**Documentation du Projet CBIR avec Authentification**

**Introduction**

Cette application est une plateforme combinant authentification sécurisée et recherche d'images basée sur le contenu (CBIR, Content-Based Image Retrieval). Elle permet :

* D'enregistrer des utilisateurs avec des photos pour la reconnaissance faciale.
* De connecter des utilisateurs via un mot de passe ou la reconnaissance faciale.
* D'utiliser des descripteurs comme GLCM pour rechercher des images similaires dans un dataset.

**Fonctionnalités Principales**

**Authentification**

* **Inscription :**
  + L'utilisateur fournit un nom d'utilisateur, un email, un mot de passe et une photo de référence.
  + La photo est analysée pour extraire un encodage facial stocké dans la base de données.
* **Connexion :**
  + Authentification classique via un nom d'utilisateur et un mot de passe haché.
* **Reconnaissance Faciale :**
  + Comparaison de l'image capturée par webcam avec l'encodage facial stocké.

**Recherche d'Images Basée sur le Contenu (CBIR)**

* **Descripteurs :**
  + **GLCM (Gray-Level Co-occurrence Matrix)** : Analyse des textures d'images.
* **Métriques de Distance :**
  + Manhattan, Euclidean, Chebyshev, Canberra.
* **Affichage des Résultats :**
  + Affiche les images les plus similaires avec leur distance calculée.

**Fichiers du Projet**

**Code Principal**

* app.py : Contient le code de l'application Streamlit avec toutes les fonctionnalités décrites ci-dessus.

**Modules Complémentaires**

* descriptor.py : Fournit les fonctions pour calculer les descripteurs d'images.

**Autres Fichiers**

* signatures.npy : Fichier contenant les signatures pré-calculées des images du dataset.
* authentification.db : Base de données SQLite pour les informations des utilisateurs.
* datasets/ : Répertoire contenant les images pour la recherche CBIR.
* reference\_images/ : Répertoire pour stocker les images de référence des utilisateurs.

**Installation**

**Prérequis**

* Python 3.9 ou supérieur.
* Streamlit pour l'interface utilisateur.

**Installation des Dépendances**

1. Créez un environnement virtuel :
2. python -m venv cbir\_env
3. Activez l'environnement :
   * Windows : cbir\_env\Scripts\activate
   * macOS/Linux : source cbir\_env/bin/activate
4. Installez les dépendances :
5. pip install -r requirements.txt

**Configuration**

**Base de Données**

1. Créez une base de données SQLite :
2. python setup\_db.py
3. Vérifiez que authentification.db contient les tables nécessaires.

**Répertoires**

Assurez-vous que les répertoires suivants existent :

* datasets/
* reference\_images/
* uploaded\_images/

**Lancer l'Application**

1. Démarrez l'application avec Streamlit :
2. streamlit run app.py
3. Accédez à l'application dans votre navigateur à l'adresse : <http://localhost:8501>

**Hébergement avec GitHub Desktop**

**Étapes**

1. **Initialisez un dépôt local :**
   * Copiez tous les fichiers du projet dans un dossier.
   * Ouvrez GitHub Desktop et sélectionnez "File > New Repository".
2. **Ajoutez les fichiers :**
   * Vérifiez que tous les fichiers du projet (code, datasets, etc.) sont inclus.
3. **Publiez sur GitHub :**
   * Cliquez sur "Publish Repository".
4. **Hébergement Streamlit Cloud :**
   * Configurez votre application pour Streamlit Cloud si nécessaire.

**Améliorations Futures**

* Intégration d'autres descripteurs comme Haralick et BiT.
* Ajout d'une authentification avec OAuth pour Google et Facebook.
* Hébergement sur des services comme Heroku ou AWS.

**Dépendances**

Voici un résumé des dépendances utilisées :

* numpy
* opencv-python
* scipy
* scikit-image
* streamlit
* face\_recognition

Pour une liste complète, référez-vous au fichier requirements.txt.

**Auteur**

Amine Ben Hassine Étudiant en Informatique, Montréal

Pour toute question, contactez-moi directement ou ouvrez un ticket sur le dépôt GitHub.