



Projet 8 – Participer à une compétition Kaggle

TENSORFLOW – HELP PROTECT THE GREAT BARRIER REEF
ELISE ANDRO

Table des matières

Description de la compétition par Kaggle	2
But de la compétition	2
Le contexte	2
Evaluation des soumissions	3
Les données	4
Exploration et traitement des données	5
Exploration des données : plusieurs questions sont posées	5
Traitement des données	6
Les modèles testés	7
Qu'est-ce que l'algorithme YOLO ?	7
Résultats	8
Conclusion	8
Liens utiles	9

Description de la compétition par Kaggle

Compétition choisie : TensorFlow – Help Protect the Great Barrier Reef

But de la compétition

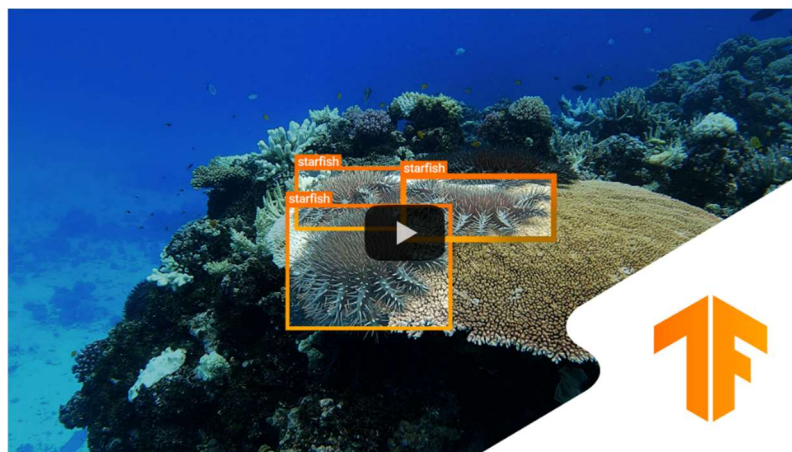
L'objectif de ce concours est **d'identifier avec précision les étoiles de mer en temps réel** en construisant un **modèle de détection d'objets** entraîné sur des vidéos sous-marines de récifs coralliens.

Votre travail aidera les chercheurs à identifier les espèces qui menacent la grande barrière de corail australienne et à prendre des mesures éclairées pour protéger le récif pour les générations futures.

Le contexte

La magnifique grande barrière de corail australienne est le plus grand récif de corail du monde et abrite 1 500 espèces de poissons, 400 espèces de coraux, 130 espèces de requins, de raies et une grande variété d'autres espèces marines.

Malheureusement, le récif est menacé, en partie à cause de la surpopulation d'une étoile de mer en particulier - l'étoile de mer à couronne d'épines mangeuse de corail (ou **COTS** en abrégé). Des scientifiques, des opérateurs touristiques et des gestionnaires de récifs ont mis en place un programme d'intervention à grande échelle pour contrôler les épidémies de COTS à des niveaux écologiquement durables.



Pour savoir où se trouvent les COTS, une méthode traditionnelle d'étude des récifs, appelée « Manta Tow », est réalisée par un plongeur tuba. Remorqués par un bateau, ils évaluent visuellement le récif, s'arrêtant pour enregistrer les variables observées tous les 200 m. Bien que généralement efficace, cette méthode est confrontée à des limites claires, notamment l'évolutivité opérationnelle, la résolution des données, la fiabilité et la traçabilité.

La Great Barrier Reef Foundation a mis en place un programme d'innovation pour développer de nouvelles méthodes d'enquête et d'intervention afin de fournir un changement radical dans le contrôle des COTS. Les **caméras sous-marines collecteront des milliers d'images** de récifs et la technologie de l'IA pourrait considérablement améliorer l'efficacité et l'échelle à laquelle les gestionnaires de récifs détectent et contrôlent les épidémies de COTS.

Pour faire évoluer les systèmes d'arpentage vidéo, l'agence scientifique nationale australienne, CSIRO s'est associée à Google pour développer une technologie innovante d'apprentissage automatique capable d'**analyser de grands ensembles de données d'images avec précision, efficacité et en temps quasi réel.**

Evaluation des soumissions

Cette compétition est évaluée sur le **score F2 à différents seuils d'intersection sur union (IoU)**. Le score F2 accorde plus d'importance au recall qu'à la précision, car dans ce cas, il est logique de tolérer certains faux positifs afin de s'assurer que très peu d'étoiles de mer sont manquées.

Formule du F2 score :

$$F2 = 5 \cdot \frac{precision \cdot recall}{4 \cdot precision + recall}$$

La métrique balaie les seuils d'IoU dans la plage de 0,3 à 0,8 avec un pas de 0,05, calculant un score F2 à chaque seuil. Par exemple, à un seuil de 0,5, un objet prédit est considéré comme un "hit" si son IoU avec un objet True est d'au moins 0,5.

Un vrai positif est la première soumission dans un échantillon avec une IoU supérieure au seuil par rapport à une bounding box qui ne match pas.

Une fois que toutes les bounding box de soumission ont été évaluées, toutes les bounding box de soumission sans correspondance sont des faux positifs ; toutes les bounding box de solution non appariées sont des faux négatifs.

Le score F2 final est calculé comme la moyenne des scores F2 à chaque seuil IoU. À l'intérieur de chaque seuil d'IoU, la métrique de concurrence utilise une micro moyenne ; chaque vrai positif, faux positif et faux négatif a le même poids les uns par rapport aux autres vrai positif, faux positif et faux négatif.

Dans votre soumission, les participants sont également invités à fournir un niveau de confiance pour chaque bounding box. Les bounding box sont évalués en fonction de leurs niveaux de confiance. Cela signifie que les bounding box avec une confiance plus élevée seront vérifiées en premier pour les correspondances avec les solutions, ce qui détermine quelles boîtes sont considérées comme des vrais et des faux positifs.

Les données

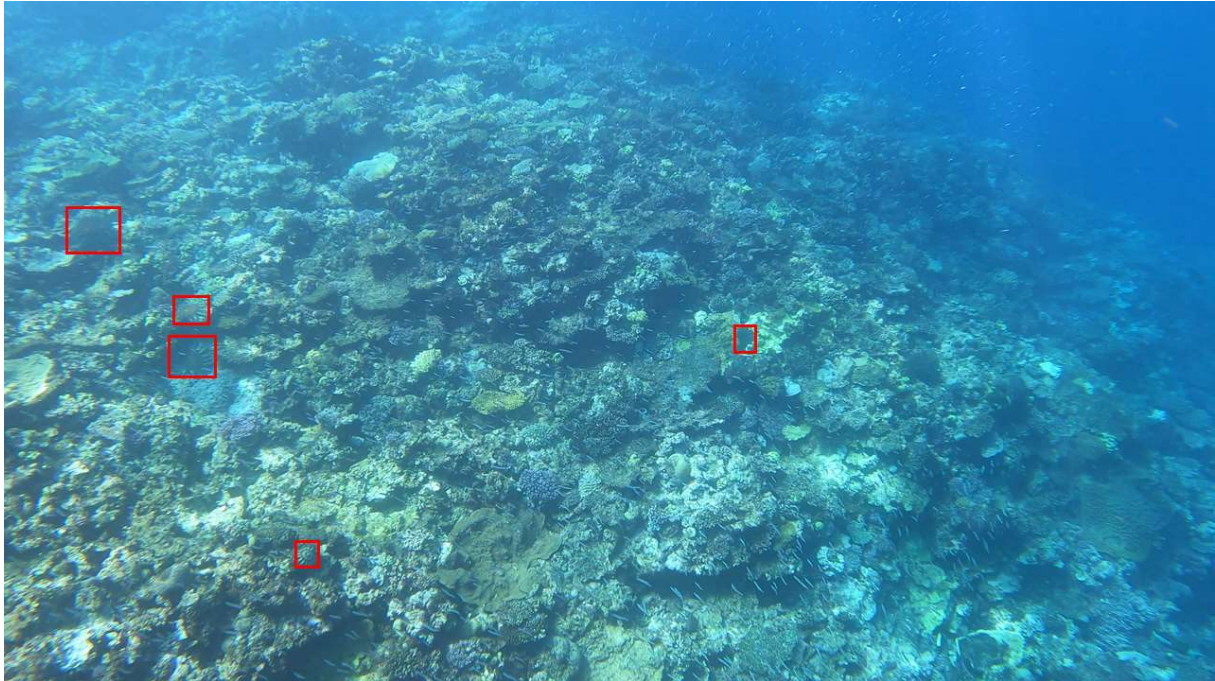
Les organisateurs de la compétition mettent à disposition un jeu de données pour l'entraînement des modèles. Ce jeu est composé de trois vidéos de plongeurs sous-marins, filmant la barrière de corail. Les vidéos sont divisées en image (séquences). Le jeu de données est donc la liste des images des trois vidéos.

Pour chaque image, on a les informations suivantes :



- `video_id`- Numéro d'identification de la vidéo dont faisait partie l'image. Les identifiants vidéo ne sont pas ordonnés de manière significative.
- `video_frame`- Le numéro de trame de l'image dans la vidéo. Attendez-vous à voir des lacunes occasionnelles dans le numéro de trame à partir du moment où le plongeur a fait surface.
- `sequence`- ID d'un sous-ensemble sans lacunes d'une vidéo donnée. Les identifiants de séquence ne sont pas ordonnés de manière significative.
- `sequence_frame` - Le numéro de trame dans une séquence donnée.
- `image_id` - Code d'identification de l'image, au format '{video_id}-{video_frame}'
- `annotations`- Les cadres de délimitation de toutes les détections d'étoiles de mer dans un format de chaîne pouvant être évalué directement avec Python. N'utilisez pas le même format que les prédictions que vous soumettrez. Non disponible dans test.csv. Un cadre de délimitation est décrit par la coordonnée en pixels (x_min, y_min) de son coin supérieur gauche dans l'image ainsi que sa largeur et sa hauteur en pixels.

	video_id	sequence	video_frame	sequence_frame	image_id	annotations	img_path
0	0	40258	0	0	0-0	[]	../input/tensorflow-great-barrier-reef/train_i...
1	0	40258	1	1	0-1	[]	../input/tensorflow-great-barrier-reef/train_i...
2	0	40258	2	2	0-2	[]	../input/tensorflow-great-barrier-reef/train_i...
3	0	40258	3	3	0-3	[]	../input/tensorflow-great-barrier-reef/train_i...
4	0	40258	4	4	0-4	[]	../input/tensorflow-great-barrier-reef/train_i...

Exemple d'image avec ses annotations :

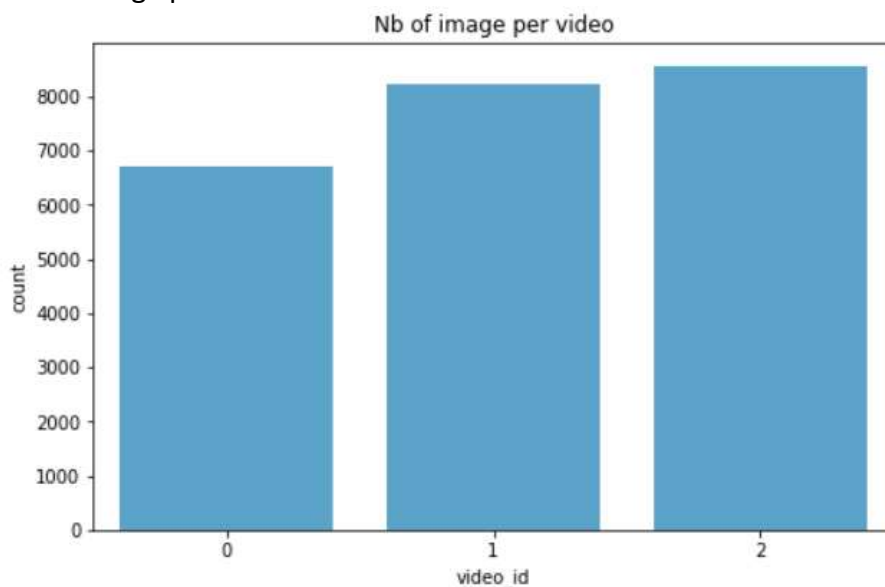


Exploration et traitement des données

Pour l'exploration et le traitement des données, je me suis inspirée du notebook «  Data Analysis and Prediction  » de KARTIK KHANDELWAL.

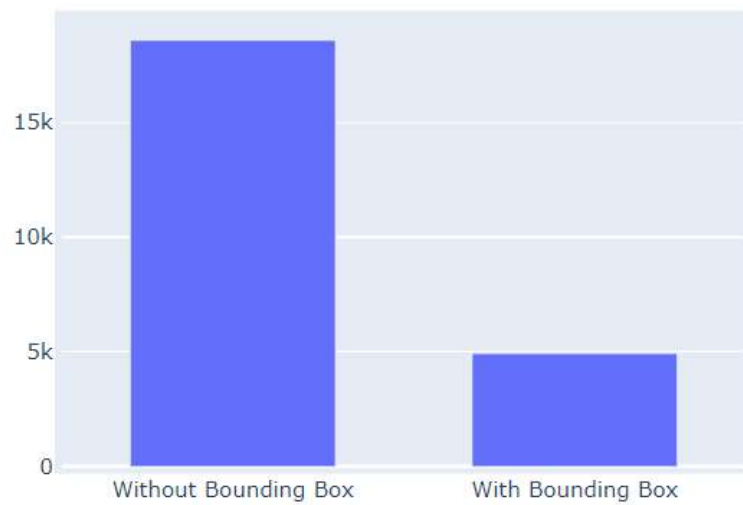
Exploration des données : plusieurs questions sont posées.

Combien y a-t-il d'image par vidéo ?



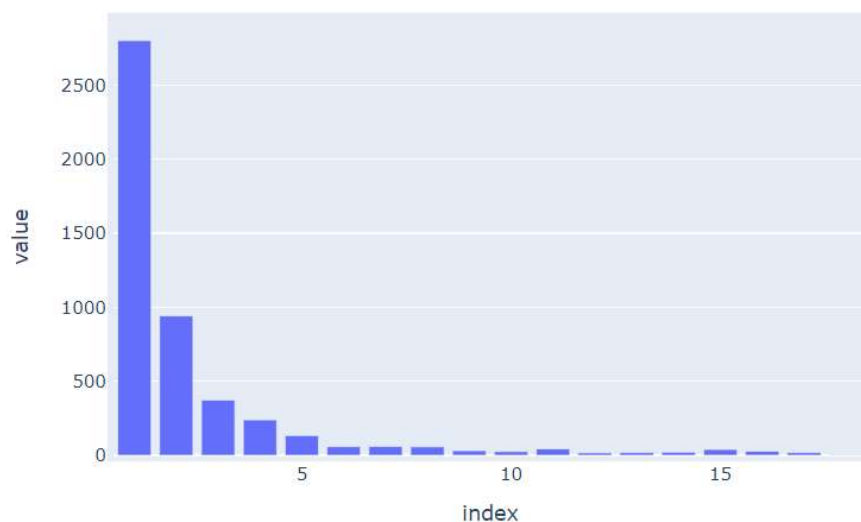
Combien d'image contiennent des étoiles de mer (COTS) ?

Image with Starfish



Combien y a-t-il de COTS sur les images qui en contiennent ?

Count of Bounding Boxes per image



Les images ont-elles toute la même taille ?

Oui, elles sont toutes au format 1280 x 720.

Traitement des données

Peu de traitement a été effectué sur ce projet.

Il est néanmoins nécessaire de retravailler les formats des annotations pour les mettre au format demandé par les modèles comme input : [x,y,width,height]

Et de préparer les chemins d'accès vers chaque image pour les phases d'entraînement des modèles (colonne img_path).

Les modèles testés

Qu'est-ce que l'algorithme YOLO ?

YOLO ou You Only Look Once, est un algorithme de détection d'objets en temps réel populaire. YOLO combine ce qui était autrefois un processus en plusieurs étapes, en utilisant un seul réseau neuronal pour effectuer à la fois la classification et la prédiction des boîtes englobantes pour les objets détectés. En tant que tel, il est fortement optimisé pour les performances de détection et peut fonctionner beaucoup plus rapidement que d'exécuter deux réseaux de neurones distincts pour détecter et classer les objets séparément. Bien que les itérations suivantes présentent de nombreuses améliorations, l'idée de base derrière l'architecture reste la même. YOLOv1, appelé simplement YOLO, peut fonctionner plus rapidement que la détection d'objets en temps réel à 45 images par seconde, ce qui en fait un excellent choix pour les applications qui nécessitent une détection en temps réel. Il regarde l'image entière à la fois et une seule fois - d'où le nom You Only Look Once - qui lui permet de capturer le contexte des objets détectés. Cela réduit de moitié le nombre de détections faussement positives qu'il effectue sur les R-CNN qui regardent séparément différentes parties de l'image. De plus, YOLO peut généraliser les représentations de divers objets, ce qui le rend plus applicable à une variété de nouveaux environnements.

YOLO est basé sur l'idée de segmenter une image en images plus petites. L'image est divisée en une grille carrée de dimensions $S \times S$. La cellule dans laquelle réside le centre d'un objet. Chaque cellule prédit les boîtes englobantes et un score de confiance pour chaque boîte. La valeur par défaut de cette architecture est que le modèle prédit deux cadres de délimitation. Le score de classification sera compris entre 0 et 1, 0 étant le niveau de confiance le plus bas et 1 le plus élevé ; si aucun objet n'existe dans cette cellule, les scores de confiance doivent être 0, et si le modèle est complètement certain de sa prédiction, le score doit être 1. Ces niveaux de confiance capturent la certitude du modèle qu'il existe un objet dans cette cellule et que la boîte englobante est exacte. Chacune de ces boîtes englobantes est composée de 5 nombres : la position x , la position y , la largeur, la hauteur et la confiance. Les coordonnées (x, y) représentent l'emplacement du centre de la boîte englobante prévue, et la largeur et la hauteur sont des fractions par rapport à la taille de l'image entière. La confiance représente l'IOU entre la boîte englobante prévue et la boîte englobante réelle, appelée boîte de vérité terrain. L'IOU signifie Intersection Over Union et est la zone de l'intersection des boîtes de vérité prédite et terrain divisée par la zone de l'union des mêmes boîtes de vérité prédite et terrain.

En plus de générer des cadres de délimitation et des scores de confiance, chaque cellule prédit la classe de l'objet. Cette prédiction de classe est représentée par une longueur de vecteur unique, le nombre de classes dans l'ensemble de données. Cependant, il est important de noter que si chaque cellule peut prédire n'importe quel nombre de boîtes englobantes et scores de confiance pour ces boîtes, elle ne prédit qu'une seule classe.

Les algorithmes Yolo ont été améliorés au fur et à mesure des années. Pour la compétition Kaggle, deux modèles ont été testés : YoloV5 sorti en 2020 et YoloX sorti en 2021. Pour les deux modèles, il s'agit de modèle pré-entraîné à utiliser en transfer learning.

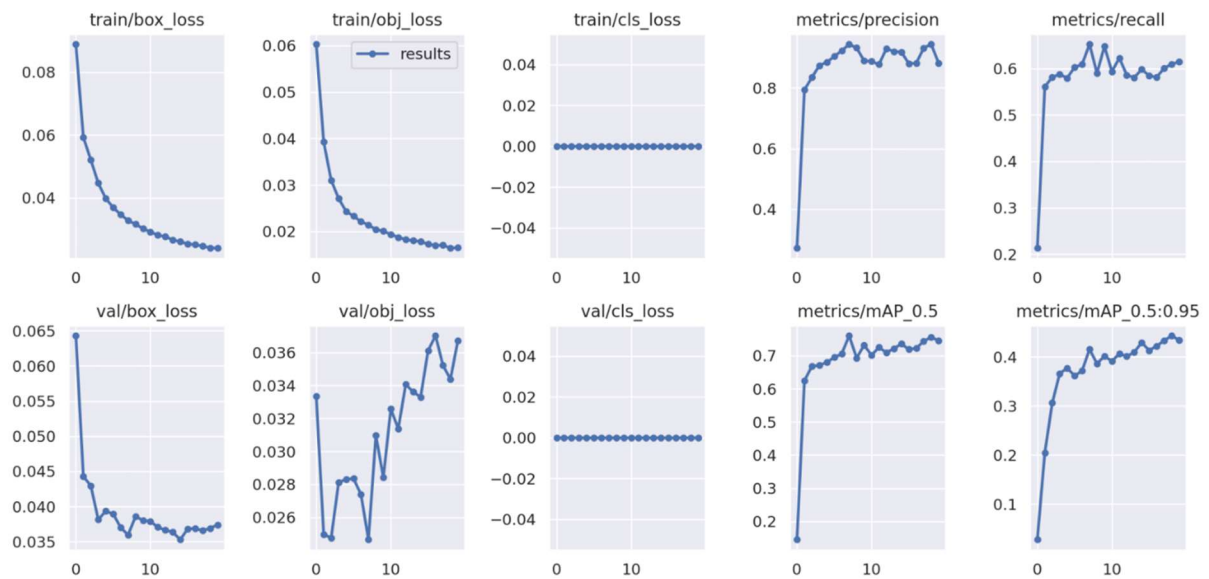
Résultats

Résultats des soumissions à la compétition Kaggle.
Les deux modèles ont été entraînés sur 20 epochs

Score de YOLOv5 : 0.355

Score de YOLOX : 0.507

Résultats détaillés de l'entraînement de YOLOv5



Conclusion

Le modèle YOLOX, bien qu'entraîné sur un nombre limité d'époches, a permis d'obtenir un score plus élevé (0.507). En comparaison des autres soumissions de compétiteurs au moment de la soutenance de ce projet, le score est 224^e sur 609. Le meilleur score étant de 0.619.

Liens utiles

Kernel Kaggle

<https://www.kaggle.com/elisead/gbr-yolox-train-infer>

<https://www.kaggle.com/elisead/gbr-yolov5-train>

Git

<https://github.com/EliAd152693/OpenClassroomsProjet8>