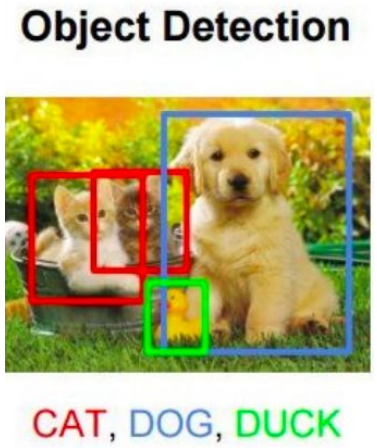
Projet d’apprentissage semi-supervisé

Détection d’animaux

Sommaire

I. Introduction



- Sujet : Détection d'objet par boîte englobante d'animaux
  - Obtenir la boîte englobante (x, y, w, h) et la classe de l'animal
- Outils :
  - Dataset avec 50 000 images
  - 10 classes d'animaux différentes [hamster, cochon d'inde, guépard, jaguar, chat, chimpanzé, lynx, loup, orang outan, coyote]
- Difficultés :
  - Images très pixellisées
  - Certaines classes d'animaux se ressemblent beaucoup
  - Images à annoter à la main (Beaucoup trop!)
- Etapes :
  - Méthode supervisée
  - Méthode semi supervisée

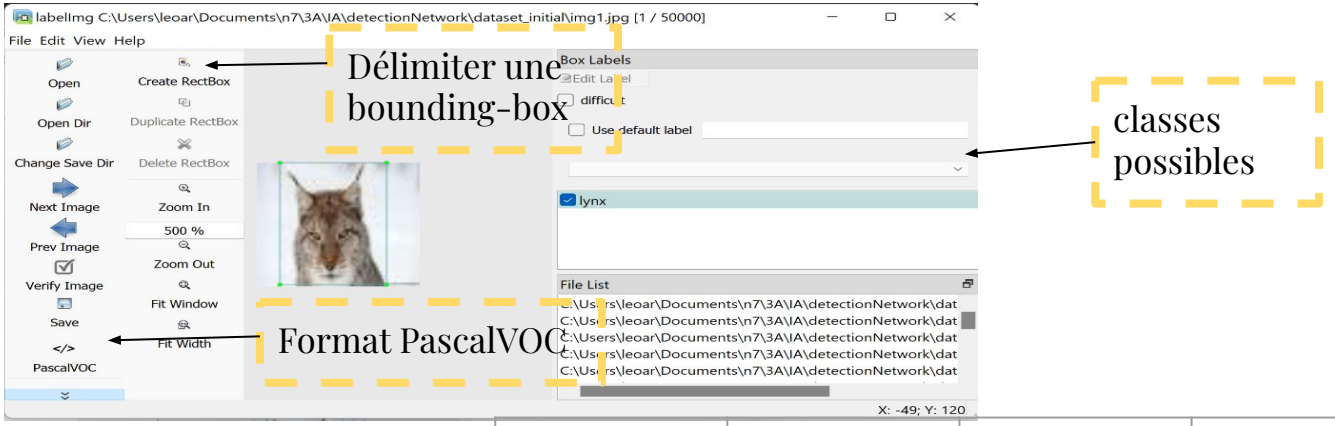
https://github.com/tzutalin/labelImg

II. Annotations

1. Labellisation

1200 images labellisées !

```
python3 labelImg.py [IMAGE_PATH] [PRE-DEFINED CLASS FILE]
```



convert_PVOC_to_yolo.py

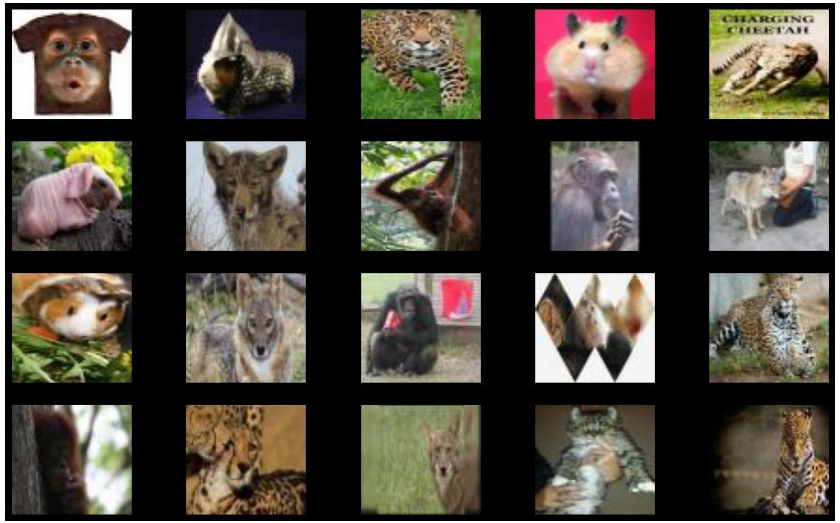
Classe	xc	yc	width	height
5	0.4921875	0.5078125	0.984375	0.953125

II. Annotations

2. Robotflow

- Hebergement les données
- Créer un dataset (hiérarchie des dossiers)
- Gros avantage importation simple des données

```
rf = Roboflow(api_key="[redacted]")
project = rf.workspace().project("projet-ia")
dataset = project.version(1).download("yolov5")
```



III. Modèle supervisé

1. Quelle architecture de réseau de neurones utiliser pour **la détection** ?

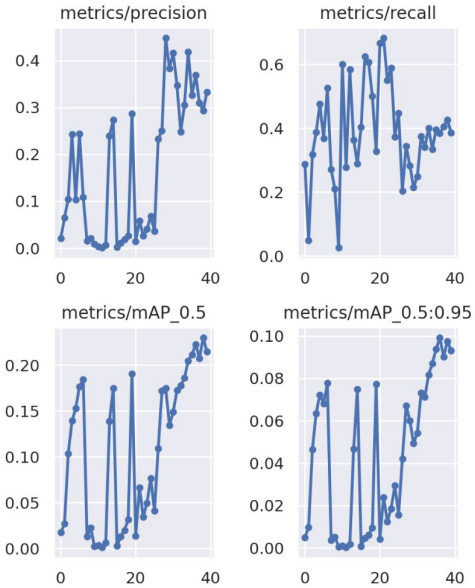
Model	mAP (%)	FPS		
		TeslaT4	*1660 Ti*	*Jetson Nano*
YOLOv3	**54.3**	80	21	8
YOLOv5s	37.6	**100**	**28**	**15**
MobileNet-SSD V2	33.7	94	26	**15**

- Conclusion sur la performance des réseaux:
  - Accuracy : Yolov3 > Yolov5s > SSD
  - Rapidité : Yolov5s > SSD > Yolov3
  - + Récent : Yolov5s > Yolov3 > SSD

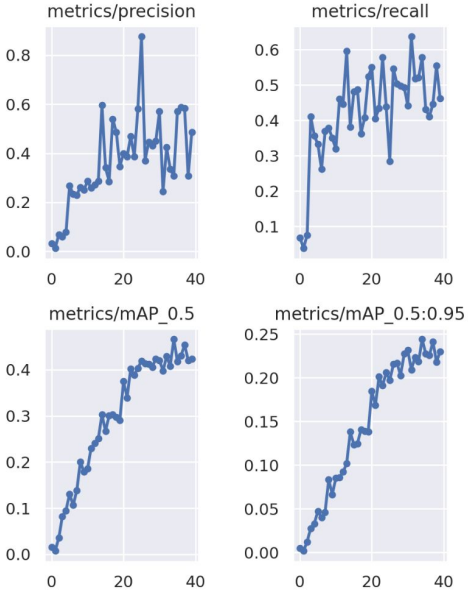
} Yolov3, Yolov5 ?

III. Modèle supervisé

1. Quelle architecture de réseaux de neurones utiliser pour **la détection** ?



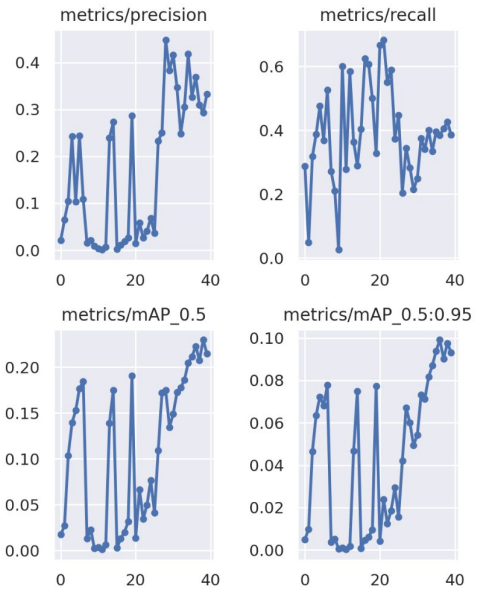
Yolov3



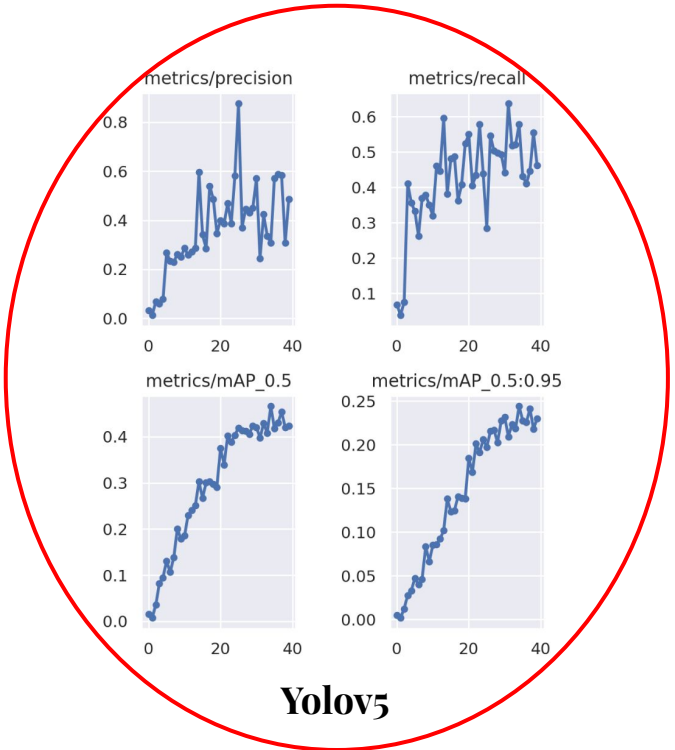
Yolov5

III. Modèle supervisé

1. Quelle architecture de réseaux de neurones utiliser pour **la détection** ?



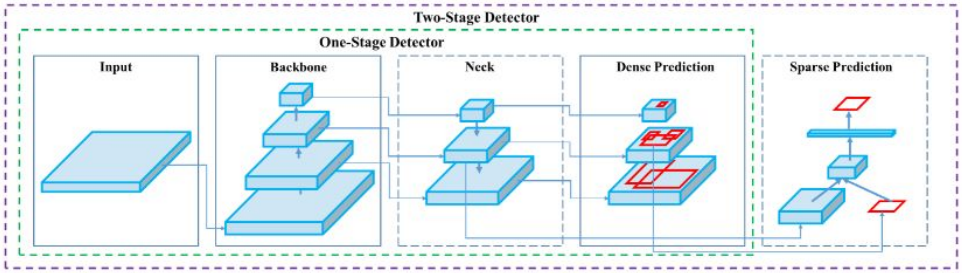
Yolov3



Yolov5

III. Modèle supervisé

2. Architecture de Yolov5



Input: { Image, Patches, Image Pyramid, … }

Backbone: { VGG16 [68], ResNet-50 [26], ResNeXt-101 [86], Darknet53 [63], … }

Neck: { FPN [44], PANet [49], Bi-FPN [77], … }

Head:

Dense Prediction: { RPN [64], YOLO [61, 62, 63], SSD [50], RetinaNet [45], FCOS [78], … }

Sparse Prediction: { Faster R-CNN [64], R-FCN [9], … }

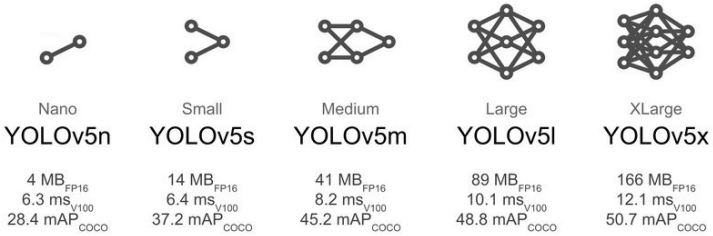
```
  0          -1  1      3520  models.common.Conv                    [3, 32, 6, 2, 2]
  1          -1  1     18560  models.common.Conv                    [32, 64, 3, 2]
  2          -1  1     18816  models.common.C3                      [64, 64, 1]
  3          -1  1     73984  models.common.Conv                    [64, 128, 3, 2]
  4          -1  2    115712  models.common.C3                      [128, 128, 2]
  5          -1  1    295424  models.common.Conv                    [128, 256, 3, 2]
  6          -1  3    625152  models.common.C3                      [256, 256, 3]
  7          -1  1   1180672  models.common.Conv                    [256, 512, 3, 2]
  8          -1  1   1182720  models.common.C3                      [512, 512, 1]
  9          -1  1    656896  models.common.SPPF                    [512, 512, 5]
 10          -1  1    131584  models.common.Conv                    [512, 256, 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    361984  models.common.C3                      [512, 256, 1, False]
 14          -1  1     33024  models.common.Conv                    [256, 128, 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     90880  models.common.C3                      [256, 128, 1, False]
 18          -1  1    147712  models.common.Conv                    [128, 128, 3, 2]
 19      [-1, 14]  1         0  models.common.Concat                  [1]
 20          -1  1    296448  models.common.C3                      [256, 256, 1, False]
 21          -1  1    590336  models.common.Conv                    [256, 256, 3, 2]
 22      [-1, 10]  1         0  models.common.Concat                  [1]
 23          -1  1   1182720  models.common.C3                      [512, 512, 1, False]
 24    [17, 20, 23]  1     35061  models.yolo.Detect                    [8, [[10, 13, 16, 30, 33, 23], [30, 61, 62, 45, 59, 119],
[116, 90, 156, 198, 373, 326]], [128, 256, 512]]
```

Model Summary: 270 layers, 7041205 parameters, 7041205 gradients, 15.9 GFLOPs

https://github.com/ultralytics/yolov5

III. Modèle supervisé

2. Architecture de Yolov5



Yolov5n sur dataset sans augmentation de données

	metrics/precision	metrics/recall	metrics/mAP_0.5	metrics/mAP_0.5:0.95
count	40.000000	40.000000	40.000000	40.000000
mean	0.327655	0.365876	0.220188	0.102420
std	0.108199	0.124732	0.107349	0.057499
min	0.020937	0.036118	0.007938	0.002257
25%	0.268373	0.283242	0.136635	0.051715
50%	0.347655	0.404820	0.262300	0.119835
75%	0.389143	0.449782	0.310615	0.157095
max	0.544430	0.549290	0.345690	0.171810

Yolov5s

	metrics/precision	metrics/recall	metrics/mAP_0.5	metrics/mAP_0.5:0.95
count	70.000000	70.000000	70.000000	70.000000
mean	0.331811	0.439746	0.264273	0.124017
std	0.098899	0.101768	0.098375	0.054416
min	0.020761	0.027545	0.011898	0.003048
25%	0.274272	0.405672	0.207590	0.090602
50%	0.363945	0.453390	0.284375	0.132305
75%	0.402757	0.499335	0.350465	0.173705
max	0.499950	0.680190	0.378990	0.192490

Yolov5m

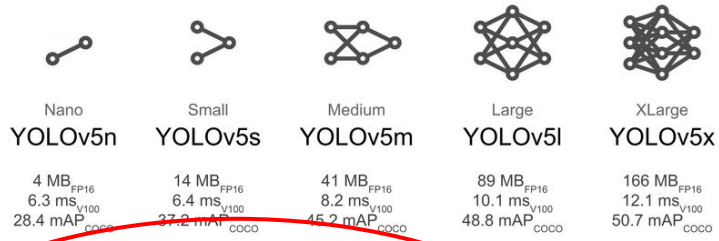
	metrics/precision	metrics/recall	metrics/mAP_0.5	metrics/mAP_0.5:0.95
count	40.000000	40.000000	40.000000	40.000000
mean	0.118601	0.301396	0.063517	0.021660
std	0.088886	0.114355	0.041027	0.017767
min	0.002943	0.080265	0.002758	0.000576
25%	0.041110	0.198870	0.037900	0.009227
50%	0.107205	0.346165	0.062297	0.018257
75%	0.192570	0.375920	0.088397	0.031453
max	0.303280	0.519500	0.194250	0.088579

Yolov5x

	metrics/precision	metrics/recall	metrics/mAP_0.5	metrics/mAP_0.5:0.95
count	40.000000	40.000000	40.000000	40.000000
mean	0.030153	0.448207	0.018139	0.005594
std	0.044927	0.161762	0.018048	0.006252
min	0.002960	0.044059	0.002546	0.000783
25%	0.011323	0.335715	0.009336	0.002871
50%	0.015212	0.507955	0.013082	0.003780
75%	0.021034	0.577562	0.018516	0.005630
max	0.197450	0.647690	0.103440	0.036297

III. Modèle supervisé

2. Architecture de Yolov5



Yolov5n sur dataset sans augmentation de données

	metrics/precision	metrics/recall	metrics/mAP_0.5	metrics/mAP_0.5:0.95
count	40.000000	40.000000	40.000000	40.000000
mean	0.327655	0.365876	0.220188	0.102420
std	0.108199	0.124732	0.107349	0.057499
min	0.020937	0.036118	0.007938	0.002257
25%	0.268373	0.283242	0.136635	0.051715
50%	0.347655	0.404820	0.262300	0.119835
75%	0.389143	0.449782	0.310615	0.157095
max	0.544430	0.549290	0.345690	0.171810

Yolov5s

	metrics/precision	metrics/recall	metrics/mAP_0.5	metrics/mAP_0.5:0.95
count	70.000000	70.000000	70.000000	70.000000
mean	0.331811	0.439746	0.264273	0.124017
std	0.098899	0.101768	0.098375	0.054416
min	0.020761	0.027545	0.011898	0.003048
25%	0.274272	0.405672	0.207590	0.090602
50%	0.363945	0.453390	0.284375	0.132305
75%	0.402757	0.499335	0.350465	0.173705
max	0.499950	0.680190	0.378990	0.192490

Yolov5m

	metrics/precision	metrics/recall	metrics/mAP_0.5	metrics/mAP_0.5:0.95
count	40.000000	40.000000	40.000000	40.000000
mean	0.118601	0.301396	0.063517	0.021660
std	0.088886	0.114355	0.041027	0.017767
min	0.002943	0.080265	0.002758	0.000576
25%	0.041110	0.198870	0.037900	0.009227
50%	0.107205	0.346165	0.062297	0.018257
75%	0.192570	0.375920	0.088397	0.031453
max	0.303280	0.519500	0.194250	0.088579

Yolov5x

	metrics/precision	metrics/recall	metrics/mAP_0.5	metrics/mAP_0.5:0.95
count	40.000000	40.000000	40.000000	40.000000
mean	0.030153	0.448207	0.018139	0.005594
std	0.044927	0.161762	0.018048	0.006252
min	0.002960	0.044059	0.002546	0.000783
25%	0.011323	0.335715	0.009336	0.002871
50%	0.015212	0.507955	0.013082	0.003780
75%	0.021034	0.577562	0.018516	0.005630
max	0.197450	0.647690	0.103440	0.036297

Métriques

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

TP = True positive

TN = True negative

FP = False positive

FN = False negative

The general definition for the Average Precision (AP) is finding the area under the precision-recall curve above

$$\text{AP} = \int_0^1 p(r)dr$$

III. Modèle supervisé

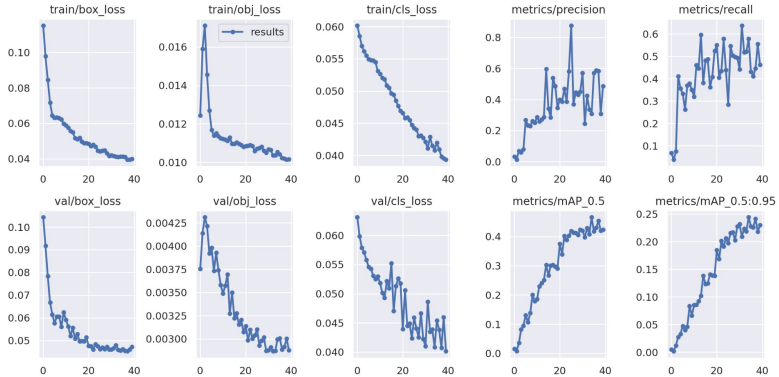
1200 données labellisées !

3. Entraînements et métriques

Entraînement **sans** augmentation de données

```
# Train YOLOv5s on ANIMALS DATASET for 40 epochs
!python train.py --img 64 --batch 8 --epochs 40 --data /content/detectionNetwork/yolov5/Projet-IA-1/data.yaml --weights yolov5s.pt
```

Métriques



	metrics/precision	metrics/recall	metrics/mAP_0.5	metrics/mAP_0.5:0.95
count	40.000000	40.000000	40.000000	40.000000
mean	0.360152	0.420681	0.297729	0.145385
std	0.179933	0.134756	0.137160	0.078336
min	0.013184	0.038409	0.007488	0.001903
25%	0.260390	0.367360	0.196990	0.084771
50%	0.343780	0.440040	0.321390	0.154650
75%	0.473397	0.506050	0.414435	0.217287
max	0.875160	0.636700	0.466220	0.244230

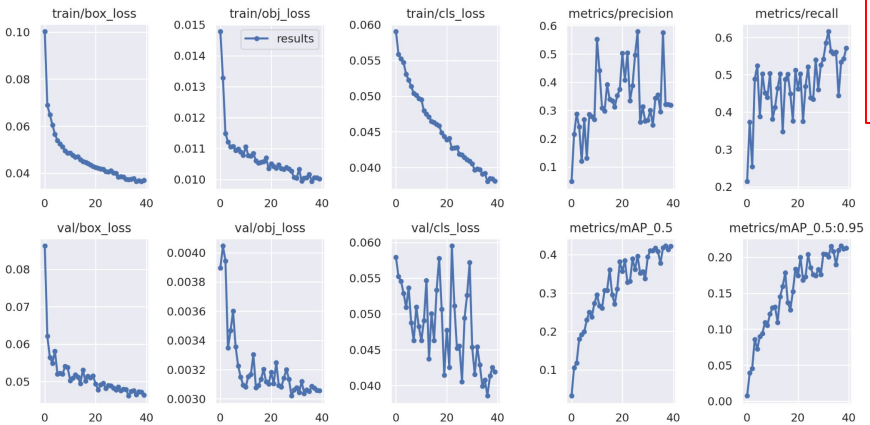
III. Modèle supervisé

3. Entraînements et métriques

Entraînement **avec** augmentation de données (Flip and Rotate 90°)

```
# Train YOLOv5s on ANIMALS DATASET for 40 epochs
!python train.py --img 64 --batch 8 --epochs 40 --data /content/detectionNetwork/yolov5/Projet-IA-4/data.yaml
```

Métriques



Entraînement précèdent

mean	0.360152	0.420681	0.297729	0.145385

	metrics/precision	metrics/recall	metrics/mAP_0.5	metrics/mAP_0.5:0.95
count	40.000000	40.000000	40.000000	40.000000
mean	0.331159	0.470459	0.313921	0.152733
std	0.114618	0.084971	0.095753	0.054088
min	0.048496	0.214250	0.032593	0.007453
25%	0.267555	0.437280	0.265232	0.118325
50%	0.316075	0.488115	0.333980	0.173510
75%	0.378410	0.528440	0.390257	0.199990
max	0.579730	0.615830	0.422630	0.215740

III. Modèle supervisé

4. Transfert Learning et métriques

Sans augmentation de données

Entraînement précèdent

mean	0.360152	0.420681	0.297729	0.145385

	metrics/precision	metrics/recall	metrics/mAP_0.5	metrics/mAP_0.5:0.95
count	60.000000	60.000000	60.000000	60.000000
mean	0.472389	0.543565	0.496698	0.271417
std	0.164020	0.086081	0.010753	0.011298
min	0.266930	0.318770	0.470980	0.235130
25%	0.324650	0.486170	0.492225	0.265207
50%	0.502320	0.531030	0.496855	0.275360
75%	0.569172	0.621258	0.502107	0.278383
max	0.917080	0.659640	0.522400	0.287320

Avec augmentation de données

Entraînement précèdent

mean	0.331159	0.470459	0.313921	0.152733

	metrics/precision	metrics/recall	metrics/mAP_0.5	metrics/mAP_0.5:0.95
count	60.000000	60.000000	60.000000	60.000000
mean	0.612493	0.408656	0.391823	0.203770
std	0.073271	0.042352	0.012378	0.009050
min	0.377880	0.358510	0.366410	0.188420
25%	0.602680	0.387193	0.383430	0.197527
50%	0.630345	0.394345	0.390305	0.202900
75%	0.640705	0.405980	0.399300	0.208230
max	0.739090	0.564870	0.423900	0.227080

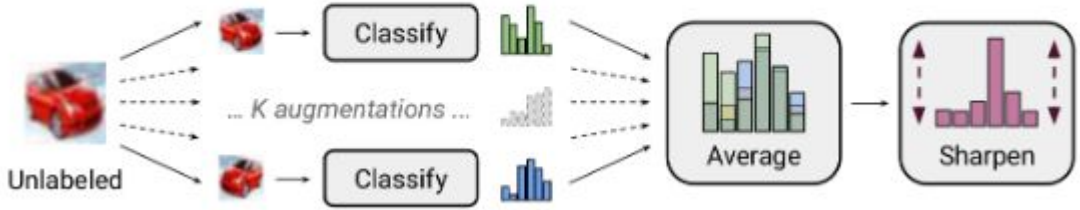
III. Modèle supervisé

5. Prédictions sur des données de tests

IV. Modèle semi supervisé

1. MixMatch

Cas de la classification :

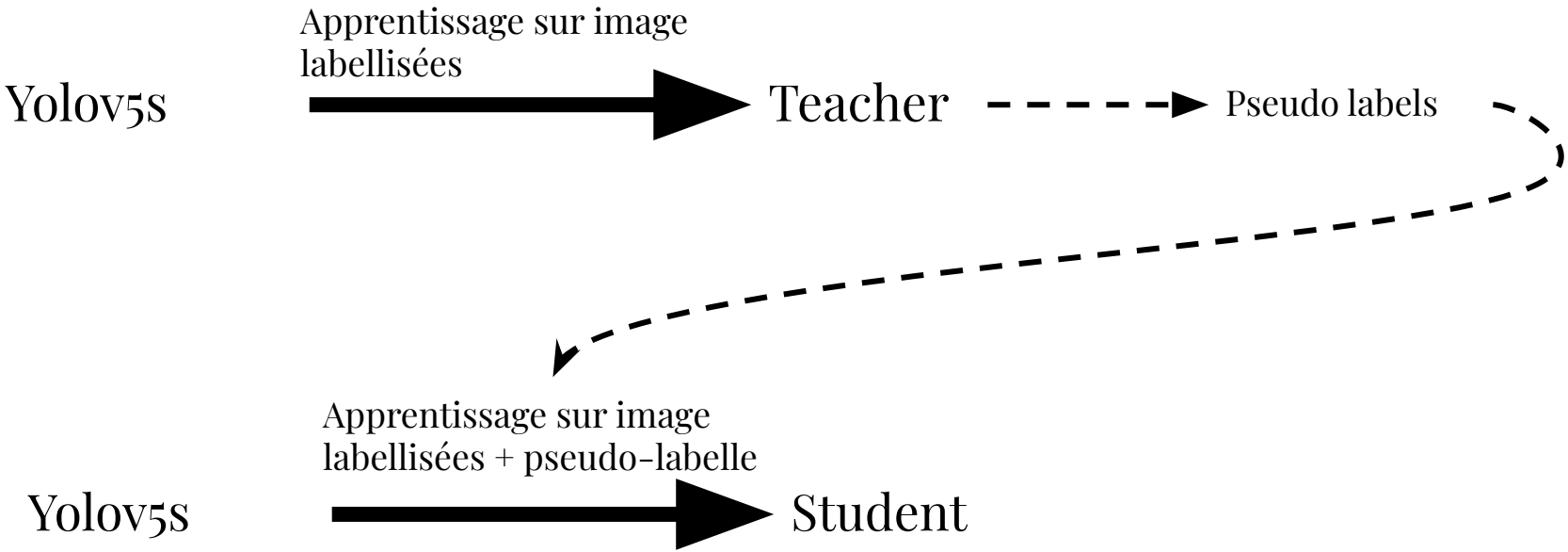


Difficulté avec la détection:

- pas toujours de prédiction
- nécessité d'effectuer une transformation inverse pour moyenner les bounding box

IV. Modèle semi supervisé

2. Teacher-Student



IV. Modèle semi supervisé

2. Teacher-Student

Teacher

	metrics/precision	metrics/recall	metrics/mAP_0.5	metrics/mAP_0.5:0.95
count	40.000000	40.000000	40.000000	40.000000
mean	0.327655	0.365876	0.220188	0.102420
std	0.108199	0.124732	0.107349	0.057499
min	0.020937	0.036118	0.007938	0.002257
25%	0.268373	0.283242	0.136635	0.051715
50%	0.347655	0.404820	0.262300	0.119835
75%	0.389143	0.449782	0.310615	0.157095
max	0.544430	0.549290	0.345690	0.171810

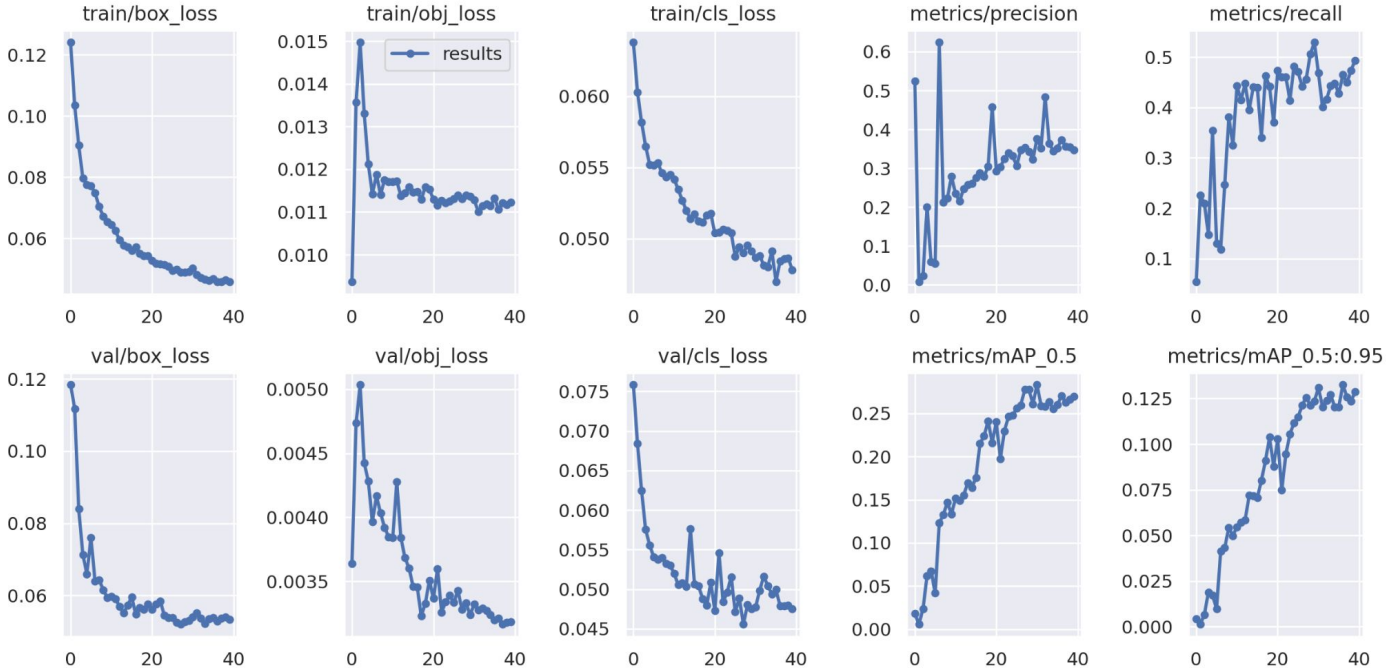
Student

	metrics/precision	metrics/recall	metrics/mAP_0.5	metrics/mAP_0.5:0.95
count	40.000000	40.000000	40.000000	40.000000
mean	0.300003	0.389194	0.193817	0.083575
std	0.121746	0.117200	0.082252	0.041802
min	0.007887	0.053662	0.005872	0.001319
25%	0.255352	0.366090	0.148528	0.054569
50%	0.314610	0.441045	0.226595	0.092661
75%	0.352492	0.461602	0.259543	0.121327
max	0.624100	0.529480	0.283070	0.132560

+483 pseudo labels avec un seuil de confiance de 0.7 parmis 6k images non labellisées

IV. Modèle semi supervisé

2. Teacher-Student



IV. Modèle semi supervisé

2. Teacher-Student

Student 40 epochs

	metrics/precision	metrics/recall	metrics/mAP_0.5	metrics/mAP_0.5:0.95
count	40.000000	40.000000	40.000000	40.000000
mean	0.300003	0.389194	0.193817	0.083575
std	0.121746	0.117200	0.082252	0.041802
min	0.007887	0.053662	0.005872	0.001319
25%	0.255352	0.366090	0.148528	0.054569
50%	0.314610	0.441045	0.226595	0.092661
75%	0.352492	0.461602	0.259543	0.121327
max	0.624100	0.529480	0.283070	0.132560

Student 60 epoch

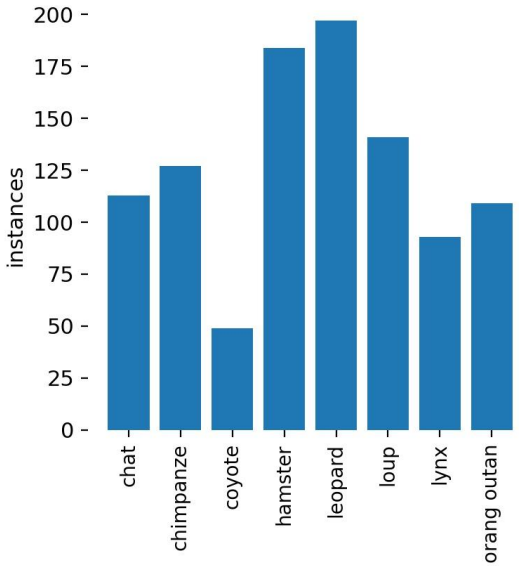
	metrics/precision	metrics/recall	metrics/mAP_0.5	metrics/mAP_0.5:0.95
count	70.000000	70.000000	70.000000	70.000000
mean	0.331811	0.439746	0.264273	0.124017
std	0.098899	0.101768	0.098375	0.054416
min	0.020761	0.027545	0.011898	0.003048
25%	0.274272	0.405672	0.207590	0.090602
50%	0.363945	0.453390	0.284375	0.132305
75%	0.402757	0.499335	0.350465	0.173705
max	0.499950	0.680190	0.378990	0.192490

V. Conclusion

- Modèle supervisé - > 1200 données, 60 epochs, Transfert Learning, augmentation de données (flip, rotate 90°)
- Modèle semi-supervisé - > 1200 données labellisées, 6000 données non-labellisées, Teacher-Student,

VI. Perspectives

- Annoter davantage d'images
- Pré-entraîner le réseau sur un dataset plus pertinent
- Égaliser les instances de classes



Matrice de confusion

Merci de votre attention !