

TP Preparation image dataset



Plan

- I. Presenation de dataset
- II. Lecture des donnes
- III. Application filtre gaussien
- IV. Supprimer les duplications

Presentation de dataset

Abstract

This data consists of 640 black and white face images of people taken with varying pose (straight, left, right, up), expression (neutral, happy, sad, angry), eyes (wearing sunglasses or not), and size

Nom	Modifié le	Type	Taille
an2i	29/05/2021 18:54	Dossier de fichiers	
at33	29/05/2021 00:56	Dossier de fichiers	
boland	29/05/2021 00:56	Dossier de fichiers	
bpm	29/05/2021 00:56	Dossier de fichiers	
ch4f	29/05/2021 00:56	Dossier de fichiers	
cheyer	29/05/2021 00:56	Dossier de fichiers	
choon	29/05/2021 00:56	Dossier de fichiers	
danieln	29/05/2021 00:56	Dossier de fichiers	
glickman	29/05/2021 00:56	Dossier de fichiers	
karyadi	29/05/2021 00:56	Dossier de fichiers	
kawamura	29/05/2021 00:56	Dossier de fichiers	
kk49	29/05/2021 00:56	Dossier de fichiers	
megak	29/05/2021 00:56	Dossier de fichiers	
mittchell	29/05/2021 00:56	Dossier de fichiers	
night	29/05/2021 00:56	Dossier de fichiers	
phoebe	29/05/2021 00:56	Dossier de fichiers	
saavik	29/05/2021 00:56	Dossier de fichiers	
steffi	29/05/2021 00:56	Dossier de fichiers	
sz24	29/05/2021 00:56	Dossier de fichiers	
tammo	29/05/2021 00:56	Dossier de fichiers	
.anonr	20/10/1995 17:54	Fichier ANONR	1 Ko

an2i_left_angry_open - Copie - Copie	06/10/1995 19:41	Portable Gray Map...	42 Ko
an2i_left_angry_open - Copie	06/10/1995 19:41	Portable Gray Map...	42 Ko
an2i_left_angry_open	06/10/1995 19:41	Portable Gray Map...	42 Ko
an2i_left_angry_open_2	19/10/1995 04:07	Portable Gray Map...	4 Ko
an2i_left_angry_open_4	07/10/1995 21:54	Portable Gray Map...	1 Ko
an2i_left_angry_sunglasses	06/10/1995 19:44	Portable Gray Map...	42 Ko
an2i_left_angry_sunglasses_2	19/10/1995 04:07	Portable Gray Map...	4 Ko
an2i_left_angry_sunglasses_4	07/10/1995 21:54	Portable Gray Map...	1 Ko
an2i_left_happy_open	06/10/1995 19:39	Portable Gray Map...	42 Ko
an2i_left_happy_open_2	19/10/1995 04:07	Portable Gray Map...	4 Ko
an2i_left_happy_open_4	07/10/1995 21:54	Portable Gray Map...	1 Ko
an2i_left_happy_sunglasses	06/10/1995 19:42	Portable Gray Map...	42 Ko
an2i_left_happy_sunglasses_2	19/10/1995 04:07	Portable Gray Map...	4 Ko
an2i_left_happy_sunglasses_4	07/10/1995 21:54	Portable Gray Map...	1 Ko
an2i_left_neutral_open	06/10/1995 19:38	Portable Gray Map...	43 Ko
an2i_left_neutral_open_2	19/10/1995 04:07	Portable Gray Map...	4 Ko
an2i_left_neutral_open_4	07/10/1995 21:54	Portable Gray Map...	1 Ko
an2i_left_neutral_sunglasses	06/10/1995 19:42	Portable Gray Map...	42 Ko
an2i_left_neutral_sunglasses_2	19/10/1995 04:07	Portable Gray Map...	4 Ko
an2i_left_neutral_sunglasses_4	07/10/1995 21:54	Portable Gray Map...	1 Ko
an2i_left_sad_open	06/10/1995 19:40	Portable Gray Map...	42 Ko
an2i_left_sad_open_2	19/10/1995 04:07	Portable Gray Map...	4 Ko
an2i_left_sad_open_4	07/10/1995 21:54	Portable Gray Map...	1 Ko
an2i_left_sad_sunglasses	06/10/1995 19:43	Portable Gray Map...	42 Ko
an2i_left_sad_sunglasses_2	19/10/1995 04:07	Portable Gray Map...	4 Ko
an2i_left_sad_sunglasses_4	07/10/1995 21:54	Portable Gray Map...	1 Ko
an2i_right_angry_open	06/10/1995 19:41	Portable Gray Map...	42 Ko



Lecture des données

In computer programming, **glob** patterns specify sets of filenames with wildcard characters. For example, the Unix Bash shell command `mv *.txt textfiles/` moves (`mv`) all files with names ending in `.txt` from the current directory to the directory `textfiles`. Here, `*` is a wildcard standing for "any string of characters" and `*.txt` is a glob pattern

Cv2 : open Cv

Hashlib : pour le hash

```
9
10 import glob
11 import cv2
12 from hashlib import md5
13
14
```

lectures tous les les images pgm (**P**ortable **G**ray **M**ap **I**mage) d'un dossier appliquons la conversion a niveau de gris.

```
14
15 # ----- read images, converted to gray scale, resize it -----
16 image_list = []
17 t=0
18 for filename in glob.glob('faces/an2i/*.pgm'):
19     im=cv2.imread(filename)
20     im= cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
21     im=cv2.resize(im,(round(224),round(224)))
22     image_list.append(im)
23     t +=1
24
25
```

Application filtre gaussien

Avec Open-cv on peut appliquer filtre gaussien sur toutes les images.

```
28 # ----- filtre gaussien -----
29 list_image_filtred=[]
30 for image in image_list:
31     list_image_filtred.append(cv2.GaussianBlur(image,(5,5),0))
32
33 print("size of list image filtres : ",len(list_image_filtred))
34
35
36
```

Supprimer les duplications

On utilise la notion de hash des images pour determiner les images dupliquons, puisque les images qui sont dupliquons ayant le même hash md5.

Simultanément, stocke les éléments uniques dans une liste finale

```
40
41
42 image_list =list_image_filtred
43 # ----- remove duplicated images -----
44 print("intial list : ",len(image_list))
45 hashes = dict()
46 list_image3=[]
47
48 for image in image_list:
49     h = md5(image).hexdigest()
50     if h not in hashes:
51         hashes[h]=1
52         list_image3.append(image)
53
```

Enregistrer tous les éléments dans un dossier finale.

```
55  
56 |  
57 print("l'image finale : ",len(list_image3))  
58 i=0  
59 for image in list_image3:  
60     cv2.imwrite("./result/"+str(i)+".png",image)  
61     i+=1  
62  
63 # -----
```

