

# Tarea 1

CC5213 – Recuperación de Información Multimedia

Profesor: Juan Manuel Barrios

Fecha: 22 de marzo de 2021

El objetivo de esta tarea es implementar un **buscador de imágenes derivadas o duplicadas**. 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.



Figura 1: Ejemplo del conjunto **R** con imágenes originales



Figura 2: Ejemplos del conjunto **Q** con imágenes derivadas

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  $q$ .

En la Figura 1 se ven imágenes originales obtenidas del dataset **mirflickr25k**<sup>1</sup> 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) y reflejado horizontal (flip).

Para la tarea se 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 datos:

```
python tarea1-procesar.py [dir_imágenes_R] [datos_R]
```

Para cada imagen en `dir_imágenes_R` se calcula su descriptor de contenido y al finalizar guarda en la carpeta `datos_R` uno o más archivos con los nombres de las imágenes procesadas y sus descriptores.

---

<sup>1</sup> MIRFLICKR: <http://press.liacs.nl/mirflickr/>

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

```
python tareal-buscar.py [dir_imágenes_Q] [datos_R] [resultados.txt]
```

Se cargan los vectores en `datos_R`, se calculan los vectores de las imágenes en `dir_imágenes_Q`, para cada vector de **Q** se obtiene el vector más cercano de **R** y finalmente genera un archivo de texto con el resultado.

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 es la distancia entre sus descriptores.

## Evaluación

Junto con este enunciado encontrará varios conjuntos de prueba y un programa que ejecuta su tarea y muestra el resultado obtenido. La evaluación se ejecuta con:

```
python tareal-test.py
```

Su tarea será evaluada en los conjuntos de imágenes publicados y en otros conjuntos similares. Su tarea no puede demorar más de 30 minutos en procesar cada conjunto de imágenes de prueba.

Además del código fuente de su tarea, deberá entregar un archivo `Readme.txt` donde debe explicar brevemente su implementación: el o los descriptores implementados, método de búsqueda, librerías usadas, forma de compilación y resultados obtenidos.

Se evaluarán tres aspectos de la tarea: orden del código fuente (20%), claridad del archivo `Readme.txt` (20%) y la calidad de los resultados logrados (60%).

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

## Entrega

El plazo máximo de entrega es el **lunes 5 de abril** hasta las 23:59 por U-Cursos. No se aceptarán tareas atrasadas.

La tarea la puede implementar en **Python 3**, **C++ 11** o **Java 8** usando la librería **OpenCV**. Puede utilizar cualquier función de OpenCV y SciPy.

Enviar solo el código fuente de su tarea (archivos `.cpp`, `.py` o `.java`) y el archivo `Readme.txt`. No enviar imágenes ni descriptores ya calculados.

**La tarea es \*individual\*. En caso de detectar copia entre estudiantes o plagio de Internet se asignará nota 1.0 a los involucrados.**