Animal detection with Faster R-CNN: Transfer Learning and Domain Adaptation

By Abdiel Fernandez, Rose Guay Hottin, Kevin Lessard, Santino Nanini

Task description

Improve animal **detection** with the *Faster R-CNN* model, *in unseen regions/locations*

https://arxiv.org/pdf/1807.04975.pdf

Detection
(Bounding box)

(No classification)

Faster R-CNN = Faster Region-based Convolutional Neural Network

Task description

Improve animal detection with the *Faster R-CNN* model, in unseen regions/locations

https://arxiv.org/pdf/1807.04975.pdf

Why ?

Detection improvements
(new environments)

can lead to

Classification improvements
(new environments)

Many applications improved

Dataset description

Loc 1

Loc 2

Loc 3

Caltech Camera Trap (CCT) dataset
(~243.000 images from 140 locations)

Specificities about locations :

- Cis locations = locations seen during training
  (10 locations for both valid and test)
- Trans locations = locations not seen during training
  (9 locations for test, 1 for valid)

Dataset description

Loc 1

Loc 2

Loc 3

From :
Caltech Camera Trap (CCT) dataset
(~243.000 images from 140 locations)

we divided :

Training set :
12.099 images

Valid set :
3198 images (cis = 1665 , trans = 1533)

Test set :
30.729 images (cis = 12.696, trans = 18.033)

Problem

Poor generalization in new environments **= Domain shift** problem

Top 5 labels and confidence

(A) **Cow: 0.99**, Pasture: 0.99, Grass: 0.99, No Person: 0.98, Mammal: 0.98

Cows in “common” contexts (e.g. Alpine pastures) are detected and classified correctly

(B) No Person: 0.99, Water: 0.98, Beach: 0.97, Outdoors: 0.97, Seashore: 0.97

Cows in “uncommon”contexts (beach, waves and boat) are **not** detected

(C) No Person: 0.97, **Mammal: 0.96**, Water: 0.94, Beach: 0.94, Two: 0.94

Cows in “uncommon”contexts (beach, waves and boat) are **poorly** detected

Summary

PROBLEM :

"...we find that generalization to new locations is poor, especially for classification systems."

Ref : **Recognition in terra incognita, Beery et.al., arxiv, 2018.**

https://arxiv.org/pdf/1807.04975.pdf

SOLUTION :

Improve : especially for unseen locations/environments (trans locations) to increase model's "adaptability" (reduce gap between cis and trans locations)

Detection
(Bounding box)

(No classification)

Ref : **Recognition in terra incognita, Beery et.al., arxiv, 2018.**

How can we improve the detection ?

- Domain adaptation techniques (Domain space alignment)
- Data augmentation

Baseline model

We used Faster R-CNN

Faster R-CNN is:

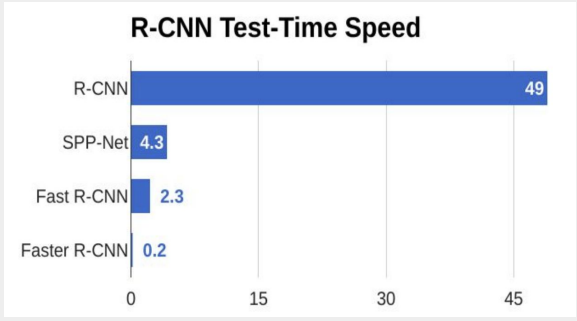
- Way Faster than typical R-CNN and Fast R-CNN
- Really good for detection

Composed of:

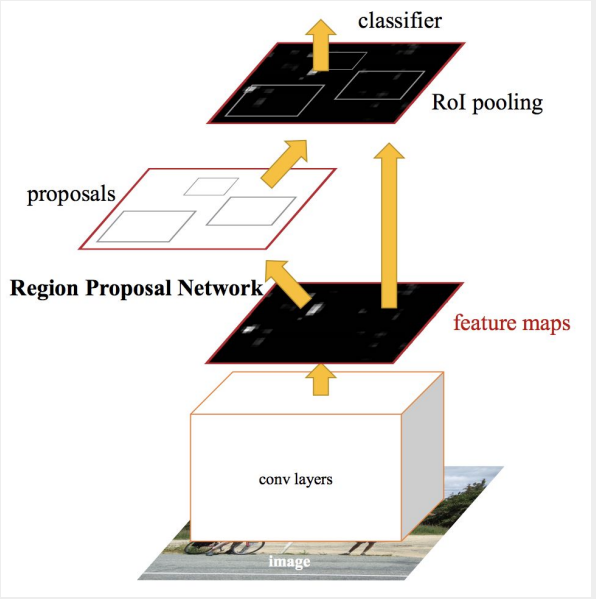
- Some convolution layers
- Region Proposal Network (RPN)
- Region of Interest pooling (ROI)
- Backbone of ResNet-50

Tuned parameters from the paper:

- “Batch size of 1 on the same training subset”
- “SGD with a momentum of 0.9”
- “Learning rate of 0.0003 (decaying)”

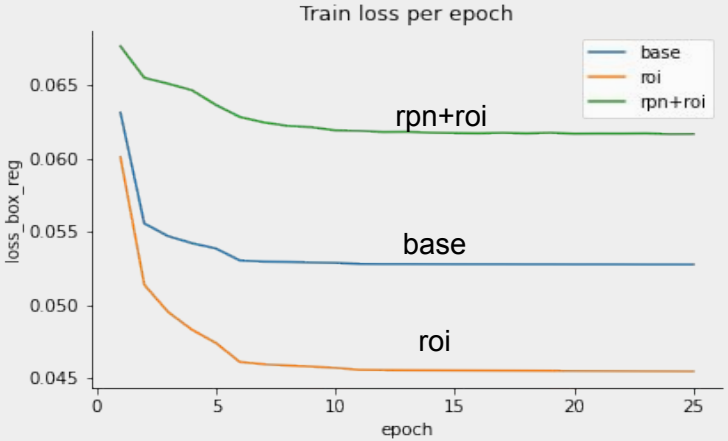


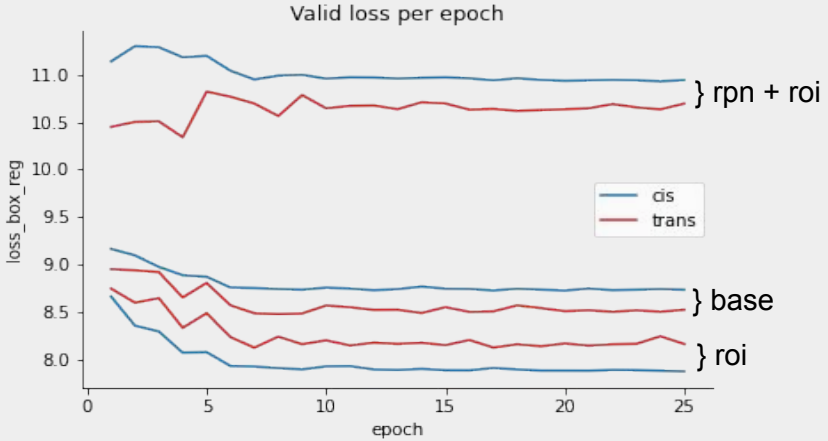
https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e



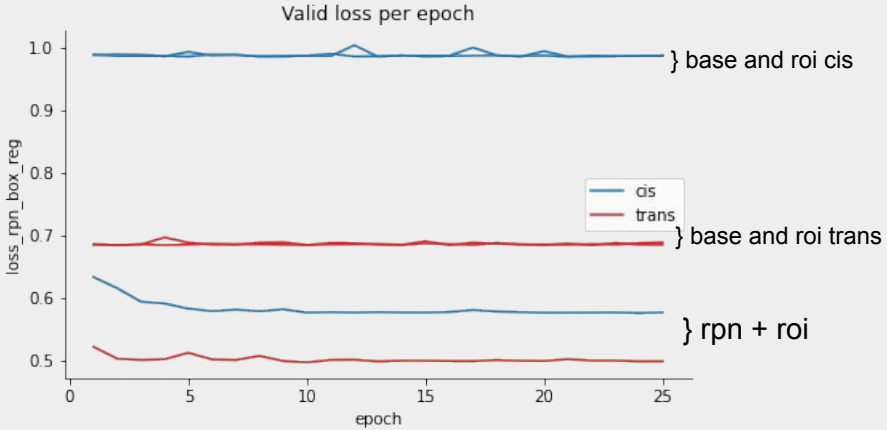
https://arxiv.org/pdf/1506.01497v1.pdf

Training of the different layers

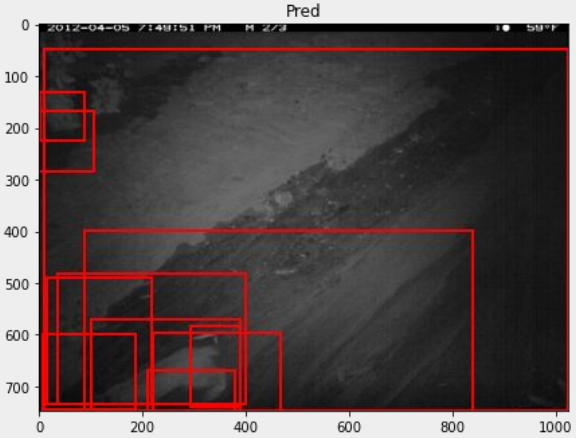




Trans validation has only 1 location on the original splits used in the paper; it may explain why trans in lower than cis.



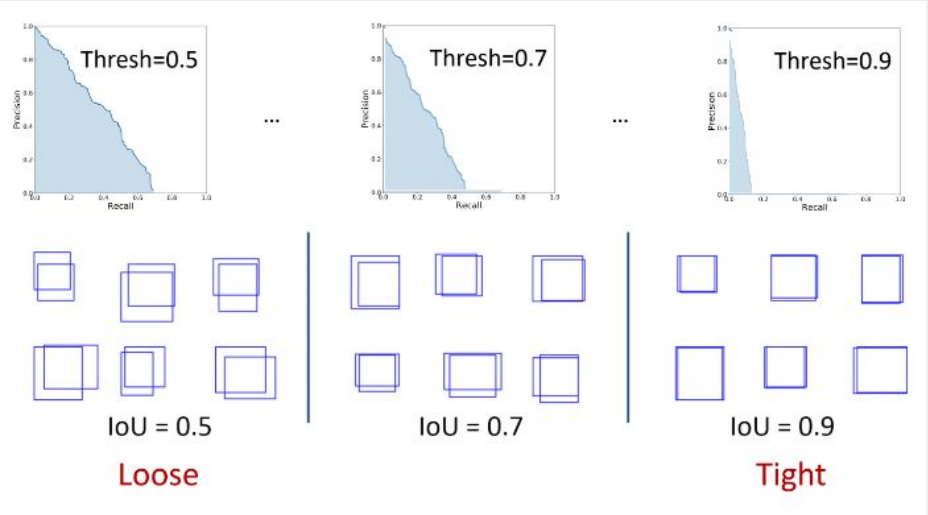
Examples on images of predicted boxes (NMS@0.05)



Evaluation metrics

Coco-evaluator

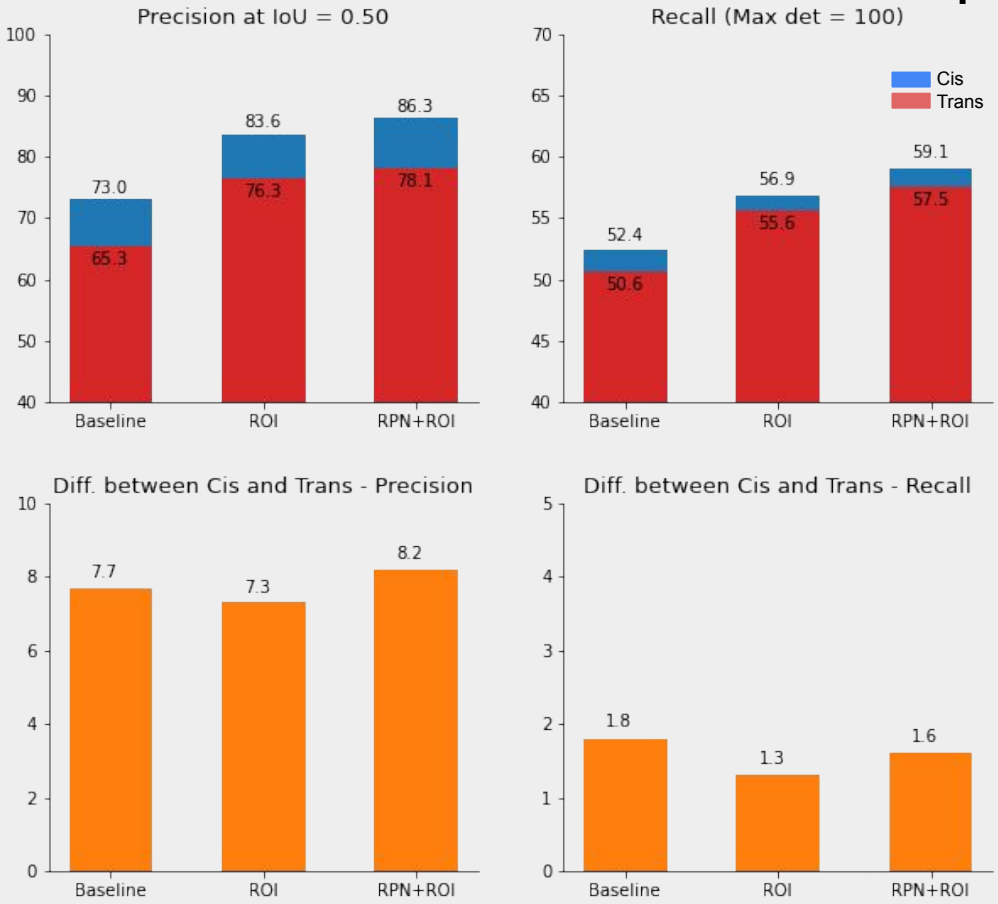
- mean Average Precision (mAP)
- Intersection over Union (IoU)
- Max of 100 predictions
- NMS with IoU=0.35



https://kharshit.github.io/blog/2019/09/20/evaluation-metrics-for-object-detection-and-segmentation

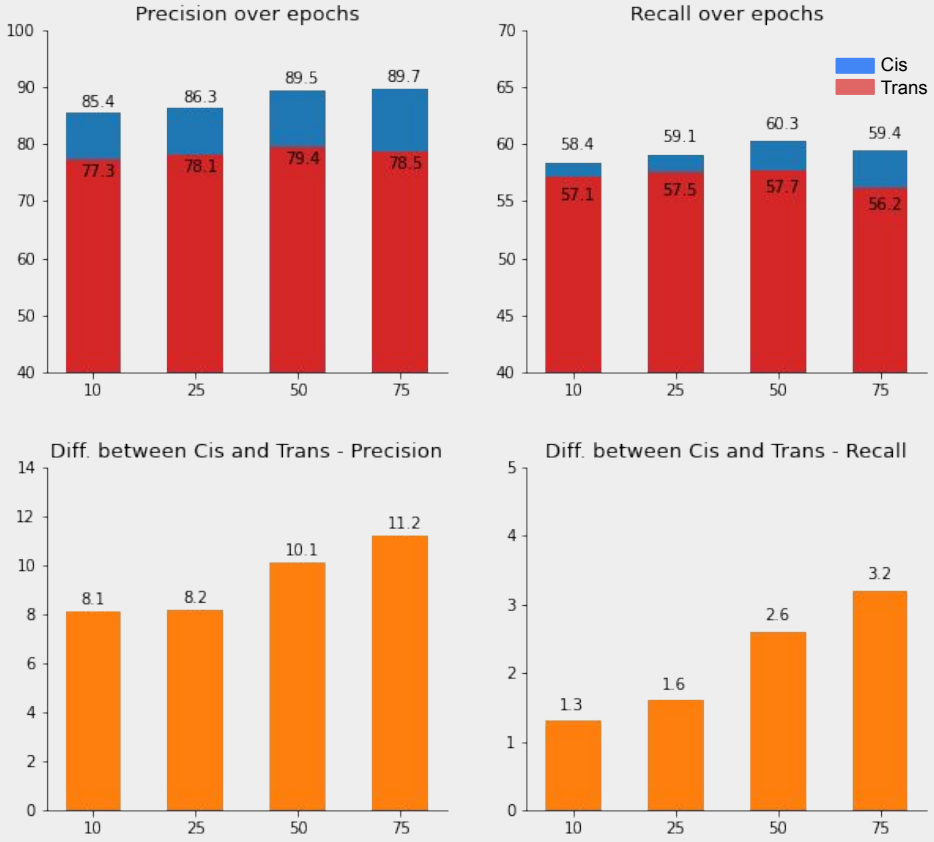
Used the same metrics as the paper for better comparison

Evaluations for different levels of depth



Evaluations for different amount of epochs

(for our new baseline - RPN+ROI trained layers)



- GPU: GTX 1080 ti
- ~23 minutes per epoch
- 1 hour per evaluation

Our new baseline becomes the model with RPN and ROI layers trained on 50 epochs

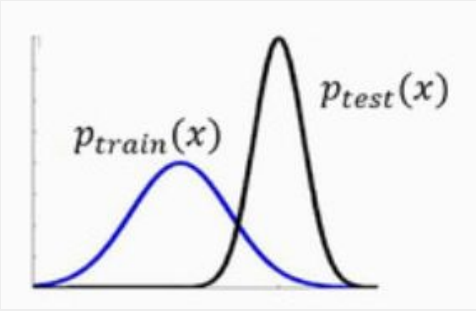
Unsupervised space alignment

Challenges of domain shift

- Source and target data lie in the same D-dimensional space but are drawn according to different marginal distributions (Fernando et al., 2013).
- ~~i.i.d assumption~~
- Performance decreases on out-of-distribution target domain.
- Less work for object detection than classification (Raj et al., 2015).

Cis-Locations		Trans-Locations		Error Increase	
ResNet	Inception	ResNet	Inception	ResNet	Inception
77.10	77.57	70.17	71.37	30%	27.6%

Detection mAP at IoU=0.5; Beery et al. (2019), Table 2.



Unsupervised Domain Adaptation using Subspace Alignment

Based on :

- Fernando, Habrard, Sebban and Tuytelaars (2013)
    - Difference with our implementation: **detection vs classification**.
- Raj, Namboodiri and Tuytelaars (2015)
    - Difference with our implementation: **Faster R-CNN vs Fast R-CNN and class agnostic**.

Subspace generation

NonAdaptedFasterRCNN ← FasterRCNN fine-tuned on Source data

FeatSrc = [], FeatTgt= []

for each image ∈ **Source_Image** :

    obtain representation (*D* =1024) for all proposed regions with **NonAdaptedFasterRCNN**

    for each region, if **IoU with GroundTruth** > 0.5:

        stack representation in FeatSrc matrix

for each image ∈ **Target_Image** :

    obtain representation (*D* = 1024) for all proposed regions with **NonAdaptedFasterRCNN**

    for each region, if **ConfidenceScore** > 0.6:

        stack representation in FeatTgt matrix

Subspace generation and alignment

$X_S \leftarrow$ PCA (FeatSrc,*d*)

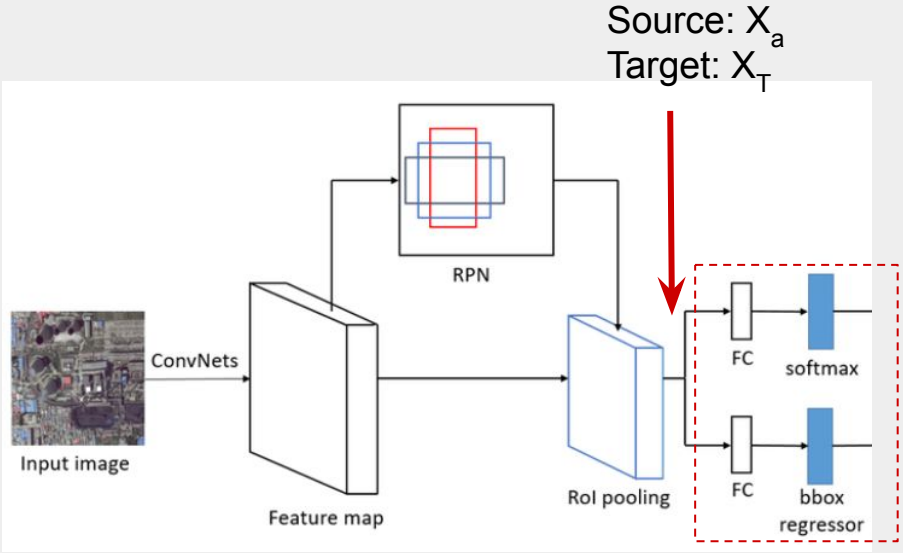
$X_T \leftarrow$ PCA(FeatTgt, *d*)

M* = $argmin_M(\|X_S M - X_T\|_F^2)$ = X_S' X_T

$X_a \leftarrow X_S M$

To project source feature representation (z_s) in target aligned source subspace: $\mathbf{z_s X_a}$

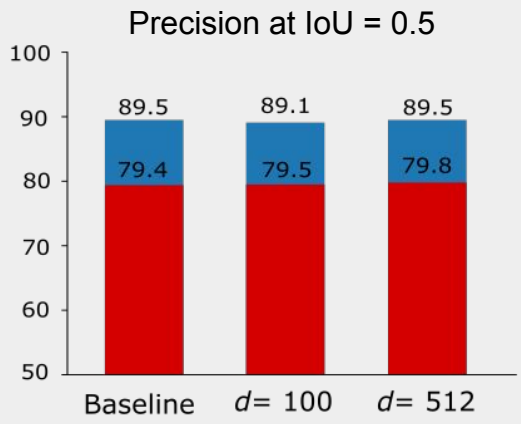
To project target feature representation (z_T) in target subspace: $\mathbf{z_T X_T}$

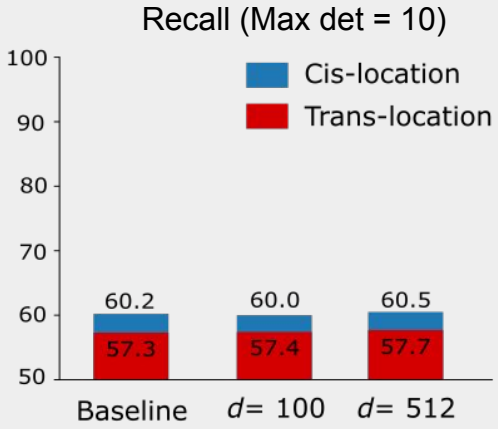


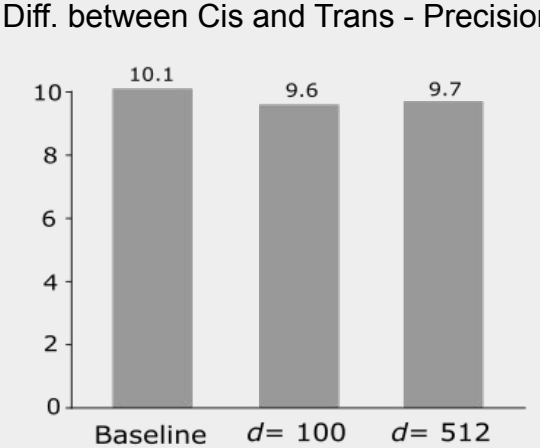
Simplified Faster RCNN architecture, adapted from Zhang and Deng (2019)

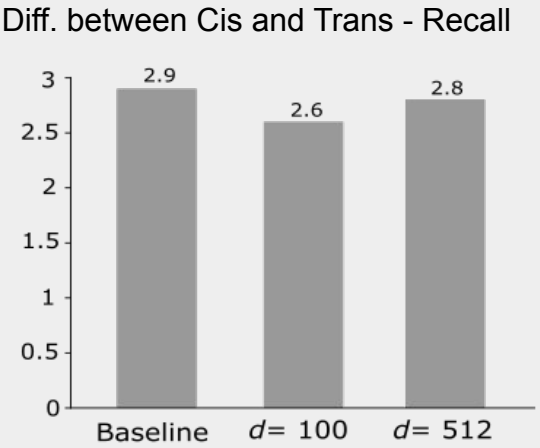
AdaptedFasterRCNN ⟵ Train NonAdaptedFasterRCNN Predictor on Source data projected in target aligned source subspace.

Subspace Alignment - Results









Data Augmentation

Default method: **Data normalization,** is included into the model architecture.
Default values of Mean and Std (by channel) in the model were replaced with the values from our training set.

Two methods (from `torchvision.transforms`)

- HorizontalFlip (p = 0.5)
- ColorTransformation {RandomInvert(p = 0.5), ColorJitter([.2,.3], [0.8,0.9],[.1,0.12]))}

Two training strategies for data augmentation (proposal)

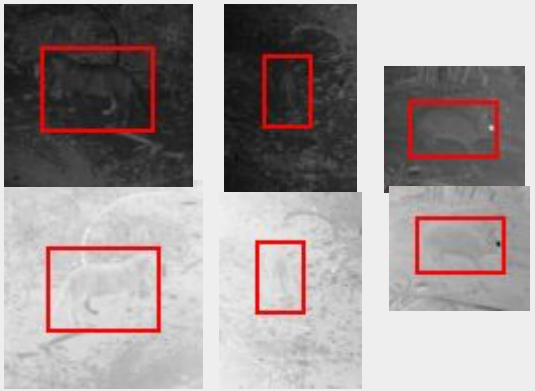
- In-line (both data augmentation methods with p = 0.5.)
  Advantages: *LESS DATA at the same time, LESS TIME consuming.*
  Desadvanages: *We DO NOT show all the possibilities to the model.*

- Off-line (original data + fullTrainingSet*HorizontalFlip + fullTrainingSet*ColorTransformation (we still trying…)
  Advantages: *We show all the possibilities to the model.*
  Desadvanages: *A LOT of data at the same time(+ 36K images), TIME CONSUMING(about 1h per epoch)*

Examples of ColorTransformation effect.

Why this transformation? A lot of image in the dark, as part of the nature of the capture process.

You're right … images don’t see the same thing! But we hope this help.



Conclusion

Summary

For trans locations (unseen locations during training)
(in general)

original paper ("*Recognition in Terra Incognita*") :
Precision = 70 %

Our baseline (RPN + ROI model) :
Precision = **79.4 %**

We have a better precision as the one in the reference paper, **our model adapts itself a bit better for unseen locations, reducing the domain shift problem**

Domain adaptation using subspace alignment

We did not use the class labels to generate subspaces

Could explain why we did not get improvements for trans locations

Data augmentation

There is No improvements on our baseline coming from the transformation applied in-line and solely.

NOTE:
Training results with Off-line and the combined methods are still in progress.

Futures perspectives/ideas

1) Use an intermediate domain generated with CycleGAN to mimic trans locations (to get a kind of intermediate training data that could generalization afterwards). Try it with our data.

   Article reference : “Progressive Domain Adaptation for Object Detection, Hsu et.al., 2020

2) Do the adaptation in a sub-network trained by “gradients” that allow more capacity/flexibility for the adaptation to trans locations

   Article reference : “Strong-Weak Distribution Alignment for Adaptive Object Detection, Saito et.al., 2019

References

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

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

Hsu, Han-Kai, et al. "Progressive domain adaptation for object detection." *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. (2020).

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

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

Saito, Kuniaki, et al. "Strong-weak distribution alignment for adaptive object detection." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. (2019).

Thank you!

Questions ?