Compte Rendu 3 : SAE 501

Implémentation de l'IA via flutter

Nous avons pu durant cette semaine avancer sur l'implémentation de l'IA sur notre application, nous sommes initialement parti pour la première version du projet avec le modèle **MobileNet**, or nous nous sommes rendu compte que la version que nous avons utilisé de celles-ci ne parvenait pas à encadrer les objets reconnues ce qui ne correspondait pas à nos attentes, nous avons donc opté pour l'intégration du modèle **YOLO**, mieux adapté pour fournir des boites de détections autour des objets. De plus, YOLO se trouve être plus performant pour les tâches en temps réel que MobileNet et a aussi une version optimisée pour les appareils mobiles.

Nous avons donc ajouté à notre projet le modèle **YOLO v5** en format **TFLite** ainsi qu'un fichier **labels.txt** contenant les noms des objets que l'IA est capable de reconnaître.

L'ajout de plugins et bibliothèques a été nécessaire pour implémenter l'IA, nous avons utilisé:

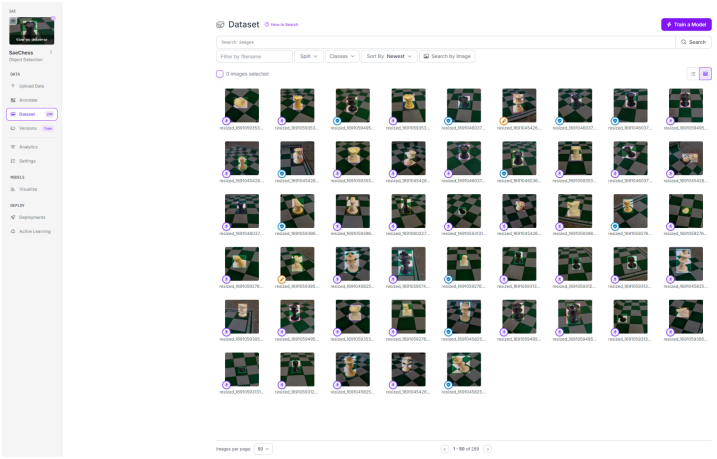
- Tflite : pour charger et utiliser le modèle YOLO en Tflite
- Caméra : pour l'accès à la caméra en direct

Pour l'affichage des résultats de la détection, notre objectif était de visualiser les résultats dans un cadre étiqueté du nom résultat, pour cela nous avons utilisé **RecognitionPainter**, sa mise en place se fait par le biais de la création d'une classe de peinture personnalisée (en utilisant **CustomPainter**), soit RecognitionPainter dans notre cas, celle-ci doit se voir capable de dessiner des cadres autour des objets détectés et d'afficher leurs labels.

Afin de tester notre implémentation du modèle Yolo, nous avons tenté de build une apk afin de la tester sur nos téléphones personnels. Lors de l'utilisation de la commande "flutter build apk", nous obtenons diverses erreurs. Le problème est que "tflite_flutter" est en conflit avec nos autres dépendances dans les versions nécessaire de gradle et de flutter sdk.
Nous avons essayé de régler ce problème, mais nous pensons changer "tflite_flutter" par "object_detection_flutter", ce qui pourrait résoudre notre problème. Nous sommes actuellement en train d'étudier la question pour savoir si cette solution est viable.

Entrainement de l’IA

Au début, nous comptions réaliser les annotations des images via LabelImg directement mais nous avons rencontré quelques problèmes et étant que nous avions déjà repéré d’autres alternatives, nous avons décidé de nous diriger vers **Roboflow**.



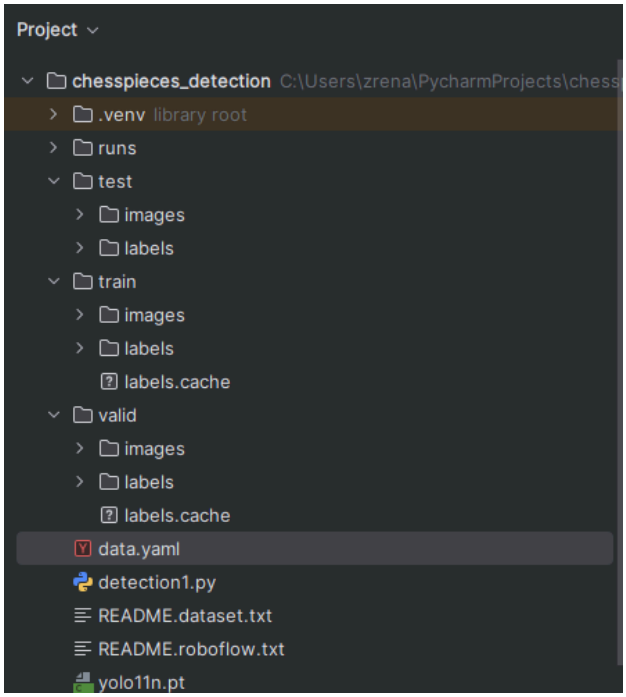
Une fois le dataset importé sur roboflow il a fallu commencer à mettre des Labels sur nos images pour la reconnaissance. À noter que pour faciliter les choses sur ce test, on a décidé de prendre un set d’images qui été déjà toutes resized à la taille 640*640.

Annotées les images est un processus assez lent qui consiste à encadrer sur les images les objets que nous souhaitons reconnaître, dans notre le cas des pièces de jeu d'échecs.
Étant donné que c’est la première fois que nous effectuons ce genre de manipulation, nous avons pris comme décision de se limiter à reconnaître uniquement le type de pièce sans prendre en compte la couleur des pièces.

Une fois les images correctement annotées, et divisées en sous fichier pour avoir des données de TRAIN / VALIDATION / TEST, on a pu importer nos données sur notre IDE avec cette commande :
` curl -L "https://app.roboflow.com/ds/JY4932mtDC?key=f5Y21UXzXS" > roboflow.zip; unzip roboflow.zip; rm roboflow.zip ` ;

sous le format adéquat pour qu’elles fonctionnent avec YOLOv11, ce qui nous a donné un fichier data.yaml avec le nom des différentes classes (['Bishop', 'King', 'Knight', 'Pawn', 'Queen', 'Rook']) et les chemins vers les autres fichiers.

Voilà à quoi ressemble l’arborescence : les données et leurs labels.



Ensuite, en important YOLO depuis la librairie ultralytics, on a pu commencer à entraîner notre propre modèle d’IA pour reconnaître les pièces d’échecs. Thomas sera chargé de cette tâche étant donné que Renaud et Steven possèdes des GPU AMD et que l'entraînement d’IA est beaucoup plus simple et efficace avec des GPU Nvidia.

Membres du groupe :
- Renaud Zell
- Thomas Ducret
- Julien Maaroufi
- Steven Lefebvre