A multi-domain generalizable animal detection model: exploration of potential solutions to improve the SOA

Deliverable 5

Presented by
Abdiel Fernandez - abdielfer@gmail.com
Rose Guay Hottin - guayhottin.rose@hotmail.com
Kevin Lessard - kevin.lessard@umontreal.ca
Santino Nanini - santino.nanini@umontreal.com

Department of Computer Science and Operations Research
Faculty of Arts and Science

Work presented to Alex Hernandez-Garcia
As part of the course IFT3710/6759
Projets avancés en apprentissage automatique

April 4, 2022

Deliverable 5 - Results so far and conclusions

Evaluation metrics

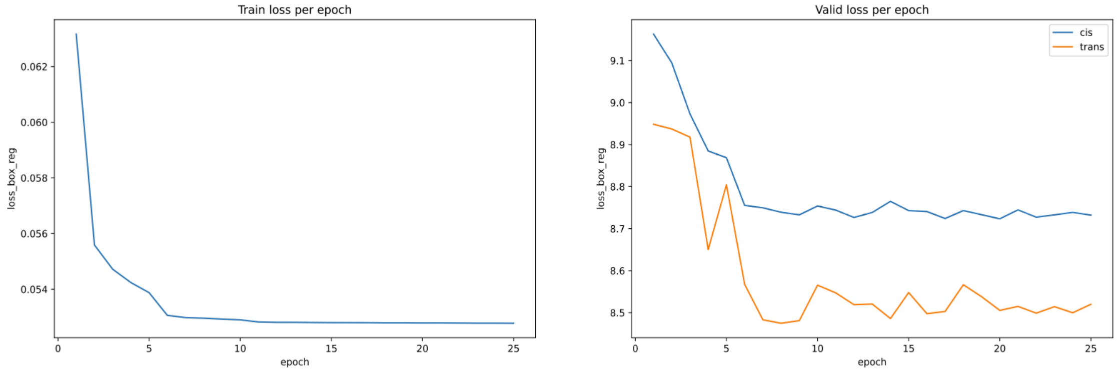
During training, the Faster R-CNN returns 5 different loss terms. The global loss, the classification loss, the regression loss (for bounding boxes), the objectness loss (binary classification at the level of the region proposal network (RPN); object or background) and the regression loss of the RPN [1]. Since our task only consists of animal detection (animal or background) and we fine-tuned layers that are “higher” than the RPN, we choose to show and use the final regression loss as the indicator of training.

Before evaluation on test sets we applied non maximum suppression (NMS) to filter out overlapping predicted bounding boxes. We then used different COCO object detection metrics to have a global idea of the performance of the model. As in our baseline paper [2], for comparisons across models and domains, we considered a predicted bounding box to be correct if its intersection over union (IoU) with a ground truth is greater than 0.5.

Baseline model - Fine-tuned pre-trained Faster R-CNN

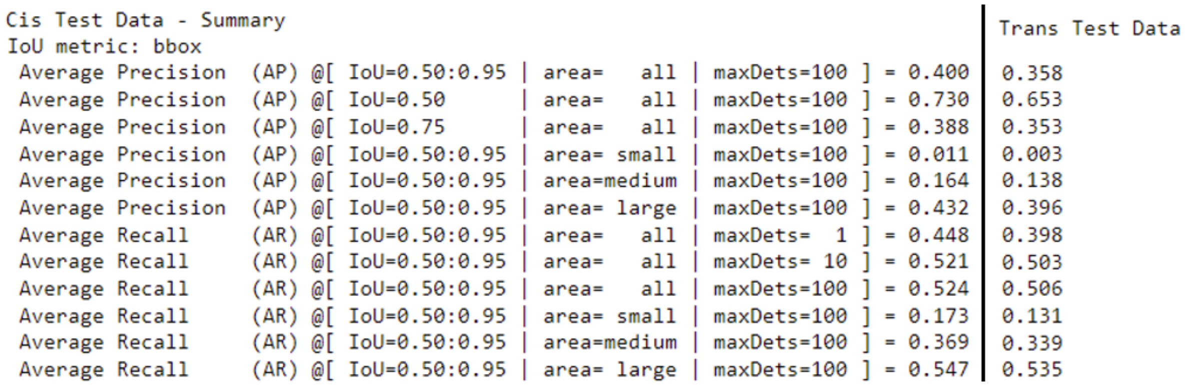
Our baseline model was trained with 25 epochs on the 12099 training images in single batches. We use data augmentation that performs an horizontal flip with probability 0.5. For each epoch, the model was evaluated on the cis-location and trans-location validation datasets in batches of 10 which consists of 2 different subsets of locations. Only the Faster R-CNN predictor was trained which is the last part of the model and is made of two fully connected layers of size (1024,2) and (1024,8) that give in parallel respectively the classification scores and the bounding box deltas (total number of trainable parameters = 10 250).

Train logs



We can clearly see that the trend drops rapidly for both graphs until about 8 epochs and then stabilizes.

COCO evaluator results



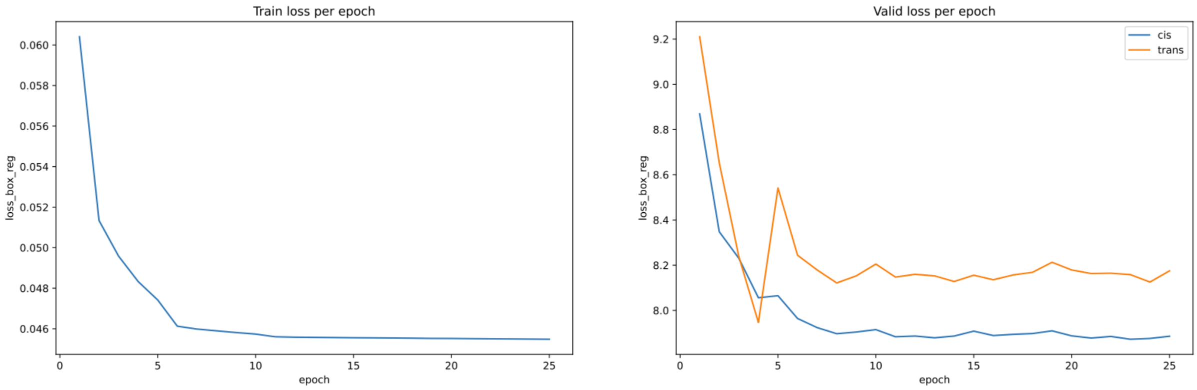
```
Cis Test Data - Summary                                                          | Trans Test Data
IoU metric: bbox                                                                 |
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.400  | 0.358
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.730  | 0.653
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.388  | 0.353
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.011  | 0.003
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.164  | 0.138
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.432  | 0.396
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.448  | 0.398
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.521  | 0.503
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.524  | 0.506
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.173  | 0.131
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.369  | 0.339
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.547  | 0.535
```

The state-of-the-art model [2] obtained accuracy of 0.771 for the cis-locations and 0.702 for the trans-locations. The detection accuracy of our baseline model for an IoU of 0.50 is 0.73 for cis-locations and 0.653 for trans-locations. Our baseline model is actually a bit worse probably because we used ResNet-50 instead of ResNet-101 as the backbone. Still, this is a very manageable baseline for the time it takes to train and our goal is to improve it.

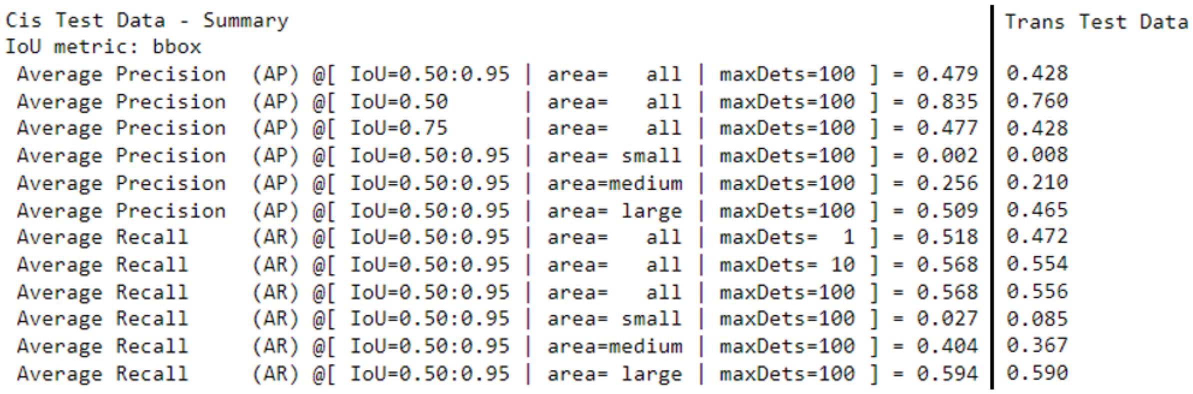
Deeper model - Deeper fine-tuned pre-trained Faster R-CNN

Seeing the early saturation in the training and validation losses, we decided to fine-tune deeper in the model to test if its performance could be improved. Using the same training setting as previously (batch size, data augmentation, number of epochs, etc), we trained the entire region of interest (ROI) heads which is made of the Region of interest pooling, two fully-connected layers and the Faster R-CNN predictor (total number of trainable parameters = 13 905 930).

Train logs



COCO evaluator results



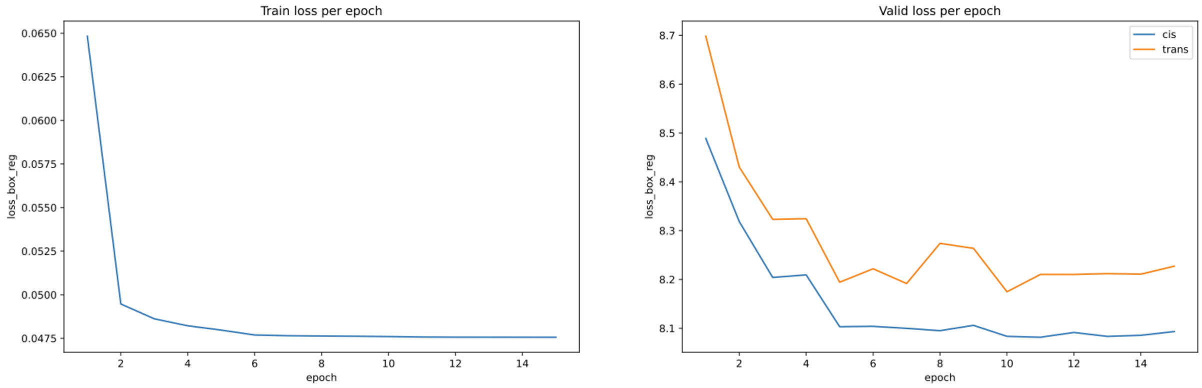
```
Cis Test Data - Summary                                                                       Trans Test Data
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.479   0.428
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.835   0.760
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.477   0.428
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.002   0.008
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.256   0.210
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.509   0.465
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.518   0.472
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.568   0.554
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.568   0.556
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.027   0.085
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.404   0.367
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.594   0.590
```

By training deeper, the average precision at IoU=0.50 increased from 0.73 to 0.835 on the cis-location test set and from 0.653 to 0.76 on the trans-location test set. Recall metrics also improved. This is likely due to the increase in capacity that comes with a “deeper fine-tuned model”.

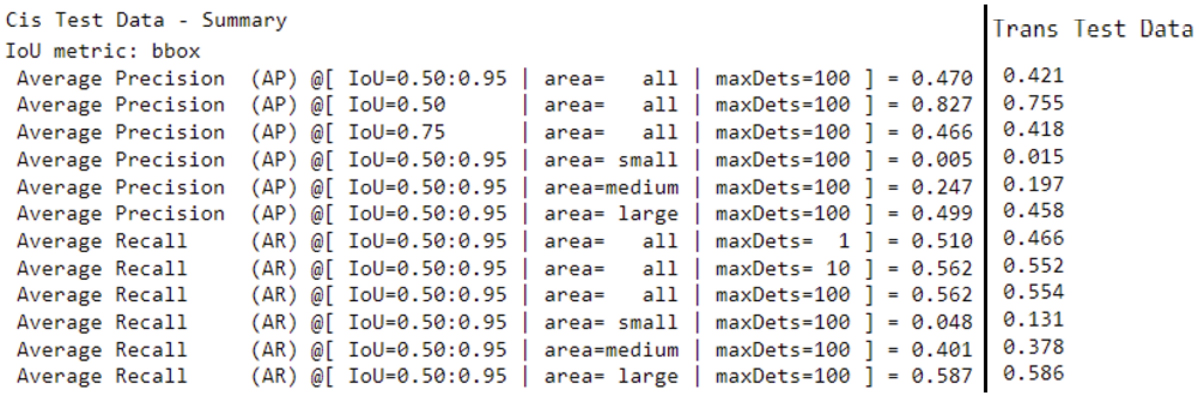
Method 3 model - Subspace alignment based domain adaptation

As a way to improve the baseline model’s performance on unseen camera location (trans-location datasets), we tried a subspace alignment based domain adaptation method inspired by [3] and [4]. Briefly, this technique consists of applying a principal component analysis (PCA) to reduce the dimensionality of the source and target space made of feature representations (here we choose to use the output of the box head to form the subspaces which are vectors of size 1024). Only the 100 eigenvectors corresponding to the 100 largest eigenvalues were kept for each domain. Then, a matrix mapping the reduced source subspace to the target one is obtained in a closed-form equation. During training, the source data is projected to the target-aligned source subspace and during evaluation, the target data is projected to the target subspace. This implies that the input size of the layers following the box head (right after the transformation is applied) has to be reduced from 1024 to 100. Using our fine-tuned deeper model, we initialized a Faster R-CNN predictor of size 100 and trained it from scratch for 15 epochs.

Train logs



COCO evaluator results



```
Cis Test Data - Summary                                                          Trans Test Data
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.470  0.421
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.827  0.755
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.466  0.418
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.005  0.015
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.247  0.197
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.499  0.458
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.510  0.466
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.562  0.552
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.562  0.554
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.048  0.131
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.401  0.378
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.587  0.586
```

Our first results show that the method was not able to reduce the performance gap between the cis-location and the trans-location test sets. Although there might be a small improvement in average precision and recall for small objects for trans-locations, the results are very similar as before the domain adaptation method. We will try variations of the method like adding more layers after the transformation to the aligned subspace or extracting the features lower in the network to increase the capacity of the adapted part of the model and try to improve our results.

References

[1] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems* 28 (2015).

[2] Beery, Sara, Grant Van Horn, and Pietro Perona. "Recognition in terra incognita." *Proceedings of the European conference on computer vision (ECCV)*. 2018.

[3] Fernando, Basura, et al. "Unsupervised visual domain adaptation using subspace alignment." *Proceedings of the IEEE international conference on computer vision*. 2013.

[4] Raj, Anant, Vinay P. Namboodiri, and Tinne Tuytelaars. "Subspace alignment based domain adaptation for rcnn detector." *arXiv preprint arXiv:1507.05578* (2015).