Rapport SAE 501

Semaine 6

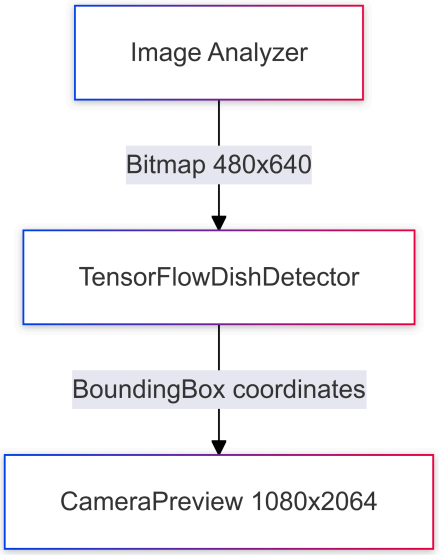
MALORON Arthur,
DOTTO Matis,
MANICK Luc,
BELMADANI Abdourrahmane

Sujets abordées

Mobilenet

Au cours de cette semaine, nous avons finalement pu trouver la cause de nos problèmes de boundingBox en utilisant le modèle Mobilenet V1.

Mais avant d'expliquer le problème, il faut rappeler la structure de notre processus de détection d'objets.



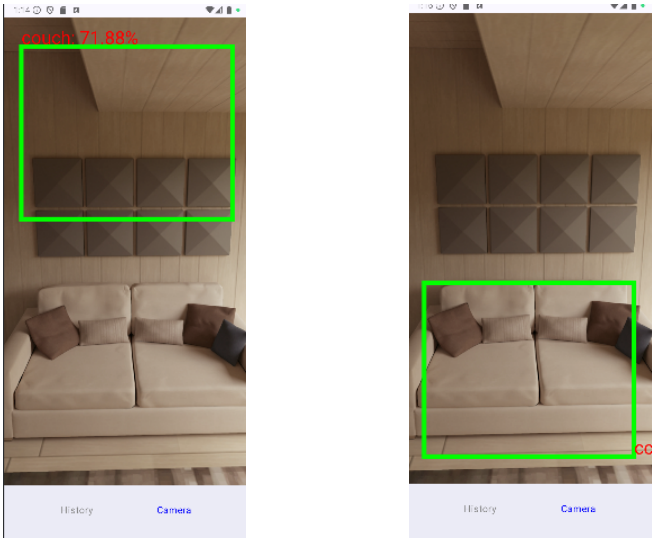
La dernière fois, nous nous étions concentrés sur la différence de dimensions entre la Bitmap que nous renvoie l'ImageAnalyzer et notre CamerPreview (l'affichage), mais nous avons découvert que notre code pour le redimensionnement de notre boundinbox était correct.

Le problème était tout autre et il peut être remarqué si l'on affiche la Bitmap renvoyée par l'ImageAnalyzer en prenant la rotation de l'image en compte :

Comme on peut le voir, l'image subit une rotation à 90° comparé à ce qui est affiché dans notre application.

Nous avions pris la rotation en compte dans l'analyse de l'Image par notre modèle, mais pas dans le redimensionnement de la boundingBox étant donné que les coordonnées que nous récupérions étaient donc "dans le bon sens".
Cependant, c'était sans compter sur le fait que les dimensions que nous utilisions pour redimensionner les coordonnées de cette boundingBox devaient aussi être tournées de 90°.

Notre boundingBox avait maintenant la forme qu'on attendait d'elle, mais sa position ne semblait pas être correcte (image de gauche). Nous avons trouvé que cela ressemblait à la bonne position mais inversée sur l'axe Y, et nous avons donc essayé d'inverser les coordonnées de la boundingBox en Y, ce qui nous a donné le résultat attendu (image de droite).



Nous avons pu confirmer que cette inversion nous donne bien les bons résultats après plusieurs tests, bien que nous ne comprenions pas réellement pourquoi.

Nous avons donc finalement décidé de merge la branche mobilenet sur la main.

Entrainement du modèle

Nous avons utilisé les images de Food-101 disponibles sur Kaggle pour entraîner le modèle. Après avoir traité les 100 000 images (1000 images par classe avec 101 classes différentes), nous nous sommes rendus compte que le modèle atteint un taux de reconnaissance de 100 % pour n'importe quelle image.

```
[Running] python -u "c:\Users\Utilisateur\OneDrive\Bureau\Info\BUT 3\SAE_python\mon_projet_analyse_images\scripts\test.py"
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
R�sultat pour l'image Capture d��cran 2024-11-03 170930.png:
Classe pr�dite : 0, Confiance : 100.00%
R�sultat pour l'image Capture d��cran 2024-11-03 171001.png:
Classe pr�dite : 0, Confiance : 100.00%
R�sultat pour l'image Capture d��cran 2024-11-03 171018.png:
Classe pr�dite : 0, Confiance : 100.00%
```

Les problèmes sont sans doute dus à un surapprentissage ou à un mauvais entraînement du modèle.

Objectifs hebdomadaires

- Comprendre pourquoi l'entraînement du modèle se déroule mal.
- Entraîner le modèle de la bonne façon et l’essayer sur notre application.
- Créer l'API permettant de synchroniser les images analysées entre les appareils.

Lien du Trello

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