

Computer Vision Rapport de Projet

Documentation Utilisateur :

Introduction :

L'application mobile est une interface utilisateur s'appuyant sur l'API de recherche CNN disponible sur notre serveur distant. Cette partie énumère les différentes interactions possibles depuis le terminal mobile.

Choix de l'image à analyser

A démarrage de l'application il vous est proposé de choisir une image selon les options suivantes :

- Votre appareil photo.
- Un point de stockage interne au téléphone.
- Un point de stockage externe (exemple : Google Drive).

Ces choix sont disponibles en cliquant sur le bouton : « Select Image »

Modification sur l'image à analyser :

A la suite du choix de l'image, l'application vous redirige vers le découpage de cette image.

Vous avez également le choix de retourner l'image (opération miroir), de retourner l'image avec un angle de 90° à chaque click.

Lancement de la recherche

Pour lancer une recherche avec votre image, cliquer simplement le bouton « Search »

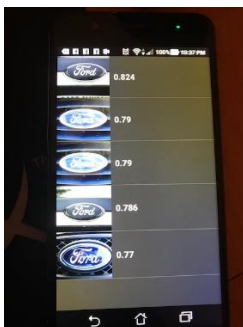
Vous allez être redirigé après un court chargement vers la visualisation des résultats.

Visualisation des résultats :

La visualisation des résultats permet de visualiser les résultats dans une liste.

Les résultats sont présentés par score décroissant, ainsi le premier résultat sera l'image la plus proche de l'image de sélection.

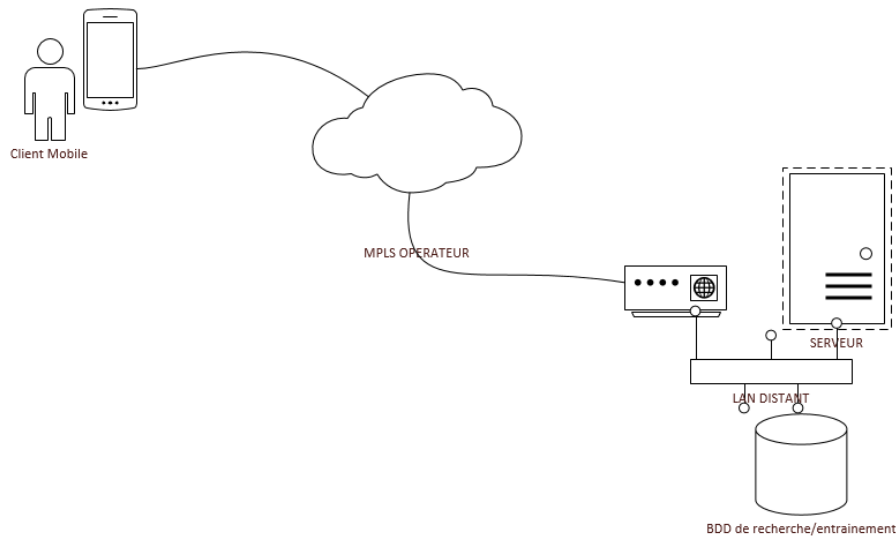
Chaque résultat de cette liste est sélectionnable pour comparaison recherche.



Architecture :

Introduction :

L'architecture générale de ce projet est de type client-serveur, le serveur est joignable par un alias DyDNS avec une NAT de l'IP publique de notre réseau distant vers le serveur distant. L'accès au serveur se fait en http (port 80) et bindé vers le port 8000 du serveur, pour un accès au serveur Django REST.



Les appels à L'API

(R -> Racine du serveur)

GET R/img_searches/<pk>

Paramètres:

Pk: integer non null et non négatif.

POST R/img_searches/

Paramètres :

Data : une image en B64

Method : ici toujours « CNN »

client_ip : pour garder une trace.

GET R/static/<img_name>.jpg

Permet de récupérer une image dans le dossier d'entraînement.

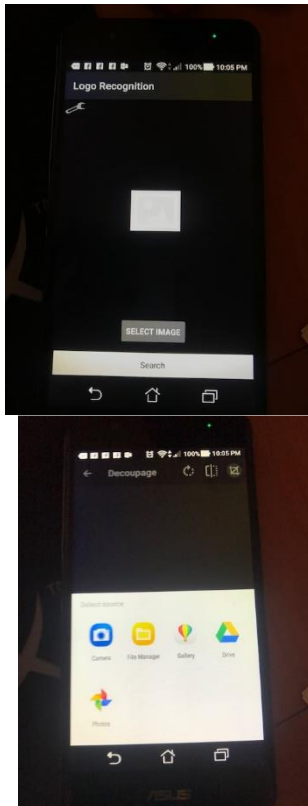
POST R/index/<int>

Permet de changer le nombre de résultats voulus par le client.

Le serveur va lancer un indexing en parallèle d'une durée relative à la taille de base.

Déroulement de la trame de recherche :

La phase de sélection :



Sur l'activité principale :

Les paramètres (version non-stable) en haut à gauche

Le bouton de sélection 'Select Image'

Le bouton de recherche 'Search'

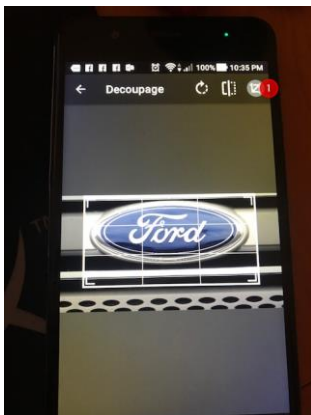
La Phase de sélection :

Comme décrit plus haut, elle laisse le choix à l'utilisateur sur l'emplacement de l'image à envoyer au serveur.

Après sélection par l'utilisateur vous êtes redirigé vers une activité que j'ai récupérée d'un autre programmeur, cette activité permet de définir la zone de l'image à sélectionner.

Figure 1 MainActivity - Selection

Découpage de l'image



La Phase de découpage

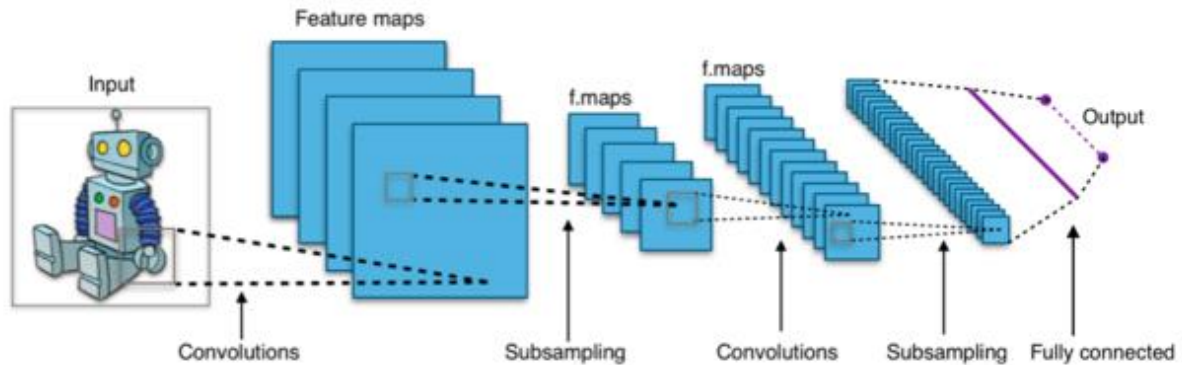
Permet de découper l'image en sélectionnant la zone spécifique à étudier.

Pour fermer le découpage après action, il suffit de cliquer le bouton en haut à droite l'écran du mobile.

Cette phase est primordiale pour l'activité de recherche, elle influe énormément sur les possibles résultats de vos recherches.

La recherche CNN

Introduction :



CNN fonctionne sous forme de transformation successive d'image entre plusieurs couches de neurones, chacune de ces couche va analyser une particularité distincte permettant de labelliser l'image cible.

Coté serveur la trame de recherche démarre après l'appel post du client (Cf requêtes de cette API)

```
0001 0.770217 , Url : b'logos32plus_004259.jpg
logos32plus_004259.jpg
[07/Apr/2019 22:36:48] "POST /img_searches/ HTTP/1.1" 201 38
-----
searching starts
-----
score : 0.824400 , Url : b'2514608364.jpg'
score : 0.789883 , Url : b'255740214.jpg'
score : 0.789883 , Url : b'flickrlogo27_255740214.jpg'
score : 0.785505 , Url : b'2513777557.jpg'
score : 0.770147 , Url : b'logos32plus_004259.jpg'
```

Le serveur Utilise Keras et tensorflow en version multithreading, il est donc possible de lancer plusieurs recherches en même temps sans faire réinitialiser le graph tensorflow.

A la suite de cette recherche, quand l'ensemble des résultats ont été stocké en base de données, le programme renvoie dans la réponse au post un id (pk) qui permettra de récupérer les résultats de sa requête.

```
logos32plus_004259.jpg
[07/Apr/2019 22:37:13] "POST /img_searches/ HTTP/1.1" 201 38
[07/Apr/2019 22:37:14] "GET /img_searches/l31/ HTTP/1.1" 200 648
[07/Apr/2019 22:37:14] "GET /static/flickrlogo27_255740214.jpg HTTP/1.1" 200 155363
[07/Apr/2019 22:37:14] "GET /static/2514608364.jpg HTTP/1.1" 200 76596
[07/Apr/2019 22:37:14] "GET /static/255740214.jpg HTTP/1.1" 200 155363
[07/Apr/2019 22:37:14] "GET /static/2513777557.jpg HTTP/1.1" 200 67345
[07/Apr/2019 22:37:14] "GET /static/logos32plus_004259.jpg HTTP/1.1" 200 388252
```

Cette image montre ensuite les requêtes du client pour récupérer les images depuis le serveur de données (static).

Le nombre de requêtes client pour récupérer les ressources est proportionnels au nombre de résultats voulu par recherche (voir les différents fichiers de configurations).

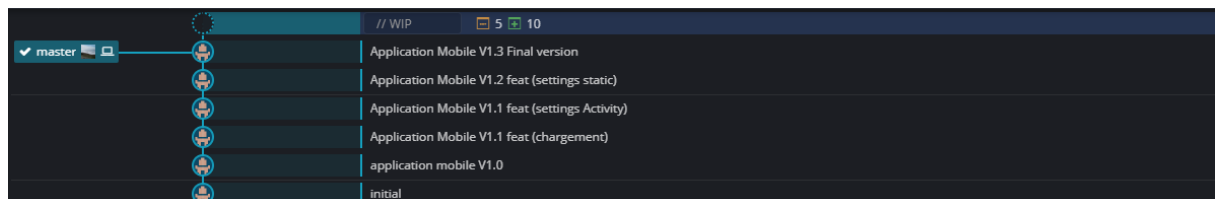
Le nombre de résultats peut être modifié (cf requêtes API).

Bilan

Ce module m'a permis de m'initier au deep learning grâce à la méthode CNN, et cela grâce au langage de programmation python. Le fait que j'ai codé entièrement ce projet, que ce soit l'application mobile mais aussi la partie serveur.

Cela m'a été profitable car j'ai connaissance de l'ensemble de l'architecture que j'ai mise en place.

J'ai principalement utilisé Git comme point de sauvegarde de mon projet, travaillant en solo je n'ai pas eu de conflit sur le projet en général sauf pour réunir les deux parties à la fin.



Cependant, je n'ai pas eu le temps de tester l'ensemble des paramètres de recherche.

Exemple : faire varier le nombre de couches de convolution, effectuer des changements de luminosité et d'adaptation d'image avancé pour mon répertoire d'entraînement.

L'architecture Client-serveur n'est plus opérationnelle, elle ne fonctionne que sur un réseau local, je n'ai pas réussi à trouver la cause de ce problème.

J'aurais voulu ajouter d'autres fonctionnalités à l'application mobile, notamment sur la partie paramétrage de la recherche : pouvoir modifier depuis l'application le nombre de résultats, les paramètres de Keras...

Ce résumé ce projet m'a donné envie de continuer à me documenter dans ce domaine et pourquoi pas, améliorer mes paramètres de recherche et mes data-sets.