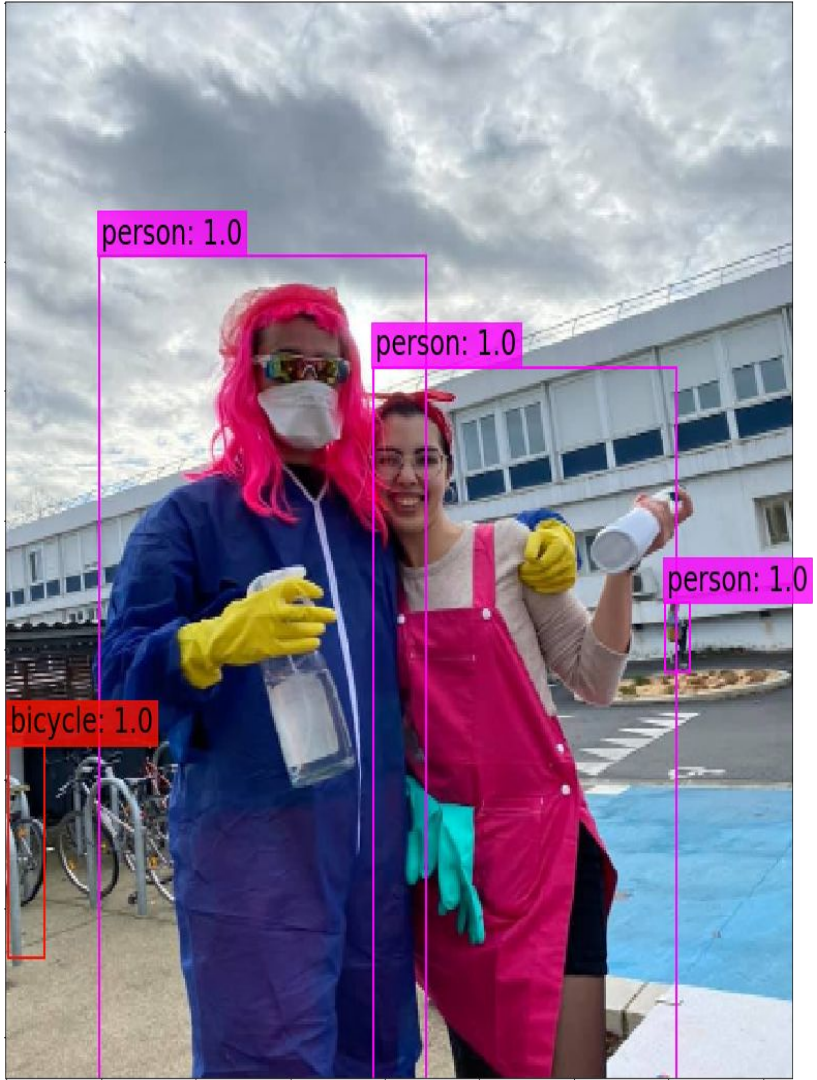
INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

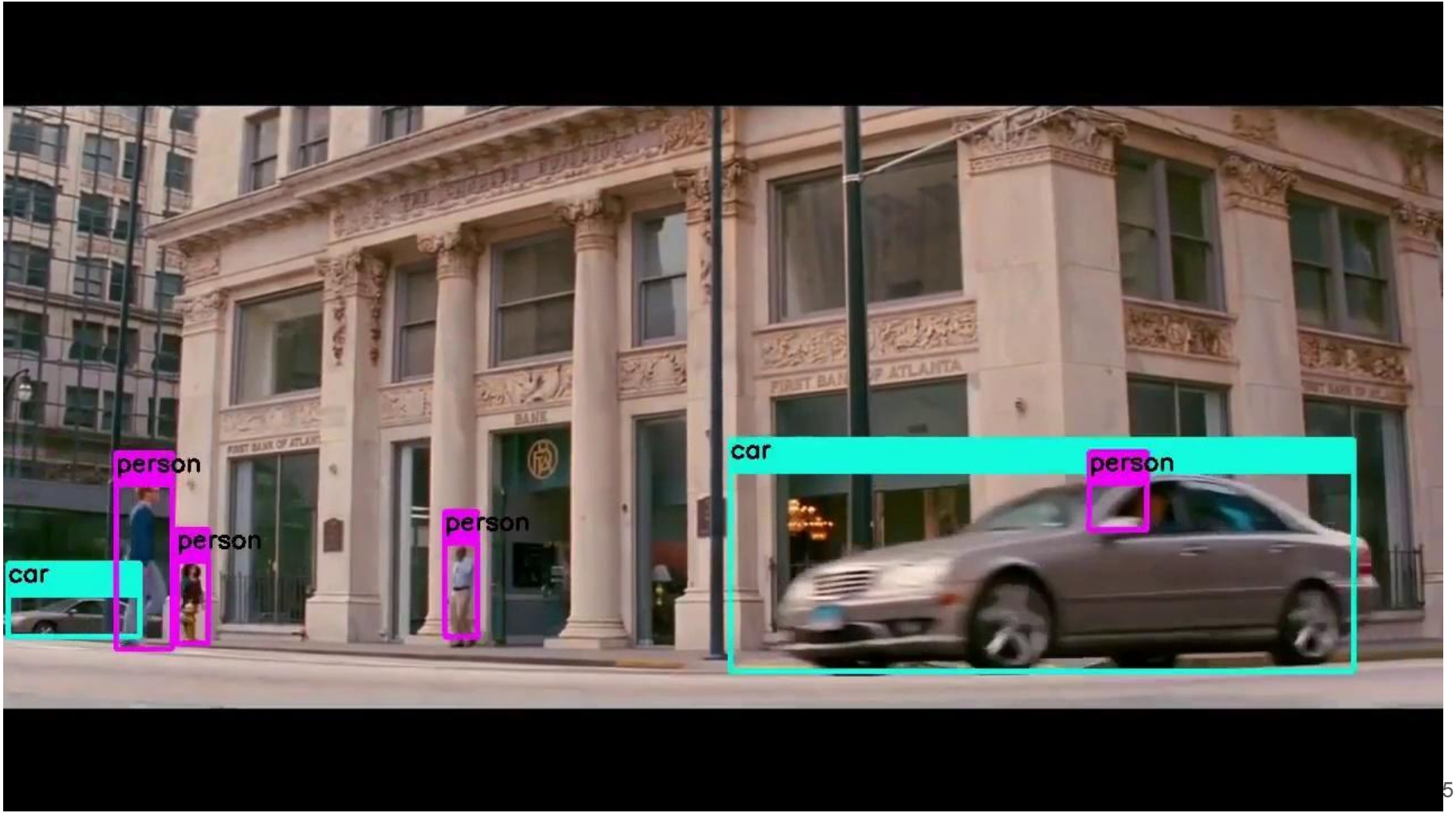
Algorithme YOLOv3

Détection d'objets et classification

Othmane Aouassar - Sébastien Ducor - Eléna Dupuy - Romain Gea - Nusaïbah Ibrahimi - Eloïse Issert

Plan


person: 1.0
person: 1.0
person: 1.0
bicycle: 1.0


person
person
car
person
car
person

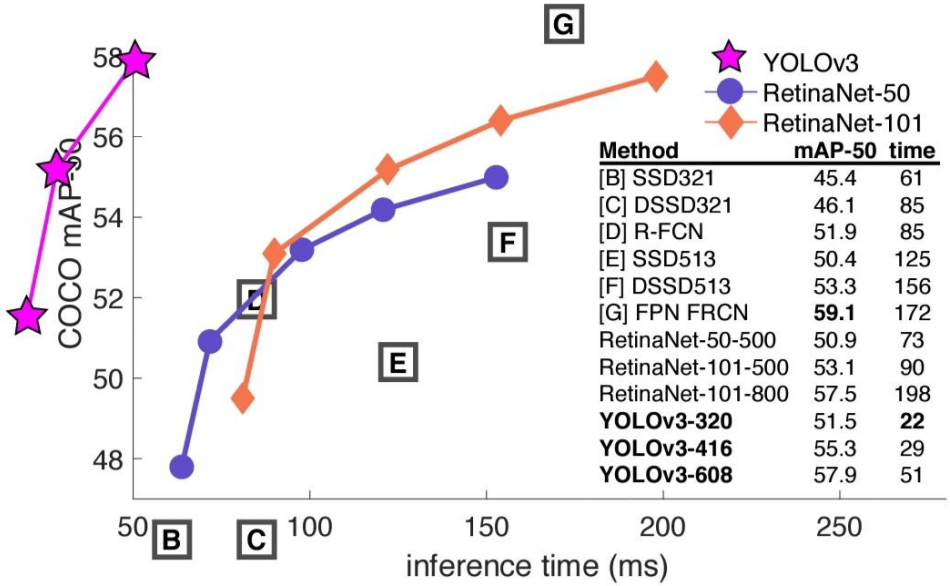
I. Introduction

YOLOv3 est un algorithme de détection basé sur la régression (prédiction de valeurs numériques)

Détection

Localisation d'objet
+
Classification d'objet

One Run



Method	mAP-50	time
[B] SSD321	45.4	61
[C] DSSD321	46.1	85
[D] R-FCN	51.9	85
[E] SSD513	50.4	125
[F] DSSD513	53.3	156
[G] FPN FRCN	**59.1**	172
RetinaNet-50-500	50.9	73
RetinaNet-101-500	53.1	90
RetinaNet-101-800	57.5	198
YOLOv3-320	51.5	**22**
YOLOv3-416	55.3	29
YOLOv3-608	57.9	51

II. Jeu de données COCO (Common Objects in Context)

Informations	Informations sur le jeu de données
Licences	Licences des images
Catégories	Identifiant unique par catégorie Peut appartenir à une super-catégorie (les catégories *rose* et *tulipe* appartiennent à la super-catégorie *fleur*)
Images	Liste des images, identifiant unique par image
Annotations	Pour chaque image, contient notamment “**bounding boxes”** (x-top left, y-top left, width, height)

80
catégories d’objets

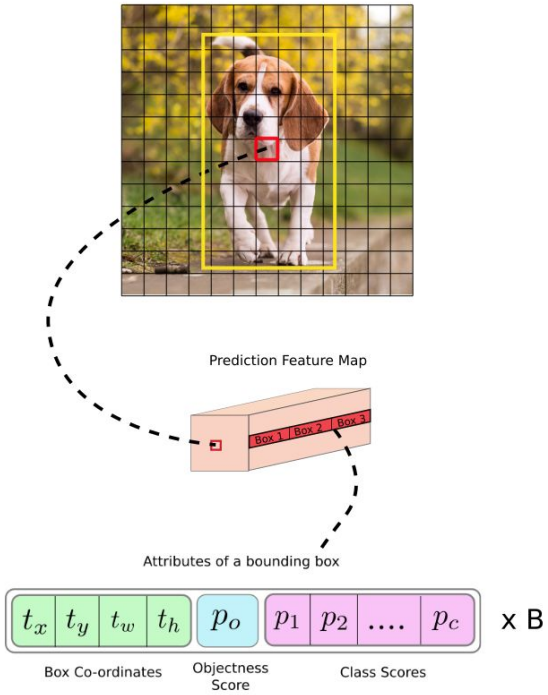
1.5 million
d’instances d’objets

IMAGE

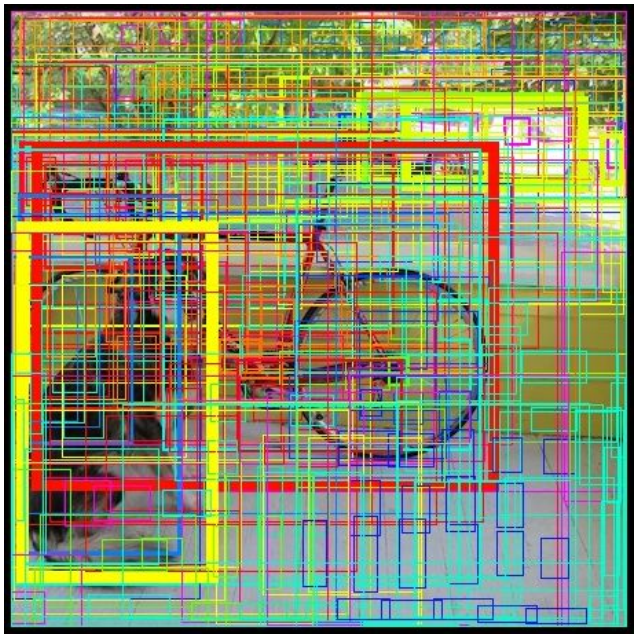
Labels :
BOUNDING-BOX
+
CLASS

III. Présentation de YOLOv3

III.1. Fonctionnement détaillé de l'algorithme



Dans notre cas, YOLOv3 prédit 3 boîtes par cellules.



III. Présentation de YOLOv3

III.1. Fonctionnement détaillé de l'algorithme

Non max suppression - étape 1

- **Suppression des boîtes selon un seuil de probabilité**

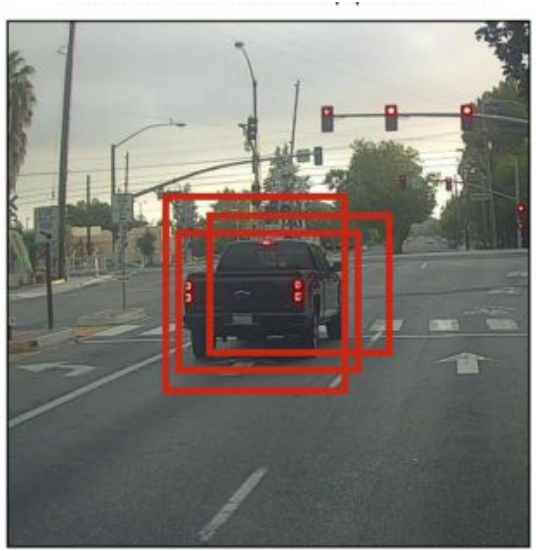
- **Probabilité de détection d'un objet p0**
  - Définie pour chaque boîte
  - Obtenue en sortie du réseau

- **p0 < 0.6** → boîte ne contient pas un objet
  → suppression de la boîte

III. Présentation de YOLOv3

III.1. Fonctionnement détaillé de l'algorithme

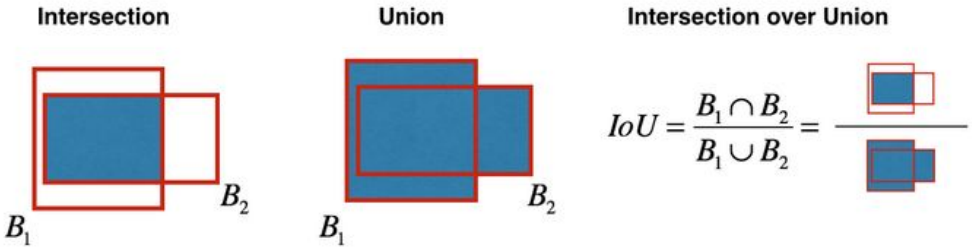
Non max suppression - étape 2



- **Suppression des boîtes selon un seuil IOU fixé**

- **IOU** : Intersection Over Union

  Critère entre deux boîtes B1 et B2

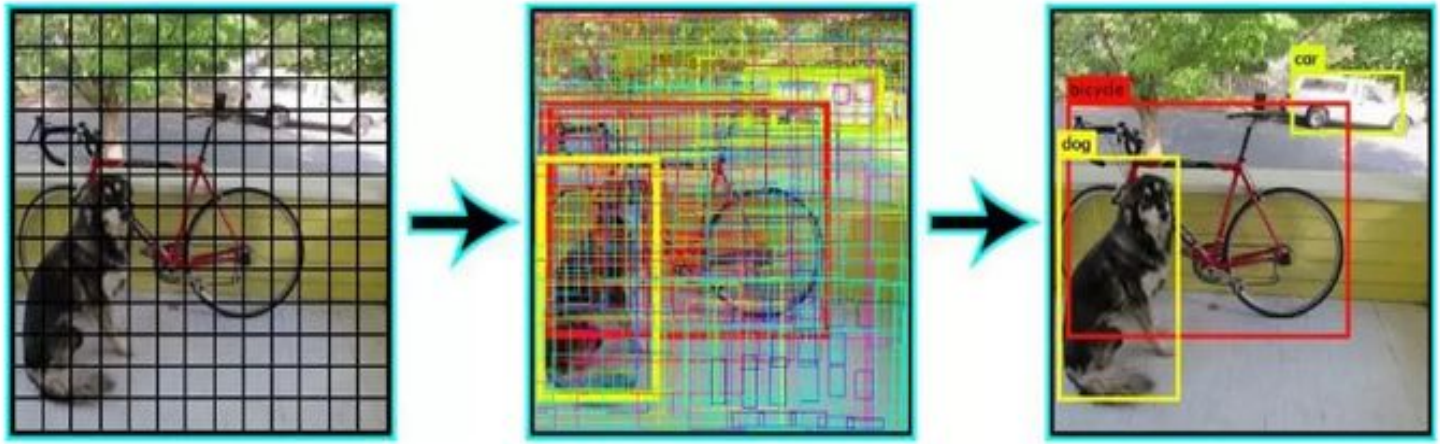


- **IOU > 0.4** ⟶ détection du même objet ⟶ suppression de la boîte

III. Présentation de YOLOv3

III.1. Fonctionnement détaillé de l'algorithme

Processus complet

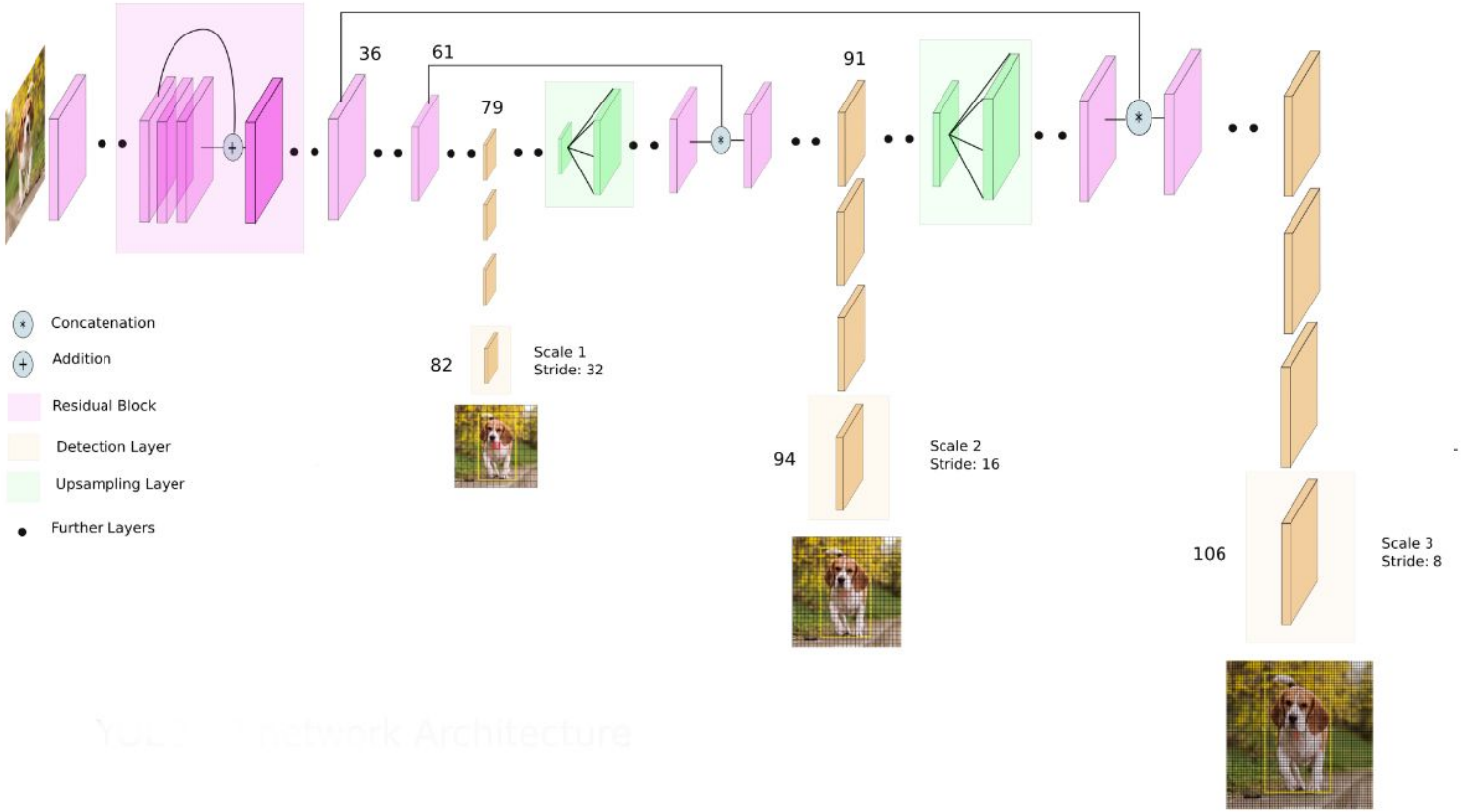


1. Poser un quadrillage sur l'image
2. Pour chaque cellule de la grille, n boîtes sont prédites avec des paramètres
3. Filtrage / Non max suppression

III. Présentation de YOLOv3

III.2. Architecture du réseau YOLOv3

Vue générale du FCNN

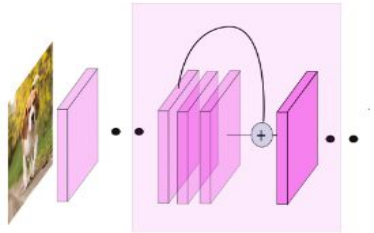


III. Présentation de YOLOv3

III.2. Architecture du réseau YOLOv3

1st **convolutional** blocks

Input image
size :
416 x 416



Concatenation

Addition

Residual Block

Detection Layer

Upsampling Layer

Further Layers

Conv 1 x 1
activ : leaky

Réduit la profondeur

Conv 3 x 3
activ : leaky

Extraction de features

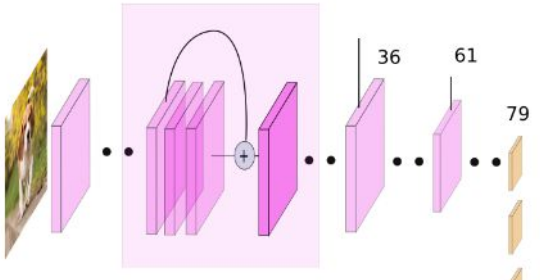
Shortcut
activ : linéaire

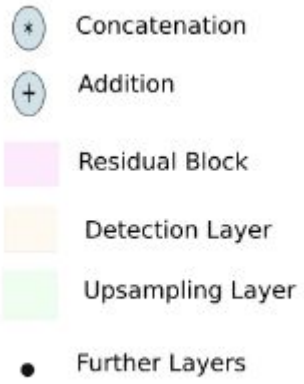
Ne pas perdre
l'information de base

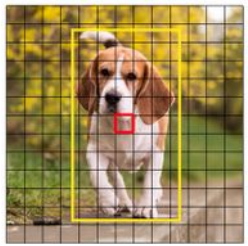
III. Présentation de YOLOv3

III.2. Architecture du réseau YOLOv3

1st **detection layer**







Pas 32

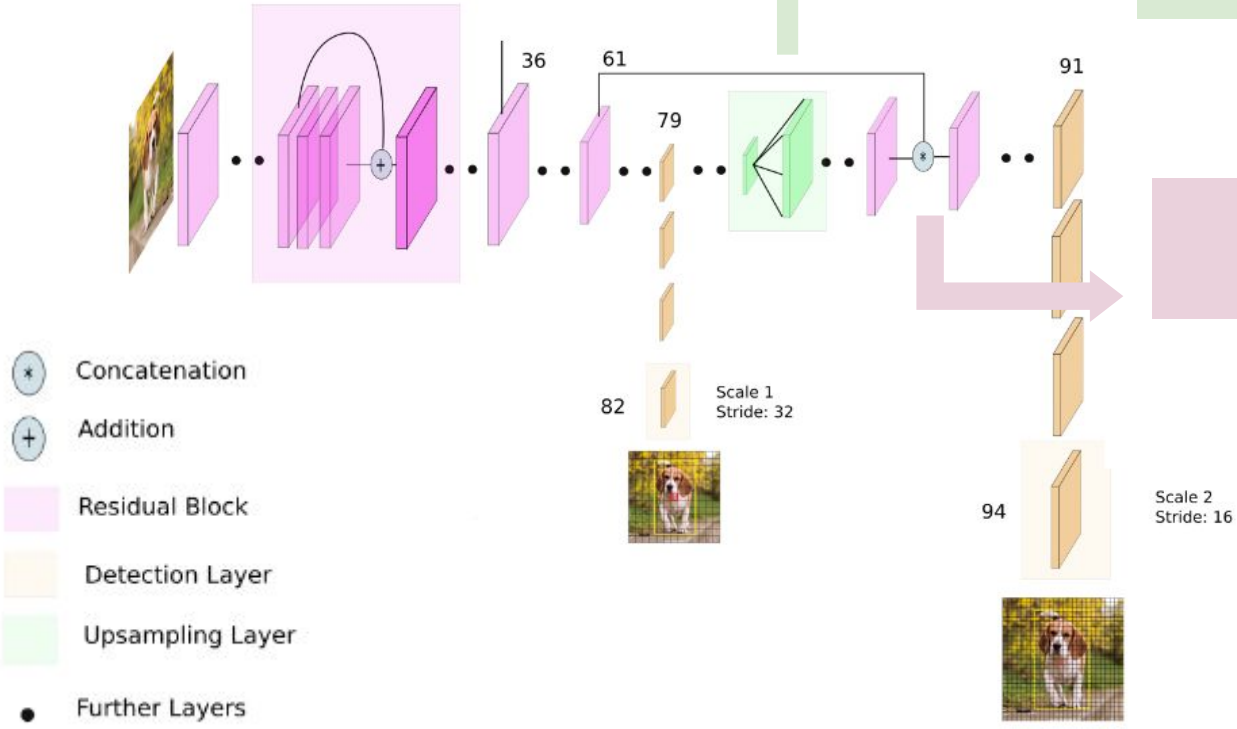
Echelle 13 x13

Détection de “gros” objets

III. Présentation de YOLOv3

III.2. Architecture du réseau YOLOv3

2nde **detection layer**



Upsampling

Augmente le nombre de features

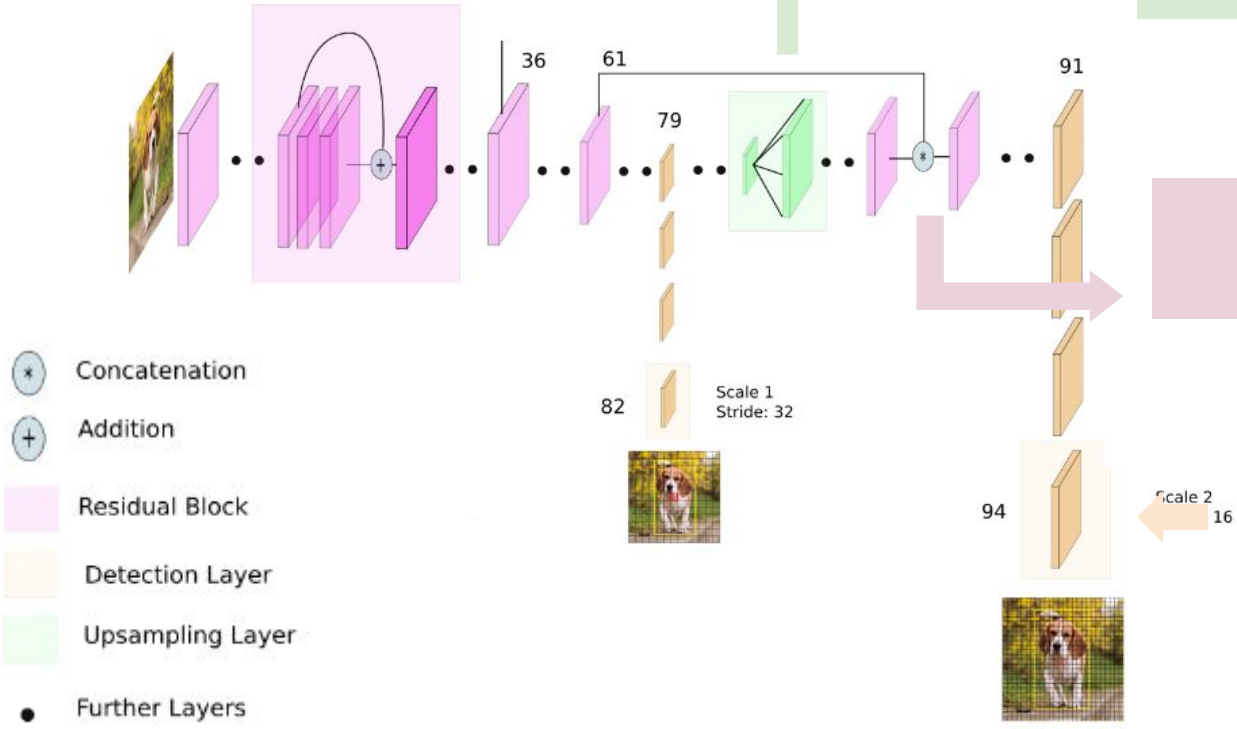
Route

Concatène de précédentes feature maps (Augmente profondeur)

III. Présentation de YOLOv3

III.2. Architecture du réseau YOLOv3

2nde **detection layer**



Upsampling

Augmente le nombre de features

Route

Concatène de précédentes feature maps (Augmente profondeur)

Pas 16

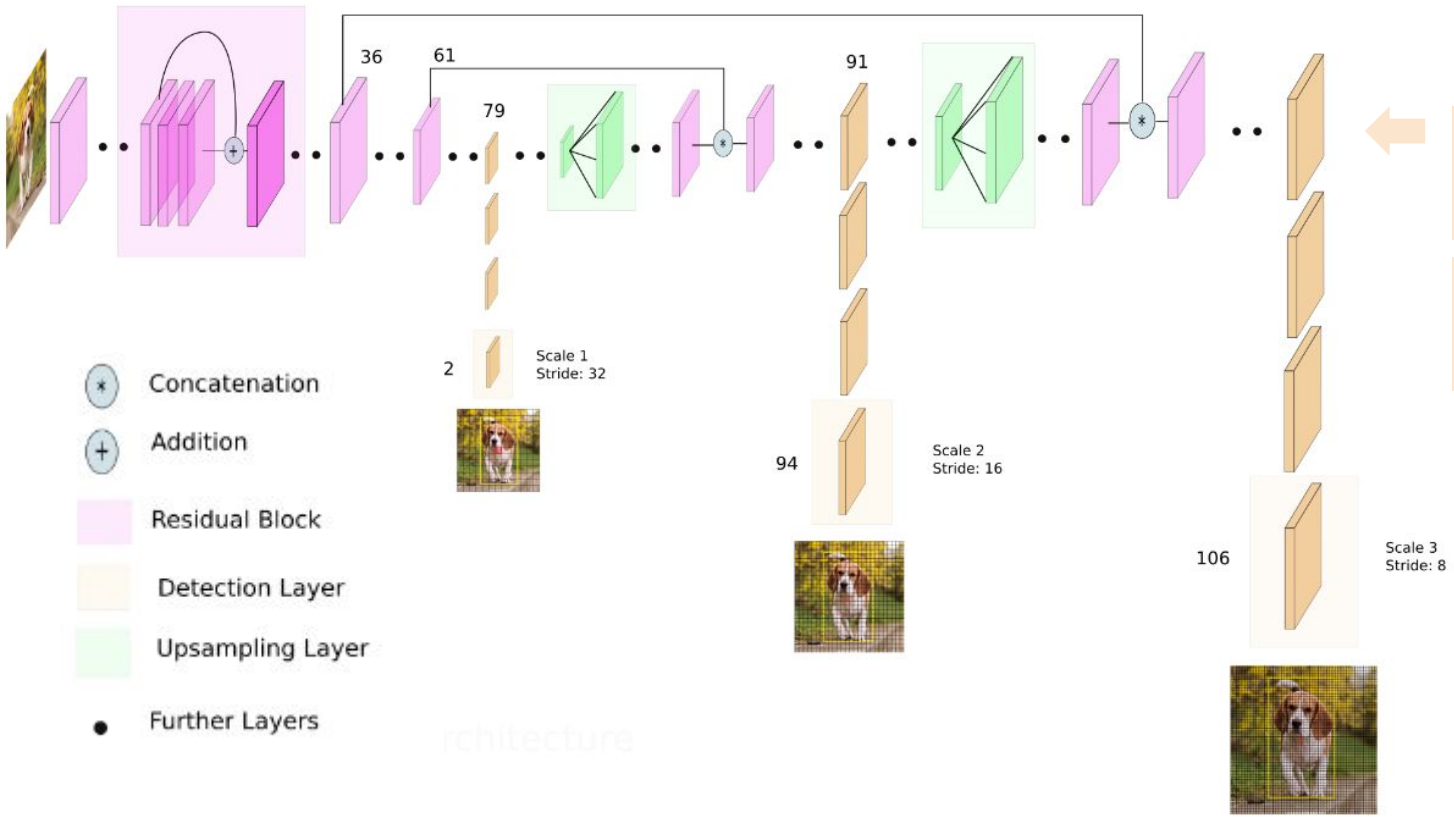
Détection de “moyens” objets

Echelle 26 x 26

III. Présentation de YOLOv3

III.2. Architecture du réseau YOLOv3

3rd **detection layer**



Détection de “petits” objets

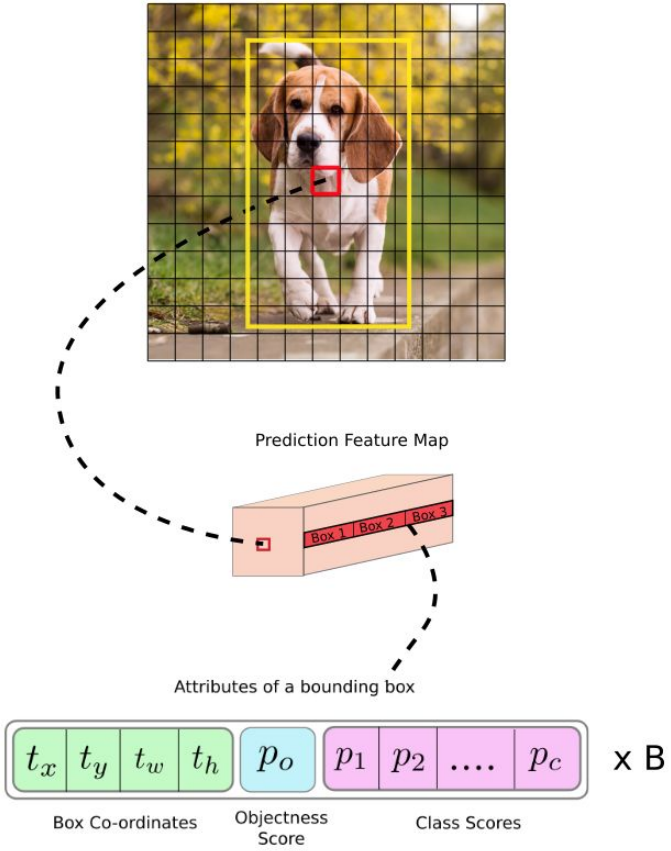
Pas 8

Echelle 52 x 52

III. Présentation de YOLOv3

III.2. Architecture du réseau YOLOv3

Output layer

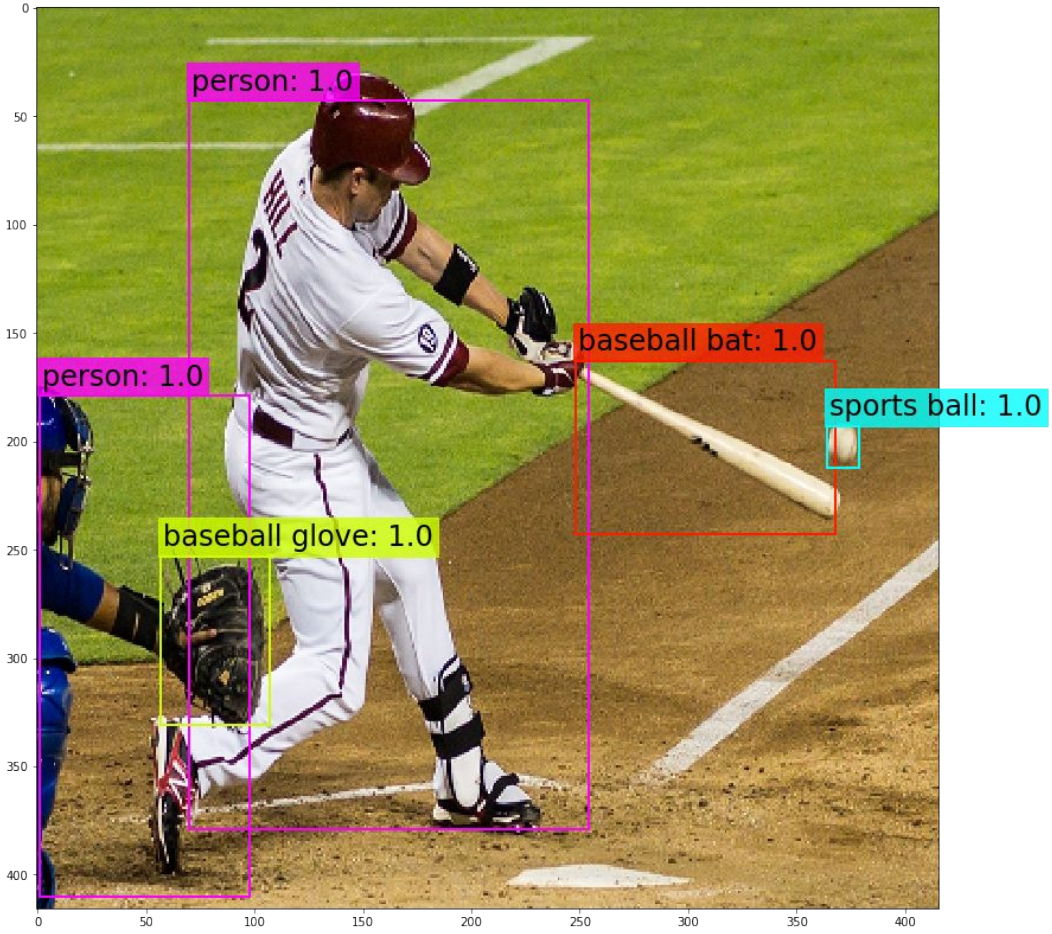


IV. Résultats obtenus

Baseball

5 objets détectés en 0.564s

1. person: 1.000000
2. person: 1.000000
3. baseball glove: 0.999998
4. baseball bat: 1.000000
5. sports ball: 1.000000



IV. Résultats obtenus

City

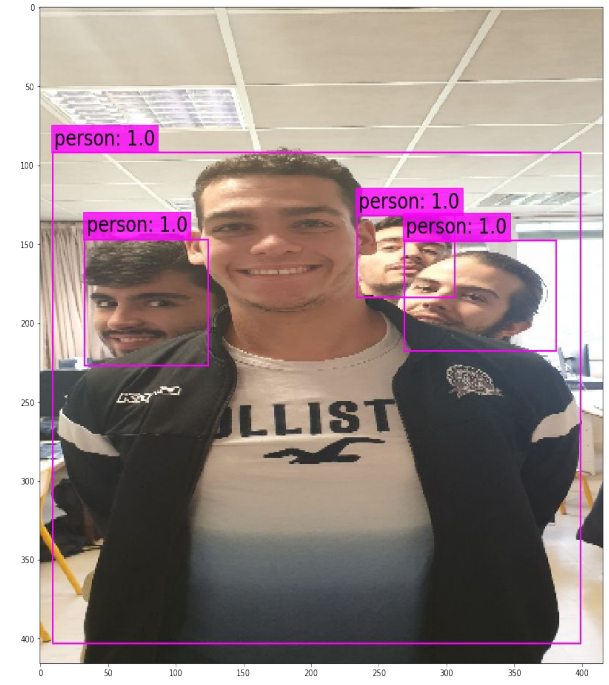
28 objets détectés en 0.627s

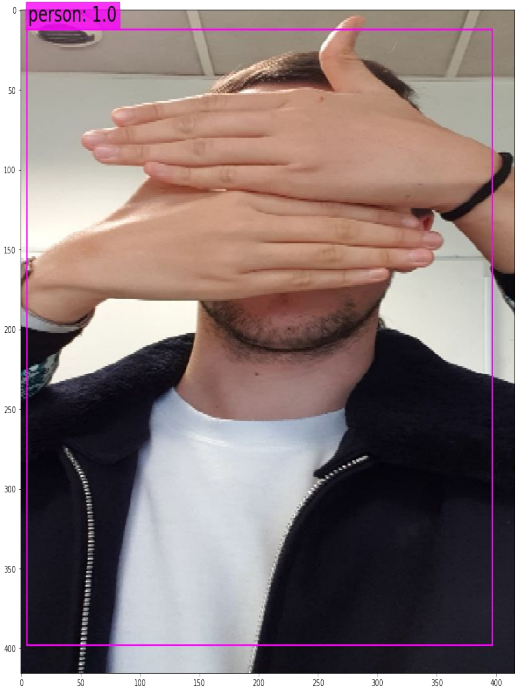
1. person: 0.999996
2. person: 1.000000
3. car: 0.707236
4. truck: 0.933031
5. car: 0.658086
6. truck: 0.666982
7. person: 1.000000
8. traffic light: 1.000000
9. person: 1.000000
10. car: 0.997369
11. bus: 0.998023
12. person: 1.000000
13. person: 1.000000
14. person: 1.000000
15. person: 1.000000
16. person: 1.000000
17. traffic light: 1.000000
18. traffic light: 1.000000
19. handbag: 0.997282
20. traffic light: 1.000000
21. car: 0.989741
22. traffic light: 1.000000
23. traffic light: 0.999999
24. person: 0.999999
25. truck: 0.715036
26. traffic light: 1.000000
27. person: 0.999993
28. person: 0.999996

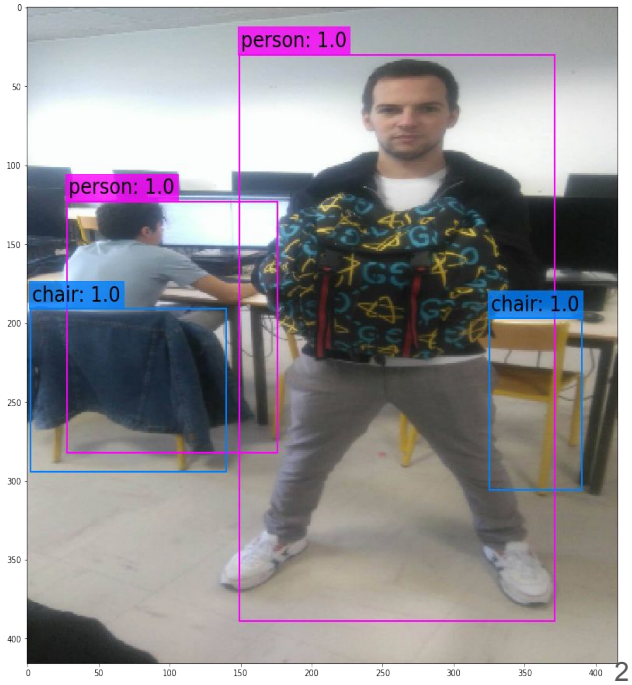
IV. Résultats obtenus

Tentatives de “tromperies”

➔ Pas d’impact sur la qualité de détection = probabilités à 1.0





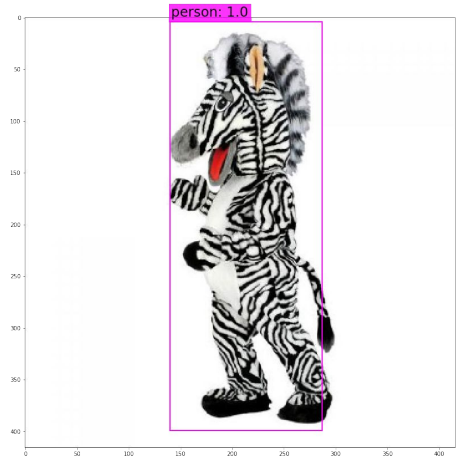


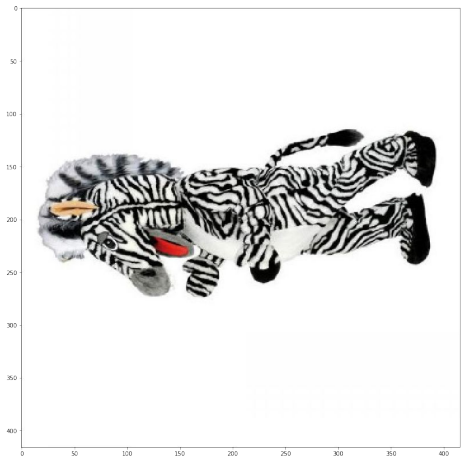
V. Limites pour la détection et la classification

La **bonne orientation** de l'image

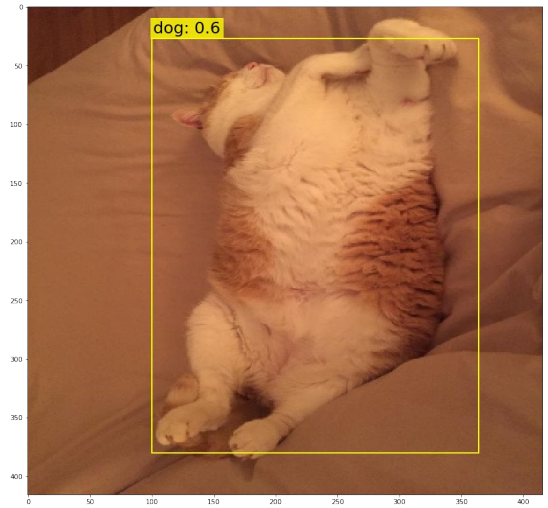
➔ Impact sur la détection et la classification

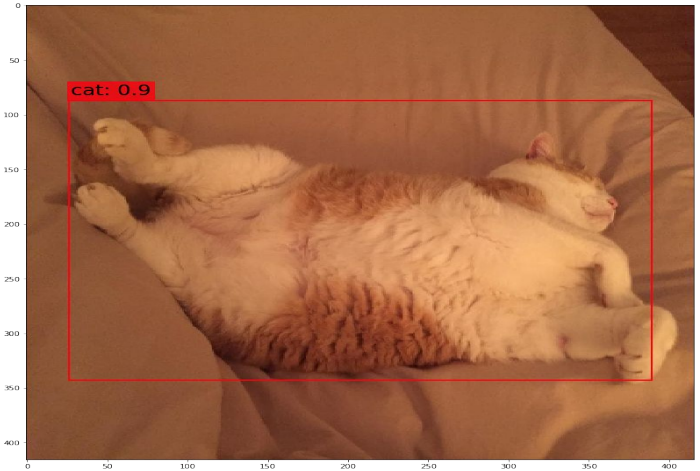
Pas de détection





Mauvaise classification





V. Limites pour la détection et la classification

Dataset rassemble **peu de classes** (80)

➔ Classification imprécise

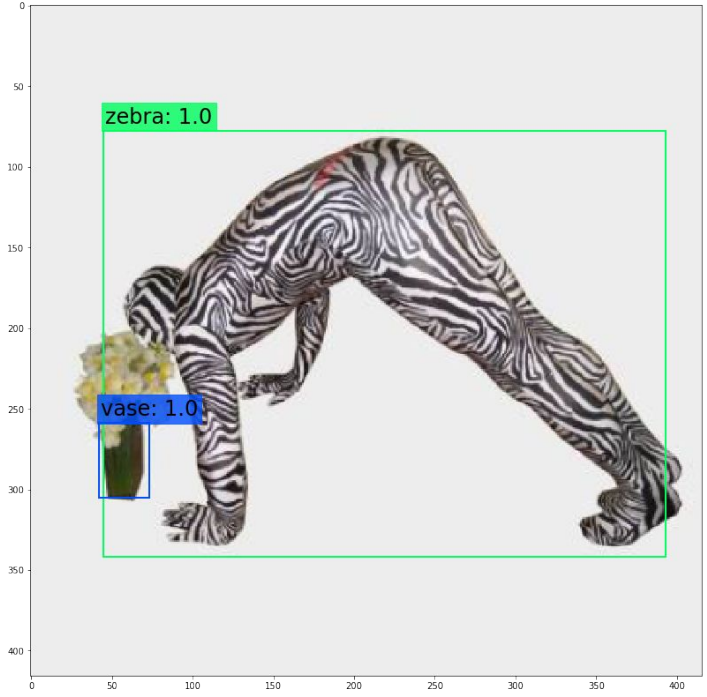
➔ Impact sur la détection

Ne différencie pas les sous-classes

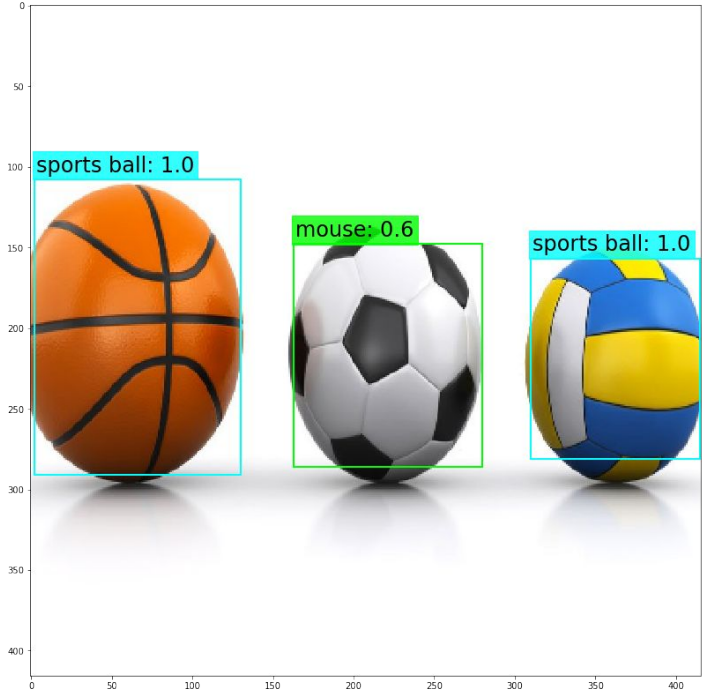
Considère une unique classe

V. Limites pour la détection et la classification

Tentatives de “tromperies” ayant fonctionnées



Mauvaise classification



V. Limites pour la détection et la classification

Mauvaise détection

“Generating adversarial patches against YOLOv2” - 20 mars 2019 - Youtube

Conclusion

- Points faibles :
  - Orientation des images est importante
  - Dataset est petit : 80 classes vs 9000 pour Imagenet
  - Peut être trompé

- Point forts :
  - Meilleur que Fast-RNN
  - Le plus rapide : prédictions en moins d’une seconde
  - Le plus puissant : image floue et chaises cachées ou individu caché

Merci de votre attention

Références

Site officiel de YOLO : https://pjreddie/darknet/yolo

Sites web :

- https://medium.com/@pratheesh.27998/object-detection-part1-4dbe5147ad0a
- https://www.kdnuggets.com/2018/09/object-detection-image-classification-yolo.html
- https://medium.com/analytics-vidhya/yolo-v3-theory-explained-33100f6d193
- https://towardsdatascience.com/an-introduction-to-implementing-the-yolo-algorithm-for-multi-object-detection-in-images-99cf240539

Articles scientifiques:

- https://arxiv.org/pdf/1612.08242.pdf
- https://arxiv.org/pdf/1506.02640.pdf