

Tarea 1

CC5213 – Recuperación de Información Multimedia

Profesor: Juan Manuel Barrios

Fecha: 15 de agosto de 2023

El objetivo de esta tarea es implementar un **buscador de imágenes duplicadas o derivadas**. Dado un conjunto de imágenes originales R y un conjunto de imágenes posiblemente duplicadas Q , se desea determinar para cada imagen $q \in Q$ si existe alguna imagen $r \in R$ tal que $T(r)=q$ para alguna transformación T válida.

Llamaremos *transformación* a una función T que recibe como entrada una imagen r y genera como salida una imagen q que es una edición de r . Una transformación será válida cuando es posible reconocer el contenido de r en q .



Figura 1: Ejemplo del conjunto R con imágenes originales



Figura 2: Ejemplos del conjunto Q con imágenes derivadas

En la Figura 1 se ven imágenes originales obtenidas del dataset **mirflickr25k**¹ y en la Figura 2 se ven imágenes derivadas correspondientes. En la tarea se evaluarán las siguientes transformaciones: bajar calidad, insertar texto, ajustar colores, recortar una zona (crop), reflejado horizontal (flip), perspectiva (homografía), agregar bordes.

¹ MIRFLICKR: <https://press.liacs.nl/mirflickr/>

Para la tarea debe implementar dos comandos:

1. Un comando que recibe el nombre de carpeta donde están las imágenes originales **R** y el nombre de una carpeta nueva donde se guardarán sus datos:

```
python tarea1-indexar.py [dir_imagenes_R] [dir_descriptores_R]
```

Para cada imagen en `dir_imagenes_R` calcula un descriptor de contenido (a su elección). Al finalizar guarda en la carpeta `dir_descriptores_R` uno o más archivos con los nombres de las imágenes procesadas y sus descriptores.

2. Un comando que recibe el nombre de la carpeta donde están las imágenes a buscar **Q**, la carpeta con los descriptores de **R** (creadas por el comando anterior) y el nombre del archivo a crear con los resultados:

```
python tarea1-buscar.py [dir_imagenes_Q] [dir_descriptores_R] [resultados.txt]
```

Para cada imagen de **Q** calcula un descriptor de contenido, que puede ser o no ser el mismo al calculado para **R** (a su elección). Luego para cada descriptor de **Q** se busca el descriptor más cercano de **R**. Finalmente, genera un archivo de texto con el resultado de imágenes más parecidas.

El archivo de texto a generar debe tener una fila por imagen en **Q** y para cada fila tres columnas separadas por tabuladores con la imagen más cercana y su distancia:

```
nombre_imagen_q \t nombre_imagen_r \t distancia_entre_q_r
```

Notar que mientras más similares sean las imágenes, menor debe ser la distancia entre sus descriptores.

Datasets

Junto con este enunciado encontrará una implementación base para `tarea1-indexar.py` y `tarea1-buscar.py`, varios conjuntos de prueba (llamados `dataset_a`, `dataset_b`, `dataset_c`) con imágenes originales y transformadas, y un programa de evaluación con la respuesta esperada para cada imagen de consulta.

Puede descargar los conjuntos de prueba y los programas base para implementar la tarea en el siguiente enlace:

<https://drive.google.com/drive/folders/1FisWVKF-mCktMy7gvmSIBKxwrMmPSQif>

Evaluación

El programa de evaluación se invoca con:

```
python evaluarTareal.py
```

Este programa invoca `tareal-indexar.py` y `tareal-buscar.py` sobre cada dataset de prueba, lee los archivos de resultados generados, los compara con la respuesta esperada, determina la distancia umbral de decisión que logra un mejor balance entre respuestas correctas y erróneas, y finalmente dependiendo de la cantidad de detecciones correctas señala la nota obtenida.

Su tarea será evaluada en los conjuntos de imágenes publicados y en otros conjuntos similares. Su tarea **no puede demorar más de 15 minutos (900 segundos)** en procesar cada conjunto de imágenes de prueba.

Existe la posibilidad de obtener **hasta 1 punto extra de bonus** para otras tareas si logra detectar la mayoría de los duplicados en los datasets de evaluación.

Entrega

El plazo máximo de entrega es el **viernes 01 de septiembre de 2023** hasta las 23:59 por U-Cursos. La tarea la puede implementar en **Python 3** usando cualquier función de OpenCV, NumPy, SciPy y otras librerías gratuitas. Debe subir sólo el código fuente de su tarea (archivos .py). No envíe datasets ni descriptores ya calculados.

Se recomienda incluir un archivo de texto señalando el sistema operativo en que realizó su tarea e incluir la salida que entregó `evaluarTareal.py` en su computador.

Opcionalmente, puede implementar la tarea en **C++ 11**. En este caso puede usar cualquier función de OpenCV y de la biblioteca estándar (std). Debe subir el código fuente de su tarea junto con los pasos necesarios para la compilación.

La tarea es *individual* y debe ser de su autoría. En caso de detectar copia o plagio se asignará nota 1.0 a los involucrados.