**Face Recognition con RTree**

Profesor Heider Sanchez

El objetivo de este laboratorio es probar el desempeño del RTree en tareas de Face Recognition.

**P1. Explorando el R-Tree**

* Seguir el tutorial 1 para instalar la librería rtree en Python.
* Analizar el performance del RTree construyendo el índice con N puntos y aplicando la búsqueda KNN.
  + En la tabla siguiente coloque la suma de los tiempos obtenidos al aplicar el KNN para K=3,6,9.

|  |  |  |
| --- | --- | --- |
| Tiempo en ms | RTree | Lineal Scan |
| N=10^2 | 5.966999999999501e-07 | 5.056999999999979e-07 |
| N=10^3 | 2.9109999999998857e-07 | 5.242699999999989e-06 |
| N=10^4 | 3.387000000000251e-07 | 8.31362e-05 |
| N=10^5 | 2.846999999990274e-07 | 0.0007299191000000019 |
| N=10^6 | 5.804999999980964e-07 | 0.008708752500000003 |

**P2.** **Explorando el Face\_Recognition**

* Descargar el [dataset de rostros](http://vis-www.cs.umass.edu/lfw/)
* Instalar la librería [Face Recognition](https://face-recognition.readthedocs.io/en/latest/readme.html) siguiendo el tutorial 2 e investigue como son obtenidos los vectores característicos en dicha librería.
* Seleccione muestra aleatoria de N pares de rostros (ej. N=5000)
  + Mostrar el histograma de distribución de distancias.
  + Usar distancia Euclidiana.

Distribución de distancias euclidianas entre imágenes para N=5000

Gráfico, Histograma

Descripción generada automáticamente

**P3**. **Face\_Recognition y KNN search**

* Adapte el RTree de Python para trabajar con los vectores característicos de Face Recognition.
* Implementar la búsqueda KNN con cola de prioridad sin índice (búsqueda secuencial).
* Realizar un análisis comparativo de los tiempos de la búsqueda secuencial y la búsqueda indexada.

|  |  |  |
| --- | --- | --- |
| Tiempo en ms | KNN-Secuencial | KNN-RTree |
| N=10^1 | 1.73600000000107e-07 | 1.6969999999894014e-07 |
| N=10^2 | 6.375999999974624e-07 | 2.961999999939735e-07 |
| N=10^3 | 4.210499999999229e-06 | 2.1370000000047184e-06 |

Se observa una ligera mejora en el rendimiento de las consultas. La mejora con el Rtree no es tan significativa por la alta dimensionalidad. En este caso, se está trabajando con vectores característicos de dimensión 128.

Se omitió el cálculo para N mayores dado que el tiempo de creación para N > 10^3 se mide en horas.

**Tutorial 1: Instalación de RTree en Python**

Más información de la librería

<https://rtree.readthedocs.io/en/latest/>

<https://github.com/Toblerity/Rtree>

1. **Instalar RTree**

[Linux]

sudo apt install python3-rtree

[Anaconda]

conda install rtree

1. **Probar los métodos de inserción y consulta**

> python rtree\_test.py

**Tutorial 2: Instalación de Face Recognition**

Más información de la librería

<https://face-recognition.readthedocs.io/en/latest/readme.html>

1. **Instalar OpenCV**

[Linux]

sudo apt install python3-opencv

[Anaconda]

conda install -c conda-forge opencv

Probar en Python

>>>import cv2

1. **Instalar Face Recognition**

[Linux]

sudo apt install python3-pip

sudo apt install python3-setuptools

sudo pip3 install dlib

sudo pip3 install face\_recognition

[Anaconda]

pip install face\_recognition

Probar en Python

>>>import face\_recognition

Si no funciona dlib, compilar la librería.

<https://gist.github.com/ageitgey/629d75c1baac34dfa5ca2a1928a7aeaf>

1. **Probar instalación desde terminal**

> face\_recognition ./fotos\_bd/ ./fotos\_test/

1. **Probar un demo con flask**

[Linux]

sudo apt install python3-flask

[Anaconda]

conda install flask

> python ws\_recognition.py