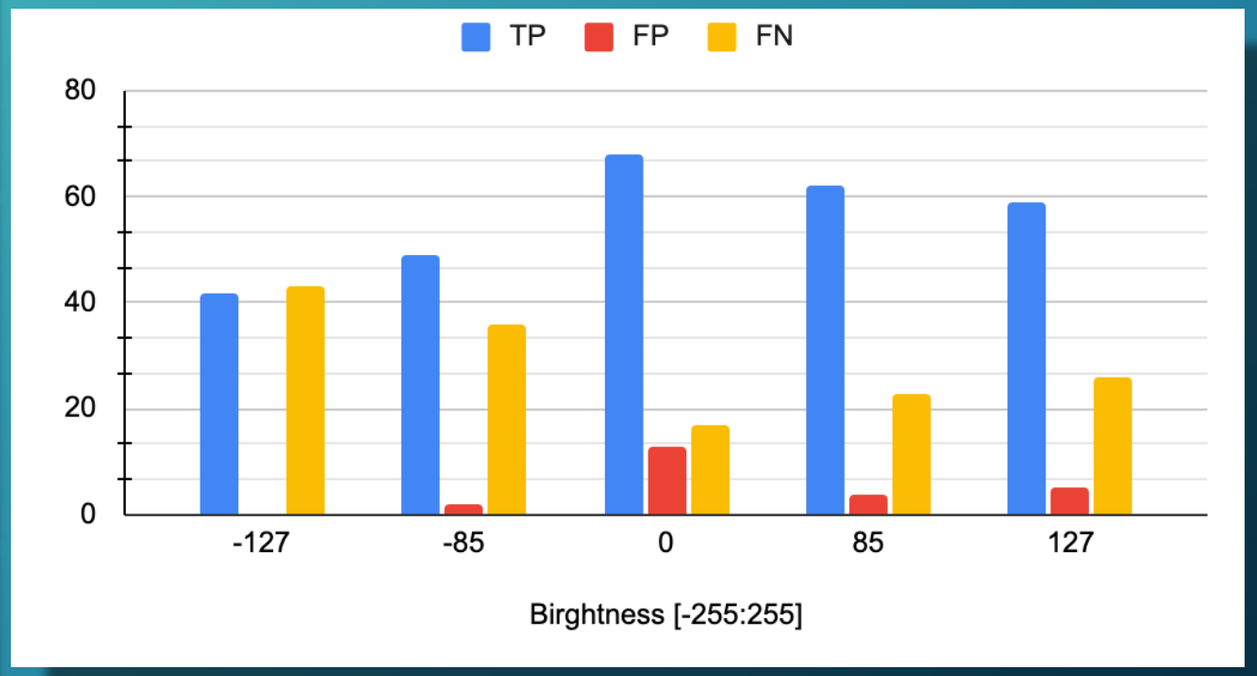
5.4. Data augmentation



(41)

MS COCO Metrics / Contrast [-255:255]	-127	-85	0	85	127
Average Precision (AP) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.378	0.454	0.499	0.476	0.253
Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets=100]	0.511	0.559	0.581	0.594	0.486

(42)



(43)

5. Experiments and results

5.5. Resources & projections

Final AP and training time per method

(44)

Architecture	Final mAP	Training time (hours)
Type A - 3K training steps	0.521	5.25
Type A - 10K training steps	0.538	17.51
Type A - 30K training steps	0.521	52.53
Type B - 3K training steps	0.525	5.25
Type B - 10K training steps	0.496	17.51
Type B - 30K training steps	0.484	52.53

Linear projections to reach 0.90 AP per method

(45)

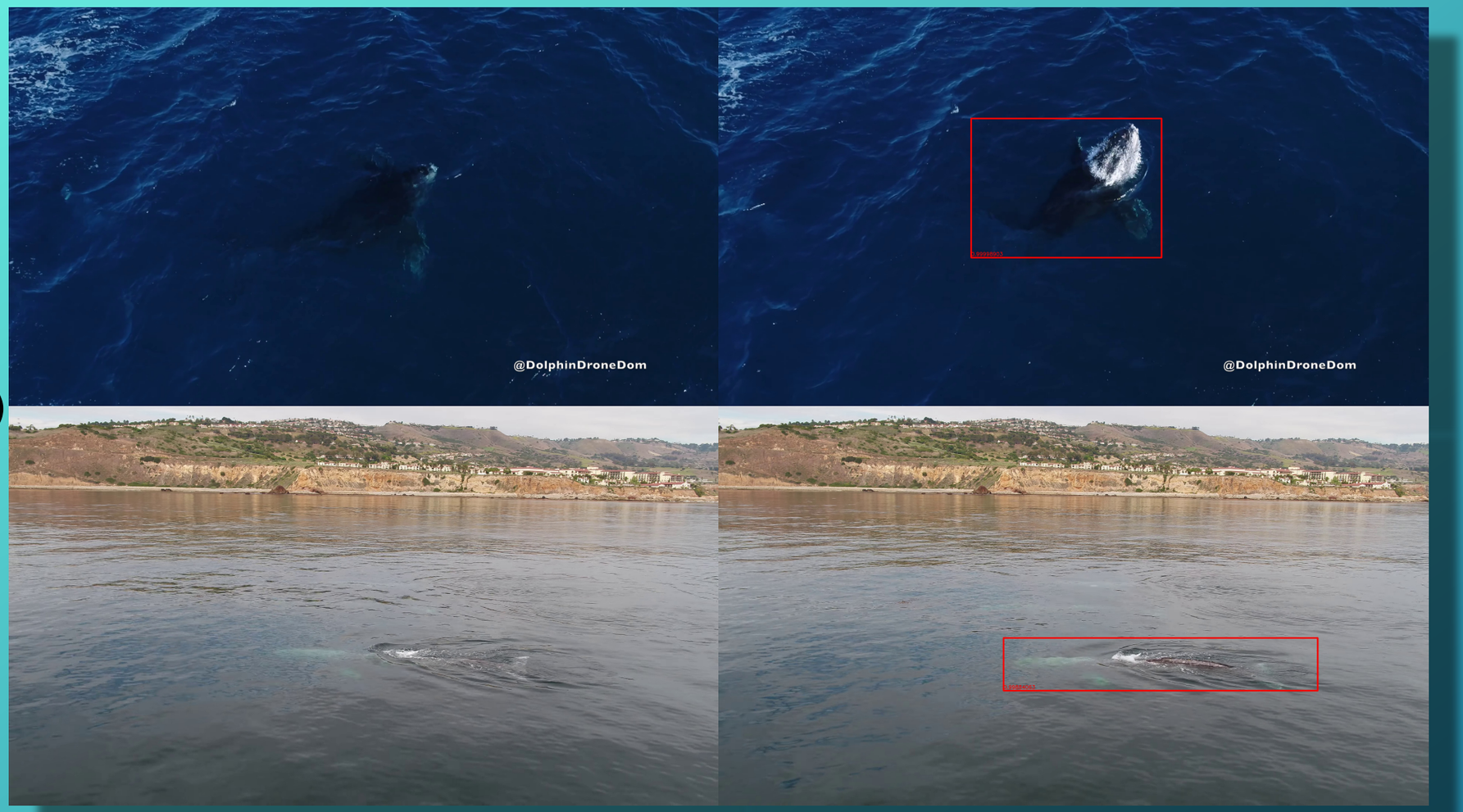
Method	Iterations	Expected training time (hours)	Expected number of training images
Type A - 3K training steps	69.39	36.45	6,939
Type A - 10K training steps	49.27	86.22	4,927
Type A - 30K training steps	65.86	345.76	6,586
Type B - 3K training steps	55.39	29.09	5,539
Type B - 10K training steps	104.53	182.92	10,453
Type B - 30K training steps	164.44	863.31	16,444

6. Conclusions

- Increased training dataset by 1438 ground-truth boxes (21% more) from 1000 drone images (23% more).
- The non-cumulative training (Type A) - 10K training steps got better AP and AR. But it’s not the most efficient.

6. Conclusions

(46)



- The network has its limits and it’s still susceptible to detecting objects as whales that are not whales. Like dolphins, boats, surfers and algae.

- Some experiments can be repeated and improved. Such as the data augmentation experiment that should have been done using a better contrast change method.

Deep Learning, I think this is the beginning of a beautiful friendship.