

YOLOV3

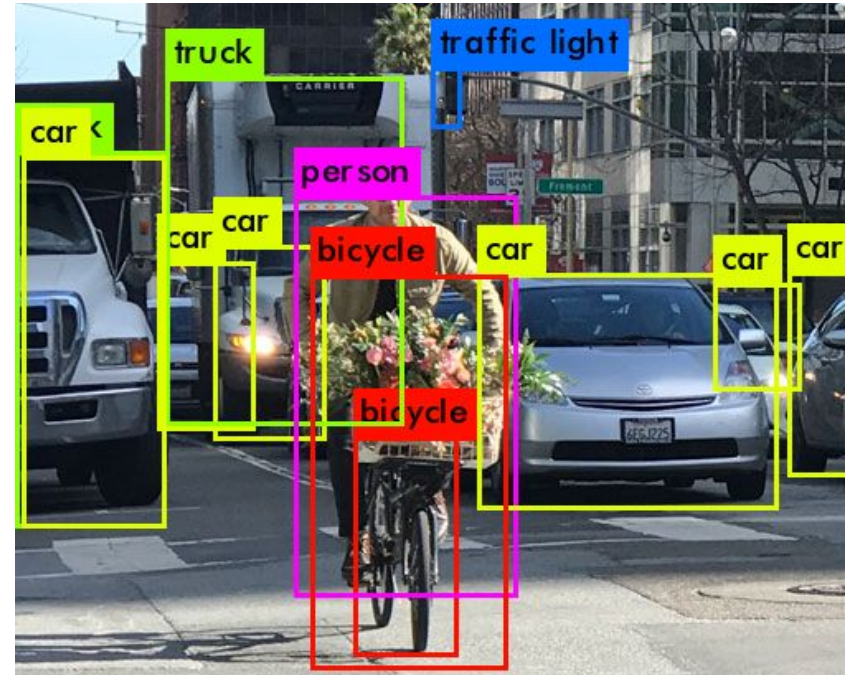
Prieur Maxime, Quarez Etienne

INSA de Rouen

29 Avril 2020

Plan

1. Problème traité
2. Méthode proposée
3. Implémentation
4. Expérience
5. Conclusion



Object detection

<https://mc.ai/part-2-fast-r-cnn-object-detection/>

Problème traité - Transfert learning

Input

- Images couleurs, taille variable
- Flux vidéo

Output

- Position de l'objet (x,y,h,w)
- Nom de la classe
- Score de confiance

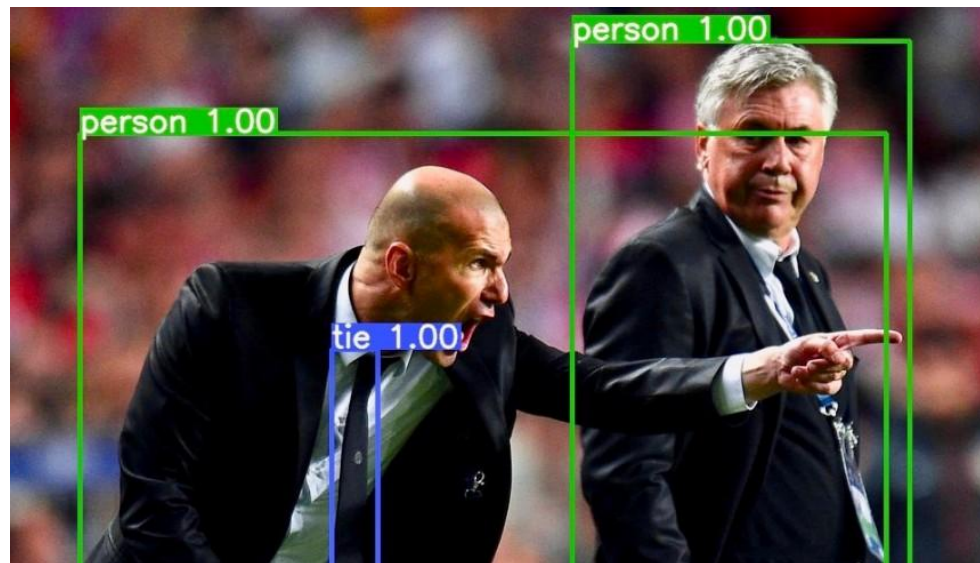


Image traité par YOLO V3

<https://pythonawesome.com/yolov3-training-and-inference-in-pytorch/>

Idée de sujet

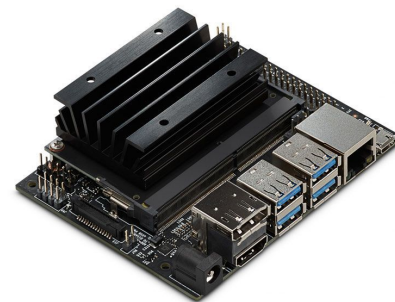


RobotMaster

<https://www.tuxboard.com/robot-educatif-dji-robotmaster/>



Malandain



Jetson Nano

~~CenterNet ?~~



YOLO V3

Méthode Proposée - You Only Look Once



Joseph Redmon



*Santosh Divvala
Allen Institute for IA*



*Ross Girshick
Facebook AI*



*Ali Farhadi
Washington University*

Yolo - 2016

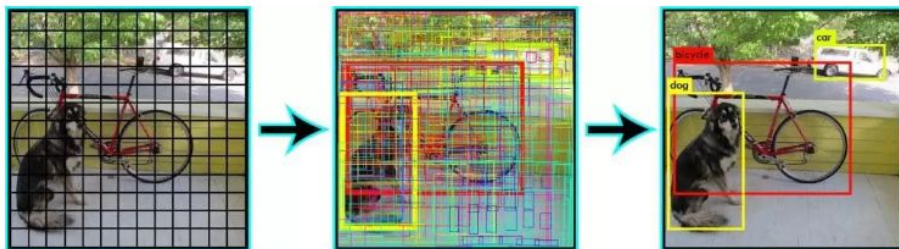


YoloV2 (and 9000) - 2016

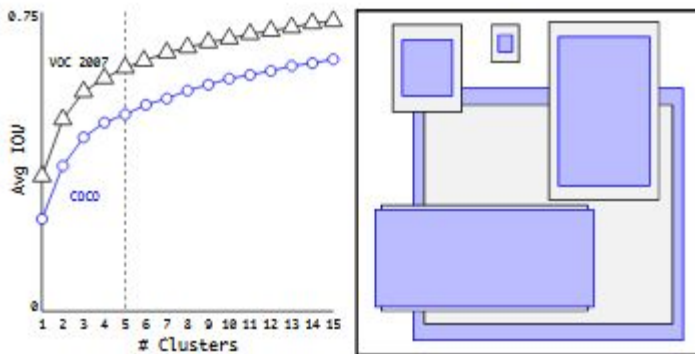


YoloV3 - 2018

YOLOV3 - Fonctionnement



Principe de détection



“Boîtes” K-means sur VOC et COCO

	Type	Filters	Size	Output
1x	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
	Convolutional	32	1 × 1	128 × 128
	Convolutional	64	3 × 3	
	Residual			
2x	Convolutional	128	3 × 3 / 2	64 × 64
	Convolutional	64	1 × 1	64 × 64
	Convolutional	128	3 × 3	
	Residual			
8x	Convolutional	256	3 × 3 / 2	32 × 32
	Convolutional	128	1 × 1	32 × 32
	Convolutional	256	3 × 3	
	Residual			
	Convolutional	512	3 × 3 / 2	16 × 16
8x	Convolutional	256	1 × 1	16 × 16
	Convolutional	512	3 × 3	
	Residual			
	Convolutional	1024	3 × 3 / 2	8 × 8
4x	Convolutional	512	1 × 1	8 × 8
	Convolutional	1024	3 × 3	
	Residual			
	Avgpool		Global	
	Connected		1000	
	Softmax			

Architecture DarkNet-53

Implémentation utilisée

- Site de l'auteur : <https://pjreddie.com/darknet/yolo/>
- Git de l'auteur : <https://github.com/pjreddie/darknet>
- Git de l'adaptation de l'architecture sous Keras : <https://github.com/qgwweee/keras-yolo3>

Expérimentation - Données

Sites de données :

- <https://www.kaggle.com/issaisasank/guns-object-detection>
- http://www.imfdb.org/wiki/Main_Page
- <https://github.com/SasankYadati/Guns-Dataset>
- <https://sci2s.ugr.es/weapons-detection>
- Extraits de films



Extrait du dataset



Interface de labelIMG

```
train.txt
../Guns-Dataset/images/162.jpg 45,34,213,71,0 177,47,277,62,0 150,62,290,99,0 3,95,28,143,0
../Guns-Dataset/images/armas(582).jpg 15,63,93,117,0
../Guns-Dataset/images/arma(287).jpg 119,103,412,346,0
```

Format des labels

Expérimentation - Procédure de transfert learning

Procédure

- Partir des poids YOLOV3 publiques
- Modèles Freeze (50 epochs)
- Modèle entier sur le reste des epochs

Paramètres

- 1300 images
- Validation set : 10% du dataset
- BatchSize : 32

Callbacks

- Tensorboard
- ModelCheckpoint
- ReduceLROnPlateau (4 epochs, delta=0)
- EarlyStopping (10 epochs, delta=0)

Expérimentation - Métriques & Loss utilisée

Métriques : MAP

Loss :

$$\text{Loss_1} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2$$

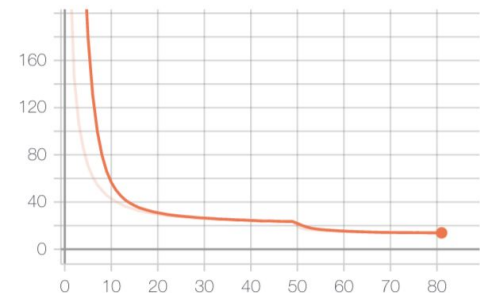
$$\text{Loss_2} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2$$

$$\text{Loss_3} = \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{noobj} (C_i - \hat{C}_i)^2$$

$$\text{Loss_4} = \sum_{i=0}^{S^2} \mathbb{1}_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2$$

Expérimentation - Résultats

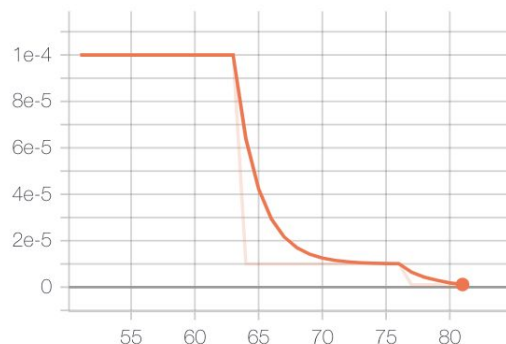
loss



Name	Smoothed	Value	Step	Time	Relative
.	13.88	13.9	81	Mon Apr 13, 00:26:33	2d 8h 34m 16s

Loss sur trainSet

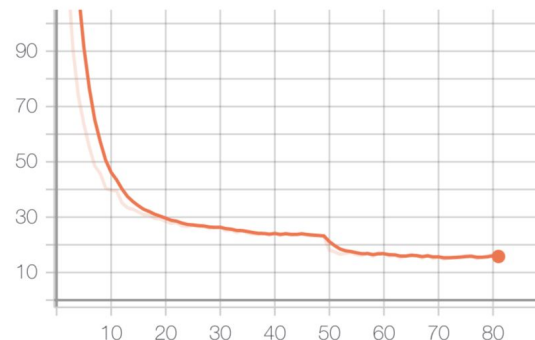
lr



LearningRate

val_lossName	Smoothed	Value	Step	Time	Relative
.	15.78	15.39	81	Mon Apr 13, 00:26:33	2d 8h 34m 16s

val_loss



Loss sur validationSet

Poids retenu

- Epoch **72**
- Loss : **14.133**
- Val Loss : **14.79**

Conclusion

- **Potentiel intéressant**
- **Diversifier/Enrichir le dataset**
- **Tester sur un GPU**
- **Faire varier les paramètres (changer des couches, tailles des batchs etc...)**
- **Essayer d'autre méthode de détection**

Bibliographie

- <https://arxiv.org/pdf/1506.02640.pdf>, *You Only look Once : Unified, Real-Time Object Detection*. **Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi**. 9 Mai 2016
- <https://arxiv.org/pdf/1612.08242.pdf>, *Yolov9000 : Better Faster Stronger*. **Joseph Redmon, Ali Farhadi**. 25 Décembre 2016
- <https://arxiv.org/pdf/1804.02767.pdf>, *Yolov3 : An Incremental Improvement*. **Joseph Redmon, Ali Farhadi**. 8 Avril 2018