

Resumen Automático de Vídeos mediante Clasificación

Iván de los Santos García
dpto. Ciencias de la
Computación e
Inteligencia Artificial
Universidad de Sevilla
Sevilla, España

ivadegar@alum.us.es

ivandega301095@gmail.com

El objetivo de este proyecto aplicar técnicas de inteligencia artificial para realizar un “clustering” o agrupación de imágenes para realizar el resumen de vídeos.

Los distintos segmentos del vídeo han sido seleccionados usando el algoritmo de K-medias. Las pruebas han sido realizadas con videos de youtube y VSUMM [1]

I. INTRODUCCIÓN

El resumen automático de vídeos (video summarization) está cobrando una gran importancia en el mundo multimedia y en Internet. Dado el gran volumen de material audiovisual que se crea a diario (desde películas y vídeos de vigilancia, a vídeos domésticos), se hace necesario técnicas que ayuden a su indexación, recuperación, procesado y archivado de los mismos.

En este artículo se propone aplicar una técnica en específico de machine learning, k-medias, para realizar el resumen de los videos haremos uso del valor del histograma de cada uno de los fotogramas del vídeo, esto nos ayudará a poder discernir entre las distintas posibles partes de nuestros videos. Para ellos habremos extraído los fotogramas, posteriormente el algoritmo ofrecerá una serie de tareas a realizar por parte de la persona que lo utilice para que pueda seleccionar tanto la carpeta de origen de los fotogramas como la de salida, así como el valor de K empleado en el algoritmo de K-medias.

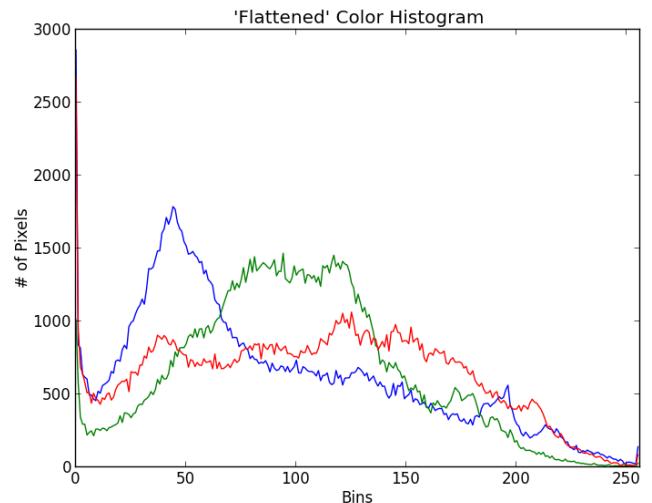


Fig. 1- Gráfico de histograma tal y como nos aparecería en nuestro proyecto

II. PRELIMINARES

A. Métodos empleados

Para la realización de nuestro Proyecto comenzaremos dividiendo el video de entrada en los distintos fotogramas a utilizar con algún software de edición de video, en mi caso ffmpeg [2].

Con un pequeño script podremos seleccionar la cantidad de fotogramas que deseamos con respecto al total de segundos del video.

Una vez obtenidos estos fotogramas realizaremos el cálculo del histograma correspondiente a los colores BGR (Blue, Green y Red).

A continuación usaremos el algoritmo de K-medias para agrupar en K clusteres los fotogramas. El valor de K, se explicará más adelante con mayor profundidad, será seleccionado por el método del codo.

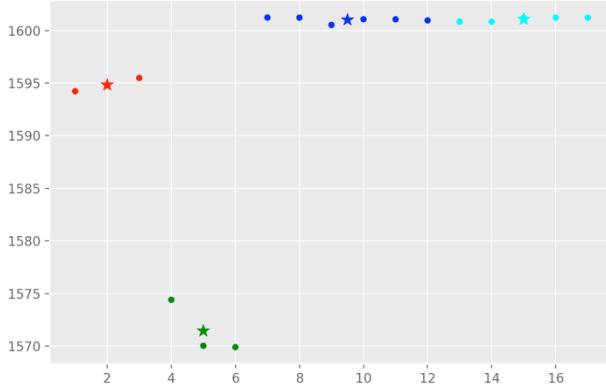


Figura 2 – Clusters o grupos encontrados por el algoritmo en un ejemplo

Una vez que estos grupos han sido seleccionados, se procederá a extraer las imágenes más cercanas esos centros para traspasarlas a la carpeta OUTPUT de salida. Habiendo finalizado así con la tarea de encontrar las partes características del vídeo para realizar el resumen.

El diagrama del proceso queda exemplificado en la figura 3.

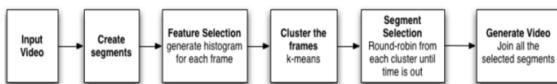


Figura 3 – Vista general

B. Trabajos relacionados

Existen técnicas más sofisticadas para la obtención del resumen de los vídeos, como la utilización de un grafo de vecinos cercanos (NGG, nearest neighbour graph en inglés), posteriormente se utiliza un grafo reverso de vecinos cercano (RNGG, reverse nearest neighbour graph) cuyas componentes fuertemente conexas darán lugar a los representantes de los futuros clústeres.

Existen además técnicas que hacen uso de redes neuronales LSTM (Long short-term memory) para la obtención de los fotogramas, siendo la salida $y_1 \dots y_n$, el grado de importancia de cada frame. [3]

III. METODOLOGÍA

A. Trabajo llevado a cabo

Para el éxito del la técnica utilizada, se ha comenzado por realizar un pequeño script que permitiera introducirlo en la carpeta de videos a usar. Este script al ejecutarlo nos preguntará por el nombre del vídeo a utilizar y posteriormente cada cuantos fotogramas por segundo deseamos obtener un fotograma para nuestro algoritmo. A menor cantidad de fotogramas menor cantidad deberá procesar nuestro algoritmo por lo tanto menor tiempo de ejecución, también teniendo en cuenta que se podría perder información sensible. Es por ello que en consideraciones se ha explicado con algo más de profundidad.

El proceso anterior nos creará una cantidad determinada de fotogramas, usando fffmpeg la estructura del nombre generado será “captura-x.png”, estos deberán ser copiados a la carpeta INPUTPATH (No es relevante el nombre seleccionado ya que el usuario podrá introducir el nombre de la carpeta que desee).

Nuestro algoritmo se apoyará de estos archivos para poder determinar el histograma de color de cada secuencia. El histograma de color es un tensor de 3 dimensiones, correspondientes a cada uno de los colores BGR, en nuestra implementación se ha realizado la media aritmética, o promedio, para cada imagen almacenada. Dando lugar entonces al vector unidimensional en el que cada hueco del vector se corresponde con su media BGR.

A este vector se le será aplicado el algoritmo de K-medias, los distintos centros serán seleccionados durante un máximo de 10 iteraciones (lo predeterminado por el algoritmo de sklearn). [4]

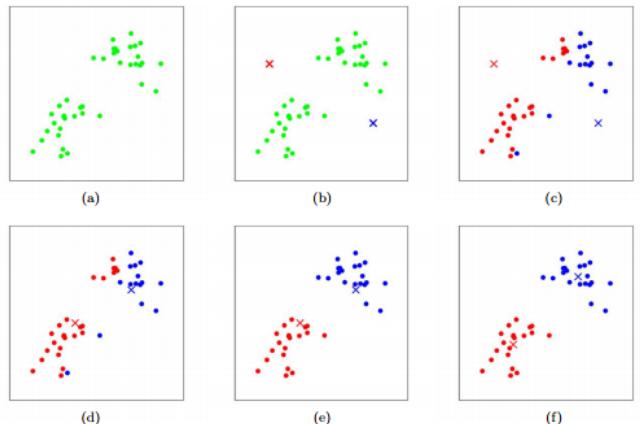


Figura 4 – Evolución del algoritmo

El algoritmo trabaja iterativamente para asignar a cada “punto” (las filas de nuestro conjunto de entrada forman una coordenada) uno de los “K” grupos basado en sus características. Son agrupados en base a la similitud de sus features (las columnas). Como resultado de ejecutar el algoritmo tendremos:

- Los “centrodes” de cada grupo que serán unas “coordenadas” de cada uno de los K conjuntos que se utilizarán para poder etiquetar nuevas muestras.
- Etiquetas para el conjunto de datos de entrenamiento. Cada etiqueta perteneciente a uno de los K grupos formados.

Los grupos se van definiendo de manera “orgánica”, es decir que se va ajustando su posición en cada iteración del proceso, hasta que converge el algoritmo. Una vez hallados los centrodes deberemos analizarlos para ver cuales son sus características únicas, frente a la de los otros grupos. Estos grupos son las etiquetas que genera el algoritmo. [5]

Por defecto el método de K-medias implementado por sklearn tiene una complejidad $O(nT)$, donde n es el número de ejemplos y T el número de iteraciones, siendo en general de complejidad lineal. [4]

La manera de seleccionar el valor de K ha sido por el método del punto de codo.

Este algoritmo funciona pre-seleccionando un valor de K. Para encontrar el número de clusters en los datos, deberemos ejecutar el algoritmo para un rango de valores K, ver los resultados y comparar características de los grupos obtenidos. En general no hay un modo exacto de determinar el valor K, pero se puede estimar con aceptable precisión siguiendo la siguiente técnica:

Una de las métricas usada para comparar resultados es la distancia media entre los puntos de datos y su centroide. Como el valor de la media disminuirá a medida de aumentemos el valor de K, deberemos utilizar la distancia media al centroide en función de K y encontrar el “punto codo”, donde la tasa de descenso se “afila”.

Aquí vemos apreciar una gráfica obtenida al ejecutar nuestro programa:

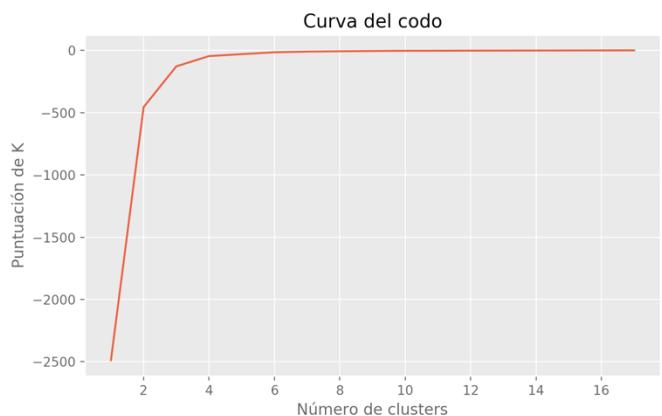


Figura 5 - El valor a seleccionar como K serán los valores entre los que la curva se genere, en este caso 2, 3 o 4.

Como podemos observar este método hace que de forma visual una persona tenga que establecer un valor aproximado para K en cada vídeo que se quiera resumir, es decir K no es escogido de manera autónoma, esto también hace que el valor escogido de K pueda ser más aproximado a su mejor valor posible.

Para realizar una valoración visual por el método del punto de codo, simplemente deberemos seleccionar un valor que se encuentre en su zona de curvamiento.

He observado realizando unas búsquedas por medio de internet que no existen formas de seleccionar el valor de K de esta manera, quizás podría ser interesante el aplicar algún método matemático si se puede obtener la curva generada por el método del punto de codo para obtener un cambio en la tangente de la función asociada a los valores y que se pueda realizar de manera autónoma por este método la selección de K.

El siguiente pseudocódigo ha sido implementado para la selección del número de puntos de codo a realizar, ya que hay casos en videos cortos (10-30 segundos) que dependiendo de los valores utilizados a la hora de seleccionar los fotogramas pueden afectar de manera que no se pueda ejecutar el algoritmo si el número de valores de puntos de codo es mayor al número de fotogramas.

$$\begin{cases} 20 & \text{número de fotogramas} > 20 \\ 10 & \text{número de fotogramas} > 10 \quad \& \quad \text{número de fotogramas} \leq 20 \\ \# & \text{e.o.c.} \end{cases}$$

Figura 6 – Decisión del número de veces a aplicar el método del punto de codo

Significando que no existe que el usuario debe probar con una mayor cantidad de fotogramas o que el video es extremadamente corto.

B. Consideraciones

- Para la selección de K hemos usado el punto de codo, podrían existir otros métodos que han sido propuestos pero no implementados como por ejemplo realizar la selección mediante alguna heurística sobre el video de entrada (longitud, tipo, mixto, etc...).
- Se ha realizado la media de colores BGR para cada fotografía en lugar de trabajar con sus colores en particular para cada canal de color, esto ha reducido la cantidad de calculo necesarios para obtener los centros pero a su vez ha hecho que al existir menos información sobre cada canal de color se hace más difícil que sea distinguir si la realización ha sido exitosa ya que se da el caso de que aunque dos fotogramas tengan colores distintos puedan acabar teniendo el mismo valor medio para el histograma. En los casos de muestra realizados he observado que esto hace también que si el video está enfocado en una temática y tiene unos valores de histograma similares sea difícil obtener partes clave ya que suelen tener un valor medio similar.

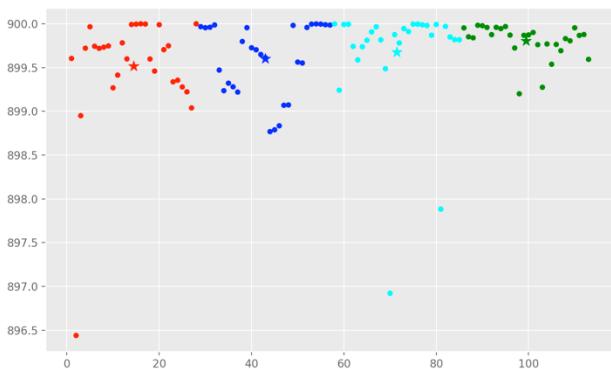


Figura 7 – Videos en los que el tipo de color es bastante similar, aun así es capaz de crear buenas imágenes para el resumen de los videos.

C. Posibles mejoras

- El script para la obtención de fotogramas de los videos podría almacenar las imágenes en la carpeta de INPUTPATH automáticamente. Se

ha considerado pero no se ha terminado llevando a cabo por falta de tiempo aunque no sería una mejora difícil de realizar con el script ya realizado copiando las imágenes obtenidas a la carpeta INPUTPATH.

- Una de las posibles mejoras para el programa es aplicar Emplear el algoritmo KNN para determinar los fotogramas que aparecen en el video resumido, además de los fotogramas clave. La idea es que el resumen del vídeo no sea una secuencia de saltos entre fotogramas, sino que sea fluido eligiendo fotogramas colindantes a los clave. Para cada fotograma clave k, se eligen S fotogramas antes y después, y se eligen aquellos que se clasifiquen igual que el fotograma k mediante KNN. Una forma similar de realizar estos pequeños resúmenes sin necesidad de aplicar KNN, por la manera en la que esta implementado el programa en este artículo, sería interesante simplemente seleccionar un número T fijo en los que además de las capturas que han sido seleccionada como centroides se seleccionan también T imágenes posteriores y anteriores para así producir resúmenes de vídeos sin cortes y con mayor eficiencia que aplicar un segundo algoritmo.
- En casos en los que el video sea especialmente largo y se obtengan una cantidad de fotogramas superior a 10.000 ejemplos, podría ser curioso aplicar un método de K-medias en batch o en grupos (llamado en sklearn minibatchKmeans), así se reduciría el tiempo de ejecución del algoritmo. [6]

IV. RESULTADOS

Los resultados del algoritmo han resultado muy buenos con distintos tipos de vídeos, en general se ha conseguido obtener unas imágenes de resumen adecuadas.

En figuras posteriores se han mostrado distintas imágenes para los videos de ejemplo utilizados.

A. Vídeo de prueba 1

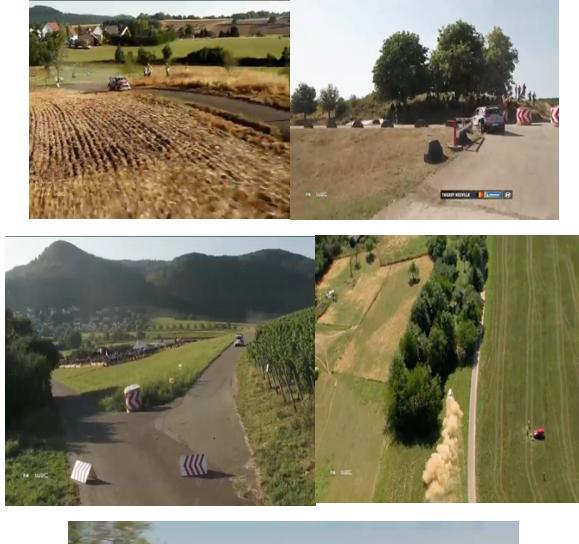


Figura 8 – carrera del World Rally Championship [7]

En esta caso se trataba de un vídeo de 2 minutos en los que solo se toman escenas fuera del vehículo, en general el algoritmo se ejecuta rápidamente.

B. Vídeo de prueba 2

En el siguiente caso se ha usado un video corto [8] de aproximadamente 30 segundos que consta de varias imágenes en distintos momentos del día que ayudará a decidir si el algoritmo está correctamente funcionando y encuentra las distintas escenas que componen el vídeo.

En este caso la K óptima se encuentra entre 4 o 5 siendo el número total de escenas del vídeo de 6. Algunas de las escenas presentan colores similares siendo normal este resultado.



Figura 9 – Escenas de la naturaleza

C. Vídeo de prueba 3

En el siguiente video [9] se ha aplicado el algoritmo a un vídeo de conducción en pista en primera persona para compararlo en cierta medida con los resultados obtenidos en la carrera de rally fuera de la vista del piloto.

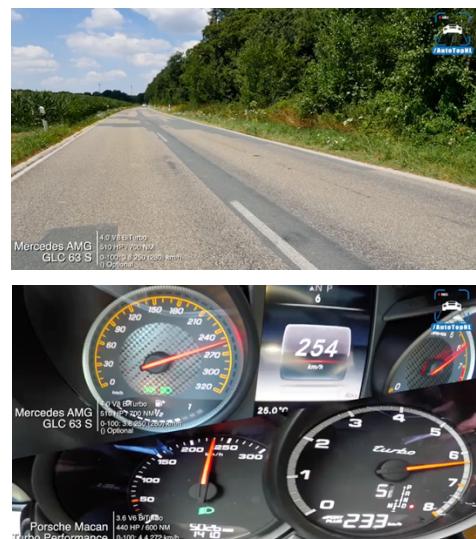




Figura 10 – Conducción en primera persona

Se puede observar que los resultados son bastante buenos ya que el video trata de una prueba en primera persona con varias escenas al exterior entre las que se podría destacar esta imagen por su intensidad en color rojo, la cual no ha podido ser clasificada como imagen para el video resumen:



Figura 11 – Escena con alta saturación no seleccionada

Por otro lado es comprensible que no haya sido seleccionada dado que es posible que un centro se encontrase más próximo a otro fotograma no tan característico dado la mayor concentración de fotogramas similares (interior del vehículo) y por tanto su mayor proximidad a ellos por el centroide.

D. Video de prueba 4

En este video de prueba, el cual ha sido el mas extenso, se ha usado un vídeo de 17 minutos el cual ha generado 1060 fotogramas. La velocidad del algoritmo es buena, lo que tarda mayor tiempo es la realización del cálculo del histograma de cada uno de los fotogramas.



Figura 12 – Imágenes de la cocina [10]

El resultado en este caso ha sido altamente satisfactorio ya que no se han obtenido multiples imágenes repetidas y se ha podido extraer partes significativas del vídeo, este vídeo poseía alrededor de 6 a 7 partes significativas.

TABLA 1. TIPO DE VIDEOS Y CARACTERÍSTICAS

Tipo Vídeo	Características		
	Duración del algoritmo	Valor K	Tiempo de duración
WRC	00:01.9	4	1:55
Naturaleza	00:00.3	4	0:18
Coche	00:01.5	6	4:09
Cocina	00:01.8	5	17:39

V. CONCLUSIONES

Finalmente, en esa sección se realizarán unas breves conclusiones extraídas al realizar este trabajo. Esta ha sido la primera vez trabajando con técnicas de machine learning (K-medias), con Python durante mi el grado universitario y las librerías usadas aportadas como referencias. La práctica 6 de Inteligencia Artificial de la Universidad de Sevilla [11] realizada en el laboratorio nos ayudó a entender un poco mejor el trabajo realizado por el algoritmo para un problema de clasificación con KNN lo que hizo posible entender para la realización de este trabajo el uso de *pandas*, *numpy* y *sklearn*, facilitando con ellos el correcto aprendizaje.

Además este proyecto ofrece un primer paso hacia el uso de algoritmos de machine learning para una implementación en un entorno profesional y con una fiabilidad aceptable dentro de los recursos consumidos para su uso.

Como hemos podido comprobar, al realizar este trabajo, el algoritmo de K-medias ha sido capaz de realizar un resumen de videos de una forma eficiente para múltiples tipos de longitud.

Sin embargo, un humano podría realizar mejores sesgos a la hora de seleccionar partes relevantes en un vídeo de forma más eficiente tratándose de un algoritmo adecuado a la hora de realizar resúmenes de vídeos “en gran cantidad” sin importar realmente la fidelidad de estos resúmenes sino más bien que puedan ser comprensibles las distintas partes del mismo.

VI. REFERENCIAS

- [1] <https://sites.google.com/site/vsummsite/home>
- [2] https://programacion.net/articulo/obtener_pantallazos_de_videos_con_ffmpeg_1201
- [3] http://home.iitk.ac.in/~kanishkg/Video_Summarization_Final_Report.pdf
- [4] <http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>
- [5] <http://www.aprendemachinelearning.com/k-means-en-python-paso-a-paso/>
- [6] <http://scikit-learn.org/stable/modules/generated/sklearn.cluster.MiniBatchKMeans.html> - [sklearn.cluster.MiniBatchKMeans](http://scikit-learn.org/stable/modules/generated/sklearn.cluster.MiniBatchKMeans.html)
- [7] <https://www.youtube.com/watch?v=swk6NGwTrpU>
- [8] <https://www.youtube.com/watch?v=668nUCeBHyY>
- [9] <https://www.youtube.com/watch?v=q58QzKMFWAo>
- [10] https://www.youtube.com/watch?v=wYGe_Ys6XBY
- [11] <https://www.cs.us.es/cursos/iais-2017/?contenido=practicas.php>