SUMUPVIDEO

Resumen Automático de Vídeos mediante Clasificación

Alfonso Alarcón Tamayo
dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla
Sevilla, España
alfonsoalarcontamayo27@gmail.com
,alfalatam

Juan Manuel de la Oliva Aguilar
dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla
Sevilla, España
juandelaoliva1@gmail.com , juaoliagu

***Resumen*:**

Este trabajo trata sobre el resumen automático de videos mediante clasificación, tendremos que, desde un vídeo obtener un resumen con la información relevante que ocurre, el cual, lo obtendremos a partir de sus fotogramas. A partir de estos, seleccionar los fotogramas clave (con contenidos relevantes del vídeo) y una vez los hemos filtrado volver a unir estos fotogramas para generar un vídeo resumen, el objetivo de este trabajo sobre inteligencia artificial es saber emplear los métodos de aprendizaje automático no supervisado mediante el uso del algoritmo de K-medias.

Como objetivo adicional del trabajo deberemos utilizar además del método de K-medias, el método KNN (K-nearest neighbors), para dotar así al vídeo resumen de una determinada fluidez a partir de obtener los fotogramas adyacentes a los seleccionados.

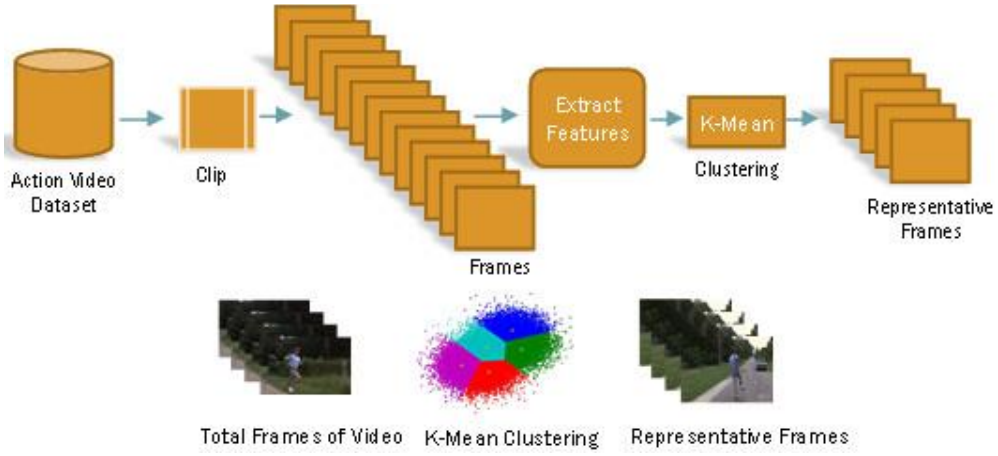
A modo de conclusión podemos mencionar que, a partir de la investigación realizada, hemos aprendido la necesidad de utilizar estas metodologías debido a los grandes volúmenes de información que se tratan día a día.

I. INTRODUCCIÓN

Los volúmenes de información que se trata en el mundo aumentan sustancialmente día a día, la necesidad de filtrar la información relevante de las montañas de datos es crucial para poder tomar decisiones correctas y en un tiempo razonable, por ejemplo, una empresa de seguridad no puede visualizar todas las horas de vídeo que realizan cada

día, lo que les importa son los eventos relevantes que ocurren en esas horas (si hay un terremoto, o entra un ladrón en una casa).

Para suplir estas necesidades se emplearán metodologías de aprendizaje automático para extraer información relevante, en nuestro caso usaremos principalmente K-means.



Video summarization using K-means

El análisis de vídeo consiste en extraer las características en "cuadros" de fotogramas clave. Estas características codifican la escena. múltiples escenas conforman una secuencia temporal de eventos, en la porción de vídeo que hemos tratado.

El problema al que el resumen automático de videos responde es la necesidad de tratar grandes volúmenes de información en poco tiempo, ya que el número de eventos que ocurren y se trasmiten a nivel mundial es casi ilimitado *[1]*, esta práctica es muy necesitada en todo tipo de campos (empresas de seguridad privadas, policial, fútbol *[2]* (para obtener sucesos destacados en momentos determinados…)

[4]Resumen de vídeos en fútbol

En el trabajo nos hemos enfocado en mantener la mayor cantidad de información relevante al sacar los fotogramas clase obteniendo un vídeo que de imágenes secuenciales que nos transmiten los sucesos que ocurren relevante en este, además a partir del *ANEXO II***,** Utilizando metodologías de K vecinos hemos sacado una secuencia de fotogramas vecinos a los claves, para así dotar al vídeo de una fluidez en los fotogramas clave *(key frames)* (Véase *[3]* y *[4]*).

II. MÉTODOS EMPLEADOS

El aprendizaje automático es un subapartado de la inteligencia artificial cuyo objetivo es desarrollar técnicas que permitan a las computadoras *"aprender"*. Se trata de crear programas capaces de automatizar comportamientos a partir de la información suministrada. Es, por lo tanto, un proceso de inducción del conocimiento. Es una disciplina que se basa en el análisis de datos. Además de en el estudio de la complejidad computacional de los problemas.

Utilizaremos tanto aprendizaje no supervisado*(K-Means)* como aprendizaje supervisado *(KNN)* en este trabajo *[5]*.

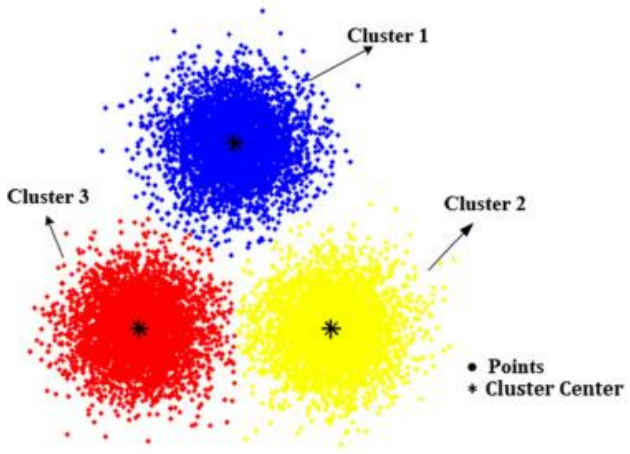
- K-Medias
- KNN

K-MEDIAS *[6]*:

El algoritmo de las K-medias es uno de los algoritmos de aprendizaje no supervisado utilizados para resolver problemas de *clusterización*. El procedimiento aproxima por etapas sucesivas un cierto número (prefijado) de clusters haciendo uso de los centroides de los puntos que deben representar, como norma general se siguen los siguientes pasos:

1. Sitúa KK puntos en el espacio en el que "viven" los objetos que se quieren clasificar. Estos puntos representan los centroides iniciales de los grupos.
2. Asigna cada objeto al grupo que tiene el centroide más cercano.
3. Tras haber asignado todos los objetos, recalcula las posiciones de los KK centroides.
4. Repite los pasos 2 y 3 hasta que los centroides se mantengan estables. Esto produce una clasificación de los objetos en grupos que permite dar una métrica entre ellos.

Aunque se puede probar que este algoritmo siempre termina, no siempre la distribución que se alcanza es la óptima, ya que es muy sensible a las condiciones iniciales. *[7]*

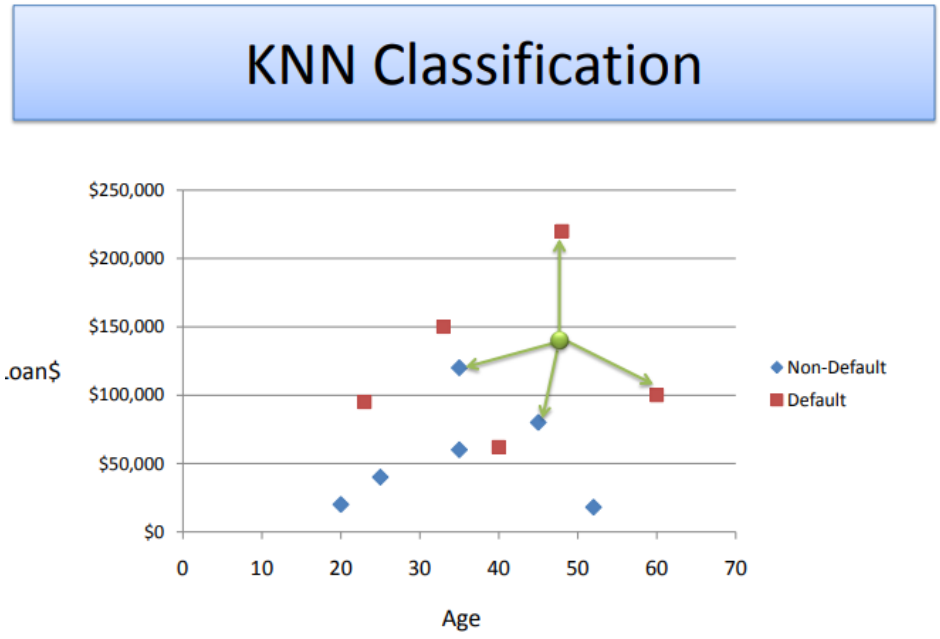


Representación de K-means

KNN:

KNN es un algoritmo simple que almacena todos los casos disponibles y clasifica los nuevos casos basados en una medida de similitud.

A través del cálculo de las distancias a "X" puntos (que en nuestro caso van a ser los fotogramas clave que escojamos), obtenemos las distancias a los centros, escogeremos aquellos puntos(frames), que más se parezcan al núcleo, que normalmente serán aquellos fotogramas adyacentes al clave *[8].*



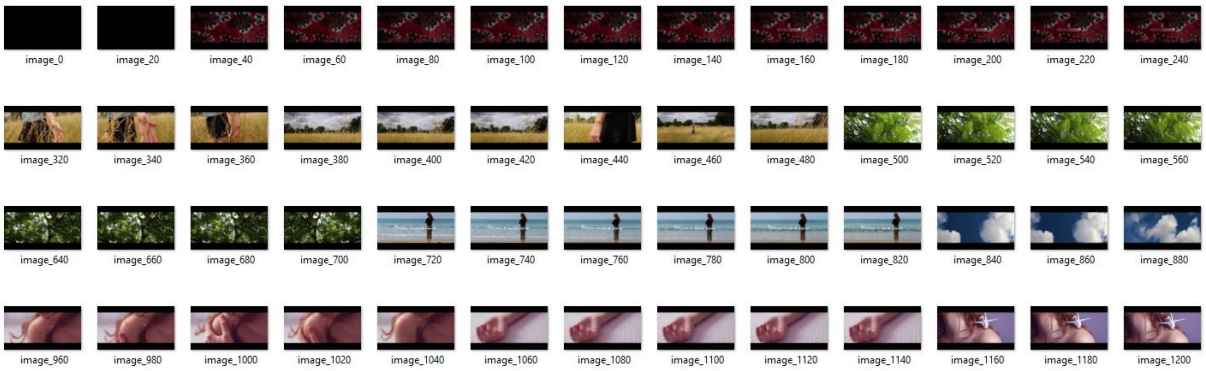
Ejemplo de clasificación por KNN [9]

III. METODOLOGÍA

Para la realización del resumen automático de vídeos el programa pedirá por pantalla el nombre del vídeo sobre el que se quiere aplicar el mismo, el cual habrá que introducirlo con su respectivo formato.

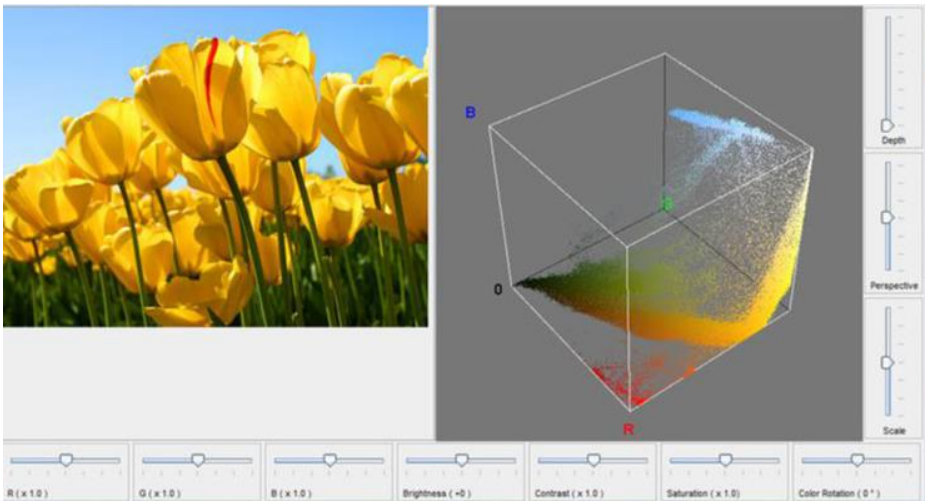
El programa manipula el vídeo para extraer la secuencia de fotogramas que lo componen, escogiendo una frecuencia de *sampleo* de una captura por cada 20 fotogramas, que es una media de 75 fotogramas por minuto de vídeo, para así optimizar el tiempo de procesamiento del programa.

Dichos fotogramas se guardarán en una ruta determinada y a su vez en una lista.



Fotogramas extraídos

Mediante la biblioteca de *OpenCV [11]* se calculan los histogramas de todos los fotogramas antes extraídos. Hay que tener en cuenta que por cada imagen o fotograma obtendremos 3 histogramas, un por cada color, rojo, verde y azul *("r", "g", "b").* Tendríamos una lista que albergaría por cada fotograma una lista tridimensional en la que se encontrarían los valores de dichos colores.



RGB Histograma [10]

Para la posterior aplicación del algoritmo de agrupamiento *Kmeans* (importado de la librería *sklearn*), es necesario calcular el número de centros o grupos que se va a escoger, y para ello se ha aplicado una heurística basada en una mezcla de la longitud de *frames* y el *frameRate* o fotogramas por segundo. Dividiendo el número de fotogramas por el valor de los fotogramas por segundo obtendríamos la duración del vídeo en segundos, los cuales a su vez los dividimos por un valor k escogido, en concreto, K=30 para sumar dos centros por cada minuto de vídeo, y al resultado final se le suman N centros más para abarcar el caso de que el vídeo a resumir fuese de una longitud muy pequeña.

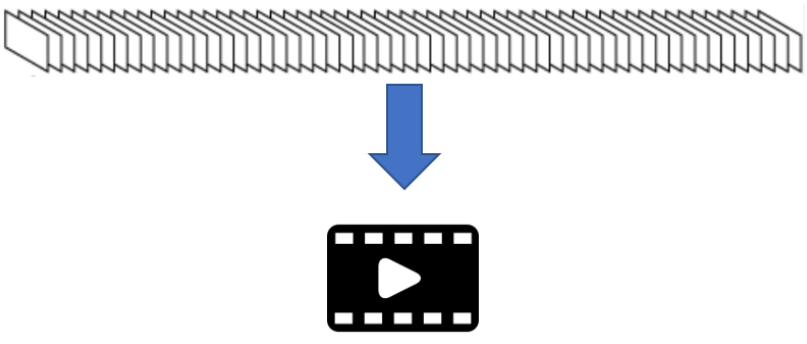
```
 Número_centros = ((nFrames / framesPorSegundo) / K) + N
K: Segundos por centro
N: número de clusters añadidos
```

Una vez conseguido los histogramas de todos los fotogramas y calculado el número de centros, aplicamos el algoritmo *Kmeans* (anteriormente descrito) y obtendremos una matriz en la que se indican en columnas las distancias de cada punto a los centros seleccionados, de tal manera, extrayendo el menor de los elementos, es decir, el que menos distancia tenga con el centro por cada columna estaremos cogiendo el fotograma más representativo de esa agrupación.

Se irán guardando dichos fotogramas clave en una lista.

Al tener la lista con los fotogramas clave el siguiente paso será extraer del vídeo original un número F de fotogramas adyacentes, tanto anteriores como posteriores para dar una cierta continuidad al vídeo final.

Finalmente se recorrerá el conjunto de fotogramas clave y sus adyacentes para escribirlos en un vídeo mediante OpenCV.



Frames to video

IV. RESULTADOS

Para la correcta realización del proyecto una parte fundamental ha sido la experimentación, tanto como para en base a ellos poder mejorar el código (sirva como ejemplo el cálculo de centros), como para poder visualizar los resultados finales obtenidos.

Los parámetros relevantes para la experimentación han sido los siguientes:

- K: o número de centros empleado para el algoritmo K-medias. El cual ha sido calculado mediante la heurística comentada en la metodología. Esta tiene en cuenta la longitud del vídeo en segundos como parámetro principal pero también tiene en cuenta el caso especial de que el vídeo a resumir sea demasiado corto por lo que le suma 'N' (véase pseudocódigo en el apartado III)
- T: número de fotogramas a saltar en el recorrido de la secuencia de entrada para calcular los fotogramas clave. Este parámetro está fijado por nuestro script a una captura cada 20 fotogramas, pero solo para la aplicación del algoritmo *K-means* ya que para el *kNN* se utilizarán los fotogramas estrictamente consecutivos.
- H: tamaño del histograma generado para cada canal de color de cada fotograma. Aquí se encuentra una de las peculiaridades de nuestro trabajo, ya que, gracias a estudios previos de asignaturas como Procesamiento de Señales Multimedia, hemos podido comprobar que la representación de colores puede ser significativa y de un buen resultado si se reducen los valores de 256 a 16, simplificando así el tamaño del histograma y a su vez la complejidad y tiempo de ejecución.

A continuación, se mostrarán una serie de experimentos realizados:

Experimento I:

En este experimento el código de nuestro proyecto nos devolvía un vídeo resumen compuesto solo por los fotogramas clave devueltos por el algoritmo de *Kmeans.*

	Original	Resultado
Vid1 "Bunny"	https://drive.google.com/open?id=1XD7pg5pxzlO3UmmDDSPdPPR8LUTSTl5v	https://drive.google.com/open?id=1dk1QBAZ_kZ39fhUZAIY62QDKsSyL7-jN
Vid2 "Simpsons"	https://drive.google.com/open?id=1kJ5CgTqrlPMc_SkW2toNbMsOkM7hZtXS	https://drive.google.com/open?id=19cCp7SteCgC6Bs9WG5DmCAW5QfXVzUMO
Vid3 "El Cuerpo"	https://drive.google.com/open?id=12ZkezqDqK6qXPHIqq1gJzBj3EFKd-pmf	https://drive.google.com/open?id=1oLNankdadqVqmuEcnwWyALnW3ktPeXkr

Experimento II:

La siguiente iteración significativa del proyecto fue aplicar el algoritmo *kNN* a los fotogramas clave buscando una serie de vecinos entre los fotogramas capturados al principio como muestras significativas del vídeo. Una vez encontrados dichos vecinos, se unían al vídeo dotando al mismo de cierto dinamismo, pero con muy poca fluidez:

	Original	Resultado
Vid1 "Bunny"	https://drive.google.com/open?id=1XD7pg5pxzlO3UmmDDSPdPPR8LUTSTl5v	https://drive.google.com/open?id=18pFdGeq8kP-O9Rkg_X-ykARrAxVkSrjL
Vid2 "Simpsons"	https://drive.google.com/open?id=1kJ5CgTqrlPMc_SkW2toNbMsOkM7hZtXS	https://drive.google.com/open?id=1AYD2fSlJAnR1FaLfvxYkjRO3N0qCmz6l
Vid3 "El Cuerpo"	https://drive.google.com/open?id=12ZkezqDqK6qXPHIqq1gJzBj3EFKd-pmf	https://drive.google.com/open?id=1y18uFTyf8oL1eoNptPhrmNftnxj5UxMR

Experimento III:

Para darle a los videos más fluidez se decidió aplicar al anterior algoritmo de *kNN*, pero esta vez sobre los fotogramas del vídeo original, teniendo que capturarlos de nuevo. Para ello también se igualó el número de fotogramas por segundo al mismo valor que el del original a la hora de escribir el vídeo. Finalmente hemos experimentado con videos de mayor tamaño llegando al último experimento con una película de 2 horas.

	Original	Resultado
Vid1 "Bunny"	https://drive.google.com/open?id=1XD7pg5pxzlO3UmmDDSPdPPR8LUTSTl5v	https://drive.google.com/open?id=15kL23fFvkLviwhCQXZsRMFoSwh6kO6X2
Vid2 "Simpsons"	https://drive.google.com/open?id=1kJ5CgTqrlPMc_SkW2toNbMsOkM7hZtXS	https://drive.google.com/open?id=1PQv-zZ-3jLoJYgRQrV0cn5lOAwFd03br
Vid3 "El Cuerpo"	https://drive.google.com/open?id=12ZkezqDqK6qXPHIqq1gJzBj3EFKd-pmf	https://drive.google.com/open?id=14aPibHpYSO2hxYICq5Z5N_Z3ovZZpACd
Vid4 "simp"	https://drive.google.com/open?id=1PvAWJzsbSVbOp9v90B52sfHU_SEI8D19	https://drive.google.com/open?id=10FrcqkdOQUEeI_INntyX4nBBSwRYQYyC
Vid5 "futurama"	https://drive.google.com/open?id=13XGlU5BakaF57JpFPkyJVaZqGKnWgM5M	https://drive.google.com/open?id=1uiwS1wprHrL0cgq5uw_mjmR0NL7Hg-r3
Vid6 "StarWars4"	https://drive.google.com/open?id=1RMCIQavK9DUTAPNLUBfEUeL8Avmbhqa_	https://drive.google.com/open?id=16hqeuV-A3m139L6Ltvc2FIq6muAYAnch

V. CONCLUSIONES

En resumen, nuestro trabajo ha consistido a partir de aprendizaje automático obtener vídeo resúmenes a partir de sus videos originales con la extracción de sus fotogramas claves *(K-means),* y sus respectivos "vecinos" para dotar al vídeo-resumen de una fluidez que le proporcionase calidad al ser reproducido *(KNN).*

Gracias a los experimentos hemos podido ir depurando el código y la estructura de la información hasta obtener la configuración óptima para el objetivo inicial.

Una de las conclusiones principales del trabajo es la importancia de dotar a las funciones o algoritmos de unas entradas y datos que tengan sentido frente al objetivo que se quiere alcanzar. Podemos ejemplificarlo con el experimento 2 al que le pasábamos al algoritmo KNN el conjunto de frames inicialmente *sampleados,* que no daba el mejor resultado posible ya que salían una serie de fotogramas que secuenciaban la imagen, pero con ciertas "pausas, por lo que optamos por sacar los k vecinos directamente del vídeo, arreglando así el problema de la fluidez.

Una posible futura mejora podría ser crear una interfaz gráfica de usuario para así facilitar el uso del programa.

Nos ha parecido un problema muy interesante para resolver (causa por la cual hemos tenido un alto nivel de involucración).

VI. BIBLIOGRAFÍA

[1] http://www.everysecond.io/youtube

[2] https://www.slideshare.net/dhan1989/goal-recognition-in-soccer-match

[3] https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4338347

[4]https://pdfs.semanticscholar.org/5c21/6db7892fa3f515d816f84893bfab1137f0b2.pdf

[5] http://www.cs.us.es/~fsancho/?e=77

[6] https://www.datascience.com/blog/k-means-clustering

[7] http://www.cs.us.es/~fsancho/?e=43

[8]https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/

[9] http://chem-eng.utoronto.ca/~datamining/Presentations/KNN.pdf

[10] https://www.mathworks.com/matlabcentral/answers/216064-3d-histogram-of-rgb-image

[11]https://docs.opencv.org/2.4/modules/highgui/doc/reading_and_writing_images_and_video.html?

https://docs.scipy.org/doc/numpy/reference/index.html

https://stackoverflow.com/