

Projet 7

Développez une preuve de concept



Sommaire

- Présentation du projet
- État de l'art de la détection d'objets
- Modèle de référence
- Algorithme YOLOv5
- Algorithme YOLOR
- Synthèse

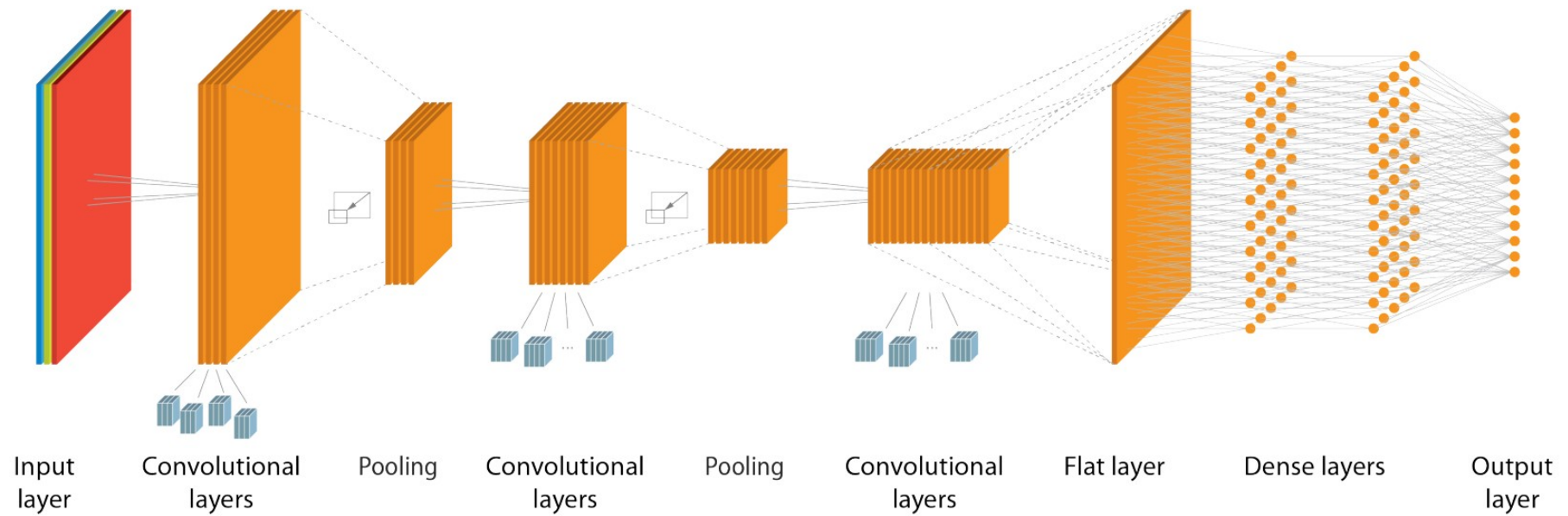
présentation

- Comment améliorer le projet P6 ?
- Actuellement : classification
- Ajout : localisation
- Pistes d'améliorations :
 - Data augmentation
 - traitement images différents(résolution)
 - test de différentes configuration sur des petits datasets
 - Tester de nouveaux modèles

Méthodologie

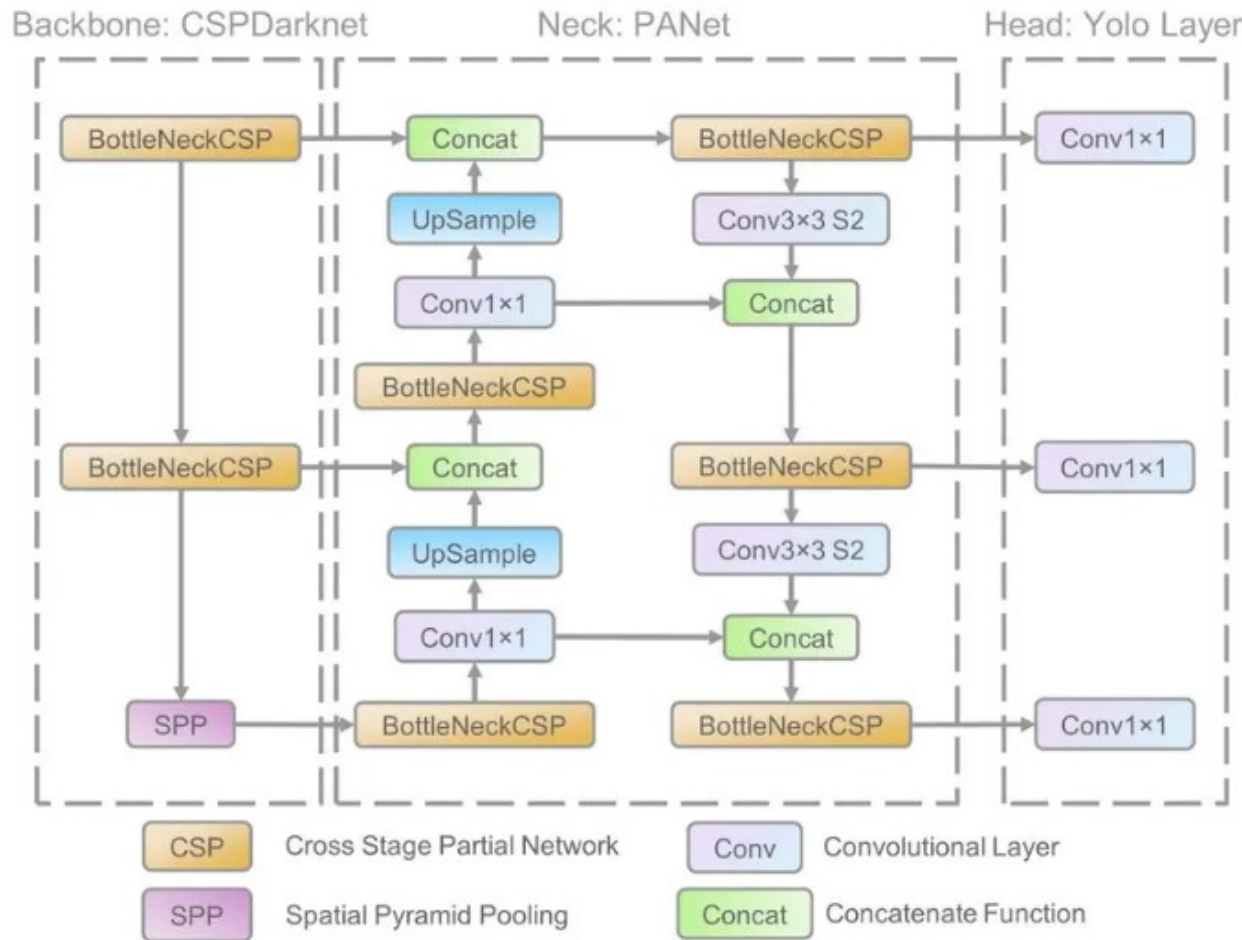
- **Dataset** : (standforddogs , road signs, whiteblood)
- **Localisation** : récupération/conversion ou création
- **Modèles de référence** Xception avec transfert learning
- **Algorithme de détection** Yolor et Yolov5

Modele de base convolution



Modele yolov5

YOLOv5 Architecture



Backbone : est principalement utilisé pour extraire des caractéristiques importantes de l'image d'entrée donnée. Dans YOLO v5, le CSP - Cross Stage Partial Networks est utilisé comme épine dorsale pour extraire des fonctionnalités riches en informations à partir d'une image d'entrée.

Neck est principalement utilisé pour générer des pyramides de caractéristiques. Les pyramides de caractéristiques aident les modèles à bien généraliser la mise à l'échelle des objets. Il aide à identifier le même objet avec différentes tailles et échelles.

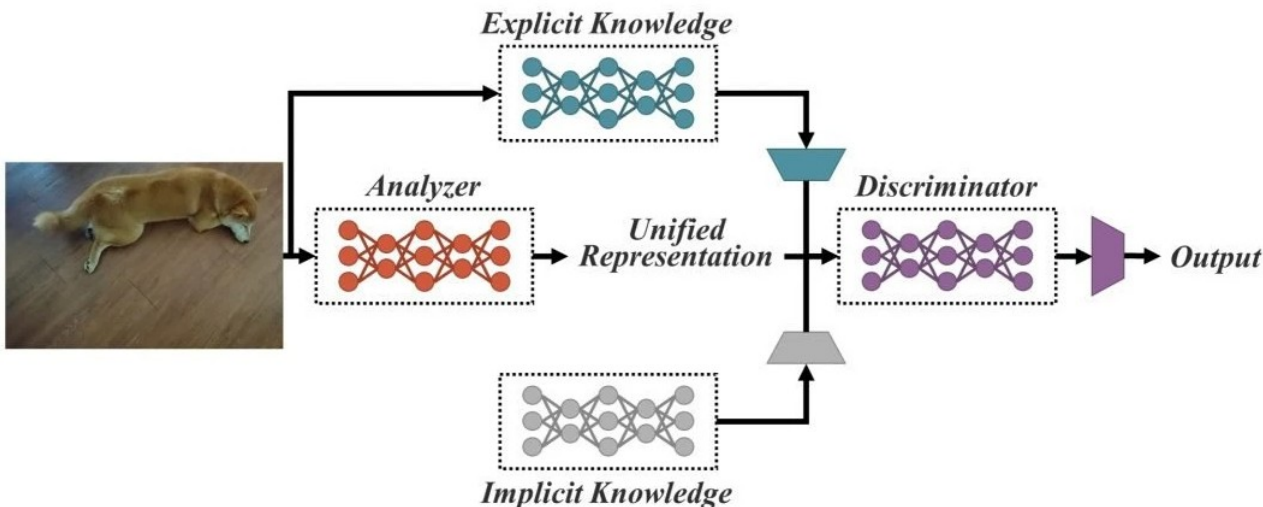
Head du modèle est principalement utilisée pour effectuer la partie de détection finale. Il applique des boîtes d'ancrage sur les entités et génère des vecteurs de sortie finaux avec des probabilités de classe, des scores d'objectivité et des boîtes englobantes.

Activation : Leaky ReLU

Optimisation: SGD Adam

Loss : binary cross entropy avec logits loss

Modèle yolor



Les réseaux de neurones convolutifs (CNN) remplissent généralement un objectif spécifique, alors qu'ils pourraient être formés pour résoudre plusieurs problèmes à la fois, ce qui est exactement l'objectif de **YOLOR**. Les CNN sont souvent créés avec un seul objectif en tête. Alors que les CNN apprennent à analyser les entrées pour obtenir des sorties, YOLOR essaie de faire en sorte que les CNN (1) apprennent à obtenir des sorties et aussi (2) ce que pourraient être toutes les différentes sorties. Plutôt qu'une seule sortie, il peut en avoir plusieurs

information explicite

Une connaissance explicite est donnée aux réseaux de neurones en fournissant des métadonnées claires ou des bases de données d'images qui sont soit soigneusement annotées, soit bien organisées. Les connaissances explicites peuvent être considérées comme des cartes mémoire pour le modèle d'apprentissage automatique, avec des définitions claires et des images/entrées correspondant à ces images.

Une fois que le modèle a parcouru le jeu de cartes mémoire, il est maintenant bien familiarisé avec la classification des images avec leurs définitions respectives, ou "classes". La connaissance explicite de YOLOR est obtenue à partir des couches superficielles des réseaux de neurones. Cette connaissance correspond directement aux observations qui sont supposées être faites.

information implicite

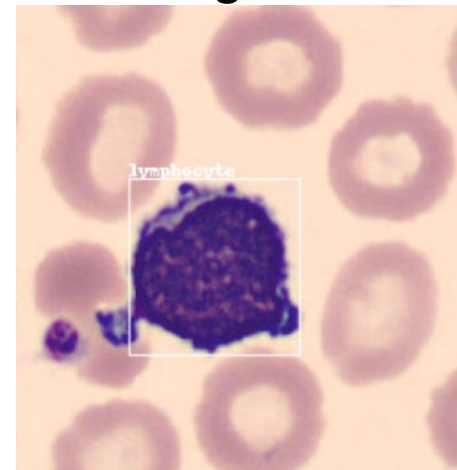
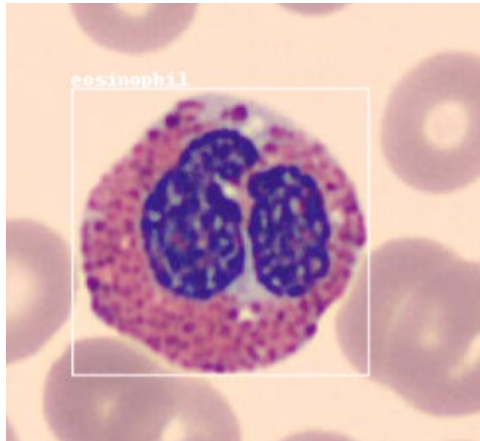
Les connaissances implicites peuvent aider efficacement les modèles d'apprentissage automatique à effectuer des tâches avec YOLOR. Pour les humains, la connaissance implicite est développée inconsciemment. Pour les réseaux de neurones, la connaissance implicite est obtenue par des caractéristiques dans les couches profondes. Les connaissances qui ne correspondent pas aux observations sont également appelées connaissances implicites.

les caractéristiques obtenues à partir de couches peu profondes sont appelées « connaissances explicites », tandis que les caractéristiques obtenues à partir de couches profondes sont appelées connaissances implicites. Les vecteurs, les réseaux de neurones et la factorisation matricielle ont été analysés comme des méthodes possibles de modélisation des informations implicites et explicites dans l'article YOLOR.

YOLOR est censé être un réseau unifié qui peut intégrer des connaissances implicites et explicites afin de créer une représentation générale.

État de l'art de la détection d'objets

- Détection d'objets = classification + localisation
- Transformation des images : encadrer les cibles
- Localisation = prédire un rectangle d'encadrement
→ coordonnées centre rectangle + hauteur + largeur



État de l'art de la détection d'objets

- Algorithmes testés :
CNN (R CNN) , yoloV5 et Yolor
- Métriques :
 - Classification → précision / recall /
 - Localisation → mAP@0.X

mAP (mean Average Precision) est une métrique d'évaluation utilisée dans les modèles de détection d'objets tels que YOLO .

Intersection sur Union (en anglais Intersection over Union), aussi appelé IoU, est une fonction qui quantifie à quel point la zone délimitante prédite p est correctement positionnée par rapport à la zone délimitante réel

Le calcul de mAP nécessite IOU, Precision, Recall, Precision Recall Curve et AP .

Lors de l'évaluation des modèles de détection d'objets, il est nécessaire de définir le degré de chevauchement des boîtes englobantes par rapport aux données de vérité terrain qui doit être considéré comme une reconnaissance réussie

Modèles de référence

- Dataset :
 - stanfordddogs (120 classes)
 - road signs (4 classes)
 - bloodcells (6 classes)
- Performances Xception

XCEPTION	StandfordDogs	Roadsigns	Whitebloods
Loss	0.3741	0.1204	0.1576
Accuracy	0.89	0.76	0.94

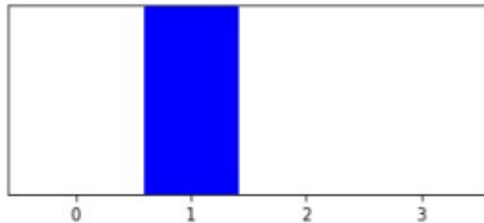
Modèles de référence

Road signs

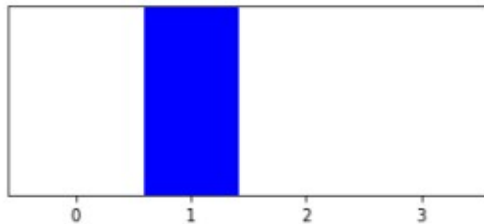
	precision	recall	f1-score	support
crosswalk	0.73	0.61	0.67	18
speedlimit	0.96	0.94	0.95	138
stop	0.60	0.75	0.67	8
trafficlight	0.73	0.92	0.81	12
accuracy			0.90	176
macro avg	0.76	0.80	0.77	176
weighted avg	0.90	0.90	0.90	176



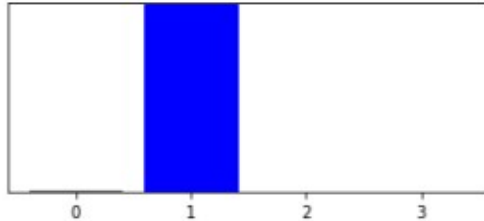
speedlimit 100% (speedlimit)



speedlimit 99% (speedlimit)

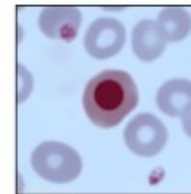


speedlimit 99% (speedlimit)

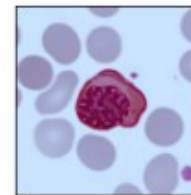
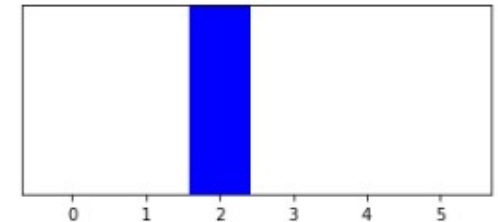


White cells

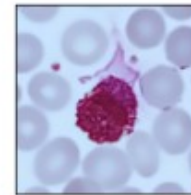
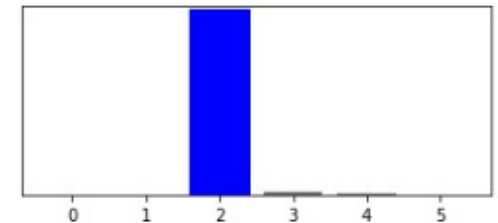
	precision	recall	f1-score	support
basophil	0.97	0.93	0.95	245
eosinophil	0.94	0.98	0.96	623
erythroblast	0.89	0.98	0.93	316
lymphocyte	0.88	0.99	0.93	228
monocyte	0.95	0.94	0.94	299
neutrophil	0.98	0.87	0.92	659
accuracy			0.94	2370
macro avg	0.94	0.95	0.94	2370
weighted avg	0.94	0.94	0.94	2370



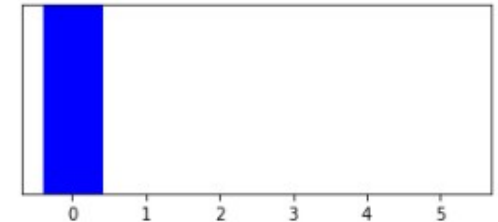
erythroblast 100% (erythroblast)



erythroblast 98% (erythroblast)



basophil 100% (basophil)



Yolov5

Rectangle d'encadrement



0 0.398 0.569 0.257 0.542

Yolo configuration :

Utilisation du Transfer Learning , fine tuning, grâce aux fichiers de configuration

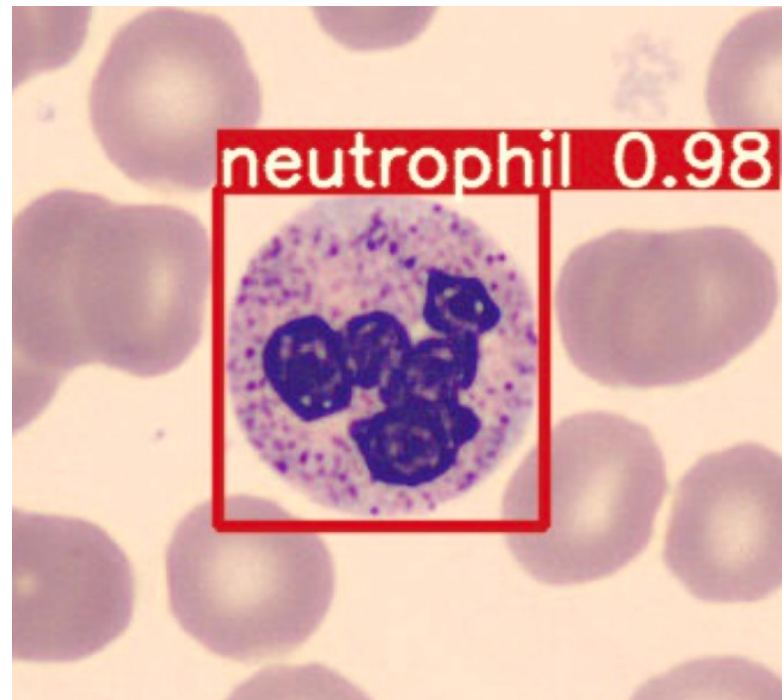
Yolov5

- Anchor boxes pour chaque images
- Préparation des données → yolov5.yaml :
 - Nombre de classes
 - Noms des classes
 - conversion des annotations aux formats pascal voc vers yolo
 - création de labels automatiquement
 - Data augmentation +Fine-tuning

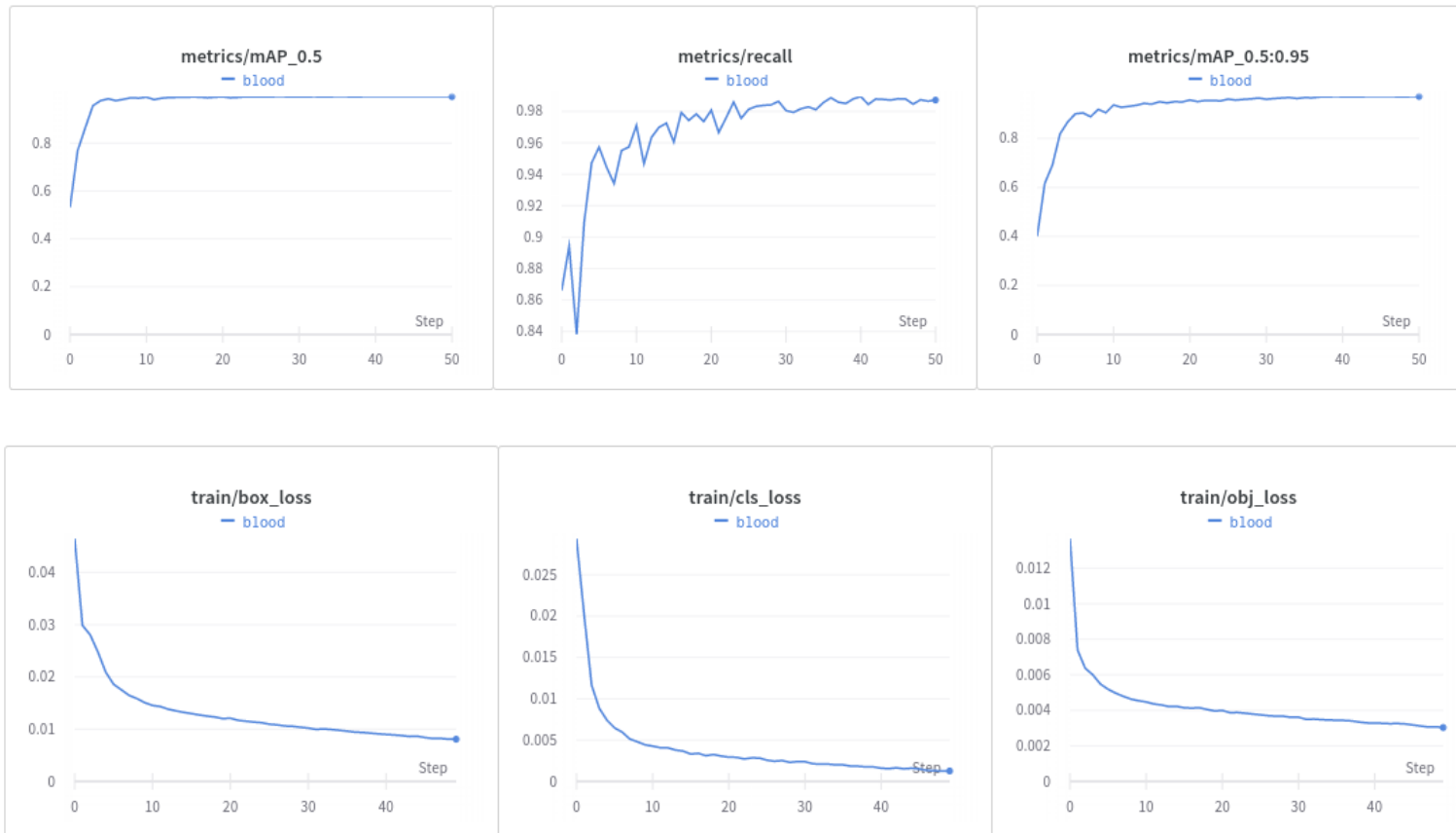
Yolov5 performances

Resultats : https://wandb.ai/ismail-azdad/YOLOR/reports/yolor-blood_cells-dataset-report---VmIldzoxOTQwNzM2

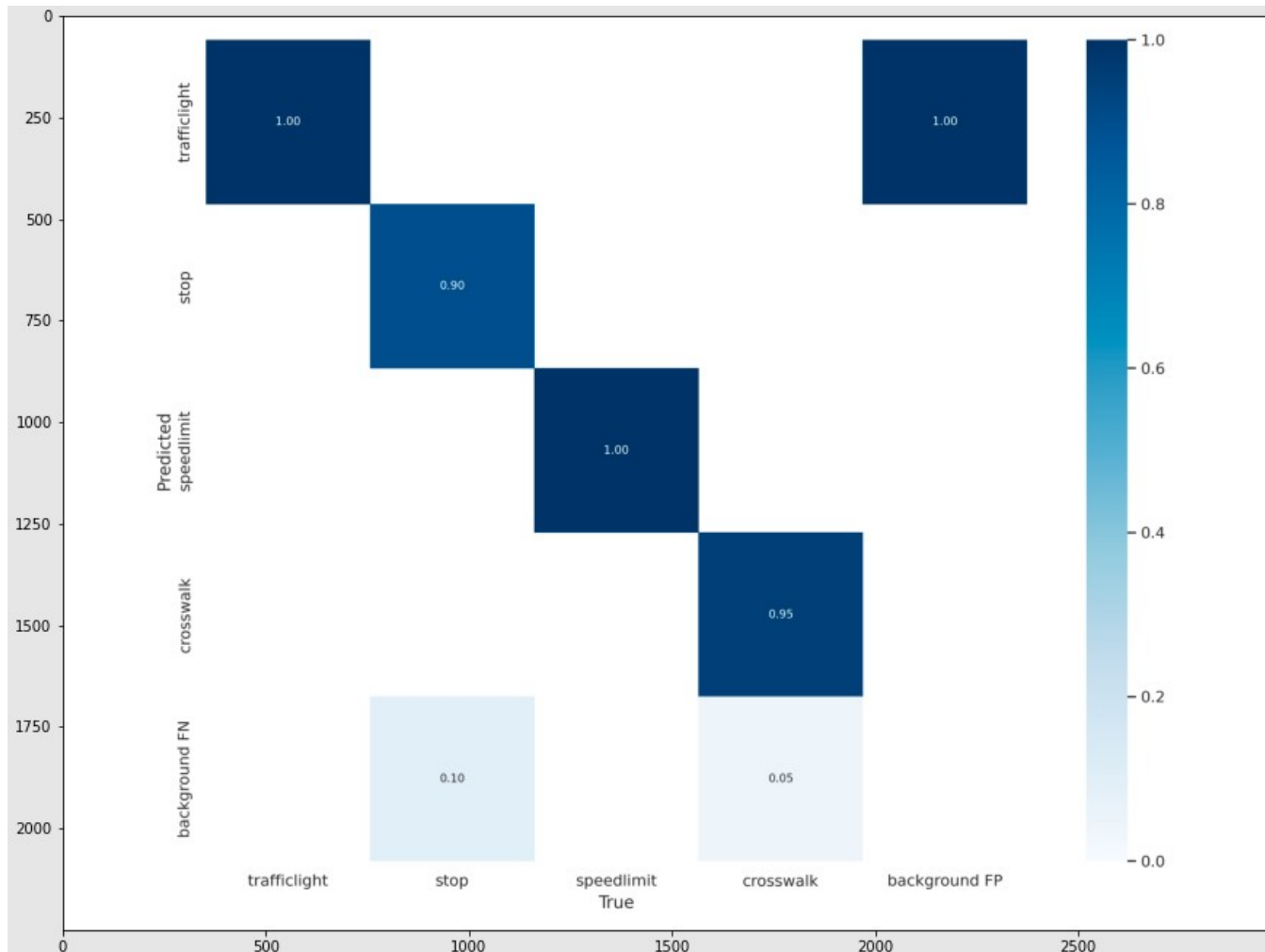
YOLOV5	StandfordDogs	Roadsigns	Whitebloods
Loss	0.00999	0.002799	0.00153
Accuracy	0.78	0.98	0.991



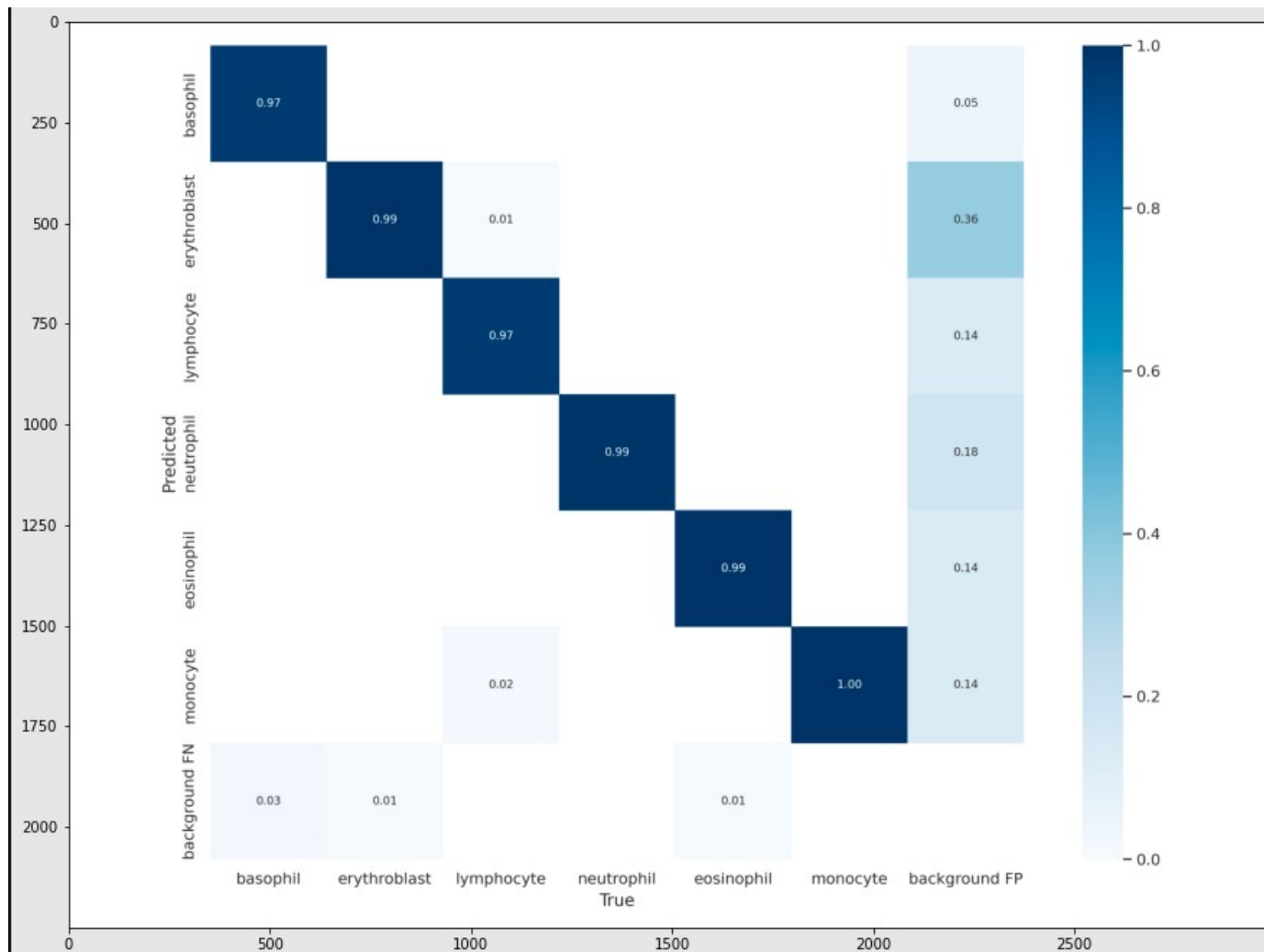
Yolov5 whitecells métrique



Matrice de confusion yolov5 roadsigns



Matrice de confusion yolov5 whitecells



Yolor

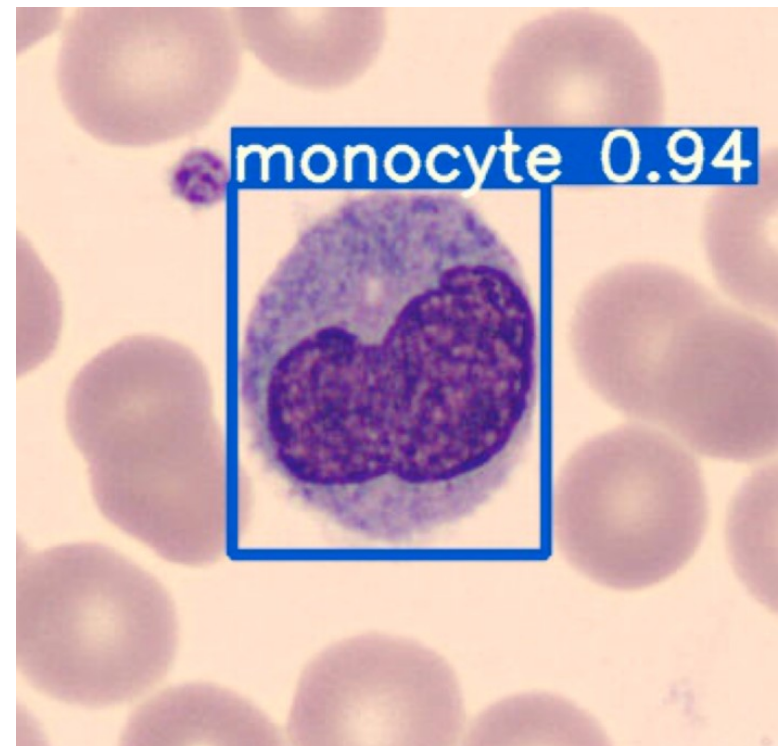
Anchor boxes pour chaque image (comme Yolov5)

- Préparation des données → fichier_conf.yaml + names :
 - Nombre de classes
 - Noms des classes
 - réglages des résolutions d images
 - path des images pour chaque jeu de données
 - conversion pascal voc to yolo format
 - creation de labels automatiquement
- Utilisation du Transfer Learning
- Data augmentation + fine tuning

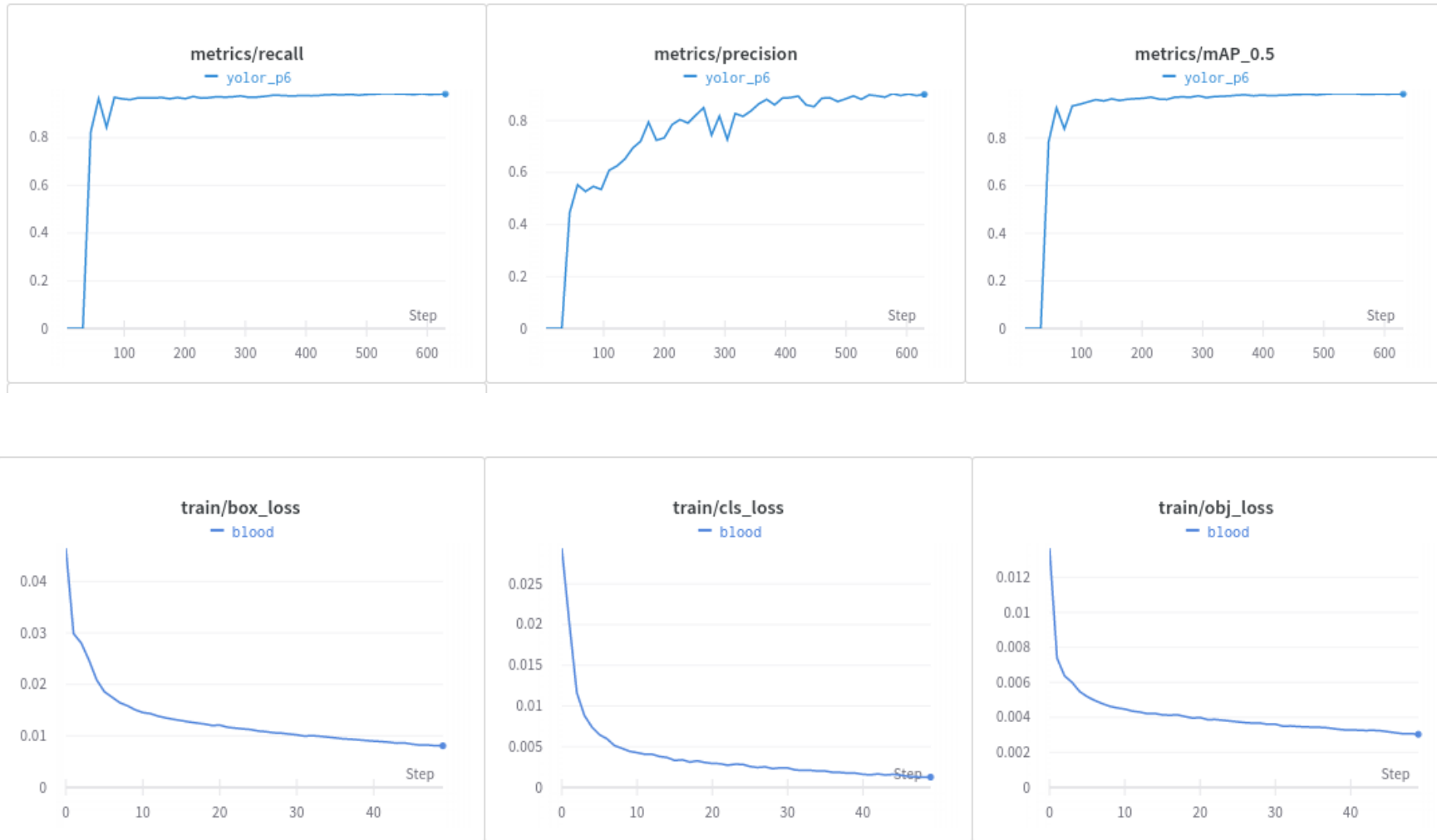
Yolor performances

Resultats : https://wandb.ai/ismail-azdad/YOLOR/reports/yolor-blood_cells-dataset-report---VmIldzoxOTQwNzM2

YOLOR	Roadsigns	Whitebloods
Loss	0.01113	0.003741
Accuracy	0.85	0.96



Yolor whitecells metric



Résultat

Liens vers les notebooks

https://github.com/ismailazdad/vision_detection_benchmark

accuracy	StanfordDogs	Roadsigns	Whitebloods
Xception	0.89	0.76	0.94
YOLOR	X	0.85	0.96
YOLOV5	0.78	0.98	0.99

synthèse

- État de l'art des architecture existantes
- Modèles références pour classification
 - Xception avec Transfer Learning
- Algorithme YOLOv5 :
 - - Ajout data augmentation
 - – Création rectangles d'encadrement
 - – Entraînement chronophage du modèle
 - – Meilleure classification des trois
- Algorithme YOLOr :
 - – Ajout data augmentation
 - – Création rectangles d'encadrement
 - – Entraînement plus rapide que YOLOv5
 - – Meilleure détection que Xception
 - – détection plus rapide

Thank you !