

Documeentación Cuarta Tarea Programada

Minor Sancho Valverde, tivin.minor10@gmail.com,
Danny Chaves Chaves, dxnnx@me.com,
Oscar Chavarria Campos, ocach14@hotmail.com

16 de noviembre de 2016

1. Introducción

La masificación del internet, consecuencia del desarrollo de tecnologías de transimisión y compresión han llevado al uso generalizado y la disponibilidad de video digital. Aplicaciones tales como bibliotecas digitales, aprendizaje adistancia, vídeo bajo demanda, vídeo digital difusión, la televisión interactiva y la información multimedia sistemas generan utilizan grandes colecciones de videos digitales. Esto ha creado la necesidad de herramientas que pueden buscar, explorar, y analizar este tipo de datos. Ejemplo de estos sistemas de análisis automático de vídeo, es el sistema ACE, desarrollado actualmente en el PRIS-Lab, en la Universidad de Costa Rica. Este sistema tiene por objetivo analizar de manera automatica videos defutbol extraídos de transmisiones televisivas. La segmentación temporal en videos digitales se refiere a la distinción de cuadros de corte o transición entre dossecuencias de video distintas.

2. Análisis del problema

Un cambio o corte en un video digital, se puede definir como un cambio abrupto o gradual entre dos cuadros consecutivos de un video: $Ut1$ y Ut . El algoritmo a implementar utiliza como descriptor el histograma $ht(z)$ para cada cuadro o imagen Ut . Un histograma se define como un arreglo que contabiliza la cantidad de apariciones de la escala de gris z , normalizado a la cantidad de pixels N en la imagen. Básicamente, el algoritmo por cada par de cuadros $Ut1$ y Ut calcula los histogramas $ht1$ y ht y cuantifica la disimilitud entre ambos histogramas, con el funcional $d(ht1;ht)$. La medida de disimilitud o distancia puede venir dada por la distancia euclidiana, de Mahalanobis o de Bathacharyya (esta última esla que se usará en el presente trabajo).

3. Diseño de la solución

Para desarrollar la solución del problema antes mencionado se utilizo el lenguaje de programación Python, ademas se utilizaron diversas librerias como: numpy, tensorflow y OpenCV. El uso de estas librerias es indispensable ya que OpenCV nos permitio cargar el video con el cual debiamos trabajar y tensorflow nos ayudo a paralelizar algunos calculo que debiamos realizar. Ademas de estas herramientas utilizamos el algoritmo de Bathacharyya el cual nos permitio encontrar el grado de disimilitud entre dos cuadros o "frames" del video.

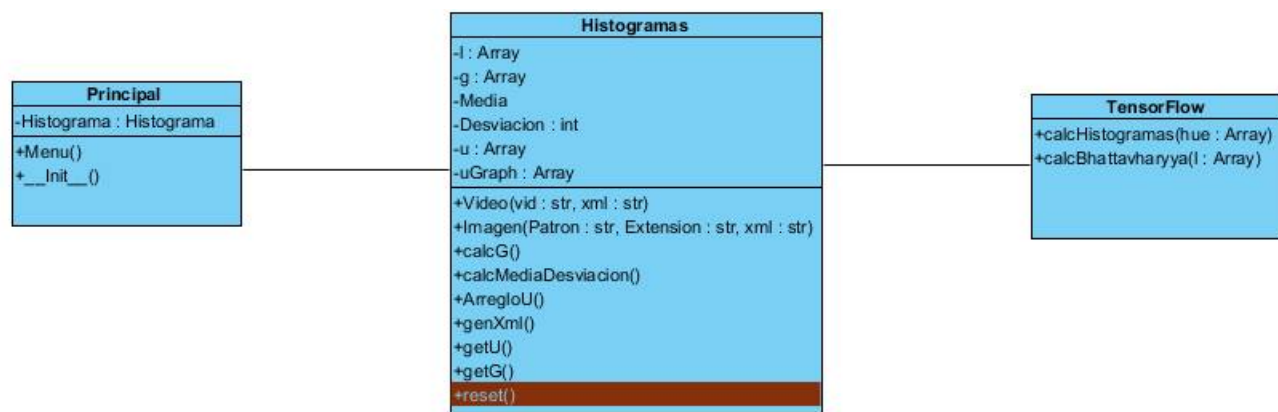


Figura 1: Diagrama de Clases

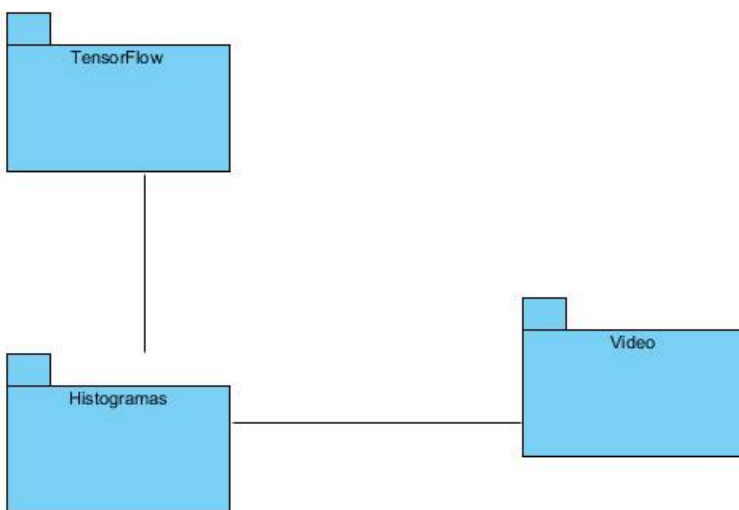


Figura 2: Diagrama de Paquetes

4. Pruebas

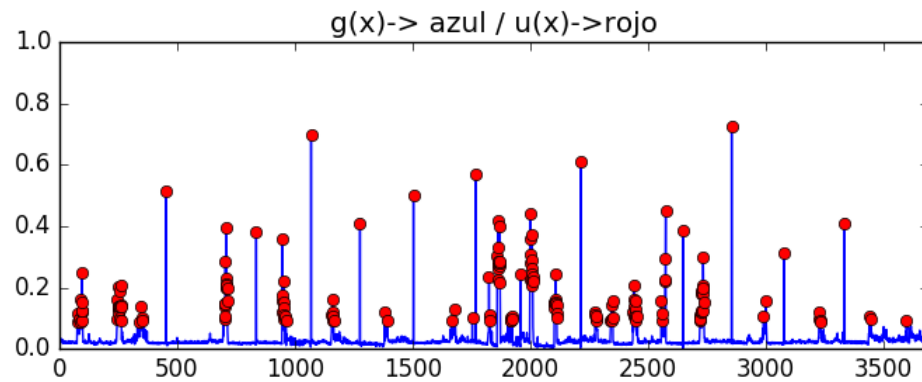
4.1. Modo de ejecutar el programa

```
[>>> execfile('Histograma-TF.py')
Tipo a revisar:
1. Video
2. Imagenes
Opcion Seleccionada: 1
Ruta al video: c.mp4
Archivo salida XML: c.xml
Analizar otra secuencia [y/n]: y
Tipo a revisar:
1. Video
2. Imagenes
Opcion Seleccionada: 2
Ruta la carpeta: c
Patron inicio (Ej: 'foto-'):
Extension: jpg
Archivo salida XML: c.xml
Analizar otra secuencia [y/n]: n
>>> █
```

1. Se debe ejecutar el archivo "Histograma-TF.py"
2. Se selecciona en que modo se quiere trabajar en modo video o imagenes.
3. Se debe seleccionar la ruta del video.
4. Se debe seleccionar el nombre del archivo .xml que se quiere generar.
5. Se puede volver a repetir los pasos anteriores o se puede salir.

4.2. Primera prueba España vs Holanda

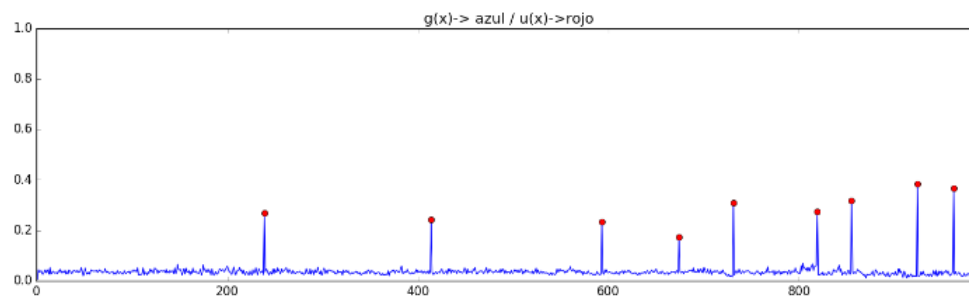
Nota: se adjuntan solo imagenes de los histogramas ya que los xml son relativamente largos, por este motivo se incluiran los mismos en el .zip.



El XML nos dice que existe un corte en los cuadros 2108-2109, lo cual podemos ver en la siguiente imagen:



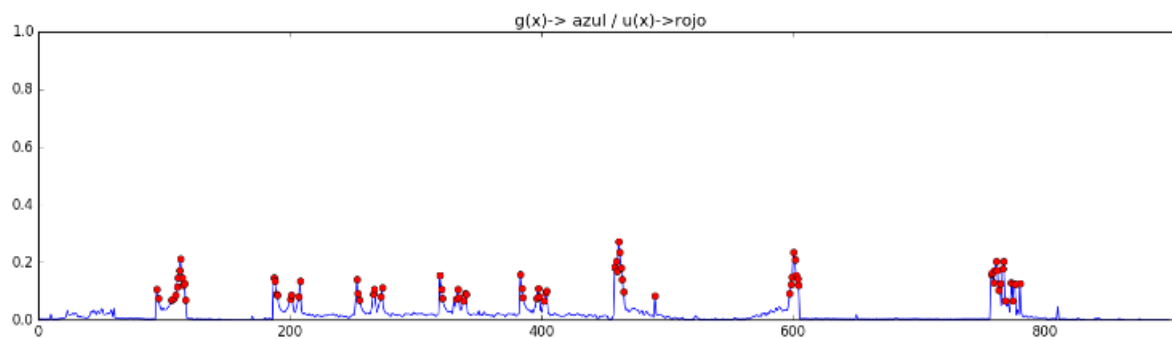
4.3. Segunda prueba Monja



El XML nos dice que existe un corte en los cuadros 855-856, lo cual podemos ver en la siguiente imagen:



4.4. Tercera prueba Educativo



El XML nos dice que existe un corte en los cuadros 597-598, lo cual podemos ver en la siguiente imagen:



5. Conclusiones

Durante la realización de este proyecto logramos apreciar el gran potencial de Python, el cual nos brindo muchas facilidades con las librerías que se utilizaron, en especial OpenCV que nos permitio manipular con facilidad el video. Otro detalle que vale la pena resaltar es la poca documentación que existe sobre tensorflow ya que es una libreria relativamente nueva.