Extension to database system for facial recognition functions

FRF

Supervised professional practice RF03-03 – 09302020 -2

Researcher:

Msc Johnny Villalobos — johnny.villalobos.murillo@una.ac.cr
www.imaginglab.una.ac.cr
imaginglab@una.ac.cr

Collaborating students

Adriana Caballeros — adriana.caballeros.bustamante@est.una.ac.cr
Enzo Quesada — enzo.quesada.rojas@est.una.ac.cr

FRF is an PostgreSql-Python extension (data types and storage functions) to allow store, detect, recognize and compare faces in digital images. FRF uses Convolutional Neural Network from Dlib's open source licensing toolkit of machine learning algorithms, and land marck points for face recognition.

FRF extension

Composite data type

points numeric[ ]

Function contents

frf_load (file)

locate human faces in a image and return the 128-dimension unique face point for each face in the image.

Parameters	- file -	image file name
Returns	- points -	set of unique image points for face

frf_compare (points1, points2)

Compare a pair of two image poits to see if they match.

Parameters	points1	image points to compare
	points2	image point to compare against
Returns	match	a true o false value

FUNCTIONALITY

SQL

Create an image table

```
create table images
        (       image_id        integer,
                image_file      text,
                image_points    points
        )
```

Population the image table

```
insert into images (image_id, image_file, image_points)
  values (   1,    'image.jpg',     frf_load('image.jpg')   )
```

Compare two image_points

1. **select** frf_compare (points1, points2)

2. **select** frf_compare(select frf_load(frf_load('know.jpg'), frf_load('unknow.jpg'))

3. **select** frf_compare (**select** image_table_1.image_point
   from image_table_1
   where image_table_1.image_id = image_id_know
   ,
   select image_table_2. image_point
   from image_table_2
   where image_table_2.image_id = image_id_know

PYTHON

Import python package

import frf_db as db

Connect to database

db.PostgresDB(host, port, db_name, user, password)

parameteres

returns

Load image

load('image.jpg')

parameteres

returns

Compare images

compare_imgs (img1, img2)

parameteres

returns

Locate fases in image

Locate_fase(img)

parameteres

returns

Encoding fases in image

encodings(img1)

Parameteres

Returns

Installation guide

PostgreSql

1. **install lenguage in PostgreSql**

2. **Download frf extension from (www. .....)**

3. **Install frf as extension on postgres**

Python

instal frf_db