Projet 7 OC : Développez une preuve de concept

Présenté par : NAMA NYAM Guy Anthony

Mentor : Julien Hendrick

05 Mai 2020

OPENCLASSROOMS

Sommaire

Contexte

- Le domaine du machine learning, et plus généralement de la data science évolue très rapidement.
- Il est donc important de rester au courant des avancées en effectuant ***une veille thématique***
- Cette dernière consiste à ***monter rapidement en compétences*** sur une nouvelle thématique et mettre en pratique ***un nouvel algorithme de façon autonome(POC)***.
- La *POC* commence par un tour d'horizon de l'état de l'art et chute sur un plan prévisionnel comprenant :
  - l'algorithme envisagé
  - le dataset sur lequel vous pensez évaluer les performances
  - Une méthode baseline
  - Arguments justifiant votre choix
  - Références bibliographiques

Thématique

- Le travail réalisé porte sur l'état de l'art des modèles de vision par ordinateur, plus précisément la classification d'images
- Les performances d'un algorithme dépend à la fois de l'architecture CNN mais aussi la stratégie de fine-tuning adoptée.
- Nous tirons pleinement profit de nouveaux modèles et une méthode de fine-tuning pendant l'étape de la data augmentation.
  1. Les poids pré-entraînés des modèles ResNext-101.
  2. La méthode FixRes pour le fine-tuning des hyper-paramètres.

Sommaire

Best model in image classification on imagenet

- Top-1 accuracy

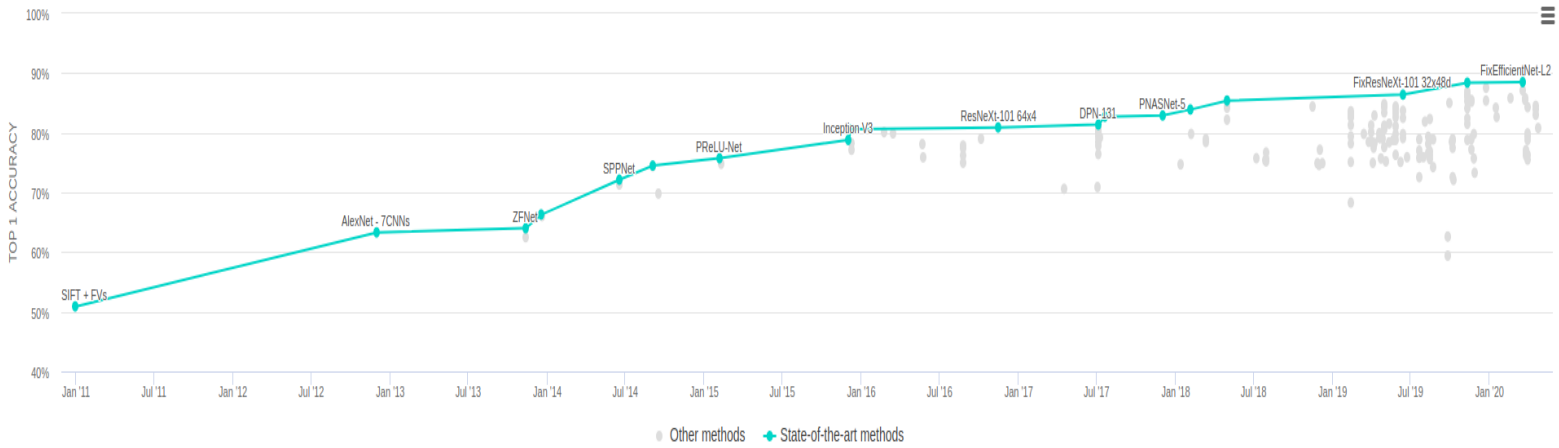


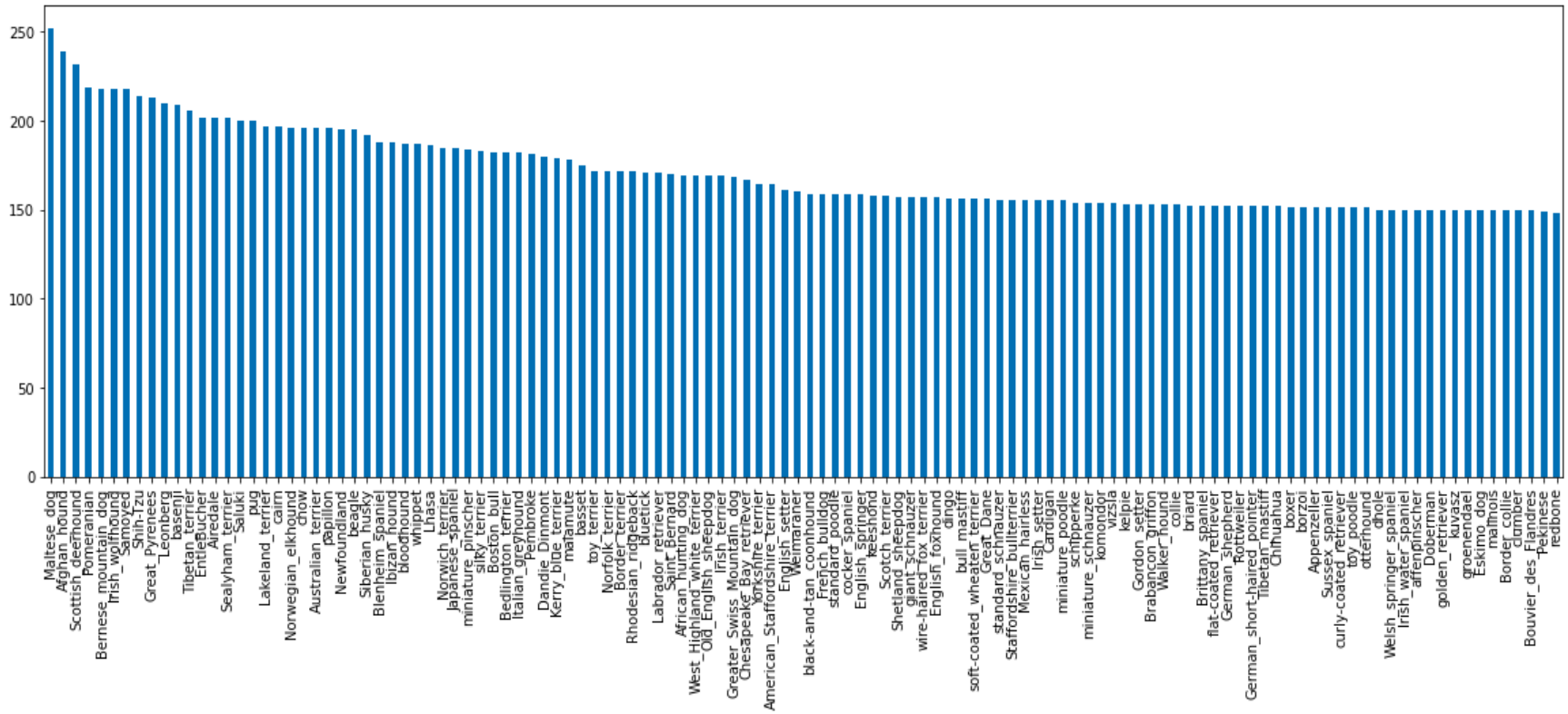
FIGURE – Image Classification on ImageNet. - source : paperswithcode

- ***FixEfficientNet-L2*** avec un ***top-1 accuracy de 88.5%*** pour ***480 millions de paramètres***
- Les poids pré-entraînés du modèle ***ResNeXt-101 32x48d*** (release) seront utilisés, modèle de ***top-1 accuracy de 85.4%*** et ***829 millions de paramètres***

Sommaire

Description - dataset Stanford Dogs

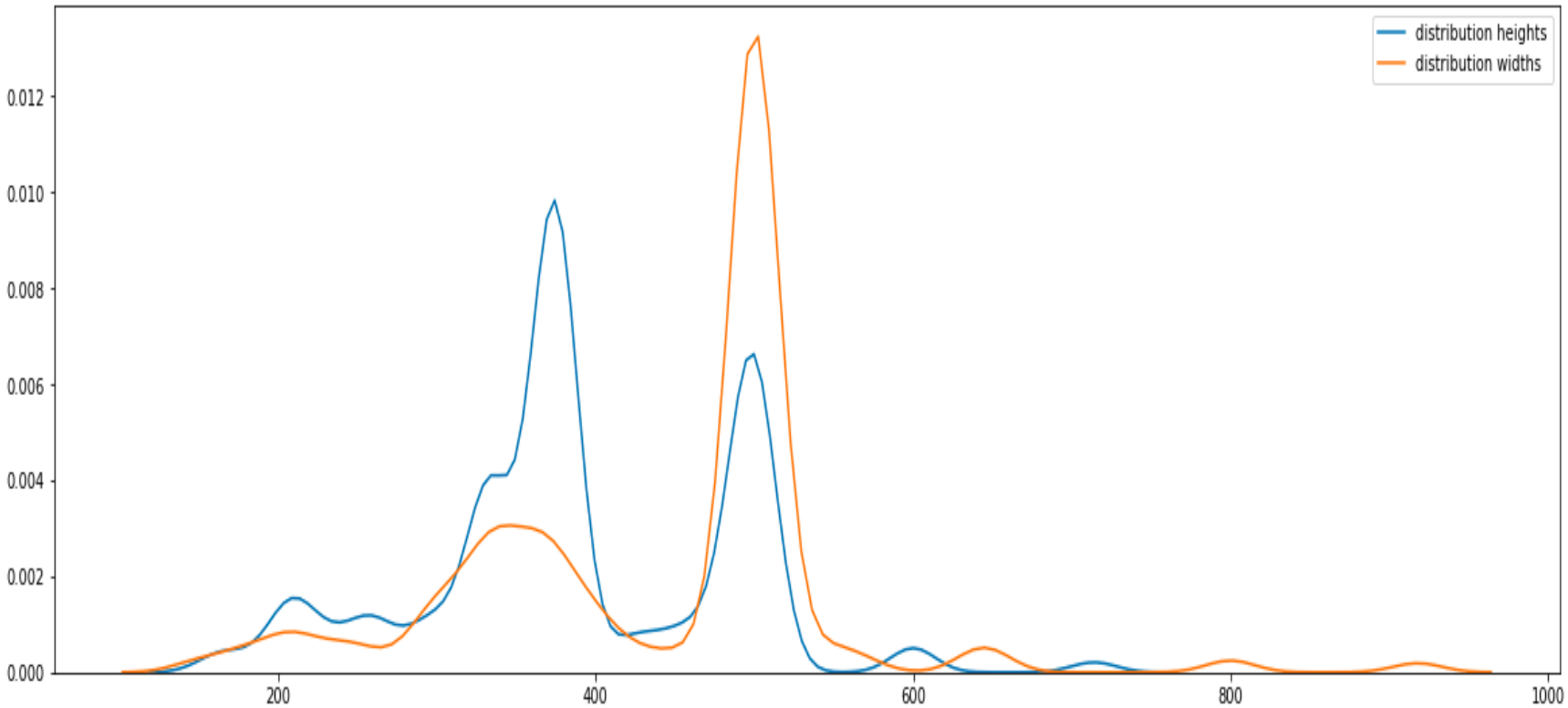
- Jeu de données avec 20580 images pour 120 classes dont la figure ci-dessous donne la distribution.



- *training set*, *validation set* et *test set* identique à la méthode baseline.

Description - dataset Stanford Dogs

- Il peut être tout aussi important d'exploiter la distribution de la taille des images.



- Largeur moyenne : 387 pixels ; hauteur moyenne : 439 pixels

Sommaire

Principe FixRes

- ***FixRes*** est une simple méthode pour corriger l'écart de résolution train-test.

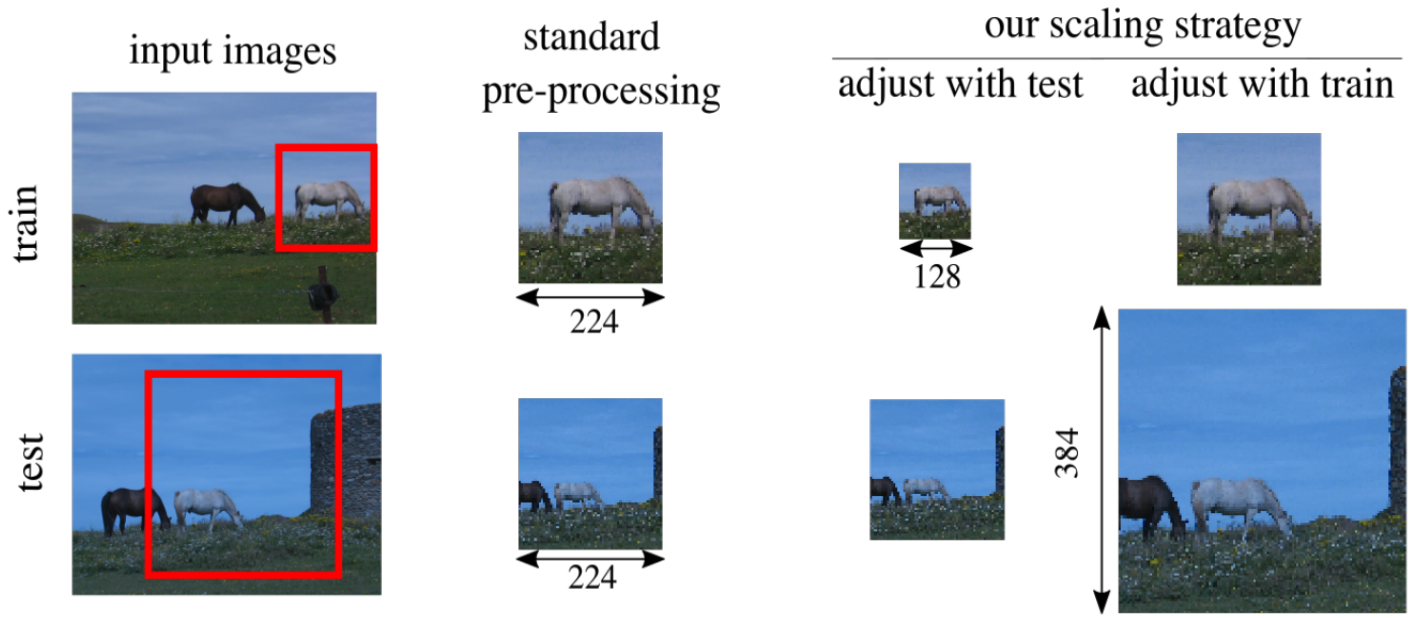


FIGURE – FixRes principe. - source : facebookresearch/FixRes

- Réduire l'invariance d'échelle pour les objets similaires.

FixRes : tranforms

- Une implémentation libre de la méthode(***transforms_v2***) est disponible à https://github.com/facebookresearch/FixRes.

```
fixRes_image_transforms = get_transforms(input_size=320,test_size=380, kind='full', crop=True, need=('train', 'val'), backbone=None)
```

- Les différentes transformations utilisent un unique recadrage(crop)
- Utiliser la transformation *train* pour *training set* et *val_test* pour *validation set* et *test set*.

```
data_images = {
    'train': datasets.ImageFolder(root=dir_train, transform=fixRes_image_transforms['train']),
    'valid': datasets.ImageFolder(root=dir_valid, transform=fixRes_image_transforms['val_test'])
    'test': datasets.ImageFolder(root=dir_test, transform=fixRes_image_transforms['val_test'])
}
```

Sommaire

Résultats et discussions

- La table ci-dessous résume l'ensemble des résultats obtenus

Train_size	***Test_size***	***Accuracy***
128	224	
224	128	89.30%
224	224	93.21%
224	320	93.48%
224	400	
320	224	94,73%
320	320	94.40%
320	380	CUDA out of memory

- ***Performance :*** première appréciation ***gain net*** par rapport à la méthode baseline de ***3.7%***
- ***Résolution :*** *Train_size* ou *input_size* à 320 et *test_size* semble meilleure pour ce dataset.

Résultats et discussions

- ***Complexité de mise en œuvre :*** la mise en œuvre de la méthode et l'algorithme d'apprentissage est simple.
- ***Temps de calcul :*** le temps reste considérable sur un seul GPU et évolue faiblement avec la résolution des images d'entraînements et de tests. Temps évalué entre 4 et 6 heures en moyenne.
- ***Framework Pytorch :*** Je dirais que *Pytorch* est moins abstractif que *Keras* mais plus puissant pour le fine-tuning des méthodes. Je pense également que les temps d'exécutions d'entraînement et d'évaluation sont similaires.

Sommaire

Conclusion

- Nous pouvons le concept théorique initial s'avère avoir un potentiel en pratique.
- Nous avons pu constater que les meilleurs modèles dépendent à la fois de leur architecture et la phase de pré-traitement.
- Nous avons atteint *top-1 accuracy* de *94.73%* sur le dataset *Stanford Dogs*
- L'atteinte de ce résultat nécessite l'usage de la méthode FixRes qui réduit l'invariance d'échelle *train-test*.
- L'usage du Framework PyTorch apporte une plus grande flexibilité dans la phase de preprocessing.

Perspectives

- Avec plus de ressources de calculs, Étendre l'espace de recherche de résolution *train-test*
- Utiliser les poids pré-entraînés des modèles de meilleur *top-1 accuracy* que *ResNeXt-101 32x48d*
- Utiliser la transformation *TenCrop* qui augmente le nombre de recadrage(*Crop*) à dix au lieu d'un seul.