

Rapport SAE 501

Semaine 9

MALORON Arthur,
DOTTO Matis,
MANICK Luc,
BELMADANI Abdourrahmane

Sujets abordées

Entraînement du modèle

Nous avons poursuivi nos efforts dans la recherche d'un moyen d'entraîner notre modèle, et tout semblait pointer vers le github de tensorflow/models évoqué la semaine dernière. Cependant, nous ne parvenons toujours pas à trouver l'environnement adapté pour le faire fonctionner.

Les dépendances du github <https://github.com/tensorflow/models> pour les modèles d'object detection sont dans un fichier devant être compilé avec protoc, puis installé en utilisant pip. La compilation n'a pas été un problème, mais il était difficile de trouver un environnement dans lequel les dépendances s'installaient correctement. Nous avons donc essayé avec différentes versions de python (3.10, 3.7, 3.9, 3.12) et aucune de ces versions n'a réglé le problème, bien que les messages d'erreurs changeaient.

Face à ces difficultés, nous avons pensé à nous tourner vers une image docker qui réglerait ce problème d'installation. Nous avons essayé de construire une image à l'aide d'un Dockerfile présent dans le github de tensorflow. Lors du lancement de ce fichier, plusieurs erreurs sont apparues. Après quelques modifications, l'image a pu être construite correctement. Une fois l'environnement lancé, tout semblait fonctionner, mais il était impossible d'exécuter des commandes. Tout semble indiquer que l'image Docker ne fonctionne pas, car le fichier Dockerfile n'est plus maintenu à jour.

Nous avons aussi essayé de créer notre propre image docker en partant de l'image de tensorflow. Cette piste nous a permis de découvrir que le docker tensorflow était sous python 3.11.6, ce qui pourrait être une piste pour notre environnement.

Nous poursuivons toujours cette piste aujourd'hui, en espérant que nous puissions rapidement faire fonctionner cet environnement.

Nous avons déjà fait la majorité des recherches nécessaires à l'entraînement du modèle et la dernière pièce du puzzle est réellement cet environnement qui nous bloque depuis quelques semaines maintenant.

Pour le modèle, nous pensons qu'il est plus judicieux de réaliser un fine-tuning du modèle SSD MobileNet v2 pré-entraîné, plutôt que de réentraîner le modèle depuis zéro, afin de l'adapter à notre propre dataset de plats, que nous avons annoté avec LabelImg. Le fine-tuning consiste à ajuster les paramètres du modèle SSD afin qu'il puisse détecter avec précision les différentes classes présentes dans notre dataset. Cette méthode permet de conserver les connaissances acquises par le modèle lors de son entraînement initial, tout en l'adaptant à notre cas d'utilisation spécifique.

Hébergement de l'API

Nous avons trouvé une solution pour héberger notre API : nous avons choisi d'utiliser une machine virtuelle mise à disposition par le service informatique de l'IUT. Une fois les identifiants reçus, nous pourrions installer un serveur Apache, configurer une base de données MySQL, et déployer notre API afin qu'elle soit fonctionnelle et accessible en permanence pour notre application Android.

Envoi des images vers l'API

Nous avons ajouté une fonctionnalité permettant d'envoyer une requête HTTP multipart, conçue pour transmettre des fichiers et des données. Cette fonctionnalité permet d'envoyer une image, accompagnée de ses labels, vers notre API. Pour l'instant, seules des données statiques peuvent être envoyées via un bouton, une solution temporaire qui permet de vérifier que la requête fonctionne bien. En attendant le déploiement de l'API, nous prévoyons de permettre à celle-ci de recevoir la requête HTTP, récupérer l'image et les labels, puis de stocker ces données.

Objectifs hebdomadaires

- Test d'entraînement de l'IA sur au moins 5 classes (100 images par classe)
- Conversion d'un modèle SSD mobilenet en .tflite
- Réussir à avoir un environnement pouvant utiliser les scripts du github de tensorflow
- Implémenter le stockage des images depuis l'API

Lien du Trello

<https://trello.com/invite/b/670e75ea832d0d8d7b7625cf/ATTI30db0fce264f5fced5a98e25047eb67DF2B89AE/s501>,