**Documentation du projet CBIR avec Authentification Multiple**

**Description**

Cette application web implémente un système de recherche d'images basée sur le contenu (CBIR) avec authentification multi-méthodes. Elle permet aux utilisateurs de s'inscrire et de se connecter via plusieurs méthodes (mot de passe, reconnaissance faciale, Google, Facebook) et d'effectuer des recherches d'images similaires en utilisant différents descripteurs et mesures de distance.

**Fonctionnalités principales**

**Authentification**

* Connexion par nom d'utilisateur et mot de passe
* Connexion par reconnaissance faciale
* Authentification OAuth (Google, Facebook)
* Gestion de profil utilisateur

**Recherche d'images (CBIR)**

* Descripteurs d'images:
  + GLCM (Gray-Level Co-occurrence Matrix)
  + Haralick
  + BiT (Bio-Inspired Texture) simplifié
  + Concaténation des descripteurs
* Mesures de distance:
  + Euclidienne
  + Manhattan
  + Tchebychev (Chebyshev)
  + Canberra
* Extraction et sauvegarde de signatures d'images
* Interface utilisateur pour la recherche et l'affichage des résultats

**Structure du projet**

projet/

├── app.py # Application principale

├── cbir\_functions.py # Fonctions pour la recherche d'images

├── animalsCbir/ # Dataset d'images organisé par catégories

│ ├── cat/

│ ├── dog/

│ └── ...

├── signatures/ # Fichiers de signatures générés

│ ├── SignaturesGlcm.npy

│ ├── SignaturesHaralick.npy

│ ├── SignaturesBit.npy

│ └── SignaturesConcat.npy

└── temp/ # Dossier temporaire pour les images téléversées

**Modules et dépendances**

* **streamlit**: Interface utilisateur web
* **face\_recognition**: Reconnaissance faciale
* **pymongo**: Base de données MongoDB
* **opencv-python (cv2)**: Traitement d'images
* **numpy**: Manipulation de tableaux numériques
* **skimage**: Extraction de caractéristiques GLCM
* **mahotas**: Extraction de caractéristiques Haralick
* **scipy**: Calcul de distances

**Architecture technique**

**Module d'authentification**

* Stockage sécurisé des identifiants (hachage des mots de passe)
* Intégration OAuth pour Google et Facebook
* Capture et encodage des visages pour l'authentification biométrique

**Module CBIR**

* Extraction et stockage des signatures d'images
* Implémentation de différents descripteurs visuels
* Algorithmes de comparaison et de mesure de similarité
* Interface de recherche et de visualisation des résultats

**Utilisation**

**Installation**

# Cloner le dépôt

git https://github.com/Moutail/cbir-authentification

cd cbir-authentification

# Installer les dépendances

pip install -r requirements.txt

# Lancer l'application

streamlit run app.py

**Configuration**

* Créer un compte MongoDB et configurer la variable MONGODB\_URI
* Organiser les images dans des sous-dossiers de animalsCbir/ par catégorie
* Configurer les identifiants OAuth pour Google et Facebook si nécessaire

**Premier lancement**

1. Créer un compte utilisateur
2. Se connecter à l'application
3. Accéder à la page "Recherche d'Images"
4. Extraire les signatures (bouton "Extraire toutes les signatures")
5. Téléverser une image et configurer les paramètres de recherche
6. Visualiser les résultats

**Algorithmes clés**

**Extraction de caractéristiques GLCM**

def glcm(image\_path):

img = cv2.imread(image\_path, 0)

co\_matrice = graycomatrix(img, [1], [np.pi/2], None, symmetric=False, normed=False)

contrast = graycoprops(co\_matrice, 'contrast')[0, 0]

# ... autres propriétés

features = [contrast, dissimilarity, homogeneity, correlation, energy, ASM]

return features**Recherche d'images similaires**

def recherche\_images(bdd\_signature, caracteristique\_requete, distance\_type, K):

img\_similaire = []

for instance in bdd\_signature:

carac, label, img\_chemin = instance[:-2], instance[-2], instance[-1]

# Calcul de distance selon la méthode choisie

img\_similaire.append((img\_chemin, dist, label))

img\_similaire.sort(key=lambda x: x[1])

return img\_similaire[:K] # Retourne les K plus similaires

**Crédits**

Ce projet se base sur les techniques de vision par ordinateur et de reconnaissance d'images étudiées dans le cadre du cours IA2: Vision artificielle et reconnaissance de formes (420-1AB-TT).

**Configuration**

1. Créez une base de données MongoDB et configurez l'URI dans le code
2. Organisez vos images dans des sous-dossiers par catégorie dans animalsCbir/
3. Configurez vos identifiants OAuth pour Google et Facebook

**Usage**

1. Inscrivez-vous et connectez-vous à l'application
2. Extrayez les signatures des images
3. Téléversez une image requête
4. Ajustez les paramètres de recherche
5. Explorez les résultats!

**Technologies**

* Streamlit
* OpenCV
* Scikit-image
* MongoDB
* Face Recognition

**Licence**

Ce projet est sous licence MIT. Voir le fichier LICENSE pour plus de détails.

https://github.com/Moutail/cbir-authentification