

L'Indexation et la Recherche d'Images Basée sur le Contenu : Une Approche avec et sans Retour de Pertinence

Salma AMGAROU, Hamza HAFDAOUI, Hanane ZAOUI

Encadré par Pr. M'hamed AIT KBIR

I. Introduction

a. Contexte et Motivation

Avec la croissance rapide des médias numériques, la quantité d'images stockées et partagées chaque jour a explosé. Gérer et retrouver ces images efficacement est devenu un défi important dans plusieurs domaines, tels que la santé, le commerce en ligne, et les réseaux sociaux.

Les systèmes traditionnels de recherche d'images reposent souvent sur des métadonnées (tags, noms de fichiers) qui peuvent être subjectives et peu fiables. Les systèmes de recherche d'images basés sur le contenu (**CBIR** : Content-Based Image Retrieval) permettent de surmonter ces limites en analysant directement le contenu visuel des images, comme les couleurs, les textures et les formes.

Cependant, les systèmes CBIR présentent encore des défis, notamment :

- Extraire des descripteurs d'image précis et efficaces.
- Traiter des bases d'images variées avec des catégories différentes.
- L'écart sémantique entre les descripteurs d'images de bas niveau et les concepts sémantiques de haut niveau

Ce projet aborde ces défis en intégrant une technique de **retour de pertinence** dans un système CBIR, permettant aux utilisateurs d'affiner progressivement les résultats de recherche en fonction de leurs préférences.

b. Objectif du Projet

L'objectif principal de ce projet est de surmonter les défis rencontrés par les systèmes **CBIR** en intégrant une technique avancée de **retour de pertinence**. Cette approche permet aux utilisateurs d'affiner progressivement les résultats de recherche en fonction de leurs préférences, en utilisant la méthode **Query-point movement**. Ce processus interactif améliore la précision des résultats en ajustant la recherche au fil des itérations, offrant ainsi une expérience plus personnalisée et efficace.

Pour mettre en œuvre cette solution, le projet consiste à, développer une application Web qui intègre les fonctionnalités essentielles d'un système CBIR. L'application permettra de :

- Calculer et afficher des descripteurs de contenu visuel pour les images.
- Effectuer des recherches d'images similaires selon deux méthodes :
 1. **Recherche simple** : Basée uniquement sur les descripteurs calculés.
 2. **Recherche avec retour de pertinence** : Utilisant la méthode **Query-point movement** pour affiner les résultats au fil des itérations.

II. Fonctionnalités et architecture du Système

a. Fonctionnalités du système

L'application Web proposée inclut les fonctionnalités suivantes, permettant une gestion et une recherche efficaces d'images basées sur leur contenu visuel :

- **Chargement, téléchargement et suppression d'images** : Les utilisateurs peuvent facilement ajouter des images à la base de données ou en supprimer selon leurs besoins. Cette fonctionnalité permet une gestion simple et fluide des images.
- **Organisation des images par catégories** : Les images peuvent être triées et organisées en fonction de catégories prédéfinies ou créées par l'utilisateur. Cela permet une gestion structurée et une recherche optimisée au sein de la base d'images.
- **Création de nouvelles images par transformation** : Il est possible d'appliquer des transformations sur les images existantes pour générer de nouvelles images. Les transformations disponibles incluent le recadrage, le changement d'échelle, et d'autres modifications de la structure visuelle.
- **Calcul des descripteurs d'images** : L'application utilise une API Flask pour calculer des descripteurs visuels essentiels, tels que l'histogramme de couleurs, les filtres de texture de Gabor, et les moments de Hu. Ces descripteurs permettent d'analyser les caractéristiques visuelles des images et sont utilisés pour les recherches de similarité.
- **Recherche d'images similaires** : Deux méthodes de recherche sont proposées :
 - **Recherche simple** : Basée exclusivement sur les descripteurs calculés, cette méthode permet

de trouver les images les plus similaires à l'image de requête.

- **Recherche avec retour de pertinence :** Cette méthode avancée utilise la technique de Query-point movement pour affiner les résultats en fonction du retour de l'utilisateur, améliorant ainsi la pertinence des résultats au fil des itérations.

b. Architecture du système

L'architecture de l'application est conçue pour garantir une gestion fluide des images et une recherche efficace par le contenu. Elle est composée de plusieurs composants clés, chacun jouant un rôle spécifique dans l'exécution des fonctionnalités du système.

Frontend

Le frontend de l'application est développé à l'aide de framework MEAN (MongoDB, Express, Angular, Node.js). Il fournit une interface utilisateur interactive et réactive permettant de :

- Charger, organiser et transformer les images.
- Effectuer des recherches simples ou avec retour de pertinence.

Backend

Le backend est conçu autour de **Node.js** avec **Express** pour gérer les requêtes des utilisateurs et la logique d'application, en complément de **Flask** et de son extension **Flask-RESTful**, qui sont utilisés spécifiquement pour le calcul des descripteurs d'image et la gestion des API. Il assure les rôles suivants:

- Gestion des requêtes utilisateur : Le backend reçoit et traite les requêtes envoyées depuis le frontend.
- Calcul des descripteurs d'images : À l'aide de bibliothèque OpenCV, le backend extrait des descripteurs visuels des images téléchargées.
- Services REST : Les fonctionnalités du système CBIR (recherche d'images similaires, calcul des descripteurs) sont exposées sous forme de services REST accessibles via des API.

Base de Données

La base de données est utilisée pour stocker les images et les descripteurs calculés. Elle est structurée comme suit :

- Images : Les images sont stockées avec leurs catégories associées et leurs métadonnées, permettant une organisation efficace des fichiers.
- Descripteurs : Les descripteurs d'images calculés sont enregistrés pour chaque image, ce qui facilite la recherche et la comparaison lors des requêtes.

III. Méthodologie et Approche Technique

a. Content-Based Image Retrieval (CBIR)

Le **Content-Based Image Retrieval** (CBIR) est une méthode permettant de récupérer des images à partir d'une base de données. Dans le CBIR, l'utilisateur spécifie une image de requête et obtient les images de la base de données similaires à cette image. Pour trouver les images les plus similaires, le CBIR compare le contenu de l'image d'entrée avec celles de la base de données.

Plus précisément, le CBIR compare des caractéristiques visuelles telles que les formes, les couleurs, les textures et les informations spatiales, et mesure la similarité entre l'image de requête et les images de la base de données en fonction de ces caractéristiques.

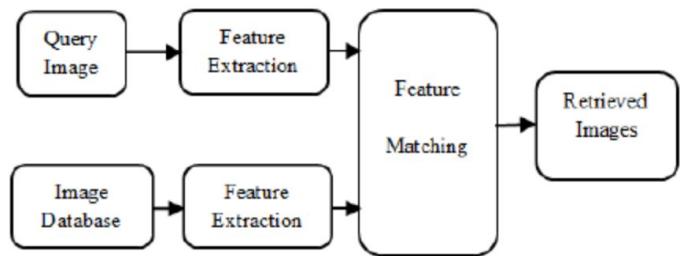


Figure 1:CBIR système

a. Descripteurs de Contenu Visuel

Le contenu visuel d'une image peut être décrit par plusieurs descripteurs qui permettent d'analyser ses caractéristiques principales. Parmi les plus courants, on trouve :

- Descripteurs de couleur : Ils capturent la distribution des couleurs dans l'image, comme les histogrammes ou les couleurs dominantes.
- Descripteurs de texture : Ils décrivent la structure de surface, souvent à l'aide de filtres de Gabor ou de matrices de cooccurrence de niveau de gris.
- Descripteurs de forme : Ils analysent la géométrie et les contours des objets, avec des outils comme les moments de Hu.
- Descripteurs de structure spatiale : Ils concernent la disposition géométrique des éléments dans l'image.

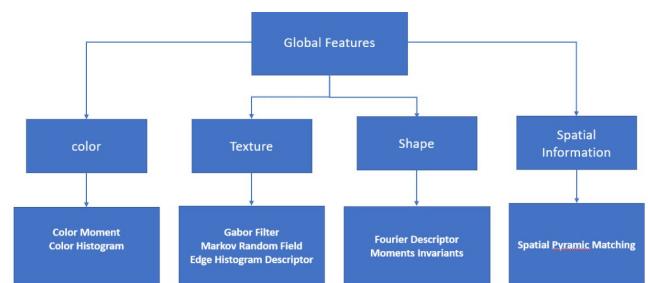


Figure 2: Descripteurs de Contenu Visuel

Pour notre système CBIR, on a choisi de travailler avec plusieurs descripteurs d'image essentiels :

- **Histogrammes de couleurs** : Ils capturent la distribution des couleurs dans l'image, permettant d'identifier les couleurs dominantes.
- **Couleurs dominantes** : L'algorithme **K-means** est appliqué pour identifier les couleurs dominantes dans l'image.
- **Descripteurs de texture avec filtres de Gabor** : Ils analysent la texture en détectant les motifs répétitifs et les détails fins dans différentes orientations et échelles.
- **Moments de Hu liés aux contours** : Ces moments décrivent les formes des objets en capturant les caractéristiques invariantes des contours, indépendamment des transformations géométriques.
- **Descripteurs GLCM (Gray Level Co-occurrence Matrix)** : Analyse la texture en mesurant la fréquence des paires de pixels à des distances et orientations spécifiques, capturant la rugosité, la régularité et les motifs directionnels.
- **Edge Histogram** : Capture la distribution des bords dans différentes orientations, fournissant une représentation des formes et structures présentes dans l'image.

Ces descripteurs permettent d'extraire des informations clés pour la recherche et la classification d'images.

b. Recherche simple

Dans notre système, la recherche simple est réalisée en mesurant la distance entre les descripteurs des images à l'aide de la **distance euclidienne**. Cette méthode calcule la similarité en fonction de l'écart géométrique entre les vecteurs de caractéristiques, permettant d'identifier les images les plus proches de l'image de requête.

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

c. Recherche avec Retour de Pertinence (Query-point Movement)

Le mouvement du point de requête (QPM) dans un système CBIR consiste à ajuster le vecteur de requête pour le rapprocher des images pertinentes et l'éloigner des images non pertinentes. L'objectif est d'améliorer la précision du "point de requête idéal" en le déplaçant vers les bons exemples et en l'éloignant des mauvais. Ce processus se fait de manière itérative, en tenant compte des retours de

l'utilisateur, ce qui permet de mieux refléter ses attentes et de perfectionner les résultats de recherche. Inspirée de la méthode de Rocchio dans la recherche d'informations textuelles, l'approche QPM utilise les ensembles d'images pertinentes (DR) et non pertinentes (DN) pour recalculer le point de requête à chaque itération et ainsi affiner les résultats.

$$Q' = \alpha Q + \beta \left(\frac{1}{N_{R'}} \sum_{i \in D_R} D_i \right) - \gamma \left(\frac{1}{N_{N'}} \sum_{i \in D_N} D_i \right)$$

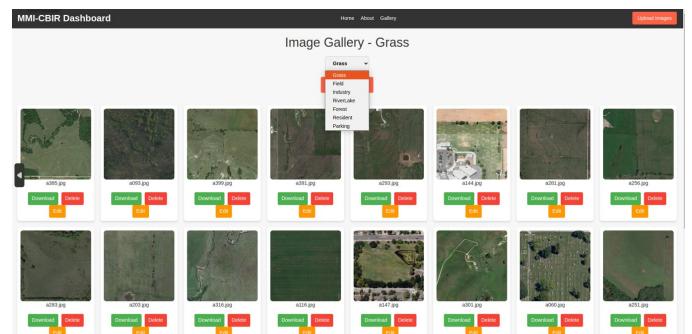
IV. Interface Utilisateur

L'interface utilisateur de l'application est conçue pour être intuitive et facile à utiliser.

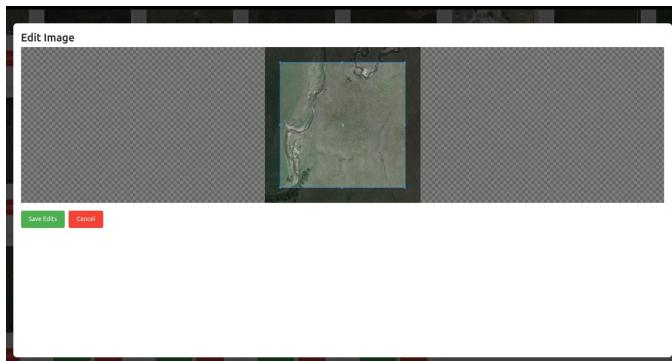
Page d'accueil et Navigation : Dès la page d'accueil, les utilisateurs peuvent accéder rapidement aux principales fonctionnalités : téléchargement d'images, à propos et Galerie.



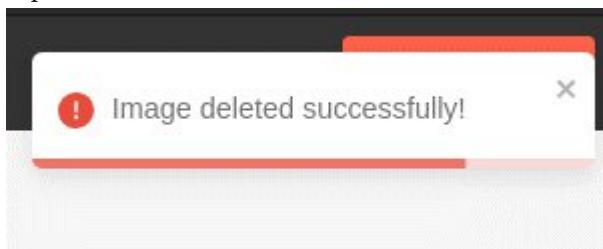
Galerie d'Images : les images sont affichées dans la galerie, organisées en différentes sections ou catégories. L'utilisateur peut facilement naviguer entre ces sections grâce à un dropdown permettant de sélectionner la catégorie souhaitée. Cette fonctionnalité améliore la navigation et permet d'accéder rapidement à des groupes spécifiques d'images.



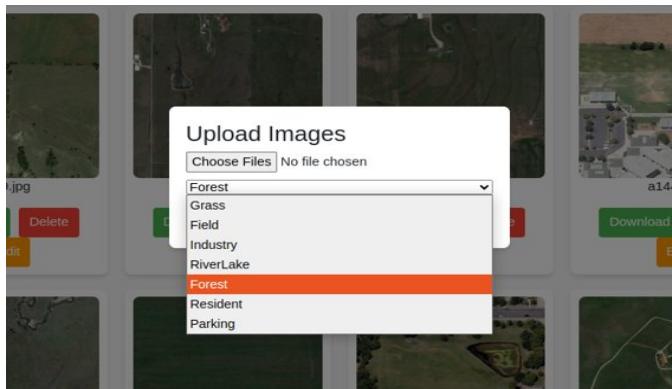
L'utilisateur peut effectuer des modifications sur une image, telles que son redimensionnement.



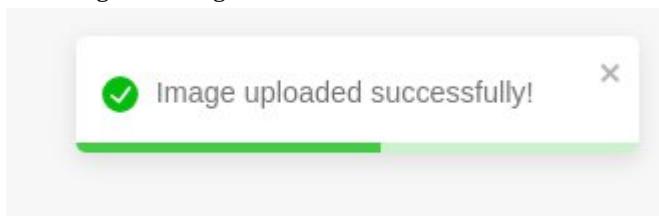
L'utilisateur peut supprimer une image, et un message de confirmation apparaît pour l'informer du succès de l'opération.



Téléchargement d'Images : L'utilisateur peut facilement télécharger une image depuis son appareil. Lors du téléchargement, l'utilisateur peut également attribuer une catégorie à chaque image via un dropdown.

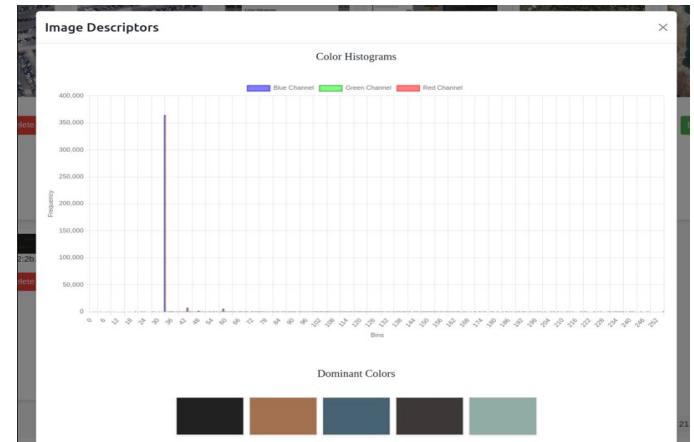


Après un téléchargement réussi, un message de confirmation s'affiche pour informer l'utilisateur que l'image a été téléchargée et catégorisée avec succès.

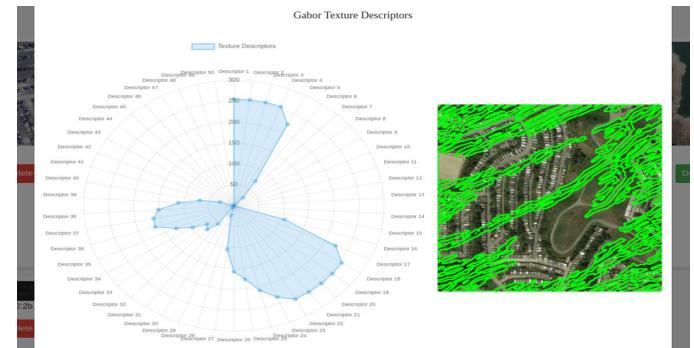


Affichage des Descripteurs : Les descripteurs calculés pour l'image de requête sont présentés de manière claire, permettant à l'utilisateur de comprendre les caractéristiques visuelles de l'image.

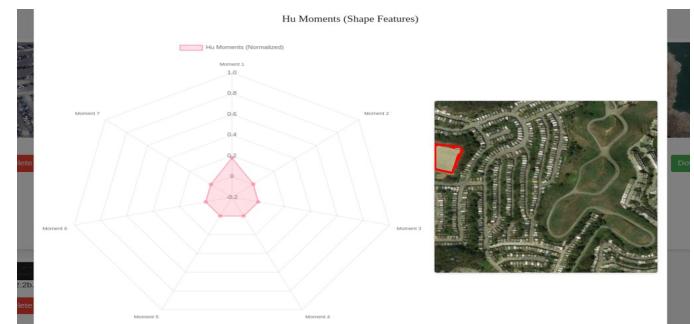
- Histogrammes de couleurs et Couleurs dominantes



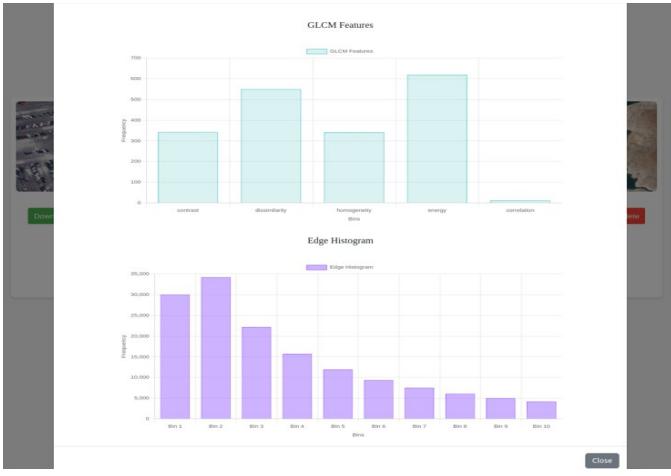
- Filtres de Gabor



- Moments de Hu



- Edge Histogram et Descripteurs GLCM



Recherche simple : L'utilisateur peut rechercher des images similaires en se basant sur les descripteurs de l'image de requête.



Recherche avec retour de pertinence : L'interface permet d'affiner les résultats de la recherche en sélectionnant les images pertinentes ou non pertinentes, grâce à la méthode de retour de pertinence (Query-point movement).

Descriptor	Updated Value
Color Histogram (Blue)	382.5
Color Histogram (Green)	382.5
Color Histogram (Red)	96.47989064035205
Dominant Colors	126.92989311354074
Gabor Texture Descriptors	376.79754515938666
Hu Moments	30.35770527371937
Edge Histogram	382.5
GLCM Contrast	382.5
GLCM Dissimilarity	81.97364693105529
GLCM Homogeneity	126.06357954893895
GLCM Energy	376.3189018891702
GLCM Correlation	30.198551723064345

V. Conclusion

Ce projet a démontré l'efficacité des systèmes CBIR pour surmonter les limites des recherches basées sur des métadonnées en exploitant les caractéristiques visuelles des images. L'intégration de la méthode Query-point Movement pour le retour de pertinence a permis d'affiner les résultats de manière interactive et personnalisée.

L'application développée, combinant MEAN stack et Flask, offre des fonctionnalités robustes, notamment le calcul de descripteurs visuels, la recherche simple, et la recherche avec retour de pertinence. Les résultats obtenus valident l'approche adoptée et ouvrent la voie à des améliorations futures, telles que l'utilisation de modèles d'apprentissage automatique pour enrichir les performances du système.

Bibliographie

- [1] Baeldung, "What Is Content-Based Image Retrieval" URL: <https://www.baeldung.com/cs/cbir-tbir>
- [2] "Efficient Relevance Feedback for Contentbased Image Retrieval by Mining User Navigation Patterns" URL: http://140.127.208.114:5050/e_learning/paper/TKDE-2011.pdf.
- [3] "ChatGPT", URL: <https://openai.com/chatgpt/overview/>
- [4] <https://github.com/palewithout/RSSCN7>
- [5] <https://cocodataset.org/>
- [6] "Invariant Moments and Image Representation" Authors: M. K. Hu