

Nom : Yonkeu Jacques Onyx

Documentation du Projet de Recherche d'images similaires

Fonctionnement :

- Pour lancer l'application, ouvrez le fichier app.py et lancez-le dans le terminal avec la commande « streamlit run app.py »
- La première étape est de s'authentifier avec la reconnaissance faciale. Si votre visage est reconnu, vous pourrez passer à l'étape suivante, sinon vous ne pourrez pas continuer.
- Une fois l'authentification faite, vous pouvez televerser une image, préciser le nombre d'images similaires que vous voulez trouver, choisir le descripteur et la mesure de distance et enfin lancer la recherche.
- Les résultats de la recherche s'afficheront juste en bas, vous n'aurez qu'à défiler pour les visualiser.

Choix Techniques :

- L'application est faite en streamlit, car c'est un Framework interactif facile à utiliser et il s'intègre bien avec le python
- Le projet est divisé en 4 fichiers python qui s'intègrent les uns dans les autres :
 - Descripteurs.py : Contient les 3 descripteurs, GLCM, Haralick, Bitdesc et la concaténation des 3
 - Extraction.py : Sert à créer les fichiers signatures de chacun des descripteurs
 - Distances.py : Contient les fonctions des mesures de distance, ainsi que la fonction qui permet d'effectuer la recherche
 - App.py : l'application centrale qui permet à l'utilisateur d'interagir avec les autres fichiers et effectuer des recherches selon les critères de son choix
- Pour l'authentification, j'ai utilisé la librairie face-recognition de python, et session_state de streamlit pour faire persister l'état de l'authentification, puisque streamlit provoque un re-render a chaque interaction, ce qui réinitialise les variables.

Dépendances :

- Le dataset
- SignaturesAll.npy pour la reconnaissance faciale
- Signatures_glcmm_RGB.npy
- Signatures_haralick_RGB.npy
- Signatures_bitdesc_RGB.npy
- Signatures_Concat_RGB.npy