

***Thibault Roussel***

**Rapport**

*Projet 7 IML*

Développez une preuve de concept

***SOMMAIRE***

1. Introduction
2. Présentation de mon choix de projet
3. Qu’est-ce que l’algorithme YoloV3 et comment fonctionne-t-il ?
4. Mon prototype
5. Comparaison des performances
6. Les avancées récentes
7. Synthèse

**1) Introduction**

Le domaine du machine learning, et plus généralement de la data science, évolue très rapidement. Il est donc important de développer l’habitude de se tenir au courant des avancées dans le domaine en effectuant une veille thématique.

On doit aussi être capable de monter rapidement en compétence sur une nouvelle thématique, en sachant effectuer une recherche et mettre en pratique un nouvel algorithme de façon autonome (POC).

Notre supérieur nous demande de trouver une méthode plus récente pour améliorer la méthode utilisée en production chez un client.

Nous devons donc choisir un des projet déjà réalisé durant ce parcours Ingénieur Machine Learning (IML) ou éventuellement un autre jeu de données ainsi qu’une nouvelle problématique et choisir un algorithme récent, l’utiliser sur nos données et comparer les performances de ce nouvel algorithme avec une approche classique ou plus ancienne.

**2) Présentation de mon choix de projet**

La thématique choisie est la détection automatique d’objets dans des images.

Ce projet vient en continuité du Projet 6 «*Classez des images à l'aide d'algorithmes de Deep Learning* » dans lequel nous avions réalisé un algorithme de détection de la race du chien sur une photo grâce, entre autre, à une approche de transfer learning, c’est-à-dire en utilisant un réseau de neurones déjà entraîné, et en le modifiant pour répondre à notre problème.

L’idée de ce projet part du constat de la présence de quelques humains, de quelques chats ou de plusieurs chiens dans les images du dataset du Projet 6. On va donc tenter de détecter la présence ou non d’un ou plusieurs chiens dans les images du dataset, les encadrer et enfin recadrer les images afin de n’avoir que le ou les chiens présents sur les photos.

La méthode utilisée est l’algorithme de détection d’objet YOLO (You Only Look Once) dans sa 3ème version. C’est l'un des algorithmes de détection d'objets le plus rapides et, nous avons 20580 images à traiter, sa rapidité est un véritable atout.

**3) Qu’est-ce que l’algorithme YoloV3 et comment fonctionne-t-il ?**

You Only Look Once (YOLO) est un algorithme de détection d’objets connu pour sa grande précision et sa rapidité.

YOLOV3 est un réseau de neurones convolutif (CNN) unique, il est uniquement composé de couches de convolution, il y en a 53. La détection d’un objet peut être faite à 3 échelles différentes grâce à 3 outputs.

En sortie de celui-ci on trouve des « features maps », des boxes contenants les objets détectés ainsi que les probabilités associées aux boxes à l’aide de la régression logistique. YOLOv3 réduit la taille des images d’un facteur 32, appelé la stride du réseau. Le réseau de neurones prend des images de 416\*416 pour fournir une feature map de taille 13\*13 .

Si des boxes se chevauchent, on garde seulement celle qui a le score d’objectivité le plus élevé.

Ce réseau de neurones convolutif YOLOV3 divise l'image en régions et prédit les cadres de délimitation et les probabilités pour chaque région.

**手机屏幕截图

描述已自动生成**YOLOV3 ne parcours pas l’image en coulissant le long de celle-ci comme les autres algorithmes de détection d’objets mais il l’analyse dans sa globalité. Il analyse donc une image qu’une seule fois avant de donner un résultat, cela le rend bien plus rapide, plus de 1000 fois plus rapide que R-CNN et 100 fois plus rapide que Fast R-CNN.

**4) Mon prototype**

On importe "yolov3.weights", "yolov3.cfg" ainsi que les classes pouvant être détectées.

On créé une fonction « detect\_and\_crop » permettant de détecter le ou les chiens présents sur une image et de rogner les images autour des chiens détectés.

Le début du code utilisé provient du site « *pysource.com* ».

La fonction « detect\_and\_crop » :

* Vérifie si un ou plusieurs chiens sont détectés sur l’image
* Récupère les coordonnées des boxes contenant les chiens
* Vérifie que les boxes ont au moins une « shape » de (150,150,3) pour conserver les images les plus qualitatives
* On enregistre l’image du premier chien dans un nouveau fichier image dans le sous fichier de la race auquel le chien appartient.
* Puis on enregistre les suivant si il y en a

Exemple de traitement d’une image :

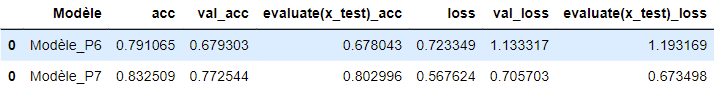
Une image contenant texte, mammifère, chien

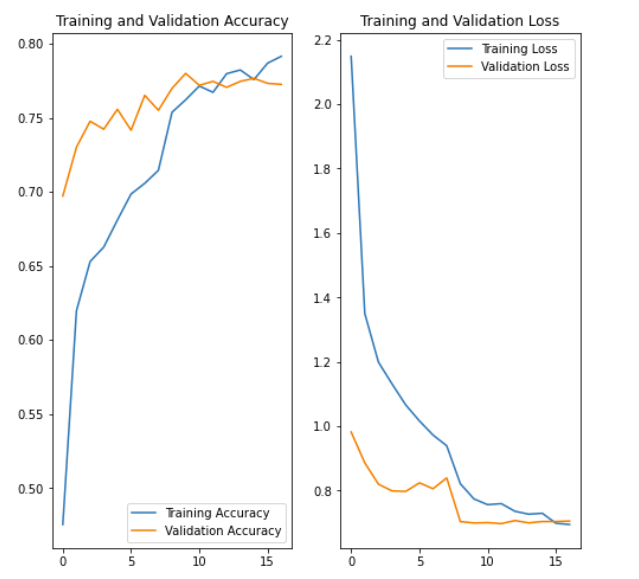
Description générée automatiquementUne image contenant chien, personne, mammifère, posant

Description générée automatiquement**Avant : Apres :**

**5) Comparaison des performances**

Tableau de comparaison des performances :



**Modèle P6** **Modèle P7**

On constate que l’accuracy sur nos données test de l’ancien modèle est de 67,8% contre 80,2% pour notre nouveau modèle dont les images ont été traité avec YOLOV3 soit une augmentation d’environ 12,4%.

**6) Les avancées récentes**

L’algorithme le plus récent est l’algorithme YOLOV5 mais je parlerais ici de YOLOV4, car la 5ème version étant très jeune, bon nombre d’articles la voie comme controversée.

Par rapport à YOLOV3, l’AP (précision) de YOLOV4 et sa FPS (cadence d'images par seconde) sont respectivement améliorées de 10 % et 12 %. Une couche de convolution a été ajouté au modèle.

Les deux versions sont très proches et, pour mon projet, j’ai choisi YOLOV3 car j’ai facilement compris comment l’utiliser. De plus, pour détecter des objets en premier plan et or vidéo, les deux algorithmes ont des résultats sensiblement identiques.

**7) Synthèse**

Nous avons trouvé une méthode plus récente pour améliorer notre réseau de neurones du P6. Les résultats ont augmenté de 12,4%. L’algorithme YOLOV3, utilisé pour détecter les chiens pour prétraiter le dataset avant le CNN Resnet50 a bien permis d’améliorer nos résultats.