

# Tarea 1

## CC5213 – Recuperación de Información Multimedia

### 21 de Marzo de 2018

El objetivo de esta tarea es implementar un detector de avisos comerciales en televisión. Dado un video con extractos de la programación emitida durante un día por un canal de televisión local, y dada una carpeta que contiene diferentes videos de comerciales, cada uno de largo entre 15 y 40 segundos, debe implementar un programa que detecte todas las apariciones de los comerciales en el video de televisión.



Figura 1: Cuadros de ejemplo de un video de televisión.

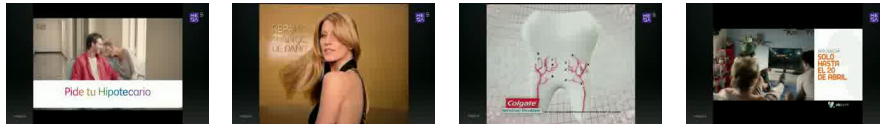


Figura 2: Cuadros de ejemplo de avisos comerciales.

Su tarea debe recibir como parámetro el nombre del video de televisión que se desea procesar y el nombre de la carpeta donde se encuentran todos los comerciales a buscar. Luego de varias etapas la tarea debe generar un archivo de texto de nombre `detecciones.txt` con todas las apariciones de los comerciales encontradas en el video de televisión. El archivo de salida debe tener formato de cuatro columnas separadas por un tabulador (`\t`), cada aparición en una línea siguiendo el siguiente formato:

```
video_television \t segundos_inicio \t segundos_largo \t video_comercial
```

Su tarea debe estar dividida en las siguientes tres etapas:.

1. **Extracción de características:** Selecciona cuadros de cada video a una tasa constante (por ejemplo, tres cuadros por segundo) y para cada cuadro seleccionado calcula un vector que representa su contenido visual. El conjunto de descriptores para cada video los puede mantener en memoria o guardar en uno o más archivos con el formato que estime conveniente.
2. **Búsqueda por similitud:** Carga en memoria los descriptores del video de televisión (conjunto  $Q$ ) y los descriptores de todos los videos de comerciales (conjunto  $R$ ). Luego para cada descriptor  $q$  de  $Q$  busca los  $k$  descriptores más cercanos en el conjunto  $R$  (con  $k \geq 1$ ) de acuerdo a alguna función de

distancia. El resultado de la búsqueda, esto es, para cada cuadro de  $Q$  los  $k$  cuadros de  $R$  más parecidos, lo puede mantener en memoria o guardar en uno o más archivos con el formato que estime conveniente.

3. **Detección de apariciones:** Carga en memoria la lista de cuadros similares para cada cuadro del video de televisión. Luego, revisa las listas correspondientes a cuadros consecutivos del video de televisión y localiza secuencias provenientes de un mismo comercial con el mismo desfase de tiempo. Se debe notar que se buscan apariciones completas de cada comercial y no apariciones parciales.

Su tarea debe reportar la mayor cantidad de comerciales emitidos, con la mejor exactitud posible en los tiempos de inicio y largo, junto con la menor cantidad de detecciones incorrectas. Encontrará videos de prueba y código fuente de ejemplo en la dirección:

<http://juan.cl/CC5213-2018/>

Además, dispondrá de un archivo de nombre `gt.txt` con la respuesta correcta para cada video de televisión y de un programa `evaluar.py` que calcula la cantidad de respuestas correctas en el archivo `detecciones.txt` generado por su tarea.

Puede comprobar cada aparición usando *VideoLAN Player* con el comando:

```
vlc "television.mp4" --start-time=[segundos_inicio] --run-time=[segundos_largo]
```

### Entrega:

- El plazo máximo de entrega es el **Martes 10 de Abril a las 23:59** por U-Cursos. No se aceptarán tareas atrasadas.
- La tarea la puede implementar en **Python 3** o **C++ 11** usando la librería **OpenCV 3**. No se permiten entregas de un IPython Notebook (.ipynb).
- Puede utilizar cualquier función de OpenCV.
- Debe entregar todos los códigos fuentes, más un archivo `Readme.txt` con las instrucciones de uso de su tarea, las librerías usadas y su versión.
- No incluya videos ni datos de prueba.

### Evaluación:

- Se evaluará el orden del código fuente, la claridad del archivo `Readme.txt` y la calidad de la detección. Cada detección reportada se considerará correcta si existe intersección con una emisión real. Cada detección falsa elimina una detección correcta.
- La tarea es **\*individual\***.