

Projet Info FA19

Application Mobile de Recherche d'images

1. Objectif du projet

L'objectif est de développer une application de recherche d'image à partir d'un smartphone. Ce type d'applications supportant la recherche d'information à partir d'une photo est très populaire dans l'industrie du mobile. Il s'agit à travers cette application de mettre en œuvre les différentes connaissances vues lors des cours à distance (développement de service REST, d'application Android, de traitements d'images)

La réalisation de cette application comporte une partie serveur et une partie cliente :

- La partie serveur est constituée d'un service REST chargé de recevoir les requêtes de recherche d'images, de les traiter et d'exposer les résultats correspondants.
- La partie client est une application pour smartphone. Cette application doit permettre de rechercher les images similaires à une image fournie par l'utilisateur.

Pour le développement de ces deux parties, nous vous conseillons d'adopter les technologies suivantes :

- Partie serveur : Python + Django + OpenCV + Keras
- Partie cliente : Android ou IOS

Ces technologies sont les seules pour lesquels nous serons en mesure de vous aider durant les séances de TP. Cependant, nous vous offrons la liberté d'autres technologies. Si vous faites ce choix, nous ne fournirons aucun support sur la ou les technologies retenues. Il est donc conseillé de choisir des technologies dans lesquelles vous êtes très à l'aise.

2. Service REST de recherche d'images

Structure du service

Le service à développer doit respecter la philosophie de conception REST qui consiste à donner accès à un espace de ressources manipulables.

Etant donné que nous souhaitons formuler des recherches d'images, nous allons matérialiser chaque recherche et les résultats associés sous la forme d'une ressource exposée en utilisant le format JSON.

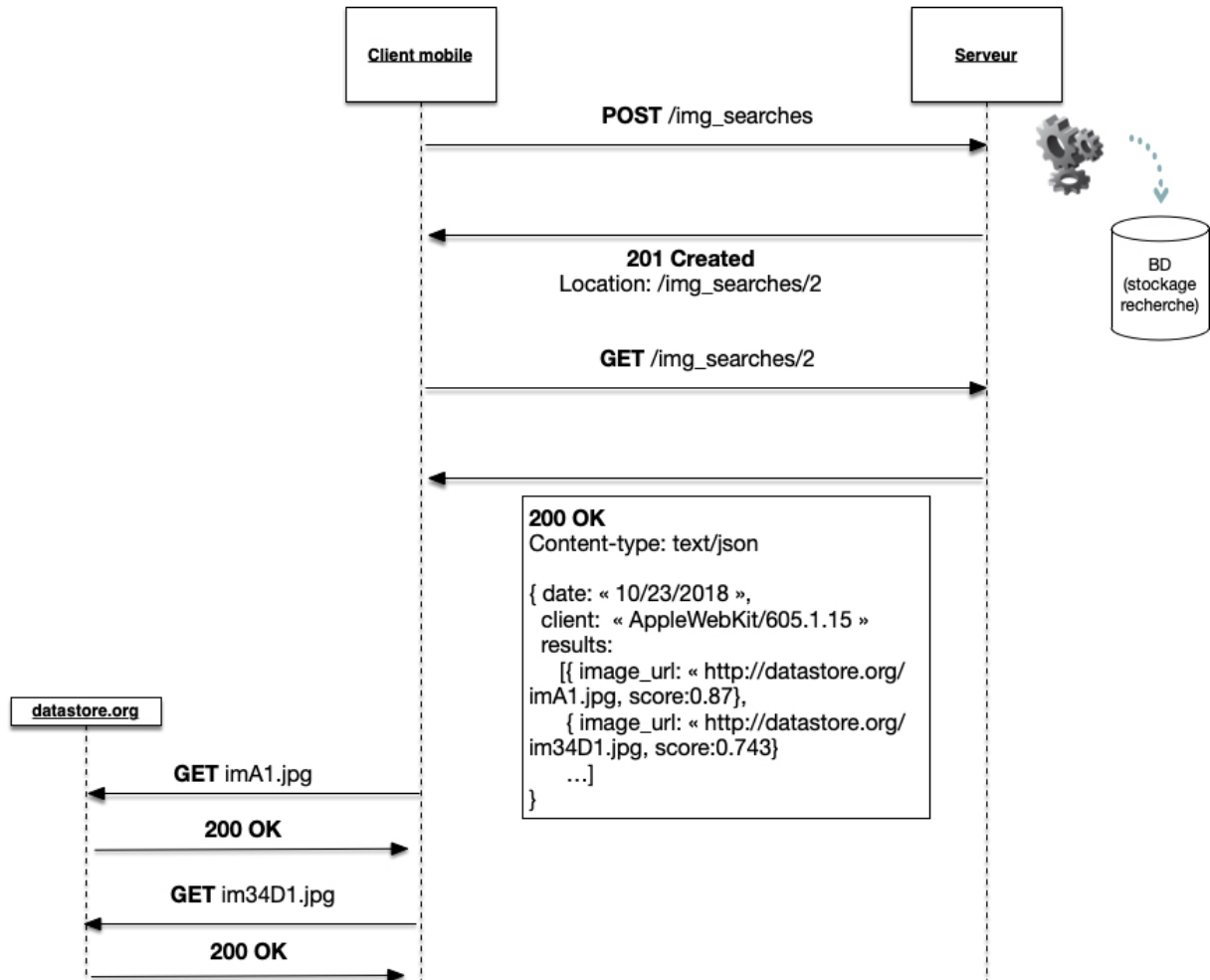
Le schéma d'interaction entre le client et le serveur pour une recherche d'images suit les étapes suivantes.

- L'envoi depuis le client d'une requête http POST contenant l'image requête fourni par l'utilisateur. Le traitement de cette requête côté serveur doit exécuter la recherche d'images similaires, réaliser le stockage du résultat de cette recherche en base puis renvoyer une réponse avec l'URL correspondant à la ressource matérialisant ce résultat ;
- L'envoi depuis le client d'une requête GET avec l'URL de la ressource retournée dans la réponse du POST pour obtenir sa représentation JSON. Cette représentation est

censée contenir les URLs des images résultants et éventuellement la date de la requête et des données relatives au client (type de client, adresse IP).

- Pour chaque URL d'image faisant partie du résultat de la recherche, l'envoi depuis le client d'une requête GET pour récupérer l'image correspondante.

La figure suivante illustre ce schéma d'interaction :



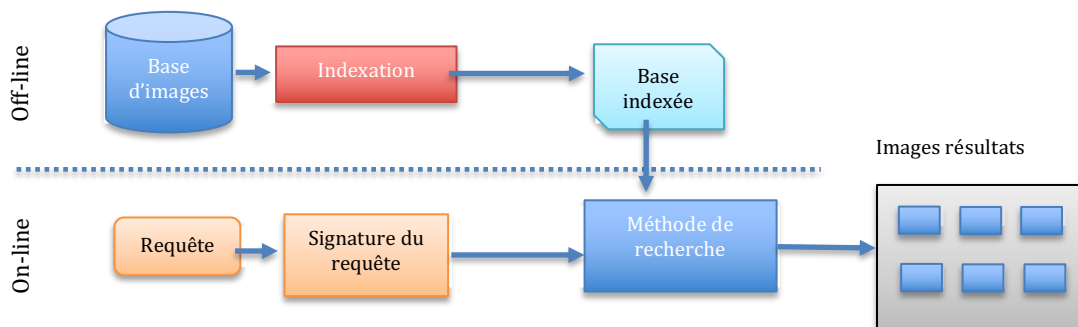
On peut voir sur ce schéma la structure du JSON contenant les informations de la requête ainsi que les images résultats. Il est utile de remarquer qu'un score est associé à chaque image de façon à pouvoir ordonner les résultats dans l'interface du client

Concernant le service REST, il n'est pas nécessaire de prendre en charge les requêtes http PUT et DELETE sur les ressources représentant les différentes recherches.

L'image contenu dans une requête ne doit pas nécessairement être stockée par le serveur.

Module de recherche d'images au sein du server

La tâche de recherche d'image (Retrieval en anglais) à développer ici correspond à un système qui permet de calculer un score de similarité visuelle à partir d'un exemple. Le système reçoit en entrée une image exemple (requête) et retourne l'ensemble des images les plus similaires à cette image, au sens de la mesure de similarité associée à la signature d'image. La recherche implique de trouver les plus proches voisins de la signature associée à l'image requête. La signature d'image dépend de la méthode employée et les caractéristiques extraites de l'image. Vous avez le choix entre les deux méthodes vues en TP : sac de mots visuels ou *Bag of visual words* en anglais (BoVW) et réseaux de neurones convolutionnelles (CNN). La figure suivante illustre ce paradigme de recherche d'image :

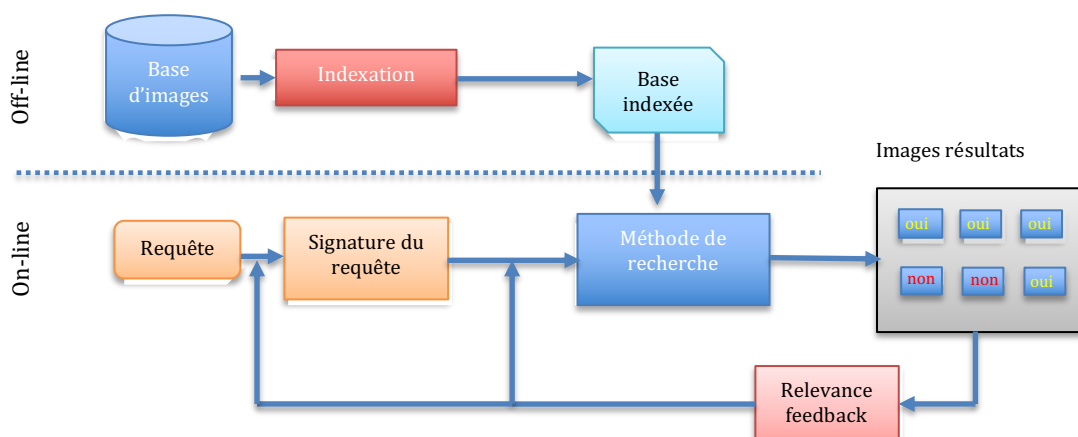


Le principe de la recherche d'image par la méthode BoVW consiste de :

- Extraction automatique de caractéristiques visuelles locales (on utilise ici le détecteur des points SIFT vu en TP)
- Construction d'un vocabulaire visuel représentatif à partir de tous les mots visuels de la base de référence
- Calcul de signatures en forme d'histogramme à partir du vocabulaire visuel
- Calcul de similarité visuelle à partir d'une requête
- Retourner les images similaires à l'image requête

Le principe de la recherche d'image par la méthode de CNN consiste à extraire des caractéristiques globales de l'image à l'aide des réseaux de neurones convolutionnelles. On peut ici créer notre propre modèle, ou simplement utiliser un modèle pré-entraîné (comme par exemple VGGNet vue en TP).

Comme extension à ce paradigme de recherche d'image, il est prévu d'ajouter un mécanisme d'interaction en forme de commentaires sur la pertinence (Retours positifs ou négatifs). Cette partie est illustrée dans la figure suivante et sera détaillée en cours.



3. Application mobile

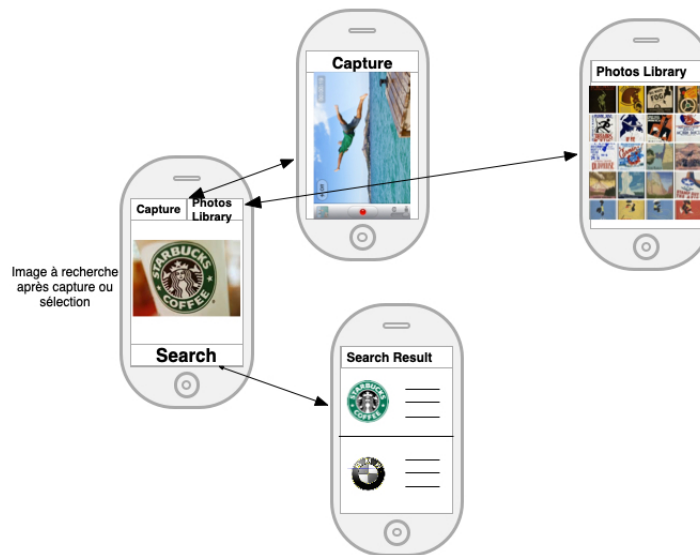
Le fonctionnement de l'application mobile et l'enchaînement des écrans sont les suivants.

Le premier écran est destiné au choix de l'image requête. Deux possibilités sont à offrir à l'utilisateur pour choisir cette image : soit l'utilisateur sélectionne une image existante dans la base du téléphone, soit l'utilisateur réalise une nouvelle photo. Dans les deux cas, un écran spécifique est à prévoir pour réaliser cette opération.

Une fois l'image sélectionnée, l'application effectue la demande de recherche d'images similaires auprès du serveur selon les principes décrits dans la section 2.

Quand le traitement de recherche est terminé, le résultat obtenu (c-à-d les images de références les plus proches) est présenté à l'utilisateur dans un nouvel écran. Pour cet écran, il faut prévoir la présentation de plusieurs images.

La figure suivante présente les différents écrans à réaliser pour l'application et leur enchaînement.



4. Méthodologie de travail

La réalisation de l'application est à faire en binôme.

Avant de coder, il est conseillé de réfléchir à l'architecture logicielle de votre application. Il y aura des étapes de validation pour chaque partie.

Pour le développement, les points suivants sont à respecter :

- Création d'un projet distinct pour la partie serveur et la partie cliente.
- Utilisation du gestionnaire de projets Git (les détails seront fournis) pour le versionnement et le travail collaboratif.
- Le code doit être suffisamment commenté.

Vous avez la possibilité d'étendre le fonctionnement de l'application avec des fonctionnalités supplémentaires (à discuter avec l'enseignant).

5. Restitution du travail

Il est demandé de rendre une archive .zip contenant les éléments suivants :

- Code source complet de deux parties et l'URL du dépôt Git
- Rapport de 4 pages max contenant :
 - une partie utilisateur
 - une partie décrivant l'architecture logicielle
 - une partie relative au traitement d'image
 - une partie bilan-conclusion
- Préparation d'une démonstration pour le dernier TP