Département Sciences du Numérique

Use of Deep learning on very high-resolution satellite imagery for detection and estimating total population size of southern elephant seal (Mirounga leonina)

PROJET DE FIN D'ETUDES

Zixing QIU

A thesis presented for the degree of
Diplôme d'Ingénieur en HPC et Bigdata
and
Diplôme Master 2 en Performance in Software, Media Scientific Computing

INP-ENSEEIHT
Parcours HPC et Bigdata Toulouse, France
September 2021

Contents

Use of Deep learning on very high-resolution satellite imagery for detection and estimating total population size of southern elephant seal (Mirounga leonina)

Zixing Qiu[1], Nicolas Sidère[2], Christophe Guinet[3], and Joris Laborie[3]

[1]INP-ENSEEIHT
[2]Laboratoire L3i - Université de la Rochelle
[3]CEBC-CNRS,UMR 7372 – Université de la Rochelle

September 2021

Abstract

The elephant seal plays an important role in the southern ocean. In recent years, the use of VHR (very high resolution) satellite imagery has made it possible to make population size predictions for elephant seals on inaccessible seal living areas. Previously, female seals on satellite images needed to be counted manually, a complex and cumbersome process that was often costly in terms of time and labour, regarding the difficulty of identifying them on satellite images of different resolutions varied. Therefore, this paper attempts to use deep learning techniques to simplify it. We segment a total of 28 satellite images, do the labelling and finally generate a database consisting of 105 images with resolutions of 0.3m, 0.4m and 0.5m respectively. We use two state-of-art models, YOLOv5 and Faster R-CNN, and use cross validation in the training and test to what extent the resolution affects the accuracy. In the end, we concluded that on both high-resolution and VHR satellite images, YOLOv5 predicted the number of female seals very close to the manual count, equal to about 87% of the latter, while Faster R-CNN was only 57%. The rate falls into 67% and 40% for images at 0.5m resolution. Taking into account the errors and noise of manual labelling, we still consider the model predictions to be valid. In the end, a linear model is proposed to reduce the gap between manual count and the model output. We provide a Google Colab sample, and also a script available on Github

1 Introduction

Southern elephant seals (Mirounga leonine) play a pivotal role in the southern ocean as widespread top predators and provide an opportunity to quantify how animals and marine ecosystems respond to environmental variation.[1] Elephant seals have been described as sentinels of coastal ecosystems (Aguirre and Tabor 2004). Their population size data is critically important for ecological/trophic modelling studies and to better estimate the overall amount of food consumed by seabirds and marine mammals and their contribution to ocean carbon and nutrients fluxes .[2] Across their range, southern elephant seal populations occur in four genetically distinct stocks (Slade et al. 1998). The South Georgia, Peninsula Valdes, and Kerguelen stocks are all either stable or increasing slightly.[3]

For Crozet and Kerguelen Archipelagos population trends are estimated from only a small part of the elephant seal breeding beaches during the breeding seasons manually. [4] However, there are still many areas on Kerguelen Island where elephant seal habitat exists, but this part is not been sensored. The proportion compares with entire population on Kerguelen Island and Crozet Archipelagos is still unknown.

Because of the extreme weather on the West coast of Kerguelen Island during the breeding period of the elephant seal, it is impossible for helicopters or boats to survey these areas. So alternative methods, such as satellite remote, ground count method must be implemented and tested.

Stock and location	Pre-1970s	1990s	2000s	Size – 2010s	References	Current status
Kerguelen						
Isles Kerguelen	157,500	143,500	153,237	153,237	Authier et al. (2011)	Stable
Heard Island	80,500	40,355	61,933	61,933	Slip and Burton (1999)	Stable?
Marion Island	3850	2009	2009	1582 (452)	McMahon et al. (2009)	Increasing
Prince Edward Island	Unknown	782		410	Bester and Hofmeyr (2005)	Decreasing
Isles Crozet and Possession I	10,500	2023	1995	1995	Guinet et al. (1999)	Increasing
Stock total	252,350	188,669	219,174	219,157		Stable

Figure 1: elephant seal population estimation in Kerguelen using in situ CTD data (Hindell et al. 2016)

Previous work using a total of 28 satellite images taken during the breeding season to count females elephant seals on the inaccessible area (80 % and n = 28 600 ± 5 300 females for Kerguelen and n = 1 370 ± 160 females for Crozet) and complement the traditional ground counts (20 % and n = 63 280 ± XX females for Kerguelen and n = 1 750 ± XX for Crozet). Counting on satellite images was carried out with GIMP 2.8 software, a powerful photographic retouching program, which allows counts to be made manually (by click and by hand). Only breeding females were counted and are easily differentiated from pups by their color and size. [1] For each harem, three successive counts were performed on the same image and by the same observer at least 6 months apart to estimate a census error and avoid biases introduced by prior knowledge. As the sample size increases, manual identification becomes increasingly costly, with

a single high-resolution satellite image sometimes containing almost a thousand sea elephants. The manual identification usually requires manual comparison with satellite images. Consequently those counts are extremely tedious and time consuming and there is an urgent need for automated counts to be able to generalize such approach. Therefore, automated tools are required to be able to automatically identify not only the location of the elephant seals, but also the sex and even the age (whether it is mature or not) of them once located.

Convolutional neural networks (CNNs) achieve impressive performance on classification tasks at real-time speeds.[9] Yet top object detection systems like R-CNN take seconds to process individual images and hallucinate objects in background noise. We believe these shortcomings result from how these systems approach object detection.[7]

Although CNN achieves good accuracy in many tasks, for example, YOLO's mAP_0.5 in 5000 COCO val2017 images is 72.0, the application in satellite images is still a huge difference compared to COCO dataset.[7] Satellite images usually have completely different characteristics from COCO, for example, the elephant seal pup is only a few pixels in size on a 0.5m level satellite image, while the measured object in COCO occupies a much larger image area. And the noise of satellite images and the influence of received meteorological factors often make the accuracy of the neural network drop dramatically.

Another issue worth investigating is that the resolution of satellite images varies with potential consequences on our ability to identify seals and to discriminate different size categories. Three resolutions were used in our experiments: panchromatic 0,50 m, 0,40 m and 0,30 m resolution respectively. There are already some works relying on satellite images to census seabirds and seals. CNN experiments have demonstrated the need of large datasets to train models and test the trained models on smaller datasets with a good accuracy. Therefore, in the field of satellite imagery, whether training with different resolutions has an impact on accuracy, to what extent high-resolution images can help improve results, and whether there is a significant difference between high- and low-resolution images in the test set still needs to be verified experimentally.

In the following, we will first describe the sources of satellite image data, how to do pre-processing of them. Secondly we will introduce the methodology, from manual count to the algorithmic implementation of image processing to census elephant seals algorithmic. Finally we compare manual counts to the automated image processing ones and draw conclusions and reason analysis associated with an adjusted predict model.

2 Related work

Hindell et al. (2016) using an at-sea habitat modelling approach according to the carrying capacity of the environment relying on primary production build up some total elephant seal's population size prediction obtained for female and male southern elephant seals from the Antarctic Peninsula (females = 159,832, males = 17,321), South Atlantic (females = 172,988, males = 45,699), Southern Indian (females = 92,049, males = 46,633), and Southern Pacific Ocean regions (females = 33,614, males = 33,614) regions. [3]

Laborie et al.(in prep)[1] provide a complete approach to prediction, from the ground count of satellite images to the building of predictive models. They estimate for the first time ever the total size of elephant seal of populations from Kerguelen and Crozet archipelagos and demonstrate that the use of high resolution satellite images is a convenient, inexpensive and very reliable method to census elephant seal during the breeding season (i.e. October, Austral spring).

The PASCAL Visual Object Classes(VOC) provides standardized image data sets for object class recognition[11] The main problems that the two PASCALs seek to solve are classification problems and detection. It also provides "Taster" challenges aims to find the approach on segmentation and pose. Other datasets such as CIFAR-10[13], MNIST[14] are widely used as training and test sets in the field of machine vision. The images from these datasets are used as global predictions, meaning that each image corresponds to only one classification problem.

ImageNet is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images.The most highly-used subset of ImageNet is the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012-2017 image classification and localization dataset. This dataset spans 1000 object classes and contains 1,281,167 training images, 50,000 validation images and 100,000 test images.[15]. Not until the year of 2012, deep learning method start to boost the ILSVRC classification dataset accuracy. Before 2012 traditional machine learning algorithms had an error rate of over 25%, in 2012 Alexnet was born and achieved the best result in the ILSVRC challenge that year. Since then neural networks have been widely discussed and developped. (more details reference Representation Learning[12])

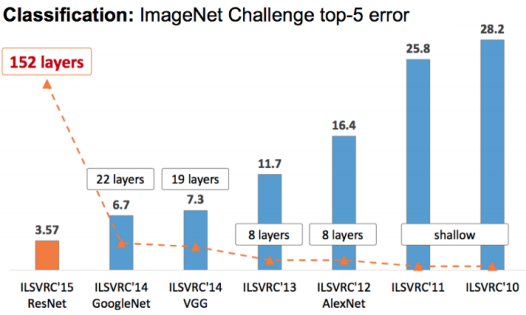

Figure 2: Top models errors between 2010-2015 of ILSVRC classification challenge (ILSVRC 2015)

3 Dataset

The images used in this paper are based on previous research by Joris Laborie, Matthieu Authier, Adrien Chaigne, Karine Delord, Henry Weimerskirch and Christophe Guinet aiming at estimating the total population size onof the Crozet and Kerguelen Archipelagos using VHR satellite imagery. A first validation work on satellite image of a small part of the Courbet Peninsula, Kerguelen Island is required , with the Pleiade PHR-1B satellite sensor by AIRBUS Defence & Space. Image processing was performed by the French National Center for Space Research (CNES) to produce a pan-sharpened orthorectified image. The study was carried out in the Indian sector of the Southern ocean, on Crozet archipelago $(-46.4260°, 51.7750°)$, regrouping more than 300 islands and islets and for a total area, composed of 5 small main islands and Kerguelen archipelago $(-49.3649°, 69.3843°)$, totaling 350 km^2. More than 7200 km^2.[1] Training set is composed from 10 images captured(see table 2)

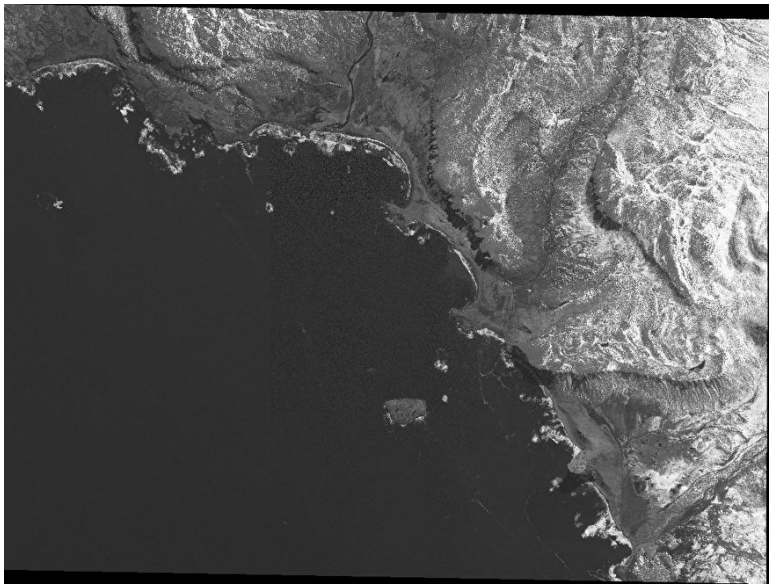

Figure 3: DigitalGlobe, 11OCT14051509P2AS-056844113050_01_P001BROWSE, one of image of Kerguelen beach area

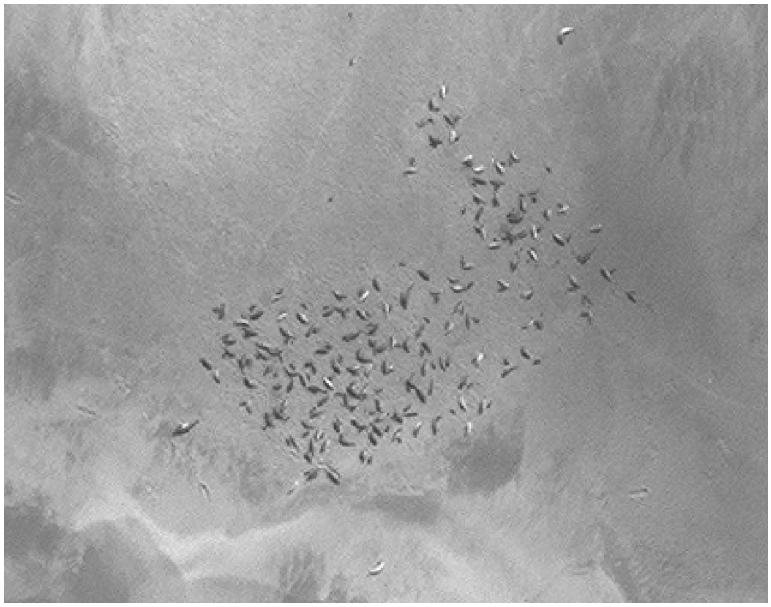

Figure 4: A closer look of elephant seal cluster on the Beach Gros Ventre

Table 1: train dataset satellite images			
State	**id**	**Resolution (m)**	**dimension**
DigitalGlobe	16OCT06053729-P2AS-056844113010_01	0.5	X: 11759 Y: 8929
DigitalGlobe	16OCT06053729-P2AS-056844113020_01	0.4	X: 8304 Y: 42255
DigitalGlobe	16OCT06053729-P2AS-056844113030_01	0.4	X: 7678 Y: 36760
DigitalGlobe	16OCT06053729-P2AS-056844113040_01	0.4	X: 11626 Y: 9291
DigitalGlobe	16OCT06053729-P2AS-056844113050_01	0.4	X: 12220 Y: 26215
Pleiades	$\text{IMG}_P HR1B\_MS\_201510280510174\_ORT\_1590203101-002$	0.3	X: 7414 Y: 7201
Pleiades	$\text{IMG}_P HR1B\_MS\_201510280510174\_ORT\_1590203101-001$	0.3	X: 29653 Y: 28801
RNN TAF	16OCT12052614-P2AS-056530250010_01	0.4	X: 12504 Y: 13275
RNN TAF	16OCT12052614-P2AS-056530250040_01	0.4	X: 17626 Y: 10454
RNN TAF	16OCT12052614-P2AS-056530250050_01	0.3	X: 12220 Y: 26215

Table 2: Test dataset			
Zone	**id Plage**	**Nom Plage**	**Taille des plages**
Baie du Noroît	82	Pointe Richard 1	800
Baie du Noroît	223	Pointe Richard 2	30
Baie du Noroît	224	Pointe Richard 3	815
Baie du Noroît	83	Pointe Berger 1	215
Baie du Noroît	222	Pointe Berger 2	70
Baie du Noroît	84	Plage Noire	1350
Anse du Gros Ventre	69	Plage Jaune	370
Anse du Gros Ventre	70	Anse du Gros Ventre	2100
Baie Bretonne 1	80	Plage des Lions Marins	2400
Baie Bretonne 1	81	Baie du Young Williams 1	130
Baie Bretonne 1	225	Baie du Young Williams 2	70
Baie Bretonne 1	226	Baie du Young Williams 3	50
Baie Bretonne 2	197	Petite Anse du Melissas	60
Baie Bretonne 2	77	Plages des Portes de l'Enfer 1	270
Baie Bretonne 2	78	Plages des Portes de l'Enfer 2	800
Baie Bretonne 2	79	Vallée de l'Octant	800
Baie Bretonne 3	76	Vallée du Sextant 2	600
Baie Bretonne 3	75	Vallée du Sextant 1	260
Baie Bretonne 3	204	Vallée du Télluromètre 2-8	1110(adjusted)
Baie Bretonne 3	73	Vallée du Télluromètre 1	350
Baie Bretonne 3	72	Plages du Glacier J. Brunhes	630

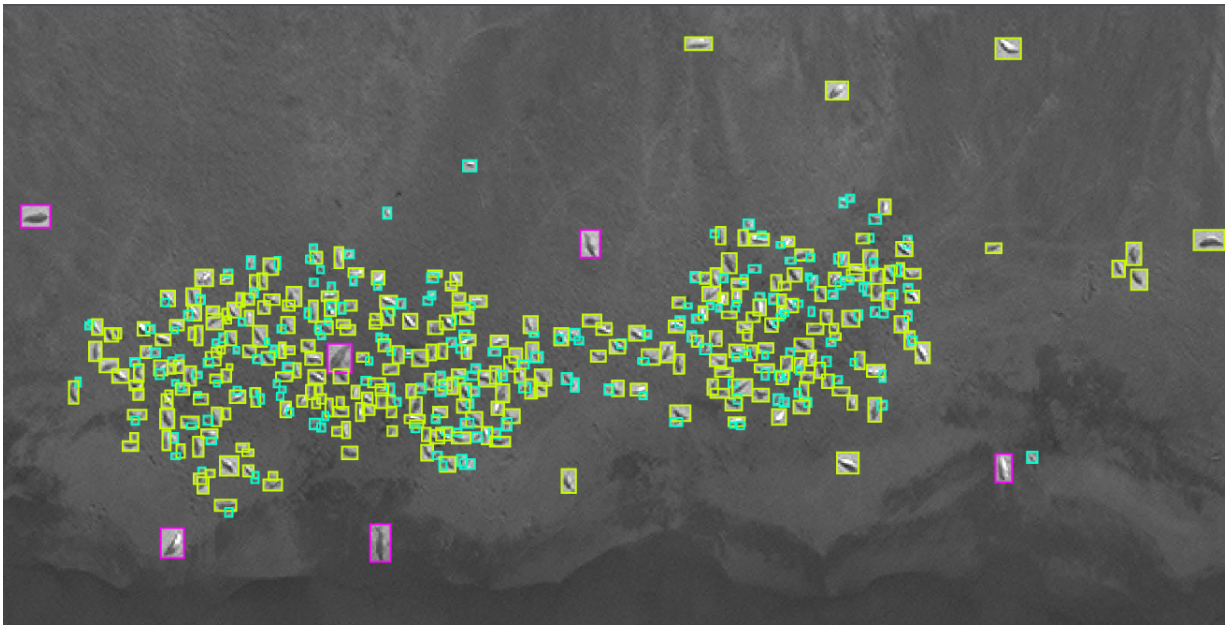

Figure 5: All individuals are identified by a bbox (boundering box), which includes their location and classification; the sky-blue bbox in the diagram represents a pup, the green one a female and the purple one a male

First we imported all the satellite images into QGIS[6] and found the part of QGIS with the elephant seal (most of the clusters were clustered on the beach, but a few on the rocky ground) and saved a screenshot of each image at 300dpi. Figure 3 and Figure 4 show an example of the original image and screenshot. To avoid confusing the elephant seal with the rocks on the beach, we compared our images by using satellite images from Google earth and Bing. To cater for different models, we used two formats, COCO[24] and YOLOv5[7] in subsequent experiments.

The female elephant seal differs significantly in size from the male, with adult males reaching over four meters in size and appearing as a light grey flattened oval on satellite images, compared to the females, which are relatively small, averaging 2.7m, and usually appear light grey, sometimes with sand applied to their bodies for protection from the sun. Similarly, the young of the elephant seal are born with dark fur and are often found in the company of females. They gather together after weaning at the periphery of the harems, they molt their black lugano fur and begin to learn to swim. The criteria for determining the seal on the image are therefore: (1) whether it is confused with a rock on Google earth or Bing Map (2) its length measured using the QGIS measurement tool (3) its position in the cluster (4) its colour. In our experiments, the threshold for pup and female body size is 1.6m, while the threshold for female and male is 3.5m.

A total of 105 images are included in the training set, including three null-labelled backgrounds in order to allow the model to distinguish between the pups and the background in particular, as the background noise is usually very similar to the pup. In total, there were approximately 4506 pup labels, 7758 female labels and 116 males appear in our dataset. We used the labelling tool provided by Roboflow to label the images, like Figure 5, all individuals appearing on the image for both train & val set and test set, were identified with a boundering box, the pup in sky blue, the female and the male in green and purple respectively. On output all images are stretched to 2048*1600 and auto-orientation is used and no augmentation is used.

4 Methodology

Overall, the prediction of elephant seal populations was divided into two approaches, using in-situ to obtain a statistical prediction model or take ground count to learn the much precise population. However, these two approaches are not necessarily either/or. The in-situ data usually only covers a portion of the island and there are many meteorological factors that can impinge on the reliability of the in-situ, so a machine learning algorithm must be used for prediction. We picked YOLO network and faster R-CNN due to their high accuracy on ILSVRC and COCO17, we also consider the stable training process is an important criteria.

4.1 Statistical analyses

In Laboris et al. (in prep)[1], article proposes a GAM model take female number counted from the satellite imagery, beach size etc into parameters, Pearson's correlation coefficient to reduce the predictants and Shapiro test to calculate residu, they use AIC as the model evaluation criteria and p-value is 0.05 as threshold and in the end cross validation by Loo. The selected model is used to predict the population on the unobserved beaches during the breeding season. All the work is realised by R and relative packages.

The model has Zone + Date + Surface_Plage + Expo_Plage + Expo_Mer = Dist_Mer + Dist_Bathy_50 as variables, this model achieves 901.1 as AIC, Loo = 507.4 and $R^2 = 0.743$ The predicted female seal population in the Plage Noire and Plage Jaune are 1243, 406. Petite Anse du Melissas, Plages des Portes de l'Enfer 1, Plages des Portes de l'Enfer 2, Vallée de l'Octant, Vallée du Sextant 2, Vallée du Sextant 1 are also preditecd by the same model with 278, 351, 681, 681, 535, 355 as result.

4.2 R-CNN family

To answer the question of how far a deep ground neural network from ImageNet can be applied to the PASCAL VOC challenge to solve classification problems, R-CNN was born.They found that CNNs are commonly used as sliding window detectors for

restricted classes of objects, such as faces and nearest pedestrians. However, the high computational cost of CNNs with large densely connected (non-convolutional) layers and the choice of detection box for objects of different shapes and characteristics make this set of solutions much less attractive.[10] The model design of R-CNN is divided into two parts, first is the region proposals, the article adopts methods such as CPMC[17], Object proposals[18], etc. to extract about two thousand possible regions in the image.After extracting the region, the article extracted 4096-dimensional feature vector from each region proposal using our own implementation of the CNN of Krizhevsky et al.[16] Then convert all regions into a 224*224 pixel CNN input, the scores of different classes are obtained by SVM after CNN features. The first filter of the neural network is used to capture oriented edges and opponent colors. In the VOC2007 experiments with animal pictures (cats and dogs there), the activitions of the feature map of the middle layer ($pool_5$) neurons show that the feature contains a rich variety of features and shapes of animal faces. In version 5 the article shows a head-to-head comparison of R-CNN and OverFeat[19], a sliding windows CNN, which shows that R-CNN significantly outperforms OverFeat, with a mAP of 31.4% versus 24.3% on ILSVRC2013.

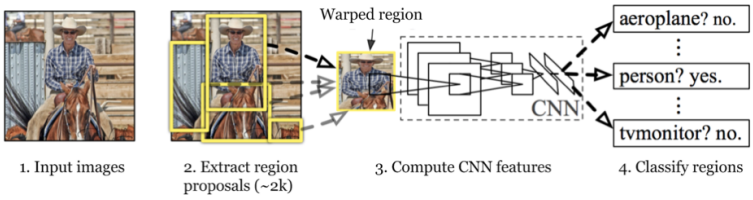

Figure 6: An overview of R-CNN object detection system (Girshick, R. et al. 2014)

4.2.1 Fast R-CNN

Although R-CNN solves the problem of how to select the region where the object is located, as Ross Girshick says[20], R-CNN has problems such as: (1)Training is a multi-stage pipeline it has to first fine tunes a ConvNet on images step to a Log loss, and use SVM as a detector, then it also has to learn bounding box. (2) Training is expensive because the SVM and bounding-box regressor training's features have to be stored in the disk and a very deep network has 2.5GPU-days to process. (3) Object detection is slow, it cost 47s/picture with VGG16. He therefore proposes a single-stage, fast R-CNN using a multi-task loss, and no need to store features on disk at all.

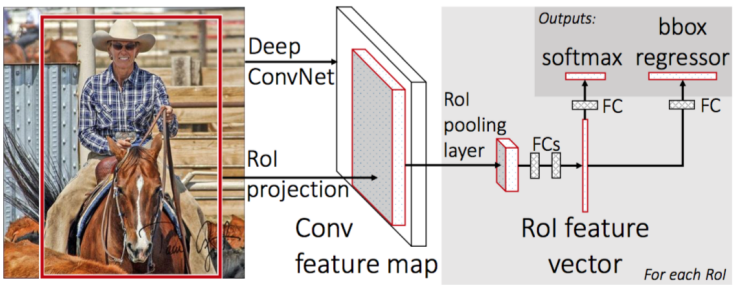

Figure 7: Fast R-CNN Basic structure (Ren, S., He, K., Girshick, R., Sun, J. (2015).)

The major difference between fast R-CNN and R-CNN is that the former takes the whole image as the input of the conv and, at the same time, extracts the region of interest (RoI) from the feature map. The whole feature map is divided into two outputs after fully connected layers.one that produces softmax probability estimates over K object classes plus a catch-all. The other one outputs four real-valued numbers for each of the K object classes as bounding-box positions for one of the K classes. The RoI pooling layer uses max pooling to project any RoI as a region of H*W on the feature mapping. The authors likewise used three database pre train models with ImageNet, and applications of the R-CNN family from larger to smaller databases have been shown to be feasible in R-CNN. Since the output consists of two components: a discrete probability distribution to indicate the correctness of the classification, and the output of the sibling layer to indicate the superiority of the bounding-box, the authors propose a global loss function to combine the two into one.

Multi-task loss: $L(p,u,t^u,v) = L_{cls}(p,u) + \lambda[u \geq 1]L_{loc}(t^u,v)$ in which $L_{cls}(p,u) = -logp_u$, assume $t^u = (t^u_x, t^u_y, t^u_w, t^u, h)$ correspond a bounding-box's 4 angles, the bounding-box Loss function can be used as : $L_{loc}(t^u,v) = \sum_{i\in x,y,w,h} smooth_{L_1}(t^u_i - v_u)$ in which $smooth_{L_1}(x) = 0.5x^2$ when $|x| < 1$ and equals to $|x| - 0.5$ otherwise.

To improve the efficiency of detection, as RoI is usually about 2000, fc layers are calculated at forward pass and the weight matrix W is decomposed by SVD $W \approx U\Sigma_t V^T$. The first of these layers uses the weight matrix $\Sigma_t V^T$(and no biases) and the second uses U (with the original biases associated with W).

4.2.2 Faster R-CNN

In recent years faster R-CNNs have been proposed to solve the problem that fast R-CNNs still take a huge amount of time in detection network.[22] The authors introduce Region Proposal Networks (RPNs). fast R-CNNs integrate R-CNNs into a single stage, but they still require two down-streams, and the layers between the two down-streams are not shared. RPNs share the The RPNs share convolutional layers to compress the time to compute the proposals for each image to 10ms. By observing that it is only necessary to add some additional convolutional layers, the feature maps of region-based detectors used by the original Fast R-CNN can be considered as RPN simultaneously regress region bounds and objectness scores at each location on a regular grid. And typically these layers are fully convolutional network[25]

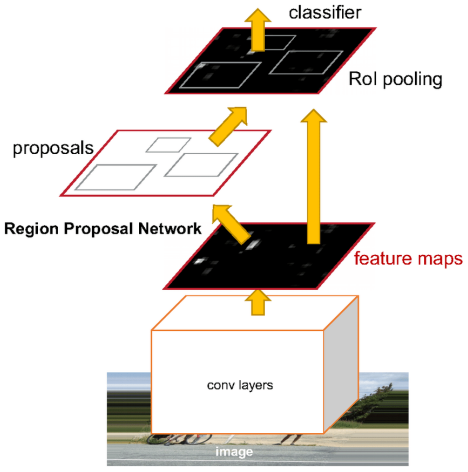

Figure 8: Faster R-CNN use a simple RPN module Ren, S., He, K., Girshick, R., Sun, J. (2015).

A Region Proposal Network (RPN) is investigated by

Zeiler and Fergus model[30] and the Simonyan and Zisserman model(VGG-16)[26]. The big idea is these two model have 5 and 13 layers can be shared with R-CNN detector network. The generation of region proposals take on the last convolutional feature mapping, an $n \times n$ window sliding on it map to lower dimension. This feature is fed into two sibling fullyconnected layers—a box-regression layer (reg) and a box-classification layer (cls).

In each sliding window, note that k maximum possible proposals of it, assume each k classes reference boxes called $anchors$. By default there are 3 scales and 3 aspect ratios. Therefor 9 anchors per windows. It has also some good properties. Like t it is translation invariant, both in terms of the anchors and the functions that compute proposals relative to the anchors. Thus the model size has been reduced because the output layer, in the case of anchors = 9, is a $(4+2) \times 9$-dim layer, far less than those in MultiBox's[31] FC output layer = $(4+1) \times 800$-dim.

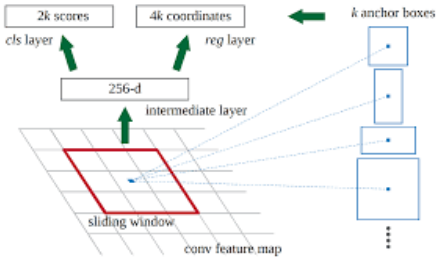

Figure 9: Each sliding window is mapped to a lower-dimensional vector and fed into two fc layers for classification and regression (Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016))

Anchors are choosed by pyramid of anchors method by using single scale and single size of filters with respecting the feature map and images. The big idea of it is to make sharing feature in the same scale possible.

The loss function is defined as:

$$L(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*)$$

The notation definition is consistent with Fast R-CNN, note thatassign a binary class label (of being an object or not) to each anchor. The authors assign a positive label to two kinds of anchors: (i) the anchor/anchors with the highest Intersection-overUnion (IoU) overlap with a ground-truth box, or (ii) an anchor that has an IoU overlap higher than 0.7 with 5 any ground-truth box. Note that a single ground-truth box may assign positive labels to multiple anchors. The whole loss like what's done in the R-CNN[10], is a combination of regression loss and classification loss. The RPN is trained end-to-end by back-propagation, for each mini-batch raise 256 anchors which let positive samples and negative one to be 1:1.

4.3 YOLO

4.3.1 YOLOv1

YOLO(You only look once)is one that, unlike the R-CNN family, only needs to run the convolutional network once, without the need to use SVM for classification, and without the need for a linear model to correct the bounding-box. A single convolutional network simultaneously predicts multiple bounding boxes and class probabilities for those boxes.[7] Figure 10 is basic YOLO architecture. It does not require an RPN, which gives it a number of advantages, as it only needs to compute a neural network for the task of detection, and its speed can reach real-time detection. Secondly YOLO is able to get a better grasp of the whole picture's essence than R-CNN, but the disadvantage is that the accuracy is poor, especially on small objects.

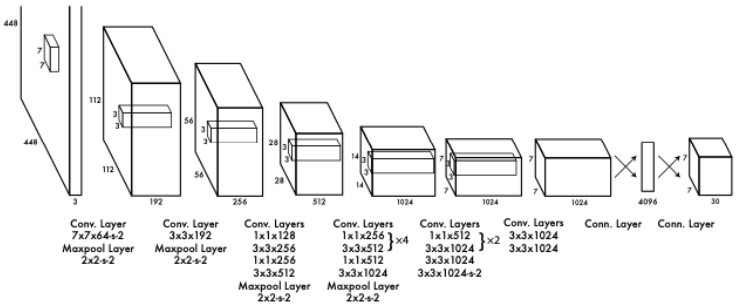

Figure 10: YOLO detection architecture, the last two layers take charge of bbox proposition and classification

It first devides images into $S * S$ grid, the grid cell which the center of the object lays in take charge of the detecting that object, and the grid cells proposed B bbox and respected confidence. Each box is given a class-specific confidence score difined as

$$Pr(Class_i|Object) * Pr(Object) * IOU_{Pred}^{truth} = Pr(Class_i) * IOU_{Pred}^{truth}$$

Where$Pr(Object) * IOU_{Pred}^{truth}$ means how confident the model consider in the box contains object, while $Pr(Class_i|Object)$ is the confidential on the predicts class probabilities. Every grid cell only have one perticular class and choose the highest IOU as class.

Instead of using sliding-windows or FCN to propose a region like R-CNN, YOLO compresses the image to 448*448 and slices it into 7*7 grids. The structure of the YOLO network is shown in figure 10, where a 448*448 compressed image is stride The first 1-20 channels are responsible for predicting the class probability, and the last 4 fc layers are responsible for predicting the box coordinates. In general, the architecture follows the idea from GoogLeNet[32] and Lin et al[33] except use a logistic activation function to constraint the output boundary between 0 and 1 echo with the image width and height. Other layers use

$$\phi(x) = \begin{cases} 1.1x, & \text{if } x > 0 \\ .1x, & \text{otherwise} \end{cases}$$

They optimize for sum-squared error in the output of our model because it's easier to optimize.

$$\sum_{i=0}^{48} (\lambda \mathbb{1}_i^{obj}((x_i - \hat{x_i})^2 + (y_i - \hat{y_i}^2) + (\sqrt{w_i} - \sqrt{\hat{w_i}})^2 + (\sqrt{h_i} - \sqrt{\hat{h_i}})^2) +$$

$$\sum_{c \in classes} (p_i(c) - \hat{p_i}(c))^2)$$

The role of λ here is to balance the weight of localisation error and classification error. The authors consider the importance of the localisation error to be more intuitive and important than the classification error. The article therefore defaults to a λ of 4.Where $\mathbb{1}$ obj $i$encodes whether any object appears in cell.

4.3.2 YOLOv5

YOLO has made major changes to the neural network and loss function in the latest version of the update, unified detection is

same as those in v1, instead of using $\phi(x)$, leaky recitified linear activation is used defined by:

$$\phi(x) = \begin{cases} x, & \text{if } x > 0 \\ .1x, & \text{otherwise} \end{cases}$$

other layers are the same. Two parameter are introduced: $\lambda_{coord} = 5$ and $\lambda_{noobj} = .5$ for solving gradient vanish through the training caused by too many uncontained object box which led to a very small gradient flow. Thus greater the bbox coordinate prediction's loss and make it domains the total loss.

The latest loss function defined as:

$$\begin{aligned} Loss = & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} [(x_i - \hat{x_i})^2 + (y_i - \hat{y_i}^2)] \\ & + \lambda_{coord} [(\sqrt{w_i} - \sqrt{\hat{w_i}})^2 + (\sqrt{h_i} - \sqrt{\hat{h_i}})^2] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} (C_i - \hat{C_j})^2 \\ & + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{noobj} (C_i - \hat{C_j})^2 \\ & + \sum_{i=0}^{S^2} \mathbb{1}_{i}^{obj} \sum_{c \in classes} (p_i(c) - \hat{p_i}(c))^2) \end{aligned} \tag{1}$$

Compare with the previous one, YOLOv5's loss function has nothing different but λ_{coord} and λ_{noobj} are parsed and there occurs a new item in the equation: $(C_i - \hat{C_j})^2$ which represents in the v5 gird cell prediction, it also predicts C conditional class probabilities, so these items can be considered as cross-entropy-loss like loss in the YOLO model.

4.4 Linear model

Although with the help of stats-of-art deep learning models, our prediction is much more closer to manual count data, there is a amount gap between these two. How to correct the prediction in order to approach and reflect the real count, we must propose a method to eliminate the gap. We assume the relation of the manual account and the model output is linear take in count of imagery resolution. So a multi variable linear model seems adapt this kind of task best.

The big idea of linear model in regression task is assume the variables have linear relation.[8] In statistical word, a linear model is presented as: $Y_{ij} = \mu + \alpha_i + E_{ij}$ with μ a constant parameter, α_j the sample set and E_i the respected residue. Thus, $Y_{ij} \sim N(\mu_i, \sigma^2)$, $[Y_{ij}]$ independent and $\mu_i = \mu + \alpha_i$. We use both qualitative and quantitative parameter which led to a general model like: $Y_{ij} = a_i + b_i x_{ij} E_{ij}$ with a decomposition effect of a_i and b_i written as $a_i = \mu + \alpha_i$,$b_i = +\gamma_i$. Then; $Y_{ij} = \mu + \alpha_i + x_{ij} + \gamma_i x_{ij} + E_{ij}$ which means the distribution of $Y_{ij}(\mu_{ij}, \sigma^2)$, note $\mu_{ij} = \mu + \alpha_i + _{ij} + \gamma_i x_{ij} + E_{ij}$.

Table 3: YOLOv5m layers and respected parameters

layer	from	n	params	module	arguments
0	-1	1	7040	models.common.Focus	[3, 64, 3]
1	-1	1	73984	models.common.Conv	[64, 128, 3, 2]
2	-1	1	156928	models.common.C3	[128, 128, 3]
3	-1	1	295424	models.common.Conv	[128, 256, 3, 2]
4	-1	1	1611264	models.common.C3	[256, 256, 9]
5	-1	1	1180672	models.common.Conv	[256, 512, 3, 2]
6	-1	1	6433792	models.common.C3	[512, 512, 9]
7	-1	1	4720640	models.common.Conv	[512, 1024, 3, 2]
8	-1	1	2624512	models.common.SPP	[1024, 1024, [5, 9, 13]]
9	-1	1	9971712	models.common.C3	[1024, 1024, 3, False]
10	-1	1	525312	models.common.Conv	[1024, 512, 1, 1]
11	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
12	[-1, 6]	1	0	models.common.Concat	[1]
13	-1	1	2757632	models.common.C3	[1024, 512, 3, False]
14	-1	1	131584	models.common.Conv	[512, 256, 1, 1]
15	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
16	[-1, 4]	1	0	models.common.Concat	[1]
17	-1	1	690688	models.common.C3	[512, 256, 3, False]
18	-1	1	590336	models.common.Conv	[256, 256, 3, 2]
19	[-1, 14]	1	0	models.common.Concat	[1]
20	-1	1	2495488	models.common.C3	[512, 512, 3, False]
21	-1	1	2360320	models.common.Conv	[512, 512, 3, 2]
22	[-1, 10]	1	0	models.common.Concat	[1]
23	-1	1	9971712	models.common.C3	[1024, 1024, 3, False]
24	[17, 20, 23]	1	43080	models.yolo.Detect	[3, [[10, 13, 16, 30, 33, 23], [30, 61, 62, 45, 59, 119], [116, 90, 156, 198, 373, 326]], [256, 512, 1024]]

5 Experiment

In this experiment we compare pretrained of YOLO and Faster R-CNN on the training set, for each we propose dataset pre-processing, we also choose parameters including learning rate, image size etc which has good performance on the training set with repected confusion matrix, mAP etc. In the following section, we will use the trained weight in this experiment to evaluate their performance on the test dataset.

5.1 YOLO

We used torch version 1.9.0+cu102, graphics card model Tesla P100-PCIE-16GB, and all training weights were saved in Google Drive. The pre-trained checkpoint we chose is Yolov5m whose structure is shown in table 3. We also tested Yolov5l and compared it to the former, the latter appeared to be overfitted during our training although it had a relatively high mAP on COCO 2017. If the training set is noisy then the final accuracy on the test set is reduced. Moreover, yolov5l requires a larger data set than Yolov5m, which does not give us an advantage for small data sets such as our seal data set.

YOLO does not support cross validation, so we wrote a script that allows us to randomly assign the 105 images to the specified dataset and to use each of the 105 images as a training set. After 300 epochs a new distribution will be allocated. Script also allows us to update the weights after each training session using a different configuration, and to freely choose the batch size etc. Note that a training cycle with 300 epochs in the script is equivalent to YOLOv5 not using the dynamically allocated 300 epochs. Not only does script support all the parameters of the original YOLO, but we have made it possible to assign data set proportions. The optimal training weight is saved at the end of each epoch and used as the initial weight for the next training.

All images were compressed to 640 pixel when used as inputs with a batch size of 16. In total, we trained three loops, each loop containing 300 epochs and the training set was redistributed in a ratio of 8:2 for each loop. SGD optimizer is been used to optimize. Other parameters are lr0=0.01, lrf=0.2, momentum=0.937, weight decay=0.0005. During the training we use wandb[27] to moniter the process. We also tested a larger batch size and image size but the results were not significantly better.

The overall mAP.5 for training reached 0.33, as the reproductive population of the seal population was largely determined by the number of females, although the number of males is also important but compared to females it's very small. This is very close to the accuracy achieved by YOLOv5l at COCO 2017 which is 0.55 if only the map of females is calculated. The main error comes from the pup classification which has only 0.09 accuracy. The male sample has been very scarce, so the map is from 0.38. This result is the same as the one given by YOLO, because the pups only take up an average of five pixels on the satellite images and they are usually confused with the background.

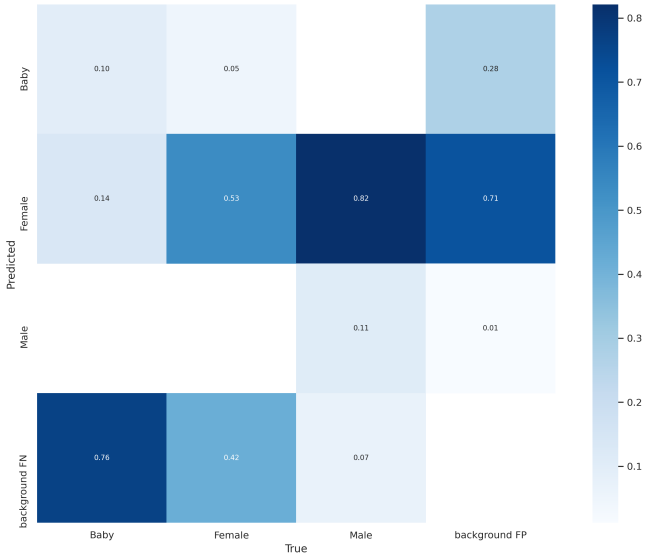

Figure 11: training confusion matrix (closer look in the appendix 10.3)

This conclusion is also supported by looking at the confusion matrix. For pup 76% of the objects were mistaken for background when a confusion occurs, while only 0.14 of the babies were mistaken for females. Thus the accuracy of pup has little impact on the overall number of females we detect. Although there are plenty of pups but we hardly find the significant errors proof that the introducing of pup could have a negative effect. The main errors that occur in the detection of females come from the background. 0.42 of the miss classification of females were not distinguishable from the background. At the experiment level,the presence of rocks could be present and mistaken with seals.

In 900 epochs of training, precision converged to 65% and recall converged to 0.35 at the last 300 epochs and with the epoch increase, YOLO learns the essence of the object distributions gets better, especially the global distribution. By comparing train dataset and model predicted images, we found that the model predicted individuals' clusters have a great similarity with original dataset, including the female seals' distribution. Aspected weather conditions can greatly affect satellite imagery. If the clouds cover the cluster then it is difficult to discern the exact outline from the image, a problem for both manual detection and detection using models. YOLO has a significantly defeat on this kind of noise(see Appendix). However, the training process is not sensitive to resolution, and image pre-processing such as brightness and gamma do not affect it, and different resolutions give similar results in training. (As shown in the figure 12)

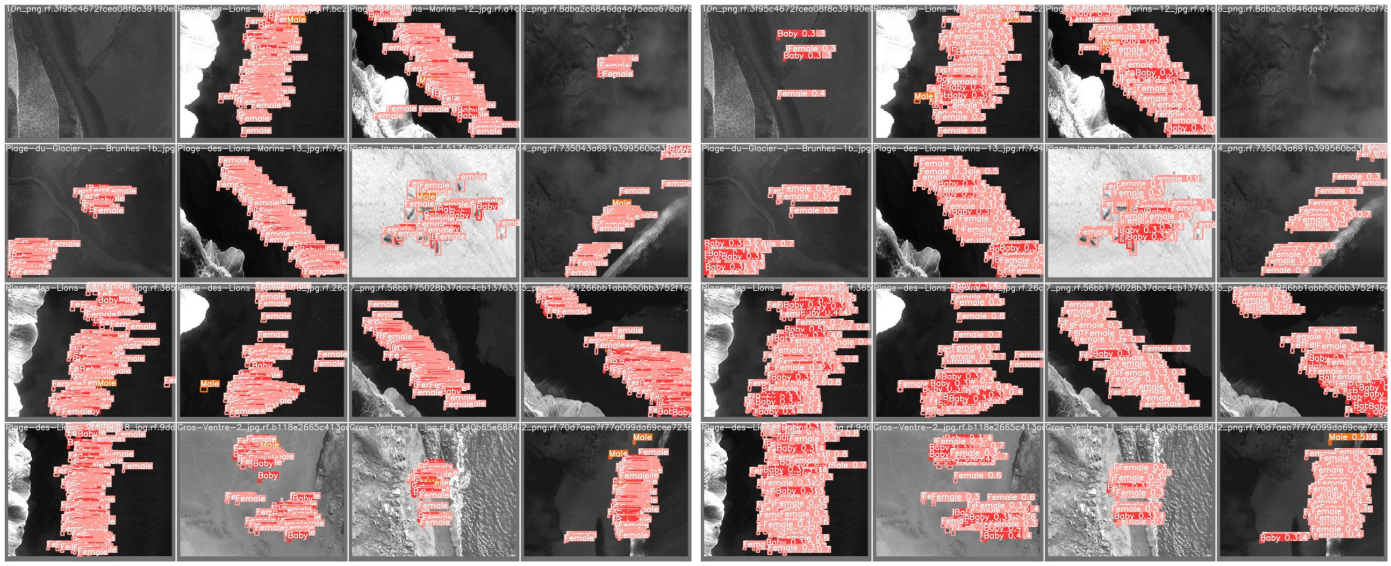

Figure 12: With the labelled image on the left and the predicted on the right, we can find a similar distribution between the two. But like the fourth one in the first row, YOLO was not able to detect any seal because of the cloud cover

5.2 Faster R-CNN

Thanks to the development of Detectron 2[23] by Facebook AI which makes it easy to train Faster R-CNNs, we trained Faster R-CNNs with the same test environment, and detectron can read COCO objects directly. The model uses faster_rcnn_X_101_32x8d_FPN_3x, although detectron also provides RPN Fast R-CNN and Retina, both of which diverged in the actual test. We pick up 0.001 as learning rate and trained 1800 iterations, over 1800 iterations we found overfitting and no accuracy improvement can be achieved. The batch size is as default 512. Tensorboard can monitor the process during training, cls_accuracy finally converge to 0.88 and false_negative achieve 0.3. Best lose is 0.9. Detectron used only 102 images due to it doesn't support null catagory. Detectron 2 supports Mask R-CNN[29], a latest approch of R-CNN labeled by mask instead of bbox and can not only do the detection and classification, but also able to segmentation. These kind of approaches fail in our task because of the size of the seal in the imagery is quite small. To label a mask with just several pixel is not practical and in practice, it can be more vague than the bbox.

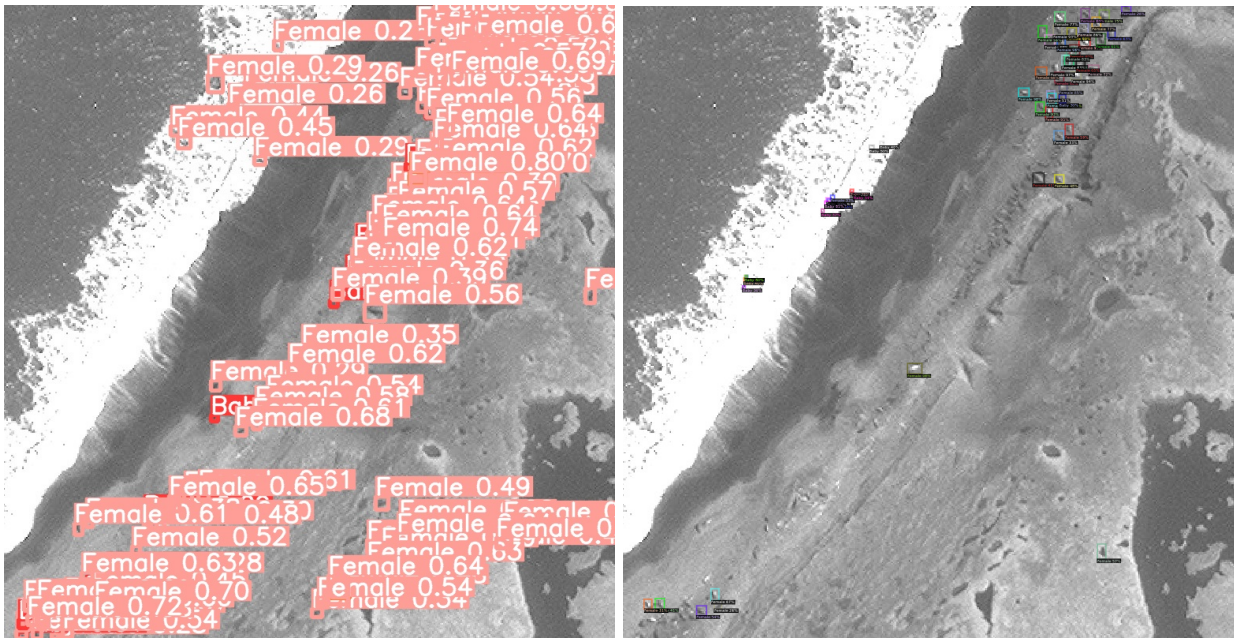

Figure 13: Pointe Richard 7, a detection test result comparison between YOLO and Faster R-CNN

6 Evaluation

In machine learning, two typical problem types are regression problem and classification task. They use totally different criterias and indicators. Like Root Mean Square(RMS) or Mean Average Percentage Error(MAPE) are widely used to evaluate the total loss alongside with the regression prediction since regression, most of the case, aims to fit a linear or a polynomial continue function to lower the difference of train set and prediction. The classification are evaluated by accuracy, precision, recall etc.

But deep learning model especially developed for detection task is not necessary the same. It has two tasks at a time and each has same importance for the total loss : (1)classification (2) localization(some model slightly changes the weights of them, like YOLO more focus on classification, but generally speaking, a loss function needs to include them to evalutate with no bias). The first task evaluates if the object predicted in one bbox has the same label as in the training or validation set. The second is to evaluate how closer the label's bbox situation and predicted bbox is. Thus, for this kind of task we use IoU(Intersection over Union) which compute the overlapped intersection of the labeled bbox and predicted bbox.[28] IoU = 1 means they perfectly overlap. The obtained IoU higher than threshold is considered as true Positive, otherwise false positive. If the labeled object is not presented by the predicted set, it is regarded as false negative. Then we have two evaluation rate formula:

$$Precision = \frac{true\ positive}{true\ positive + false\ positive}$$

$$Recall = \frac{true\ positive}{true\ positive + false\ negative}$$

and we define $F1$ curve

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

This yields:

$$precision = \frac{F1 * recall}{2recall - F1}$$

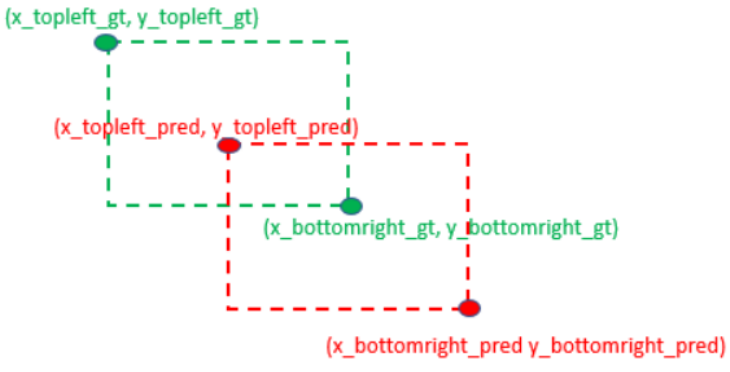

Figure 14: IoU calculates the overlapped area between the red bbox(predicted) and green one(given labeled bbox)

We calculate arithmetic mean of the interpolated precision(mAP) of each recall level which is split in 11 equal space, which means, from 0.0 to 1.0 with 0.1 step. For every recall level, the interpolated precision is the average precision among 11 recall points. The formula of mAP using 11 point interpolation: $mAP = \frac{1}{11} \sum_{recall \in \{0, 0.1, \ldots, 1\}} \rho_{Interp(recall)}$ where $\rho_{Interp(recall)}$ is interpolated precision with respected to recall level.

These guidelines give us the ability to evaluqte

7 Result

As we mentioned in the previous section, we used a total of 15 beaches for a total of 106 intercepted images covered 0.3m, 0.4m and 0.5m resolution, with weights chosen to be the best weights in the model training. All images in the YOLO test were transformed to 1280 pixel, the image size of 640 makes YOLO lose almost all detection capability, while the faster R-CNN does not have any size requirement for the input images. For both models, the threshold used in detection is 0.25, which is also the default threshold for YOLO. For all images contain beaches we counted only the total number of females and compared this with the manual count and the predictions of the GLM model. A total of three manual counts were performed and the final average was taken.

Table 4 is the comparison of manual count, Yolo output and Faster R-CNN output. Firstly the results for YOLO as in the model on the validation set as Figure 13. It is able to clearly identify the general distribution of the cluster and the females in the cluster can be identified with a high IOU, but it is still possible to observe a part of the background being identified as female, note that the part of the wave in the upper left corner the IOU is significantly lower compared to the IOU in the cluster, so this problem can be solved by using a suitable threshold. The part of the beach in the lower right corner of the image is identified as female with a higher IOU, although the number is not high but the observation shows that the misidentified objects are also very similar to females. This phenomenon can be solved by adding more female and null labels. It can be seen that the main confusion arises from the similarity between the female and the background on the satellite images.

As for Faster R-CNN, we found on more than one test set that although its recognition to female is very accurate, it can only recognize objects with high brightness, clear edges and high frequency, so most of the individuals in the cluster will be ignored. We also observed that for almost all images, regardless of their resolution, the model was able to identify only the central region of the cluster, and that the region around the cluster was often not identified. Lowering the threshold will not improve the recognition rate of this part, but rather more backgrounds will be mistaken for females.

YOLO is clearly better at detection of pup, and on VHR satellite images such as Pointe Richard 7, the Faster R-CNN fails completely on pup, and we speculate that the RPN filters out the pixel of pup. Both Vallée du Sextant images show very low recognition, as most of the area is covered with clouds, and the Faster RCNN has better recognition in the areas covered with clouds, but the number of individuals that both models can recognize is much smaller, by about 10% than the manual count,.

For all the beach in the dataset, Plage Jaune and Anse du Gros Ventre are 0.3m resolution and others are 0.4m, no low resolution image is included. YOLO obtained 7,867 females in a total dataset of 8,944 seals, which is 87% of the manual count, compared to 57% for the Faster R-CNN but much more accurate.

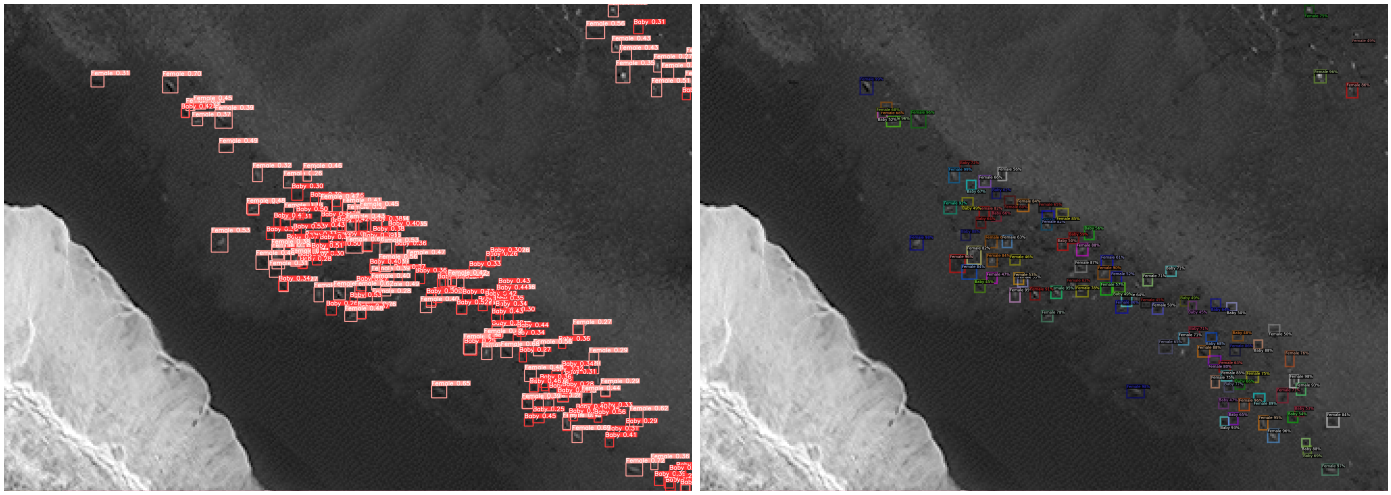

Figure 15: 11OCT14051534-P2AS-056844113040_01_P001 0.5-scale, left YOLO prediction; right Faster R-CNN prediction. Notice that in YOLO, a pup cluster appears in the middle of the female cluster, but these should have been identified as female

Table 4: Manual count, Yolo output and Faster R-CNN output

Nom Plage	manual	YOLO	YOLO%	Detec2	Detec2%
Pointe Richard 1,2,3	222+20+655=897	899	1,002229654	353	0,5389312977
Pointe Berger 1,2	37+37=74	91	1,22972973	37	0,5
Plage Noire	697	617	0,8852223816	382	0,5480631277
Plage Jaune	520	565	1,086538462	189	0,3634615385
Anse du Gros Ventre	1265	1658	1,310671937	798	0,6308300395
Plage des Lions Marins	2436	2318	0,9515599343	1740	0,7142857143
Baie du Young Williams 1	18	29	1,611111111	27	1,5
Baie du Young Williams 2	0				
Baie du Young Williams 3	0				
Petite Anse du Melissas	32	21	0,65625	44	1,375
Plages des Portes de l'Enfer 1,2	140	69	0,552	60	0,48
Vallée de l'Octant	1092	588	0,5384615385	399	0,3653846154
Vallée du Sextant 2	322	12	0,03726708075	33	0,102484472
Vallée du Sextant 1	174	17	0,09770114943	33	0,1896551724
Vallée du Télluromètre 2-8	673	506	0,7518573551	347	0,5156017831
Vallée du Télluromètre 1	254	192	0,7559055118	78	0,3070866142
Plages du Glacier J. Brunhes	350	285	0,8142857143	196	0,56
Total	**8944**	**7867**	0,8795840787	**4716**	0,579385625

Table 5: DigitalGlobe 0.5m

no	Man	YOLO	YOLO rate	Dec	Dec rate
1	207	135	0,652173913	73	0,3526570048
2	164	108	0,6585365854	65	0,3963414634
3	164	85	0,5182926829	69	0,4207317073
4	184	98	0,5326086957	79	0,4293478261
5	97	86	0,8865979381	44	0,4536082474
Total	**816**	**512**	0,6274509804	**330**	0,4044117647

To compare the extent to which high-resolution satellite images improve recognition rates, we used a 0.5m-scale image (11OCT14051534-P2AS-056844113040_01_P001, Figure 15) from which we segmented a total of five clusters and calculated them, noting that the image we used had a lot of rocky background and are not labeled after comparison with Google Earth and BingMap which is very noisy, and we calculated the number of females manually with a loose criterion because the lack of resolution fall into the plenty of blur in the edge of the individual. The problem of background and female confusion that occurs at other resolutions persists because of noise, and pup and female become more difficult to distinguish at low resolution,(table 5) for both the model and the human eye, with YOLO and R-CNN yielding 15% lower numbers compared to both high-resolution and very-high-resolution images.

The above results confirm the feasibility and reliability of neural networks for individual recognition in satellite imagery and demonstrate that different resolutions do have a significant impact on the accuracy rate. In order to obtain a numerical prediction, we use the resolution and the output of the model as variables. We assume the relationship between these variables is linear and use the multipal linear model to correct for the manual counted data. We use R function $lm()$ on two model. For yolo we use lm(formula = main s(resolution) + YOLO2, data = data) with Multiple R-squared: 0.8782, Adjusted R-squared: 0.8639 and p-value: 1.691e-08:

Table 6: YOLO prediction

Coefs:	Estimate	Std.Err	t-val	Pr($\geq \|t\|$)
(Intercept)	-271.7694	411.6392	-0.660	0.518
s(resolution)	766.0935	940.7460	0.814	0.427
output	0.9405	0.0916	10.268	1.05e-08

And for faster R-CNN we also apply a same study which lead to the result: Multiple R-squared: 0.9035, Adjusted R-squared: 0.8922 and a p-value 2.331e-09. For both adjustment we accept the result due to the p-value and after a comparison of manual, model predict and adjusted model output data.

Table 7: faster R-CNN prediction				
Coefs:	**Estimate**	**Std. Err**	**t value**	**Pr($\geq |t|$)**
(Intercept)	328.981	341.776	0.963	0.349
s(resolution)	-559.041	791.979	-0.706	0.490
output	1.360	0.116	11.729	1.43e-09

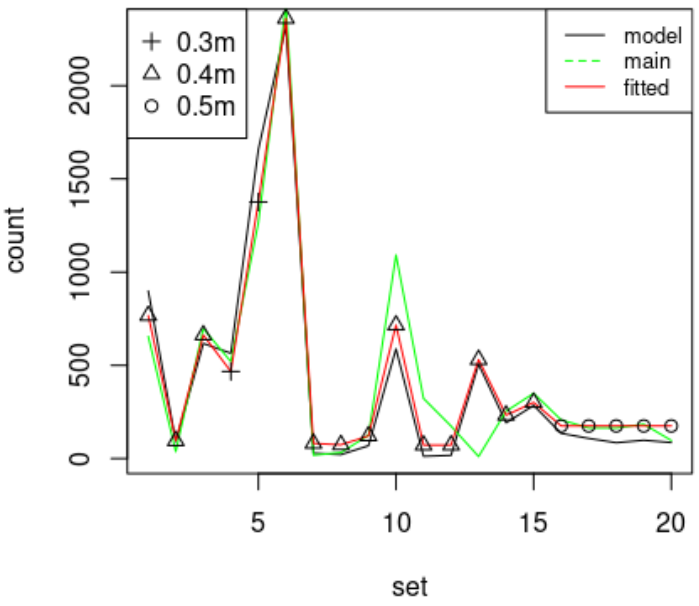

Figure 16: lm fitted data achieve comparisonnal count by comparison with manual count

The fitted output(red line) fits very well those counted by hand(green line), especially the large gap of cloudy image by adjusting them closer to the manual count line. The effect of the adjustment influence the most on 0.5m-scale images and almost overlap with the green line. The new total count of prediction is 8804 instead of 7867, Almost indistinguishable from manual counted data(8944).

8 Conclusion and future work

In this paper We first segmented the beach with the elephant seal from a total of 28 multi-resolution satellite images to form a dataset of 105 images and labelled the individuals with three categories: pup, female and male. We chose two deep learning networks, YOLOv5 and Faster R-CNN improved by adding support to cross validation and tested the model with the highest mAP: yolov5m and faster_rcnn_X_101_32x8d_FPN_3x are the choosen model. We train these two on our training set. YOLO has a very good representation rate while Faster R-CNN is almost 30% less efficiency. These two approaches also work well on the 0.5m resolution but have met under estimation on the cloudy image. For each model after an adjustment by linear model, the predicted population is comparable with manual count.

Note that we did not test the other Mask based models in the detectron 2 model zoo. the cost of their labeling on pixel-level images is very large although the mask R-CNN has been shown to have good performance in terms of prediction rate and accuracy.

There is also a need to expand the existing database. 105 images were not enough for this experiment. Finally, the problem of all models failing on images with high noise levels especially those with poor meteorological conditions needs to be addressed. Another possible way of reduce the noise impact comes from the background and the weather condition is that, in the non-breeding season, one could capture the background with no elephant seal exist, including the various weather condition images occur in the background. This could help model like YOLO better understand the different essences of individuals and background.

9 Acknowledgements

Thanks to Christophe Guinet and Nicolas Sidère for giving me this exciting opportunity to complete my PFE project at N7, it was thanks to them that I was able to not only develop my skills in the field of machine learning, but also learn a lot about marine biology and realise that there is such a large, beautiful and vivid ecology outside my living space. Both of them have also helped me a lot during my internship, and their extensive industry knowledge has been the cornerstone of my ability to complete this internship.

I am also particularly grateful to Joris Laboris, whose master's internship project provided the direction for the project and the methodology. He provided us with all the satellite images and guidelines on how to label them. He actively helped me in the best possible way during my internship.

I would like to thank Shumin Zeng, my fiancé, for her support and encourage in my life during the six months so that I could insist and concentrate on my work. She made many constructive suggestions for my approach such as encouraging me to go on a field trip to study the ecology of the elephant seal. Although this was ultimately unsuccessful.

References

[1] Joris Laborie, Matthieu Authier, Adrien Chaigne, Karine Delord, Henry Weimerskirch and Christophe Guinet, Use of very high-resolution satellite imagery for estimating total population size of southern elephant seal (Mirounga leonina) in Kerguelen and Crozet Archipelagos, and trends re-examination in the French Subantarctic islands in prep

[2] Sue E. Moore, Marine Mammals as Ecosystem Sentinels, Journal of Mammalogy, Volume 89, Issue 3, 5 June 2008, Pages 534–540, https://doi.org/10.1644/07-MAMM-S-312R1.1

[3] Hindell, M. A., McMahon, C. R., Bester, M. N., Boehme, L., Costa, D., Fedak, M. A., Guinet, C., Herraiz-Borreguero, L., Harcourt, R. G., Huckstadt, L., Kovacs, K. M., Lydersen, C., McIntyre, T., Muelbert, M., Patterson, T., Roquet, F., Williams, G. and Charrassin, J.-B.. 2016. Circumpolar habitat use in the southern elephant seal: implications for foraging success and population trajectories. Ecosphere 7(5):e01213 10.1002/ecs2.1213

[4] BARRAT, A. and MOUGIN, J. (1978) L'Eléphant de mer Mirounga leonina de l'île de la Possession, archipel Crozet (46°25' S, 51°45' E). , Vol. 42 (Issue 2), pp. 143-174. https://doi.org/10.1515/mamm.1978.42.2.143

[5] Perrin, William F.; Würsig, Bernd; Thewissen, J. G. M., eds. (24 November 2008). "Earless Seals". Encyclopedia of Marine Mammals (2nd ed.). Burlington, Massachusetts: Academic Press. p. 346. ISBN 978-0-12-373553-9.

[6] QGIS Development Team, QGIS Association 2021, QGIS_software, QGIS Geographic Information System,https://www.qgis.org

[7] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[8] Sébastien BALLESTEROS, Le modèle linéaire avec R : fonction lm(), https://rug.mnhn.fr/semin-r/PDF/semin-R_lm_SBallesteros_110308.pdf

[9] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. CoRR, abs/1502.01852, 2015.

[10] Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. IJCV, 2010.

[12] Bengio, Y., Courville, A., Vincent, P. (2013). Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 35(8), 1798-1828.

[13] Krizhevsky, A., Nair, V., Hinton, G. (2009). Cifar-10 and cifar-100 datasets. URl: https://www. cs. toronto. edu/kriz/cifar. html, 6(1), 1.

[14] LeCun, Y and Cortes, C 2010 MNIST handwritten digit database. Available at http://yann.lecun.com/exdb/mnist/

[15] Olga Russakovsky*, Jia Deng*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. (* = equal contribution) ImageNet Large Scale Visual Recognition Challenge. IJCV, 2015.

[16] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In NIPS, 2012.

[17] J. Carreira and C. Sminchisescu, "CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 7, pp. 1312-1328, July 2012, doi: 10.1109/TPAMI.2011.231.

[18] I. Endres and D. Hoiem. Category independent object proposals. In ECCV, 2010.

[19] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229.

[20] Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. FeiFei. ImageNet: A large-scale hierarchical image database. In CVPR, 2009.

[22] Ren, S., He, K., Girshick, R., Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497.

[23] Yuxin Wu and Alexander Kirillov and Francisco Massa and Wan-Yen Lo and Ross Girshick, Detectron2,https://github.com/facebookresearch/detectron2, 2019

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft COCO: Com- ´ mon Objects in Context," in European Conference on Computer Vision (ECCV), 2014.

[25] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[26] the Simonyan and Zisserman model networks for large-scale image recognition," in International Conference on Learning Representations (ICLR), 2015.

[27] Biewald, Lukas, wandb, Experiment Tracking with Weights and Biases, 2020, Software available from wandb.com, https://www.wandb.com/

[28] Evaluating performance of an object detection model Renu Khandelwal 2016, https:towardsdatascience.com/evaluating-performance-of-an-object-detection-model-137a349c517b

[29] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).

[30] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional neural networks," in European Conference on Computer Vision (ECCV), 2014.

[31]] C. Szegedy, S. Reed, D. Erhan, and D. Anguelov, "Scalable, high-quality object detection," arXiv:1412.1441 (v1), 2015.

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. CoRR, abs/1409.4842, 2014.

[33] M. Lin, Q. Chen, and S. Yan. Network in network. CoRR, abs/1312.4400, 2013.

10 Appendix

10.1 Big picture of the project

10.1.1 laboratories included

CEBC-CNRS, Centre d'Etudes Biologiques de Chizé was found in 1951, the mission of the centre is to understand how species adapt - or disappear - in the face of natural changes, or those induced by man's use of nature. Our study models are vertebrates, reptiles, birds and mammals, both terrestrial and marine. Their research is fundamentally makes an important contribution to future environmental solutions and protection measures. 3 teams are integrated in the center: AGRIPOP, ECOPHY and Prédateurs Marins. This report is a part of Prédateurs Marins whose big idea is the study on global changes' affect on the population, especially marine ecosystems by using birds and mammals. Naturally they integrate the dynamics of lower trophic levels and are considered reliable indicators of the state of the ecosystems to which they belong.(reference: CEBC and Prédateurs Marins)

Created in 1993, the Computer Science, Image and Interaction Laboratory (L3i) is the research laboratory of the digital sciences domain of La Rochelle University with label of "Équipe d'Accueil" (EA 2118) and label of Technological Research Team (ERT).The laboratory associates researchers in computer science from the IUT and the Faculty of Science and Technology of La Rochelle University. (more info see: L3i)

10.1.2 Previous and related works in the project

As mentioned in section 2 (Hindell et al. 2016) proposed elephant seal's population prediction were measured based on several criteria: their at-sea distribution and important foraging areas; their broad-scale habitat as well as the use of the sea ice zone; the different water masses used by the seals at their foraging depths using CTD; the region importance to the population; how they profit habitats reponding to the seasonal ice edge expansion and last the comparison of different stock population trends. These data came from two hundred and eighty-seven elephant seals equipped with oceanographic sensor satellite tags. To compare response variables to sex, region, seasons, the time of day with seal ID, LMM is used, use GLMM when it comes to binary variables with seal ID. The result is also showed in section 2.

My work is an extension of the Joris Laborie's master internship project with PSL University, Ecole Pratique des Hautes Etudes (EPHE): Use of very high-resolution satellite imagery for estimating total population size of southern elephant seal (Mirounga leonina) in Kerguelen and Crozet Archipelagos, and trends re-examination in the French Subantarctic islands, with Matthieu Authier, Adrien Chaigne, Karine Delord, Henry Weimerskirch and Christophe Guinet as coopertaors His article has shown me the way to my internship. His work aims to present an efficient approach to estimate the total elephant seal population on Kerguelen and Crozet archipelagos. It's in his work they started to use high resolution satellite imagery for ground counts. With them, they build a statistical predictive model to estimate breeding females on given beach.

10.1.3 My approaches

My work is to generate a dataset contains all the satellite images in the previous work, test deep learning models on this dataset. I need to demonstrate that deep learning has better convenience and comparable accuracy in such satellite images, as opposed to lengthy manual ground counts. I have to evaluate the accuracy on different resolution. Analyse the reasons for its errors and train the model to detection for future datasets. Finally, determine a way to correct the estimation bias.

Based on Joris Laborie's work, in the initial stage, I worked with him, Christophe Guinet and Nicolas Sidère to study together the feature of satellite images. I looked at the differences in the features that elephant seals show on satellite images at respected resolutions, I learned from them how to use Qgis, a geoscience tool to distinguish between females, males and pups in different resolution and weather. Qgis provides an equal scale measurement tool as a way to measure the size of individuals, and I learned from them how to set thresholds to distinguish different individuals. I also studied the influence of image pre-processing like gamma, contrast ratio etc on our satellite images.

After the premier stage, I started to contribute a dataset from the original satellite images. I split out the areas of the 28 images that contained beaches resembled seal cluster which resulting in 105 images. Some of these images have been pre-processed to make the edges of the elephant seal more visible. With the help of Roboflow annotate tool, like those in the Figure 5, I labeled elephant seals in each beach in three catalogue. For each individual I measure the size by Qgis and draw a bbox around them.

Before experiment, there were three potential options: YOLO network, Faster R-CNN and Mask R-CNN. The previous two is detection neural network while the third is designed for segmentation task. I tried to segment individuals on the satellite images, at last it failed into failure because It's hard to draw the mask upon the individuals who just showed as several pixel on the satellite images.

In the end I chose the first two state-of-art models and tested their performance in the local environment and on the cloud respectively. For YOLO pretrained model, in the local environment it shows a long training execution time, but have a good speed

in detection. yolov5m use less GPU memory but have a competitive accuracy and presentation rate. I also read the row YOLO's PyTorch script, I found that their script doesn't support cross validation during the training, so I, with repect the original YOLO's API, I wrote a script which allowed the training process auto-distribute the training and validation set. For Faster R-CNN network, I found that detectron 2 framework, powered by Facebook AI group, has already provide a mature, easy-using tool of deployment. The training and test result and correction are well explained in the section 5 and section 7. Our criteria is presented in the section 6.

Finally, to meet the requirement that the code must be simple to use, understand and use without industry knowledge, run without Linux environment, I created a Google Colab Jupyter notebook, it provide a T80 GPU with 16GB GPU memory which is sufficient for the training of our dataset. Also it doesn't have environmental requirement. In the future work, just simply need to update the dataset and replace the folder name.

10.1.4 Previous work's imagery

id	Zone	area (km²)	Prix € / Résolution panchromatique			Satellite	Filiale	Cathalogue Id	Pan-Réso (m)	Date acquisition
			pan-30cm	*pan-40cm*	*pan-50cm*					
1	Baie du Noroît	25,6		348,57		GeoEye1	Digital Globe	1050010006A08100	0.44	12/10/2016
2	Anse du Gros Ventre	25,2	407,91			WorldView3	Digital Globe	104001002371F200	0.33	12/10/2016
3	Plage Demi-Lune	25,2			300,28	WorldView2	Digital Globe	103001000EBA6200	0.52	14/10/2011
4	Baie Bretonne 1-1	25,2		343,31		GeoEye1	Digital Globe	1050010006A08100	0.44	12/10/2016
5	Baie Bretonne 2-1	28,7		391,51		GeoEye1	Digital Globe	1050010006A08100	0.44	12/10/2016
6	Baie Bretonne 3	28,3	457,16			WorldView3	Digital Globe	104001000322CE00	0.38	17/10/2014
7	Baie de la Mouche 1	41,5		565,36		WorldView2	Digital Globe	103001005E64E800	0.49	29/09/2016
8	Baie de la Mouche 2	25,8	417,31			WorldView3	Digital Globe	1040010022D4D100	0.33	06/10/2016
9	Fjord Larose	30,5	493,42			WorldView3	Digital Globe	1040010022D4D100	0.33	06/10/2016
10	Baie Larose 1	25,4			303,41	WorldView2	Digital Globe	103001000D806F00	0.55	14/10/2011
11	Ile du Port	27,2			323,85	WorldView2	Digital Globe	103001000EBA6200	0.52	14/10/2011
12	Presqu'île Joffre 1	25,7			306,97	WorldView2	Digital Globe	103001000EBA6200	0.52	14/10/2011
13	Presqu'île Joffre 2	25,2			301,02	WorldView2	Digital Globe	103001000EBA6200	0.52	14/10/2011
14	Ile Howe	25,5			303,48	WorldView2	Digital Globe	103001001CCDD700	0.55	20/10/2012
15	Nord île Foch	28,1			335,03	WorldView2	Digital Globe	103001001CCDD700	0.55	20/10/2012
16	Sud île Foch	25,8			307,81	WorldView2	Digital Globe	103001000D806F00	0.54	14/10/2011
17	Baie de l'African	45		612,48		WorldView2	Digital Globe	10300100742C9900	0.48	16/10/2017
18	Péninsule Loranchet	25,8		351,74		WorldView2	Digital Globe	1030010007481F00	0.49	20/10/2010
19	Baie du Centre	25,3		344,76		WorldView2	Digital Globe	1030010007481F00	0.49	20/10/2010
20	Baie de Recques	25,8		350,97		WorldView2	Digital Globe	1030010007481F00	0.49	20/10/2010
21	Baie de la Table	31,8	514,66			WorldView3	Digital Globe	1040010022D4D100	0.33	06/10/2016
22	Plage du Feu de Joie	26,7	431,75			WorldView3	Digital Globe	1040010023AD0800	0.33	12/10/2016
23	Baie Norvégienne	205			2448,68	Pléiade 1B	Airbus - Defence & Space	DS_PHR1B_2015102805 09422_FR1_PX_E070S5 0_0516_01260	0.50	28/10/2015
	Sous-total	824,3	2722,21	3308,7	4930,53					
	Total			10961,44						

Figure 17: 23 satellite imageries cover Kerguelen and Crozet during the reproduction season with 0.3m, 0.4m and 0.5 resolution. They are used in the previous work by Joris Laborie et al. Our work also include these in dataset captured by different satellite.

10.2 YOLOv5m training process

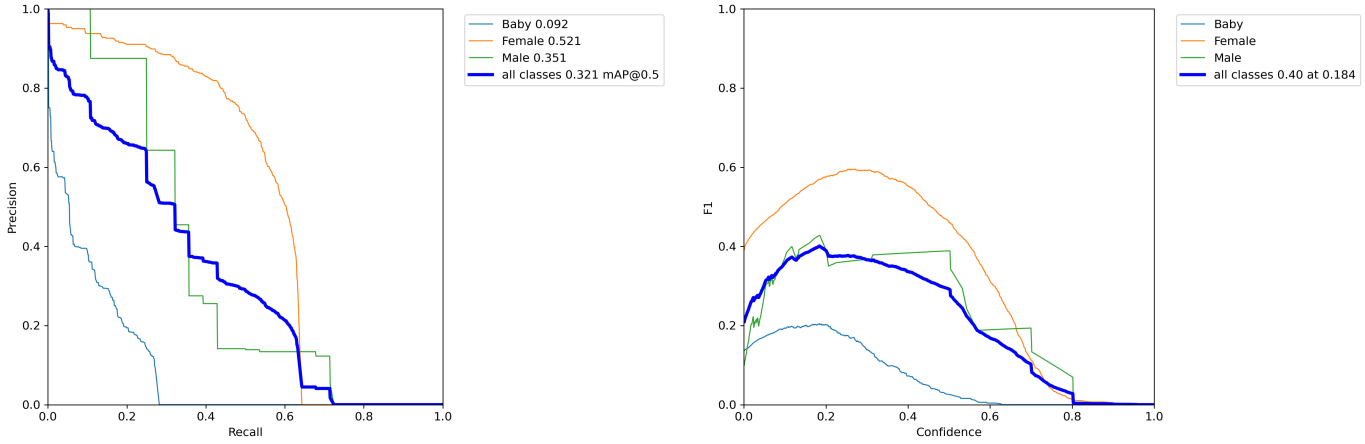

Figure 18: PR curve and F1 curve during training

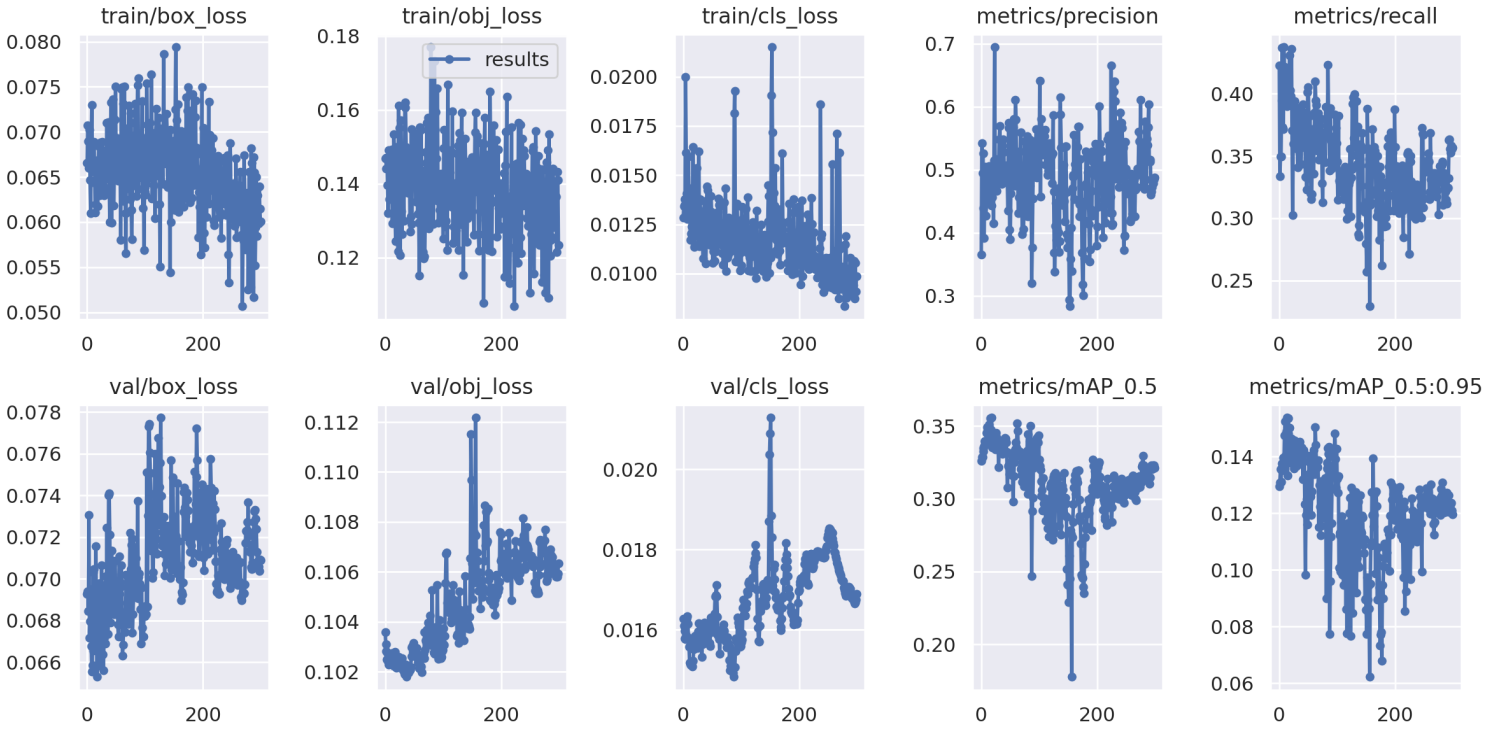

Figure 19: Result contains training losses(reg and cls), recall , precision, mAP.5 and mAP.5:.95

10.3 A closer look at confusion matrix

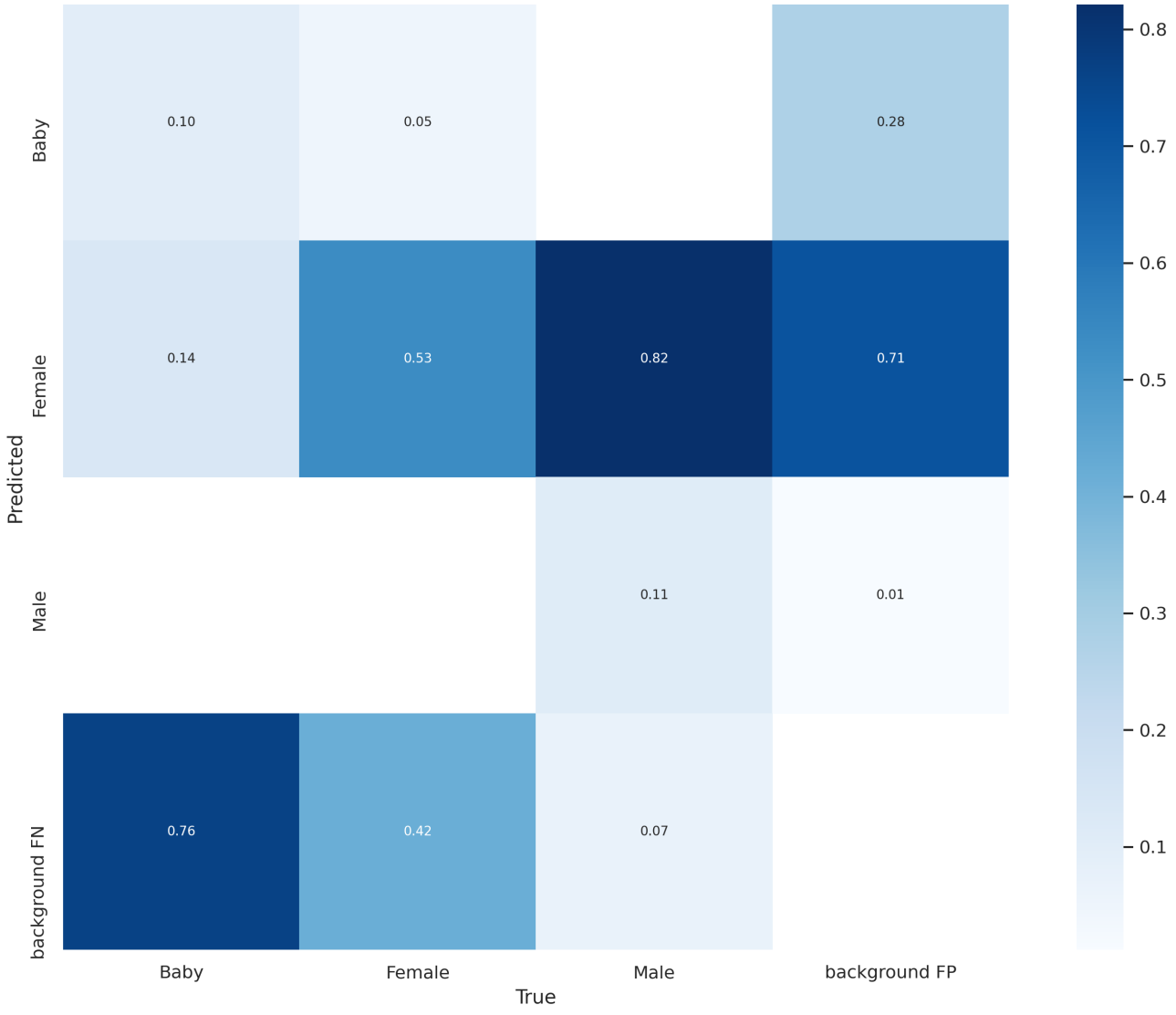

Figure 20: training confusion matrix

Figure 23 illustrates the main sources of errors generated by YOLOv5m. We see that the number of pup predictions has almost no correlation with the number of females, i.e. the introduction of pup did not have any effect on the prediction of the number of females. In contrast, a significant proportion (0.42) of the classification of females was mistaken for background. This proportion results in the number of females always being underestimated in the text. As for males, because of the inadequate sample and very small numbers, there is little effect on the other labels.

10.4 Failure case

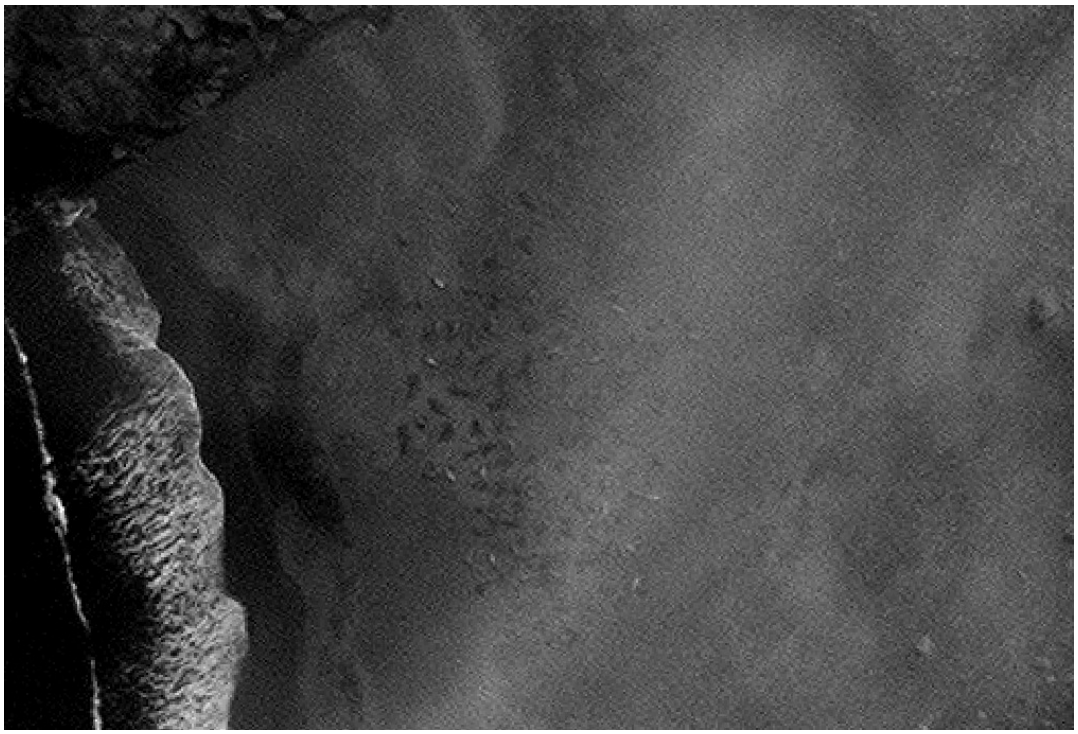

Figure 21: An example of how weather condition effect accuracy

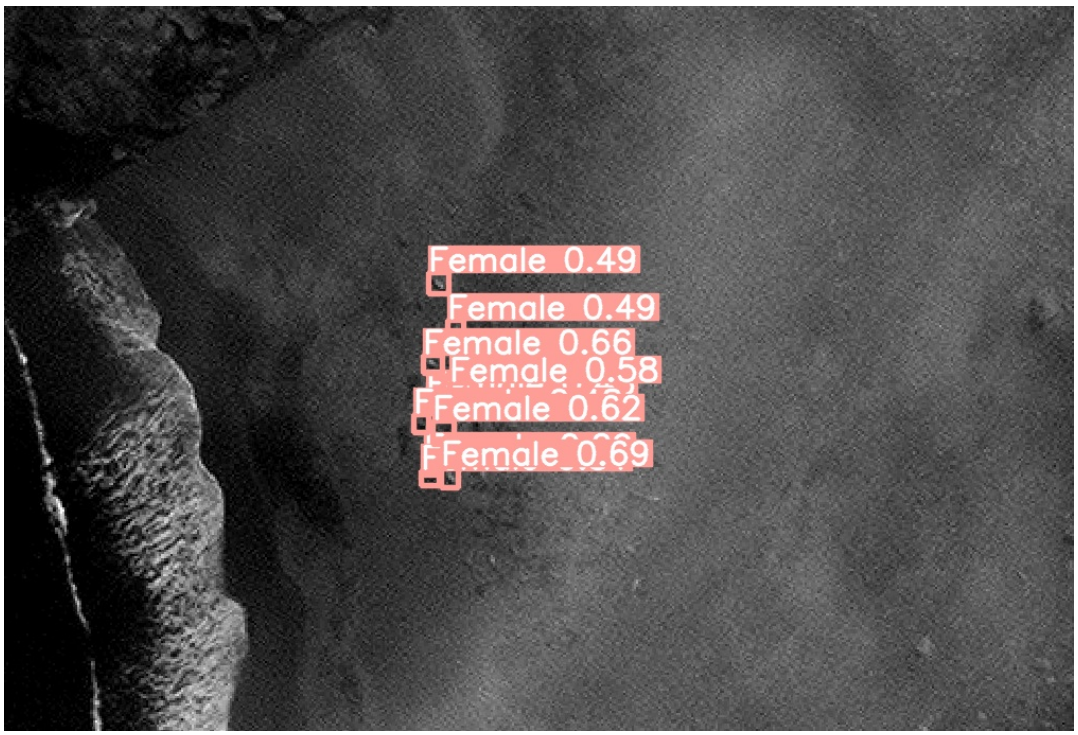

Figure 22: YOLOv5 fails on the cloudy weather condition images

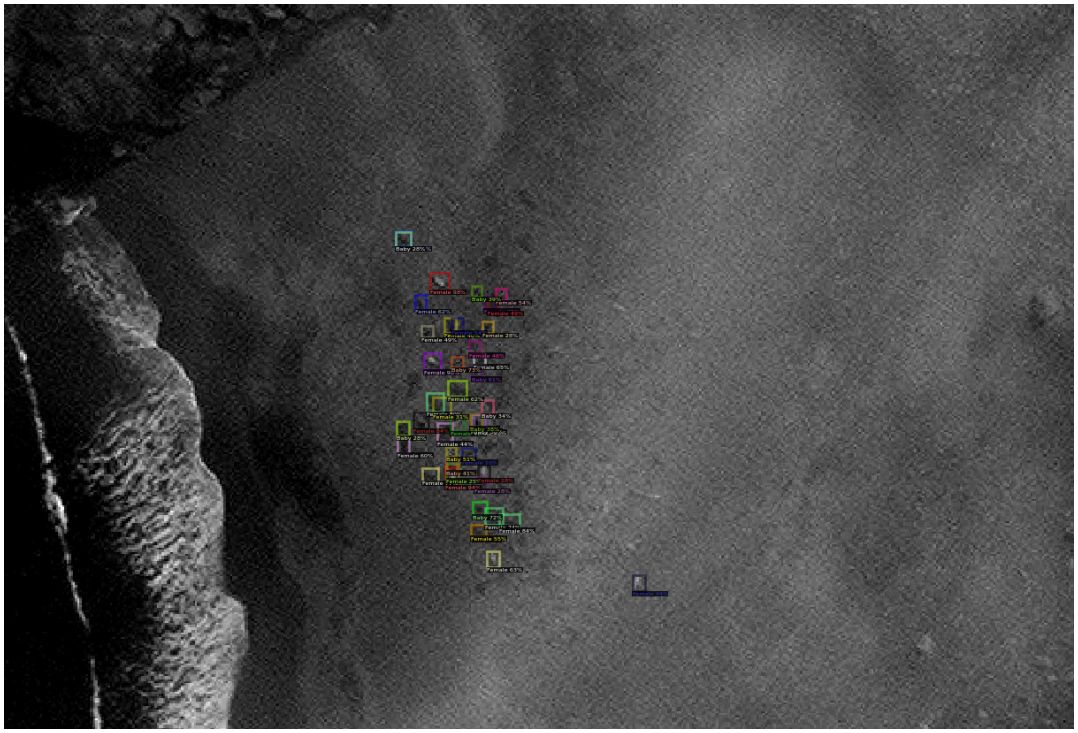

Figure 23: YOLOv5 fails on the cloudy weather condition images

Figures 20, 21 and 22 show the original satellite images, the predictions of YOLO and the predictions of Faster R-CNN, respectively. The right half of the original image is covered by clouds. We found that this part of the elephant seals are difficult to distinguish their boundary even if it is manually marked. YOLO can only identify very few individuals, while the Faster R-CNN seems to perform slightly better than the former. However, there is still a huge gap between the two for manual recognition

10.5 Faster R-CNN test result example

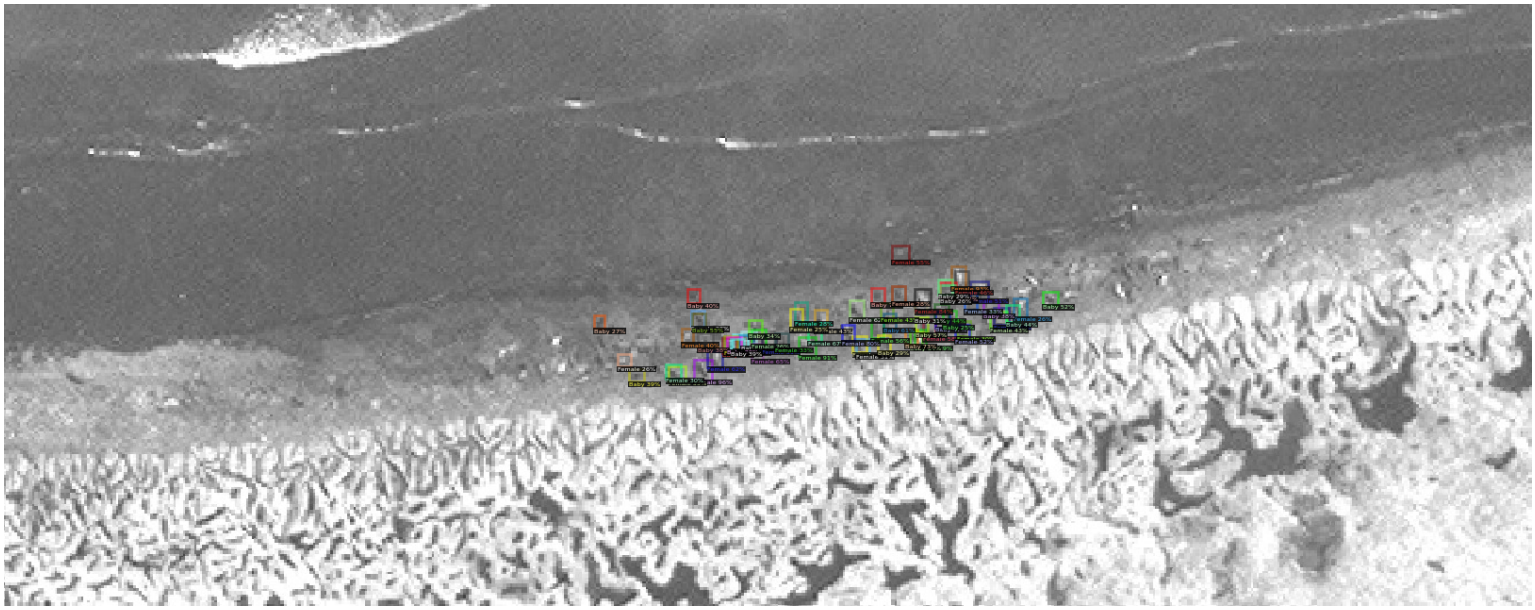

Figure 24: Pointe Richard 1

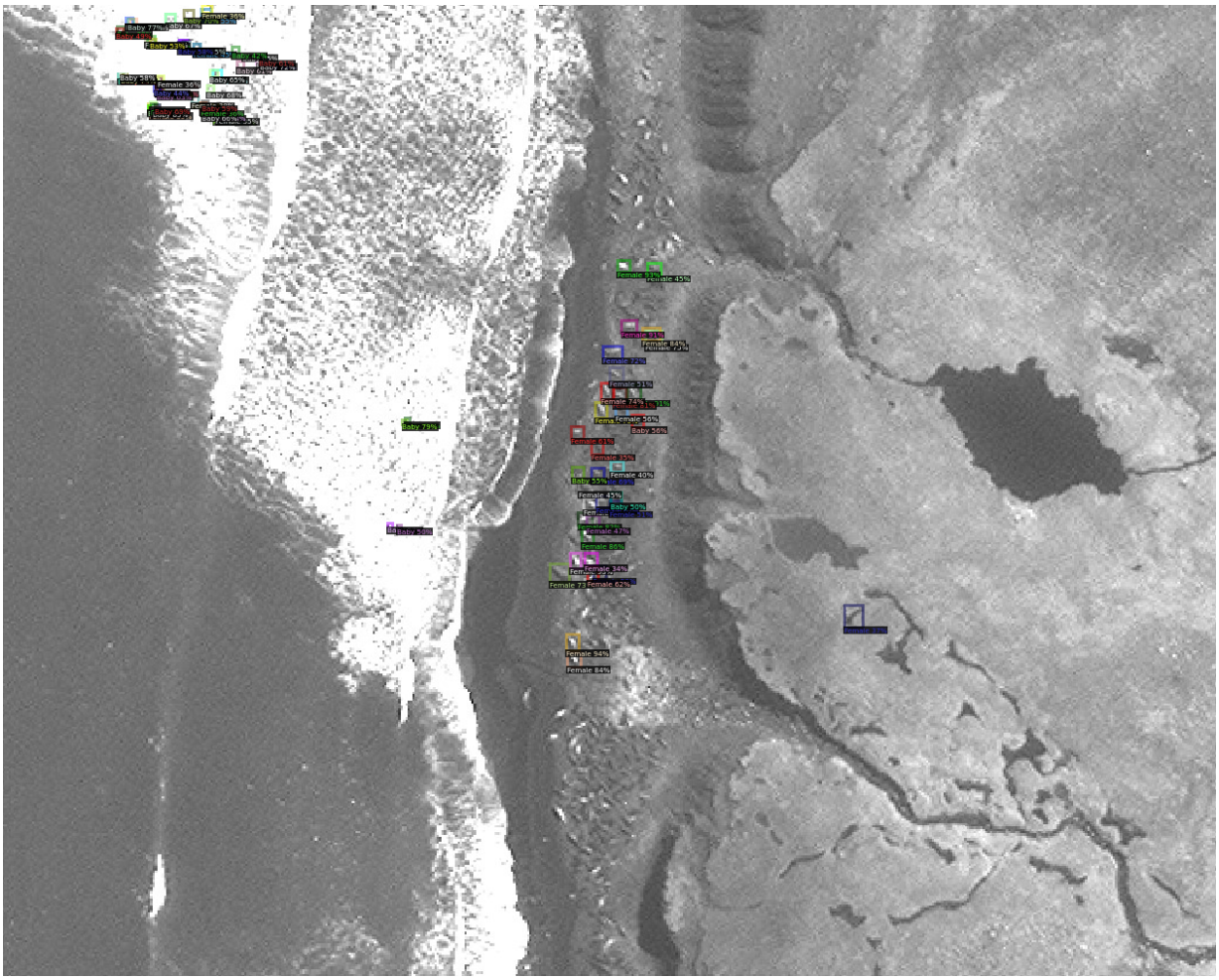

Figure 25: Pointe Richard 2

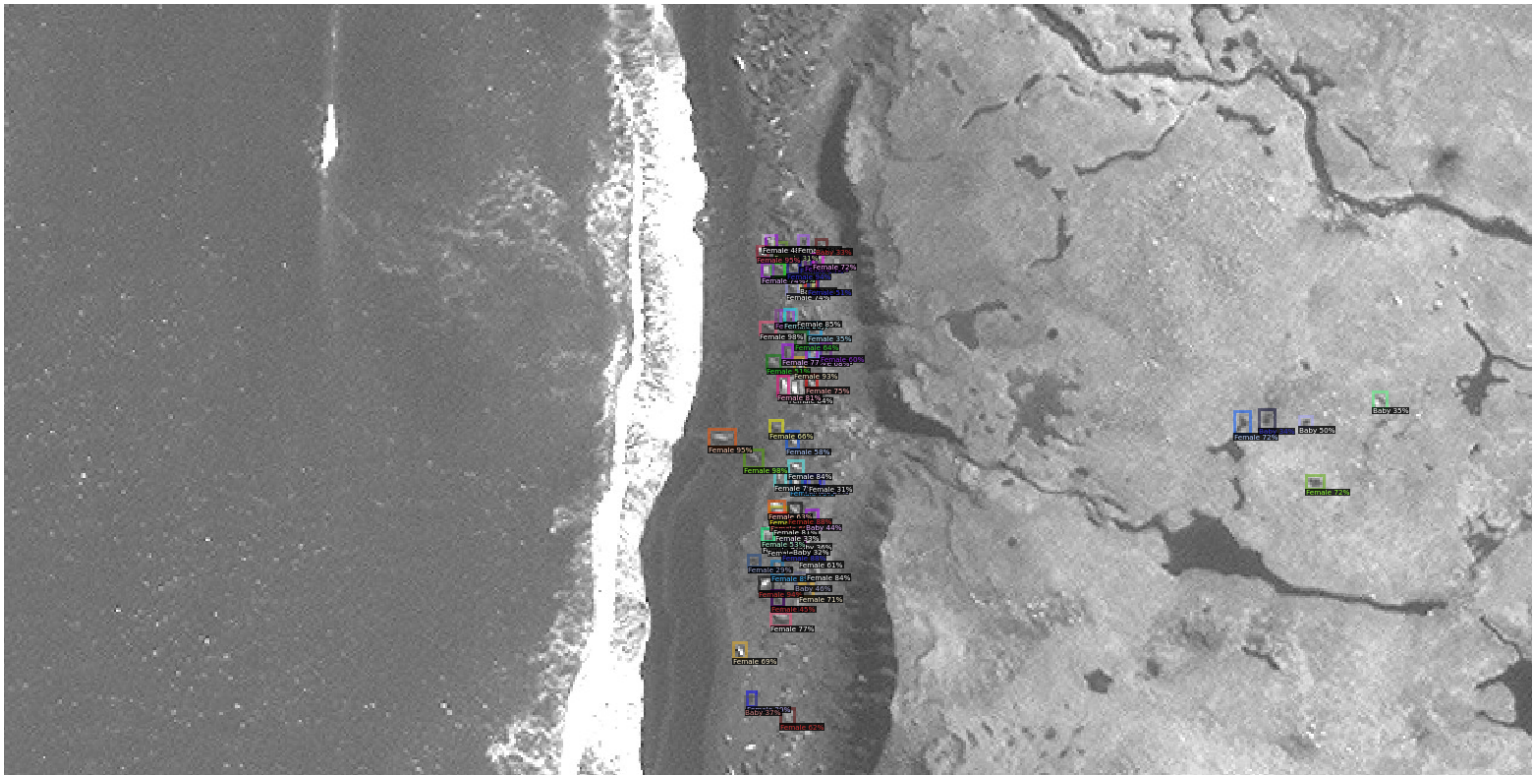

Figure 26: Pointe Richard 3

The figure 24 to 26 present a vivid example of how Faster R-CNN detect and learns the feature of test dataset. They exhibit similar characteristics. Note from these diagrams that all the identified clusters are concentrated in the centre of the image, and it seems that RPN tends to propose more regions in the centre of the image. More than that, all the detected individual have a brighter shape and grey level than the false negative individuals.