**Face Recognition con RTree**

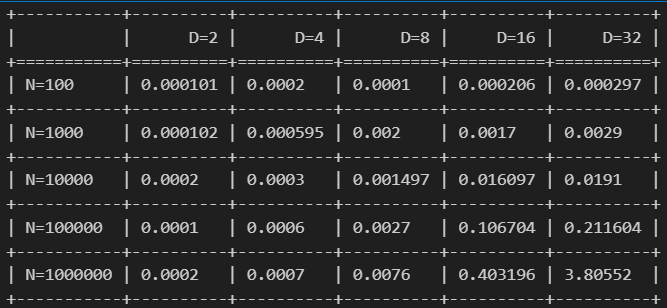
Profesor Heider Sanchez

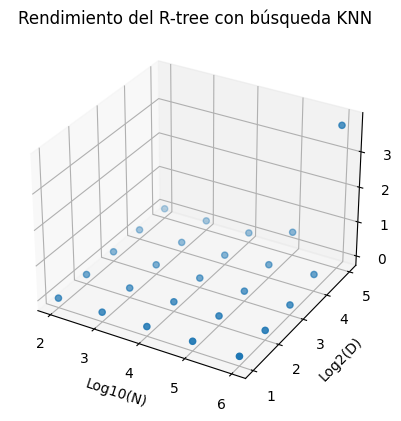
El objetivo de este laboratorio es probar el desempeño del RTree en tareas de Face Recognition.

**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 aleatorios y aplicando la búsqueda KNN.
  + En la tabla coloque la suma de los tiempos al aplicar el KNN para K=8 y variando los siguientes parámetros:
    - N = 10^2, 10^3, 10^4, 10^5, 10^6
    - D = 2^1, 2^2, 2^3, 2^4, 2^5
  + Realizar un grafico en 3 dimensiones (N, D, Técnica).
  + El R-tree debe estar en memoria principal (RAM).

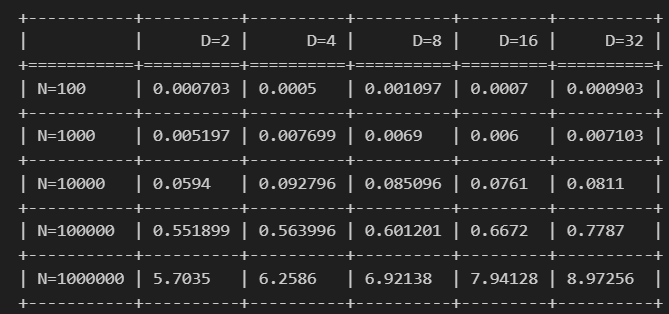
Tiempo en ms (solo tiempo de búsqueda, no considerar indexación)

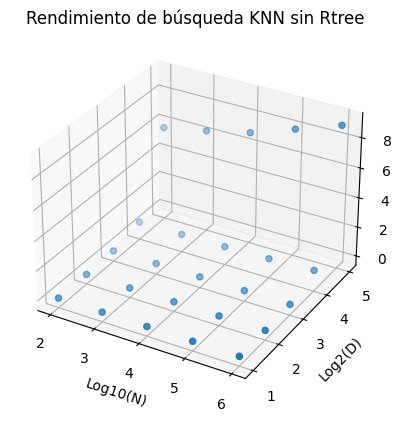
Indexando en el RTree



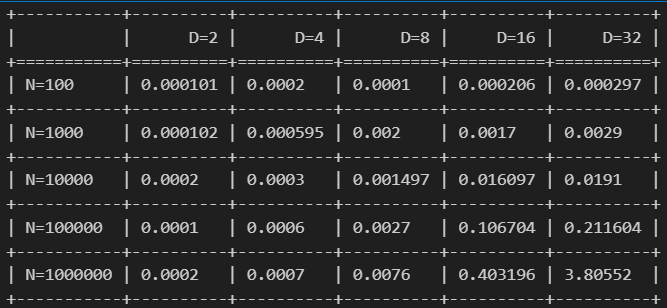
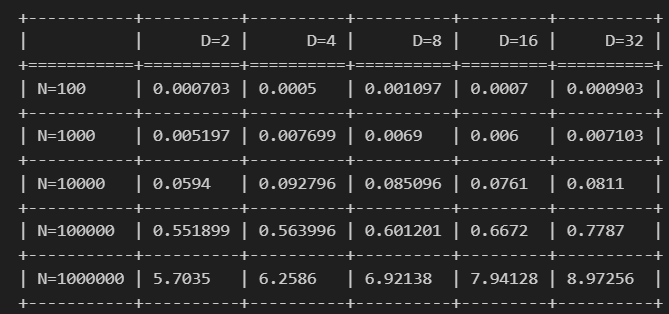
La altura de cada punto indica qué tanto demoró la búsqueda (a mayor altura, mayor tiempo).

Búsqueda Lineal (Sin índices):





Conclusiones



Sin Rtree

Con Rtree

Se observa una mejora significa del tiempo de búsqueda utilizando el Rtree, sobre todo a partir de N>1000, siendo 3.81 segundos el mayor tiempo obtenido usando Rtree y 8.972 segundos sin el Rtree. No obstante, construir el Rtree en RAM demoró varios minutos, obteniendo una duración de la experimentación de 23 minutos. Por otro lado, sin utilizar el Rtree se logró realizar todas las operaciones en 6 minutos.

**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 una muestra aleatoria de N pares de rostros (ej. N=5000)
  + Mostrar el histograma de distribución de distancias.
  + Usar distancia Euclidiana.

**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=10^2 |  |  |
| N=10^3 |  |  |
| N=10^4 |  |  |
| N=10^5 |  |  |
| N=10^6 |  |  |

**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