Projet 7 : Soutenance

Développez une preuve de concept

Gaëtan PELLETIER

Sommaire

- Présentation du projet
- État de l’art de la détection d’objets
- Modèles de référence
- Algorithme YOLOv4
- Algorithme YOLOv5
- Déploiement d’une application
- Synthèse

Présentation du projet

Présentation du projet

- Comment **améliorer** le projet P6 ?

- Pistes d'améliorations :
  - Augmentation performances prédictions
  - Ajout fonctionnalité

- Actuellement : **classification** races de chiens

- Ajout : **localisation** races de chiens

Présentation du projet - Méthodologie

- **Dataset** utilisée → images chiens **P6**
- **Localisation** des chiens sur chaque image
- **Modèles références** du P6 :
  - CNN sans pré-entraînement
  - Xception (*Transfer Learning*)
- **Algorithme de détection** de races de chiens
- **Métriques** pour évaluer classification et localisation
- Déploiement **application**

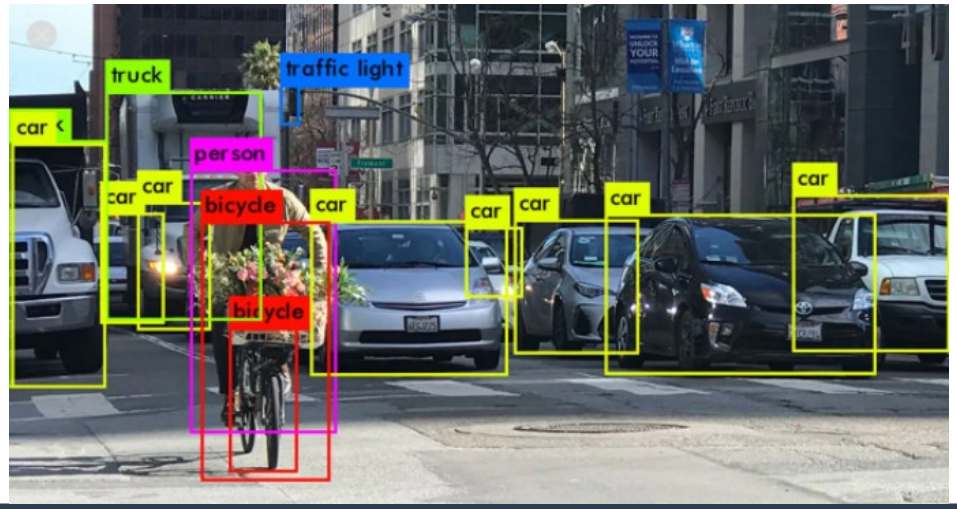
État de l’art de la détection d’objets

État de l'art de la détection d'objets

- Détection d'objets = classification + localisation
- Localisation = prédire un rectangle d'encadrement

  → coordonnées centre rectangle + hauteur + largeur

- Transformation des images : encadrer les cibles



État de l'art de la détection d'objets

- **Algorithmes développés :**
  - CNN classification + régression pour *anchor box*

    → R-CNN, Fast R-CNN, Faster R-CNN
  - FCN (*Fully Convolutional Network)*

    → R-FCN
  - Single shot

    → YOLO

- **Métriques :**
  - Classification → précision / recall / F1-score
  - Localisation → mAP@0.X

Modèles de référence

Modèles de référence

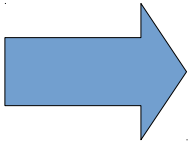
- **Dataset :**
  - Images du projet P6
  - Data augmentation
  - Nombre de classes : 10 ou 120
- **Performances modèles pour la classification :**

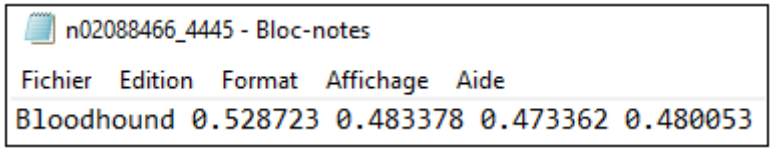
	CNN from scratch	Xception	Xception (120 breeds)
Loss	3,602	0,0115	0,3385
Accuracy	41,43 %	99,71 %	89,31 %

Algorithme YOLOv4

Algorithme YOLOv4

- **Rectangle d'encadrement** : *https://www.makesense.ai/*





- **yolov4.data :**
  - Nombre de classes + path vers leurs noms
  - path de chaque image / label pour chaque jeu de données
  - path pour sauvegarde poids d'entraînement

Algorithme YOLOv4

- **YOLOv4 utilise darknet :**

  → Freamework open source en C

- **Utilisation du *Transfer Learning***
  - dataset pour le pré-entraînement = COCO
  - Configuration non optimale (chronophage)

- **Fine-tuning avec dataset contenant 10 classes**

Algorithme YOLOv4

- **Performances** de l'algorithme :

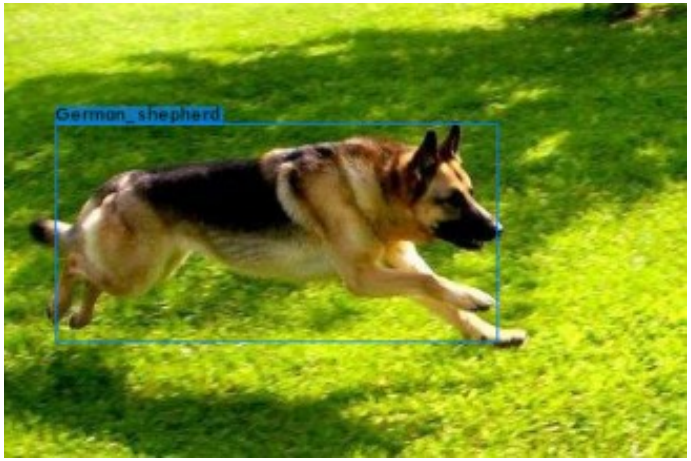
	Accuracy	Recall	F1-score	mAP@0,50
Valid. set	0,66	0,72	0,69	0,74
Test set	0,64	0,71	0,67	0,72

- **Comparaison** classification avec *baseline* :

	CNN from scratch	YOLOv4	Xception
Accuracy	41,43 %	64 %	99,71 %

Algorithme YOLOv4

- **Inférences :**



- **Limites du modèles :**
  - Entraînement chronophage avec configuration optimale
  - *Darknet* pas si évident
  - Inférences n'indiquent pas d'indice de confiance

Algorithme YOLOv5

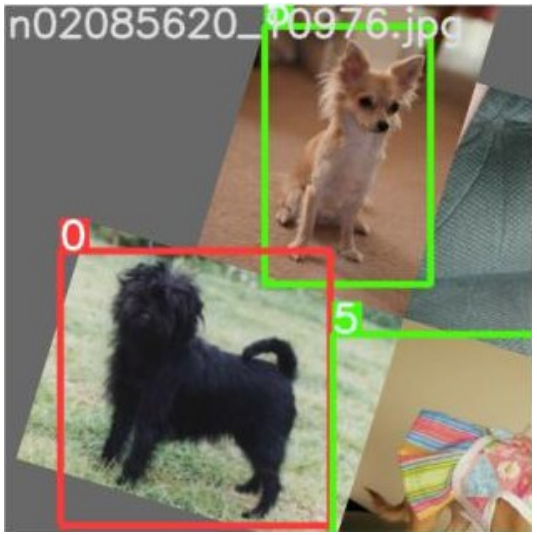
Algorithme YOLOv5

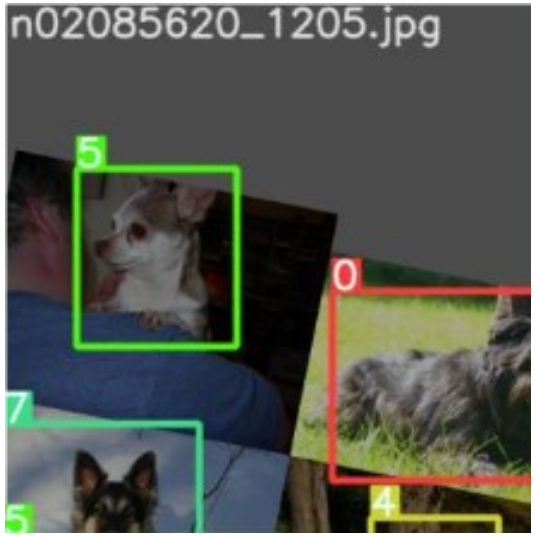
- *Anchor boxes* pour chaque image (Yolov4)
- Préparation des données → yolov5.yaml :
  - Nombre de classes
  - Noms des classes
  - path des images pour chaque jeu de données
  - cherche automatiquement path labels

    → remplace « images/ » par « labels/ » dans path pour images

Algorithme YOLOv5

- **Utilisation du *Transfer Learning***
  - dataset pour le pré-entraînement = COCO
  - Modèles avec et sans couches gelées
- **Data augmentation :**

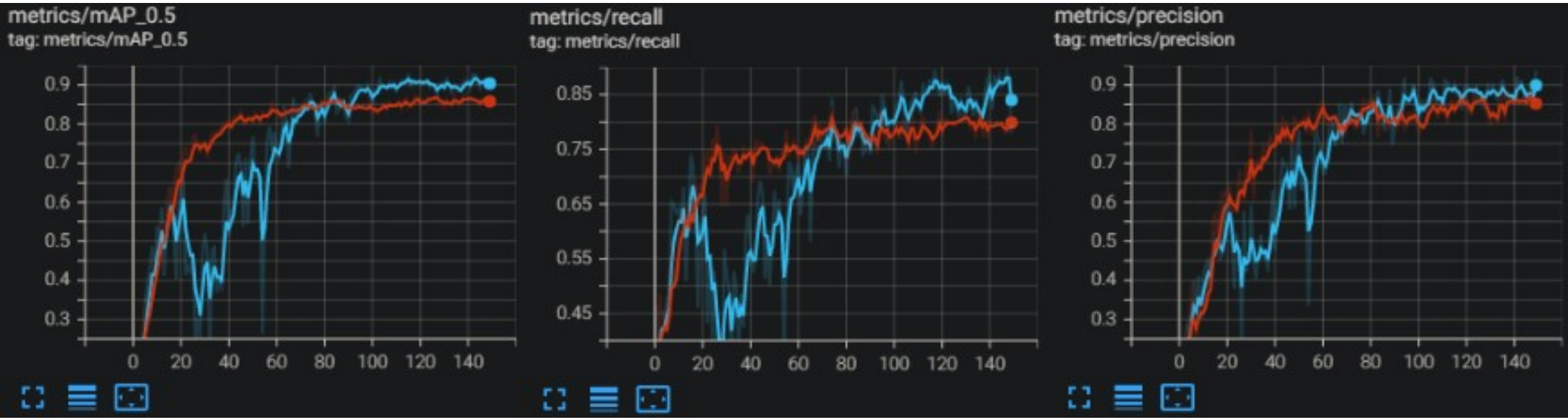




- **Fine-tuning avec dataset contenant 10 classes**

Algorithme YOLOv5

Différence d'apprentissage des modèles :

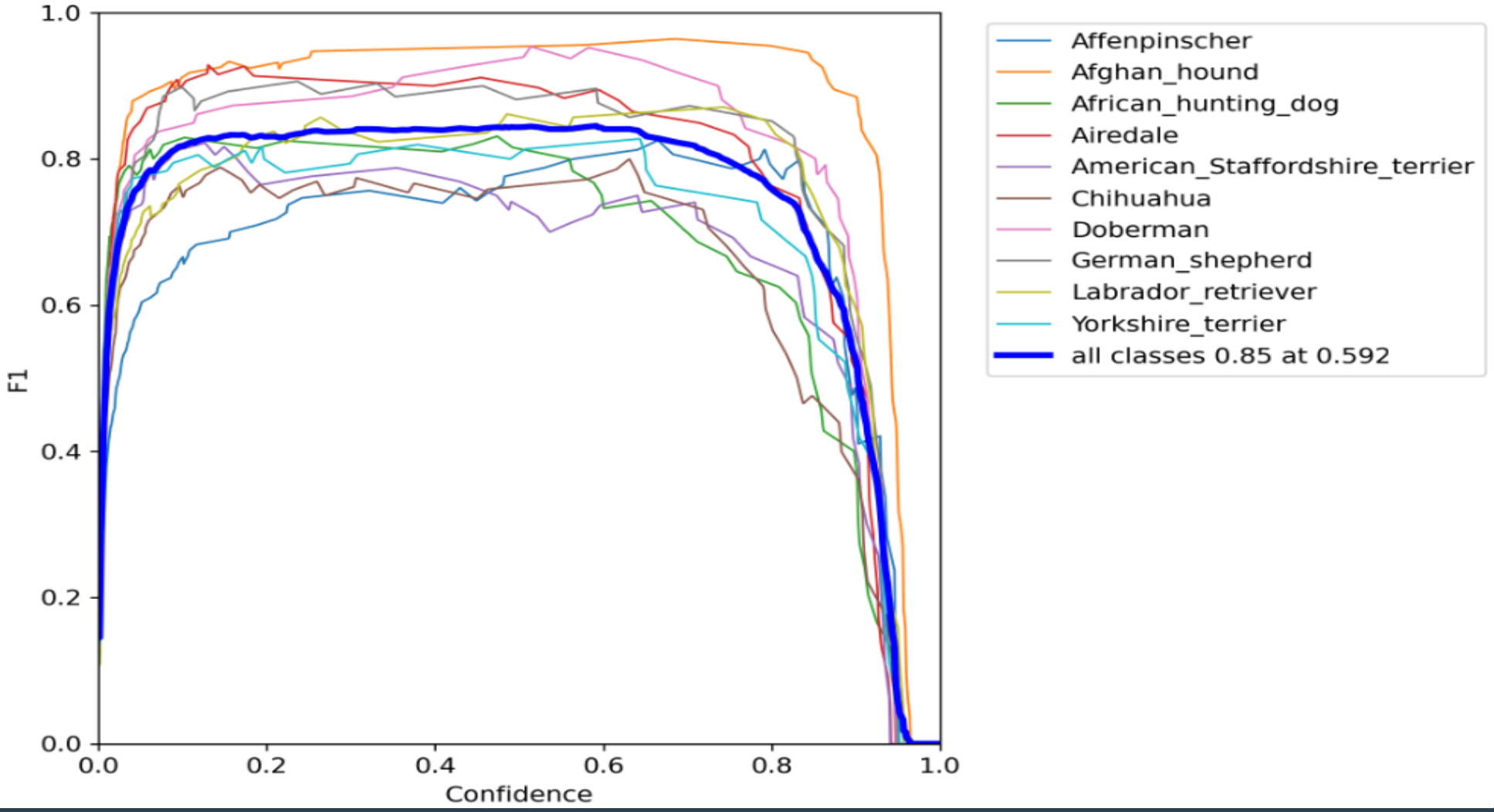


courbe rouge → YOLOv5 avec couches gelées

courbe bleu → YOLOv5 sans couches gelées

Algorithme YOLOv5

Choix seuil de confiance : *F1 curve* (*no frozen layers*)



Algorithme YOLOv5

- **Performances** des modèles :

	Accuracy	Recall	mAP@0,50
YOLOv4	0,64	0,71	0,72
YOLOv5 (couches gelées)	0,829	0,738	0,726
YOLOv5	0,937	0,779	0,9

- **Comparaison** classification avec *baseline* :

	CNN from scratch	YOLOv4	YOLOv5	Xception
Accuracy	41,43 %	64 %	93,7 %	99,71 %

Algorithme YOLOv4

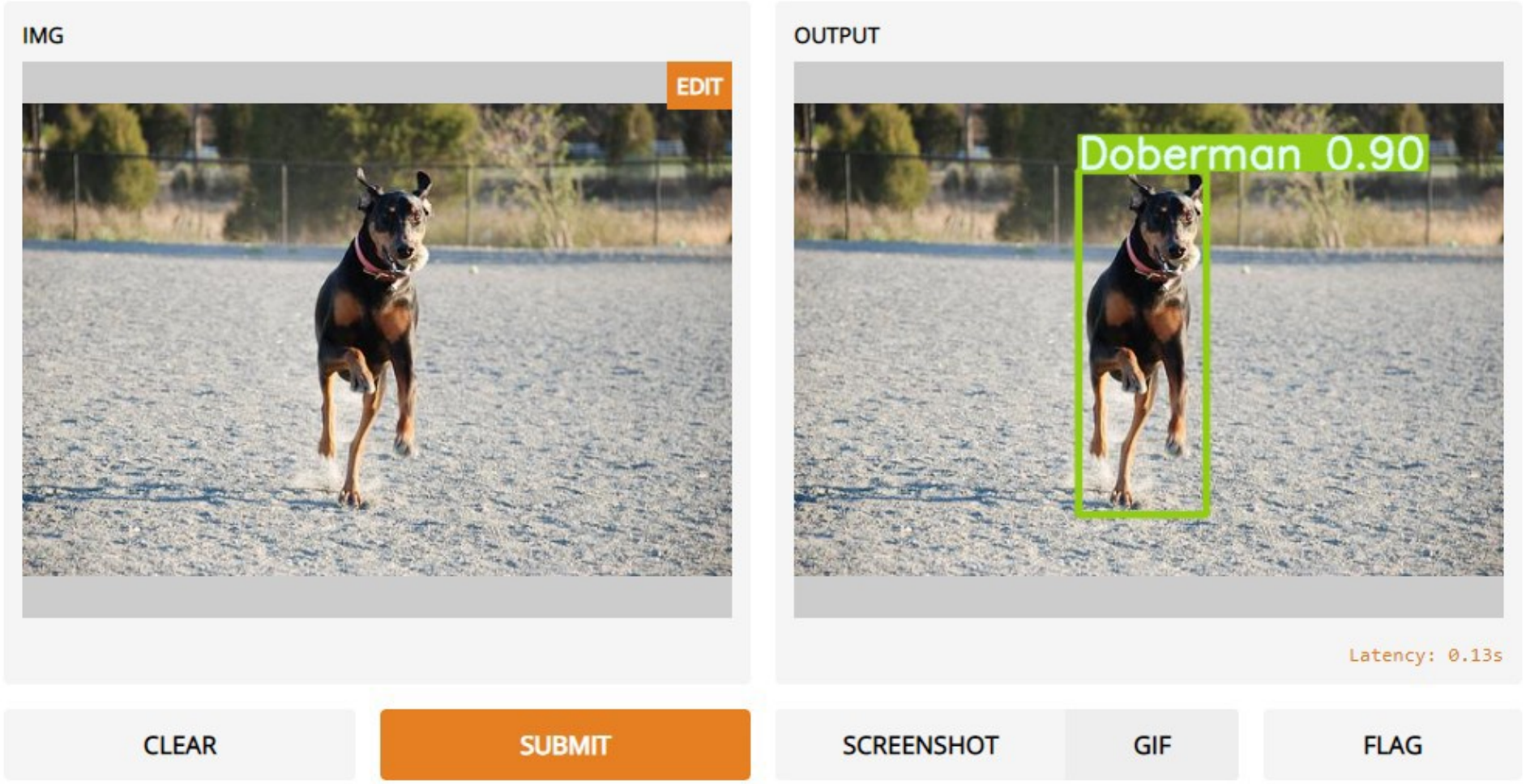
- **Inférences :**

- **Conclusions concernant YOLOv5 :**
  - Le modèle est meilleur que YOLOv4
  - Prédictions plus légitimes que Xception
    → pré-entraînement avec dataset COCO

Déploiement d’une application

Déploiement d’une application

Application avec Gradio :



Synthèse

Synthèse

- **Amélioration projet P6 : détection de chiens**
- **État de l'art des architecture existantes**
- **Modèles références pour classification :**
  - CNN sans pré-entraînement
  - Xception avec *Transfer Learning*
- **Algorithme YOLOv4 :**
  - Création rectangles d'encadrement
  - Entraînement chronophage du modèle
  - Meilleure classification que CNN sans pré-entraînement
- **Algorithme YOLOv5 :**
  - Ajout *data augmentation*
  - Entraînement plus rapide que YOLOv4
  - Meilleure classification que CNN sans pré-entraînement
  - Meilleure détection que YOLOv4
- **Déploiement de YOLOv5 dans une application**

Merci de votre attention

Annexes

Annexe 1 – Sources utilisées

- **Livre** :

Deep Learning avec Keras et TensorFlow 2nd édition, *A. Géron* (2020)

- **Papier de recherche** :

YOLOv4 : Optimal Speed and Accuracy of Object Detection, *Bochkovskiy, Wang, Mark Liao*
https://arxiv.org/pdf/2004.10934.pdf (2020)

- **Blogs** :

Deep Learning for Object Detection : A Comprehensive Review, *Joyce Xu*
https://towardsdatascience.com/deep-learning-for-object-detection-a-comprehensive-review-73930816d8d9 (2017)

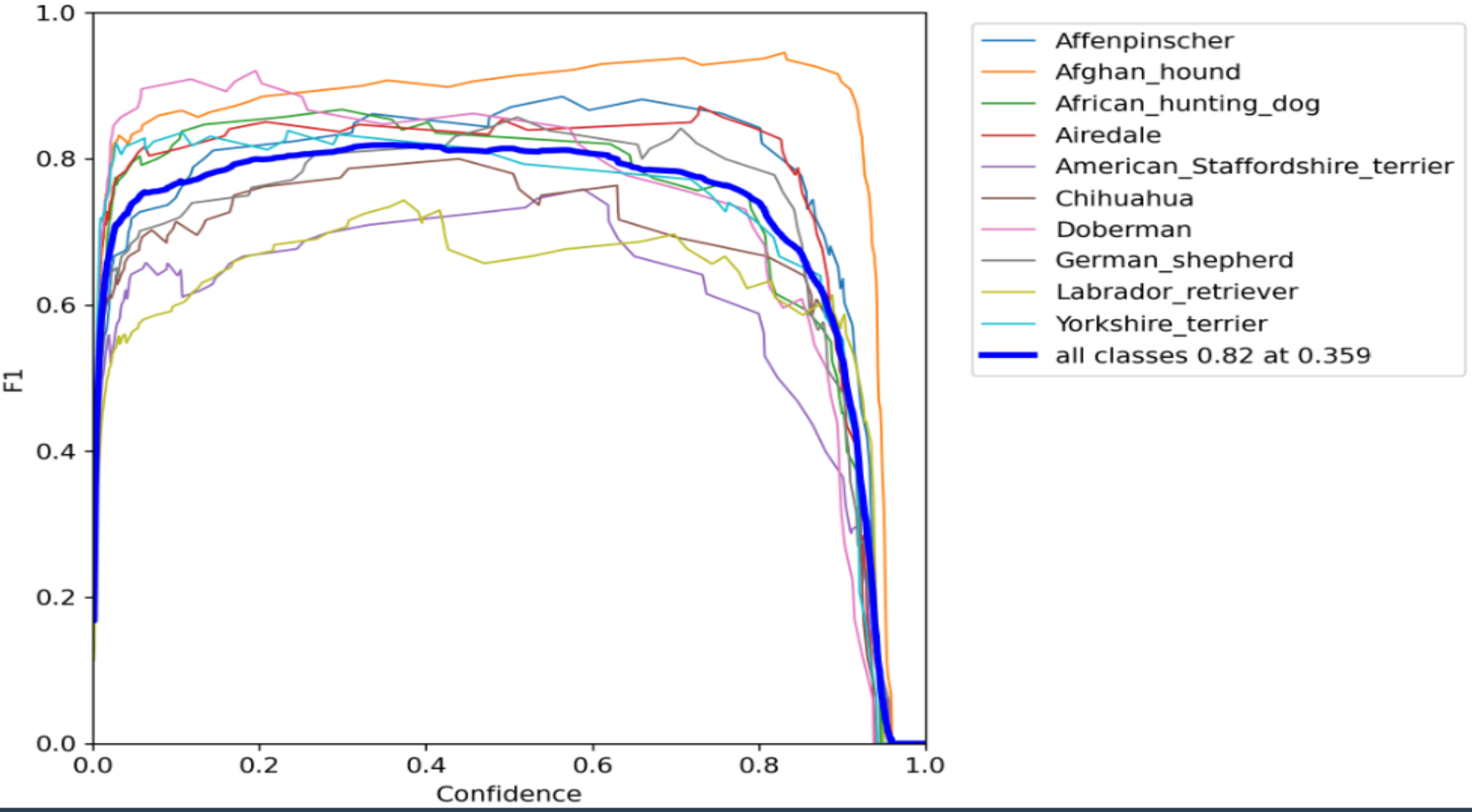
R-CNN, Fast R-CNN, Faster R-CNN, YOLO – Object Detection Algorithms, *R. Gandhi*
https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e (2018)

YOLOv4 on Google Colab : Train your Custom Datasetwith ease, *Quang Nguyen*
https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e (2020)

How to Train YOLOv5 On a Custom Dataset, *Jacob Solawetz, Joseph Nelson*
https://blog.roboflow.com/how-to-train-yolov5-on-a-custom-dataset/ (2020)

Annexes 2

Choix seuil de confiance : *F1 curve* (*frozen layers*)



Annexes 3

Matrice de confusion – YOLOv5 (best model)