**Rapport du 10/08/2021**

Le projet poulailler a pour but d’écrire un algorithme qui permet de détecter des poules sur une image. Pour y parvenir, on a fait un transfert Learning de YOLO. Le transfert Learning consiste à ré-exploiter les connaissances acquises dans une tâche source à une tâche cible et YOLO est un système(algorithme) utilisant les réseaux de neurones pour détecter et analyser les objets en temps réel.

Nous avons d’autres algorithmes de détection d’objet qui sont l’OpenCV, le tensorflow et autres, mais l’avantage de Yolo est qu’il s’adapte plus facilement aux différents codes.

Pour écrire notre algorithme de détection d’objet, plusieurs étapes sont à franchir :

1. Collecte de données (création d’un dataset) :

A cette étape, on a créé un dataset contenant un ensemble d’images de poules. On a créé deux sous dossiers images et labels dans un dossier nommé train\_data et dans chaque dossier on a créé deux dossiers train et val. Pour notre cas, on a utilisé 75% d’images d’entrainement et 25% d’images test.

Les 75% d’images test sont stockés dans le dossier images/train et les 25% sont stockés dans le dossier images/val.

1. Etiquetages :

A ce stade, on a utilisé [Makesense.ai](https://www.makesense.ai/) qui est un logiciel d’étiquetage.

On importe le dossier images/train dans le logiciel d’étiquetage, on appuie sur le bouton ‘objet detection’, ensuite on appui dans sur le bouton ‘+’ situé en haut à gauche pour créer le nom de l’étiquette puis on appuie sur le bouton ‘Start project’ pour commencer le projet d’étiquetage. L’étiquetage consiste à repérer et encadrer sur chaque image les poules. Après avoir étiqueté toutes les images qui sont dans le dossier images/train on clique sur le bouton ‘action’ et on exporte les annotations dans un fichier puis on le décompresse dans le dossier labels/train. On fait de même pour les images contenues dans le dossier images/val.

1. Entrainement du modèle :

A ce niveau, on a utilisé le laboratoire Google colab qui est un notebook permettant d’exécuter les scripts python et autres. On a choisi colab car il ne tient pas compte de la vitesse du processeur d’un ordinateur pour exécuter un code. Après être connecté sur la plateforme on a écrit un code pour mettre en place le modèle. Pour le faire, on a cloné yolov5, installé les dépendances, de-zippé le dossier train-data et ensuite entrainé le modèle. Puis, on a visualisé les métriques de performance du modèle c’est-à-dire la perte et la précision. Et enfin, on a testé la fiabilité du modèle sur des nouvelles images.