

Proyecto 2 (Hito 2): Base de Datos Multimedia

1- Introducción

El logro del estudiante está enfocado a entender y aplicar los algoritmos de búsqueda y recuperación de la información basado en el contenido.

Esta segunda parte del proyecto está enfocado a la construcción óptima de una estructura multidimensional para dar soporte a las búsqueda y recuperación eficiente de imágenes en un servicio de reconocimiento facial.

2- Backend: Servicio de Reconocimiento Facial

Implementar un web service para la identificación automática de personas a partir de una colección grande de imágenes de rostros.

El procedimiento general consiste en lo siguiente:

- Extracción de características
- Indexación de vectores característicos para búsquedas eficientes
- Algoritmo de búsqueda de los K-vecinos más cercanos

Extracción de características

Para la extracción de características se usará la librería [Face Recognition](#). En dicha librería ya se encuentra implementado las técnicas necesarias para obtener de cada imagen una representación numérica y compacta (encoding). El tamaño del vector característico es de 128 ([ver referencia](#)). La eficacia del reconocimiento ha sido probada con modelos de búsqueda basados en *deep learning* (99.38% de precisión). En esta tarea vamos a usar **solo las** funciones **básicas** de dicha librería.

Se usará una colección de referencia con más de 13 mil imágenes de rostros de personas, disponible en el [siguiente enlace](#). Algunas personas tienen más de una imagen asociada.

El grupo puede optar por otro dataset de rostros [disponibles en la web](#).

Indexación y Búsqueda

- Implementar el índice R-Tree (o usar [librería existente](#)) para indexar todos los vectores característicos que serán extraídos de cada imagen de la colección. Debe considerar el hecho de que el índice puede reducir su eficiencia para vectores con dimensiones muy altas ([maldición de la dimensionalidad](#)). Estudie como mitigar este problema si se presenta en su implementación.
- Implementar el algoritmo de búsqueda KNN con cola de prioridad, el cual recibe como parámetro el objeto de consulta y la cantidad de objetos a recuperar K.
- Implementar la búsqueda por rango, el cual recibe como parámetro el objeto de consulta y un radio de búsqueda.
- Experimento:
 - o Aplique el KNN-RTree y el KNN-secuencial haciendo crecer el tamaño de la colección (N) para comparar la eficiencia en tiempos de ejecución.

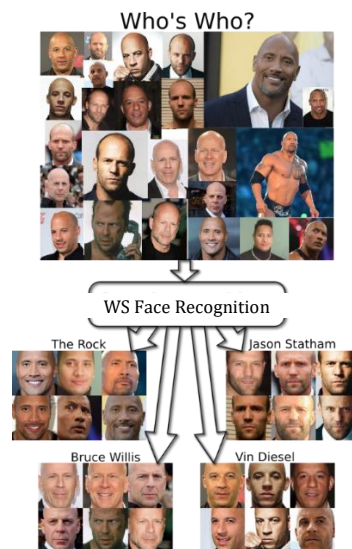
Tiempo	KNN-RTree	KNN-Secuencial
--------	-----------	----------------

N = 100		
N = 200		
N = 400		
N = 800		
N = 1600		
N = 3200		
N = 6400		
N = 12800		

* Para todo el valor de K es 8.

3- Frontend: Motor de Búsqueda

Construir una aplicación frontend que permita interactuar con el web service de reconocimiento facial. La consulta es una imagen cualquiera (puede usar fotos externas) y debe responder a las preguntas del tipo ¿Quiénes son las personas más parecidas a Bruce Willis?



- Muestre los resultados de búsqueda interactivamente.
 - o El personaje de consulta y el valor de k debe ser un dato de entrada.

4- Entregable

Los alumnos formaran grupos de máximo de tres integrantes.

El proyecto estará alojado enteramente en GitHub, GitLab o Bitbucket.

Trabajar de forma colaborativa, se considerará para su nota individual.

En el Canvas subir solo el **enlace público** del proyecto.

La fecha límite de entrega es el 02/12/2020 (no habrá prorroga).

5- Informe del proyecto

- Archivo Readme o Wiki
- Ortografía y consistencia en los párrafos.
- El informe debe describir todos los aspectos importantes de la implementación.

- Construcción del índice RTree.
 - Algoritmo de búsqueda KNN con los vectores característicos.
 - Análisis y discusión de la experimentación.
 - Incluir imágenes/diagramas de apoyo.
- Además, acompañar al informe un video de presentación del producto, en donde se visualice el programa en acción (hay que vender).
 - Los resultados deben visualizarse de forma amigable e intuitiva para el usuario.

6- Rúbrica

Va adjunto