

Proyecto Final Área Imágenes

Profesor: Claudio Pérez F.

Auxiliar: Juan Pablo Pérez.

Fecha entrega: Ultimo día de clases

El objetivo de este proyecto final es desarrollar e implementar un algoritmo de búsqueda de imágenes similares basado en su contenido (*Content Based Image Retrieval*, CBIR) utilizando distintos métodos de extracción de características. Para el desarrollo de este proyecto se deben utilizar dos bases de datos:

1. *INRIA Holidays dataset*. Debe descargar los archivos *jpg1.tar.gz* y *jpg2.tar.gz*^[a] que contienen imágenes de consulta para 500 clases distintas y las imágenes que deberán ser almacenadas en el buscador, y que están asociadas a esas 500 clases.
2. *GPR1200* (General-Purpose image Retrieval set with 1200 clases): Debe descargar el archivo *GPR1200.zip*^[b]. Contiene 1200 clases con 10 ejemplos por clase.

La figura 1 muestra las principales etapas de un sistema CBIR.

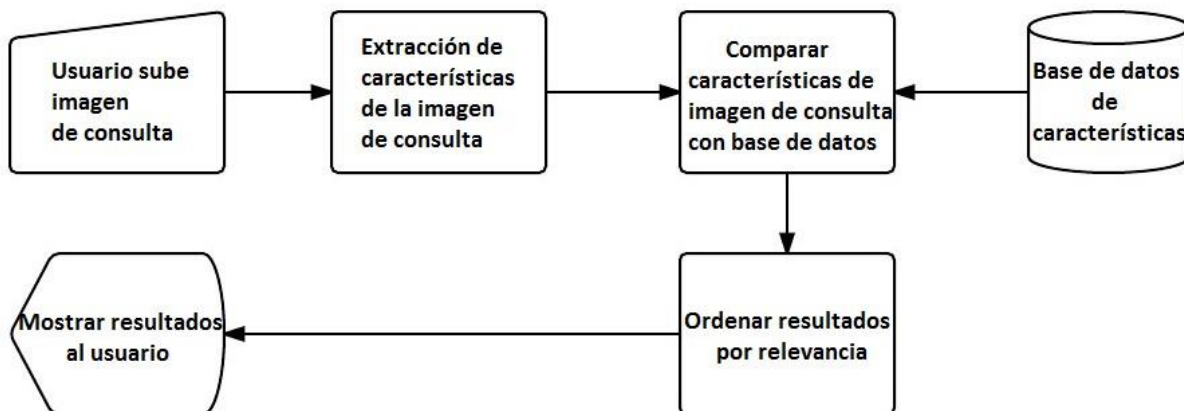


Figura 1: Etapas de un sistema de búsqueda por consulta de imágenes.

Para implementar este sistema CBIR, usted deberá comparar el desempeño de dos extractores de características distintos [1]: un método *handcrafted* o clásico (métodos basados en color, Local Binary Patterns, HOG, etc), y una red convolucional o CNN pre entrenada (VGG, ResNet, MobileNet, etc).

Este proyecto deberá ser desarrollado en Python, podrá utilizar las librerías *keras*, *scikit-image*, *OpenCV* donde encontrará los extractores de características basados en CNN's y podrá implementar un método clásico (puede usar otra librería si lo desea, pero debe especificar el método de instalación y compilación si fuese necesario).

Para desarrollar el sistema debe implementar las siguientes etapas:

1) Cálculo de vector de características

Implemente una función que calcule el vector de características de una imagen. Esta función debe recibir como parámetros (mínimos) la imagen de consulta y el tipo de extractor (clásico o CNN). Además, debe retornar un vector con las características calculadas.

Utilizando esta función, calcule y guarde en memoria las características de cada imagen de la base de datos, a modo de ahorrar tiempo en cada consulta (así se evita computar dichas características cada vez que se realiza una consulta).

Nota: No es necesario que se incluyan los vectores de características almacenados en la entrega de la tarea, pero si el algoritmo que las generó y almacenó.

2) Medida de similitud

Implemente alguna medida de similitud entre vectores de características. (distancias sugeridas: Euclidiana, Distancia Coseno).

3) Ordenar resultados por relevancia

Desarrolle un esquema de comparación para cada imagen de *img_query* que dé como resultado un conjunto de valores de similitud entre una imagen de consulta y las imágenes almacenadas en *img_database*.

4) Una vez obtenidos los valores en 3), para todas las imágenes de consulta, ordene los resultados por relevancia. Aplique la siguiente medida de ranking para obtener la posición promedio de las imágenes arrojadas por el buscador que pertenecen a la misma clase de la imagen buscada.

$$Rank = \frac{1}{N_{rel}} \sum_{i=1}^{N_{rel}} R_i,$$

Donde N_{rel} = Número de imágenes relevantes para una imagen de consulta particular (imágenes de la misma clase que la imagen de consulta).

R_i = corresponde a la i -ésima imagen relevante obtenida (posición de las imágenes de la misma clase que la imagen de consulta).

5) Para evaluar la robustez de cada uno de los métodos implementados, calcule el Rank Normalizado para los 2 métodos distintos de extracción de características.

El Rank Normalizado se calcula de la siguiente manera:

$$\widetilde{Rank} = \frac{1}{N \times N_{rel}} \left(\sum_{i=1}^{N_{rel}} R_i - \frac{N_{rel}(N_{rel} + 1)}{2} \right)$$

Donde N_{rel} = Número de imágenes relevantes para una imagen de consulta particular (imágenes de la misma clase que la imagen de consulta).

R_i = corresponde a la i -ésima imagen relevante obtenida (posición de las imágenes de la misma clase que la imagen de consulta). N = Tamaño del set de imágenes.

Los valores que se pueden obtener con el Rank normalizado están en el rango $[0,1]$, Cero corresponde al mejor desempeño (todas las imágenes buscadas están en las primeras posiciones), 0.5 corresponde al desempeño random (las imágenes buscadas aparecen aleatoriamente ordenadas)

6) Muestre resultados cualitativos de algunas consultas, identificando la imagen de consulta y los 10 primeros resultados retornados por el buscador (en orden de relevancia).

7) Optimización de tiempo de búsqueda y reducción de uso de memoria. Investigue qué estrategias son típicamente utilizadas para abordar el problema de reducción de tiempos de búsqueda y uso de memoria, mediante compresión de vectores de características, clustering u optimización en el cálculo de distancias. Implemente al menos una alternativa y compárela con los resultados anteriores, de acuerdo a la métrica implementada originalmente en términos de tiempos de búsqueda y memoria utilizada por los vectores almacenados en la base de datos.

El proyecto se debe realizar en grupos de 2 integrantes. Se debe entregar el último día de clases un reporte del trabajo realizado, además incluir los códigos necesarios con las debidas instrucciones para realizar el buscador. Debe incluir todas las referencias de trabajos revisados para el desarrollo de su proyecto.

Se debe incluir un archivo .ppt con una presentación del método empleado, los resultados obtenidos y las conclusiones.

El día del examen, cada grupo deberá presentar una ppt con todo el desarrollo del proyecto. La presentación deberá tener una extensión máxima de 10 minutos.

Los alumnos que decidan desarrollar este proyecto deben inscribirse, informando quiénes son los integrantes del grupo, mediante un mail al Profesor Claudio Pérez con copia al profesor auxiliar. **El plazo para la inscripción es hasta el 21 de octubre.** Los alumnos que no informen sobre la inscripción de este proyecto (e igualmente lo desarrollen) tendrán una penalización de 1 punto en la nota.

[a]NOTA: La base de datos *INRIA Holidays dataset* se puede encontrar en el siguiente enlace <http://lear.inrialpes.fr/people/jegou/data.php>.

La clase de cada imagen está dada en el nombre de cada archivo. Cada imagen está codificada de la siguiente manera:

1 [clase codificada en 3 dígitos] [ID_imagen codificado en 2 dígitos]. Las imágenes de consulta son las que poseen ID_imagen = 00 y las almacenadas en la base de datos son aquellas donde ID_imagen distinto a 00.

[b]NOTA: La base de datos *GPR1200* se puede encontrar en el siguiente enlace

<https://visual-computing.com/files/GPR1200/GPR1200.zip>

La clase de cada imagen está dada en el nombre de cada archivo. Cada imagen está codificada de la siguiente manera:

[clase] [nombre original].jpg. Las 12000 imágenes se utilizan como consulta sobre el resto de la base de datos.

Las imágenes de la base de datos sólo se pueden utilizar en este proyecto académico. Para cualquier otro uso se debe referenciar los trabajos:

INRIA Holidays

Herve Jegou, Matthijs Douze and Cordelia Schmid, "*Hamming Embedding and Weak geometry consistency for large scale image search*", Proceedings of the 10th European conference on Computer vision, October, 2008.

GPR1200

Konstantin Schall, Kai Uwe Barthel, Nico Hezel and Klaus Jung

GPR1200: A Benchmark for General-Purpose Content-Based Image Retrieval

Referencia

[1] Latif, Afshan, et al. "Content-based image retrieval and feature extraction: a comprehensive review." *Mathematical Problems in Engineering* 2019 (2019).