

Rapport SAE 501

Semaine 3

MALORON Arthur,
DOTTO Matis,
MANICK Luc,
BELMADANI Abdourrahmane

Sujets abordées

Organisation

Au cours de cette semaine, certains membres de l'équipe n'ont pas pu être très présents pour cause de problèmes de santé ou de problèmes personnels, ce qui nous a en partie empêché d'atteindre nos objectifs hebdomadaires.

Cependant, même dans des conditions parfaites, nos objectifs pour cette semaine étaient trop ambitieux. Nous allons donc mettre un peu plus de réflexion dans le choix de nos objectifs hebdomadaires afin de pouvoir avancer sans se précipiter.

Cette semaine, nous avons effectué beaucoup de changements, mais très peu pourront être montrés étant donné que nous n'avons pas pu prendre le temps de créer quelque chose d'assez propre pour pouvoir être montré.

C'est pourquoi à partir de maintenant nous allons ajouter une nouvelle rubrique "Objectifs hebdomadaires" pour exposer plus clairement nos objectifs pour la semaine suivante et nous permettre de mieux nous organiser autour de ces objectifs.

Intégration du modèle d'IA

Notre précédente application, utilisant un modèle de classification, était la première étape de l'implémentation de notre modèle de détection d'objets.

Cette étape avait pour but de nous familiariser avec le fonctionnement du traitement des images en Android kotlin.

Cependant, ce que nous n'avions pas prévu, c'était que la structure de cette intégration serait si différente

Pour commencer, nous avons dû modifier une bonne partie de notre structure pour utiliser l'API interpreter de tensorflow au lieu de l'objectDetector (qui avait une structure similaire au classifieur utilisé jusque là), ce changement a été nécessaire étant donné que l'API objectDetector est adaptée pour les modèles SSD mobilenet, mais pas pour les modèles Yolo.

mais il a aussi fallu faire un gros effort de conversion afin d'utiliser un modèle Yolo dans notre application. Ces efforts vont être détaillés dans la sous-catégorie ci-dessous.

Conversion du modèle

La conversion du modèle a pris un temps considérable, nécessitant plusieurs environnements, plusieurs outils, plusieurs environnements python et même plusieurs versions de ce même langage.

Cette conversion se fait en trois étapes :

- La conversion du modèle de .weights à .pb (Python 3.7 maximum)
- La conversion du .pb en .tflite (sans métadonnées) (Python 3.10)
- L'association des métadonnées au modèle .tflite (Python 3.10)

Suite à cette conversion, le modèle peut être utilisé dans l'application.

Cependant, nous avons eu plusieurs problèmes liés aux métadonnées et à la configuration de notre modèle.

En effet, notre yolov4-tiny semble ne pas avoir les métadonnées indiquées sur le repository git où il a été téléchargé, car le tensor de retour est [1, 1, 84].

Nous ne rentrerons pas dans les détails de ce que cela signifie cette semaine, mais pour résumer, d'après ce résultat, notre modèle devrait en théorie avoir 79 classes dans ses métadonnées, or la labelmap que l'on nous a fourni en contient 80.

L'erreur que nous rencontrons actuellement avec le modèle yolov4-tiny est celui-ci :

```
java.lang.IllegalStateException: Internal error: cannot fill a Java array of 336 bytes with a Tensor of 0 bytes
```

Nous n'avons pas pu trouver de solution à temps, alors nous avons essayé avec un autre modèle, un modèle yolov2-tiny. Pour confirmer notre théorie, nous avons utilisé des métadonnées erronées pour notre premier essai, et le modèle nous renvoyait bien la même erreur que notre modèle yolov4-tiny.

En revanche, lorsque nous avons utilisé les bonnes métadonnées pour le modèle yolov2-tiny, l'erreur a disparu. Le problème vient donc assurément du modèle et probablement de mauvaises métadonnées.

Autres problèmes

Notre implémentation du modèle n'est pas correcte et le modèle ne retourne pas de résultats lorsqu'il devrait en détecter, mais étant donné que nous avons réussi à faire fonctionner l'application avec le modèle yolov2-tiny assez tard, nous n'avons pas eu le temps d'explorer les raisons de cette erreur.

Objectifs hebdomadaires

- Création d'un espace regroupant toutes les étapes de la conversion du modèle (avec les environnements virtuels) ainsi que la production d'une documentation.
- Débug de l'application pour faire fonctionner la reconnaissance d'objets avec le modèle yolov2-tiny.
- Recherches sur l'entraînement de l'IA et début de la création du dataset
- Choix et création de la base de données.

Lien du Trello

<https://trello.com/invite/b/670e75ea832d0d8d7b7625cf/ATTI30db0fce264f5fcced5a98e25047eb67DF2B89AE/s501>