

Identification faciale

Présentation du projet

Ce projet va vous permettre de mettre à profit les connaissances acquises lors de l'enseignement du machine learning et du traitement d'images. Nous allons utiliser ce que l'on appelle le transfert learning. L'idée consiste à utiliser un (ou des) réseau de neurones déjà entraîné avec des milliers d'images pour apprendre une petite base d'images personnelles. Le premier réseau que nous allons utiliser est [MTCNN](#) et est spécifique à la l'extraction de visages, il a été développé à partir de la référence [1]. Ce réseau nous donne uniquement la localisation et les points caractéristiques des visages extraits. Le second réseau développé pour de l'identification faciale est [vggface](#).

Ces deux réseaux sont implémentés dans la librairie Deepface, celle-ci permet d'ailleurs de tester d'autres réseaux.

Une lecture attentive du site de [Deepface](#) est indispensable au bon déroulement de ce projet. Vous y trouverez les différents détecteurs de visage mais aussi les différents réseaux d'identification de visages, ainsi que les fonctions implémentées pour l'usage de tous ces réseaux. Vous verrez que la librairie implémente d'autres fonctionnalités que la détection et l'identification de visages.

[1] Zhang, K., Zhang, Z., Li, Z., and Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, 23(10):1499–1503.

Installation

Toutes les librairies utiles à ce projet sont contenues dans l'environnement K3_2.yml, il est installé sur les PC de l'école mais si vous voulez travailler sur vos propres PC alors il faut installer l'environnement K3_2 :

Pour ceci après avoir téléchargé le fichier K3_2.yml, dans une fenêtre de commande sous Anaconda prompt tapez :

```
conda env create -f K3_2.yml
```

Après la création de l'environnement, lancez le spyder associé à cet environnement.

Exemple

Téléchargez le fichier [exemple_etu_deepface.py](#)

Ce programme fonctionne avec une webcam, vous pouvez le modifier pour appeler à la place une image si vous n'avez pas de webcam. Ce programme permet de détecter tous les visages présents puis de les enregistrer au format jpeg. Si vous avez une webcam ce programme fonctionnera sans aucune modification, cela vous permettra de vérifier que tout est bien installé correctement. Sinon appelez la « hotline », visitez les forums etc...

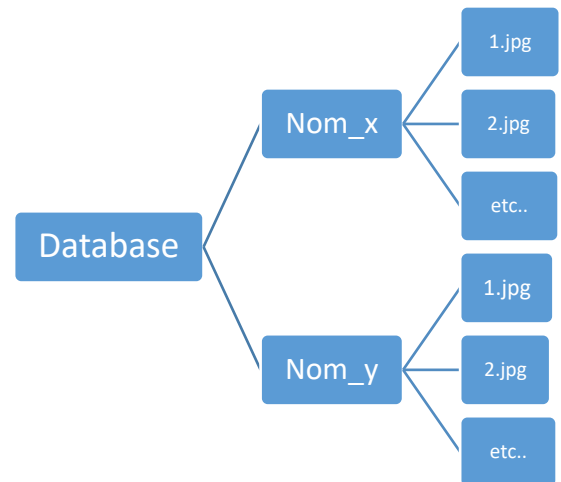
Objectifs

Nous allons créer un programme d'identification des étudiants de iese5 option ISA .

Création de la base

Pour faciliter l'échange et la gestion des données nous allons tous adopter la même convention.

Chacun donnera un répertoire à son nom dans lequel il y aura quelques photos obtenues et mises en forme avec un programme similaire à `exemple_tuto.py`. On appellera les images simplement 1.jpg, 2.jpg etc... c'est le nom du répertoire qui fera l'identification.



Lecture et parcours des images de la base

Une façon de gérer facilement cette base est donnée par ces quelques lignes :

```
employee_pictures=[]
liste_employee=np.array(["nom_x","nom_y"])
# directory du projet
ROOT_DIR = os.path.abspath("../")
# Directory des images où sont les visages à identifier
N=2 (car ici on a 2 noms)
IMAGE_DIR = os.path.join(ROOT_DIR, "mon_directorie/Database")
for i in np.arange(0,N) :
```

```
    employee_pictures.append(os.path.join(IMAGE_DIR,liste_employee[i]))
visages=dict()
```

Nous aurons dans `employee_pictures` les chemins pour accéder aux fichiers images de chaque personne.

L'accès à chaque image de ces 2 personnes se fera alors avec la double boucle :

```
for i in np.arange(0,N):
    j=0
    for file in listdir(employee_pictures[i]):
        num, extension = file.split(".")
        nom_fich= num+'.'+extension
        ..... Lire le fichier nom_fich
        visages[liste_employee[i],j] =..... appeler la fonction predict du model choisi (voir plus loin
        paragraphe model)on a le vecteur de caractéristiques du visage j de la personne i dans le dictionnaire
        visages.
        .....
    j=j+1
```

DeepFace

La librairie Deepface permet de tester différents extracteurs de visage et différents algorithmes d'identification de visages.

Il est très important d'aller sur le site de deepface voir tout ce qu'il est possible de faire, et connaître ainsi les différents modèles disponibles :

<https://github.com/serengil/deepface>

Prenez le temps de bien comprendre la librairie, ce qu'elle propose et comment l'utiliser.

Projet

- Installer l'environnement et Créer la base (à faire avant la première séance)
- Développer un système d'identification faciale à partir d'une webcam.
 - Etudiez l'influence des différents détecteurs de visage
 - Etudiez l'efficacité des différents algorithmes d'identification de visages
 - L'analyse doit se faire avec les mêmes contraintes que dans le cours de machine learning, c'est-à-dire qu'il faudra donner une analyse la plus précise possible : matrice de confusion et discussion par rapport aux effectifs.
 - Pour étudier l'efficacité de l'identification, vous serez obligé de passer par une étape de classification (avec autant de classes que de personnes), le nombre de photos par classe sera donc très important. Vous pourrez gérer vous-même votre classification en utilisant simplement le vecteur de caractéristiques en sortie du réseau puis en utilisant un svm ou autres classifieurs.
- Rendre : Démonstrateur et rapport d'analyses.