

Resumen de videos, algo no tan automático

Ángel Manuel Calzado Llamas
dpto. Ciencias de la Computación e Inteligencia
Artificial
Universidad de Sevilla
Sevilla, España
angcalla@alum.us.es
angelcalzadolamas@gmail.com

Santiago Borge Sánchez
dpto. Ciencias de la Computación e Inteligencia
Artificial
Universidad de Sevilla
Sevilla, España
sanborsan1@alum.us.es
santiaguito975@gmail.com

Resumen— El resumen automático de vídeos mediante clasificación, técnica basada en la extracción de fotogramas de un vídeo, agrupación de estos en grupos y elección de un representante de cada uno es, en el ámbito de la Inteligencia Artificial y más concretamente en el aprendizaje automático, el tema que va a abordar esta investigación.

Es un medio útil, fácil pero no preciso, los resultados varían entre una muy rápida ejecución y una calidad baja, y entre una ejecución algo más lenta y calidad alta. Con el añadido de escoger unos buenos datos de entrada.

Palabras clave— Inteligencia Artificial, Algoritmo, Aprendizaje automático, fotogramas clave o key frames, K-Medias o K-Means, Clustering, Cluster,

I. INTRODUCCIÓN

El resumen automático de vídeos, es una técnica para facilitar procesos como los de indexación, recuperación, procesado y archivado de un vídeo. Está cobrando gran importancia en el mundo multimedia e internet debido a la carga masiva de datos y los ya existentes que se encuentran en la plataforma[1].

Es un método para reducir la redundancia y generar una representación sucinta de la información de un vídeo; es decir, que contenga sólo lo relevante. Dicho resumen debe cumplir tres principios: debe contener entidades y eventos de alta prioridad, debe exhibir un grado razonable de continuidad, y debe estar libre de repeticiones. Uno de los métodos más comunes está basado en la extracción de fotogramas clave (key frames), los cuales representan los contenidos más importantes de un vídeo. Una vez se tiene una secuencia de fotogramas clave, se puede generar un vídeo resumen con dichos fotogramas[1].

La problemática consistirá en realizar un sistema que permita la ejecución paso a paso de lo descrito anteriormente. La pregunta aquí es: ¿Cómo vamos a elegir los fotogramas más importantes de un vídeo? ¿En qué datos vamos a basarnos?

Pues bien, el análisis de los fotogramas será llevado a cabo con sus histogramas de color RGB, descrito sencillamente, se trata de elegir aquellos fotogramas que representen un mayor cambio en sus representaciones de color. Para determinar que fotogramas vamos a escoger para resumir nuestro vídeo haremos uso del algoritmo de K-Medias (K-Means), es un método de agrupamiento, que tiene como objetivo la partición de un conjunto de n observaciones en k grupos en el que cada observación pertenece al grupo cuyo valor medio es más cercano[2].

La estructura del documento la dividiremos en: Preliminares, donde explicaremos la técnica usada; Metodología, el método empleado para el trabajo y donde se encuentra el trabajo realizado; Resultados, comentamos los valores obtenidos y analizamos su experimentación; Conclusiones.

II. PRELIMINARES

Ya hemos visto que debemos de encontrar una manera de seleccionar aquellos fotogramas más relevantes para añadir a nuestro vídeo resumen.

Vamos a emplear una técnica de aprendizaje automático de la rama de Clustering. Se trata de dividir un conjunto de datos de entrada en subconjuntos (clusters), de tal manera que los elementos de cada subconjunto compartan cierto patrón o características a priori desconocidas. Se trata de un tipo de aprendizaje no supervisado: no tenemos información sobre qué cluster corresponde a cada dato.

El clustering basado en distancia sigue la idea de dado el número k de grupos o clusters, buscar k puntos o centros representantes de cada cluster, de manera que cada dato se considera en el cluster correspondiente al centro que tiene a menor «distancia». De lo explicado es de lo que va a tratar el algoritmo empleado en este proyecto y del que ya hablamos en el apartado 1 de este documento llamado K-medias.

K medias[3]

► Entrada: un número k de clusters, un conjunto de datos $\{x_i\}$ N $i=1$ y una función de distancia
► Salida: un conjunto de k centros m_1, \dots, m_k

k-medias(k,datos,distancia):

1. Inicializar m_i ($i=1, \dots, k$) (aleatoriamente o con algún criterio heurístico)

2. REPETIR (hasta que los m_i no cambien):

2.1 PARA $j=1, \dots, N$, HACER:

Calcular el cluster correspondiente a x_j , escogiendo, de entre todos los m_i , el m_h tal que $\text{distancia}(x_j, m_h)$ sea mínima

2.2 PARA $i=1, \dots, k$ HACER:

Asignar a m_i la media aritmética de los datos asignados al cluster i -ésimo

3. Devolver m_1, \dots, m_k

Semejanza con los datos del problema:

► Entrada: un número de k clusters, un conjunto de fotogramas y una función distancia

► Salida: un conjunto de k fotogramas que resume el vídeo de entrada

III. METODOLOGÍA

Ahora pasamos a describir el método implementado en el trabajo, para ello hemos usado pseudocódigo de cada función, aunque también están descritas y explicadas en el código mediante comentarios y en el fichero README.txt. El pseudocódigo que aparece a continuación es el de todas las funciones desarrolladas en el trabajo excepto las funciones de distancias [4] las cuales se basan en la misma idea y queda más claro explicar a continuación las fórmulas que se emplean en cada distancia.

Distancia Euclídea: $de(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$

Distancia Manhattan: $dm(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n |x_i - y_i|$

Distancia Hamming: número de componentes en las que se difiere

PRINCIPAL: Dada una secuencia de N fotogramas en la carpeta INPUTPATH, una carpeta de salida OUTPUTPATH, los parámetros K , T , H , una técnica y el tipo de distancia.

- 1) ListaKeyFrames, ListaFramesBuenos \leftarrow *CalcularFotogramasClave*(INPUTPATH, T , K , H , técnica, tipoDistancia)
- 2) ListaImágenes \leftarrow Lista vacía
- 3) Para cada fotograma en INPUTPATH, índice=0, índice< N , índice+= T
 - a) Añadir a ListaImágenes
- 4) imagenesSolucion \leftarrow Lista vacía
- 5) Para cada fotograma de ListaKeyFrames
 - a) Añadir en imagenesSolucion la imagen con la misma posición que el fotograma en ListaFramesBuenos
- 6) Escribir cada imagen de imagenesSolucion en OUTPUTPATH
- 7) Sacamos el tiempo_final
- 8) *creaVideo*(imagenesSolucion, OUTPUTPATH)
- 9) Devolver tiempo_final

Pseudocódigo 1. Función Principal

CalcularFotogramasClave: Dada una secuencia de N fotogramas en la carpeta INPUTPATH, los parámetros K , T , H , una técnica y el tipo de distancia.

- 1) ListaFrames \leftarrow Lista vacía
- 2) Para cada fotograma en INPUTPATH, $f=1$, $f \leq N$, $f+=T$
 - a) Si técnica es igual a media:
 - i) Añadir a ListaFrames *media_histogramas*(fotograma, H)
 - b) Si no, si técnica es igual a concatena:
 - i) Añadir a ListaFrames *concatena_histogramas*(fotograma, H)
- 3) ListaCentroides \leftarrow *AplicaKmedias*(ListaFrames, K)
- 4) ListaKeyFrames \leftarrow *CalculaCentroidesClases*(ListaCentroides, ListaFrames, tipoDistancia)
- 5) Devolver ListaKeyFrames y ListaFrames

Pseudocódigo 2. Función CalcularFotogramasClave

media_histogramas: Dada una imagen o fotograma y el parámetro H

- 1) sumaHistogramas \leftarrow Lista vacía
- 2) Llenar sumaHistogramas con H ceros
- 3) Para cada histograma de cada color:
 - a) Para cada posición del histograma
 - i) Sumar esa posición en la misma posición de sumaHistogramas
- 4) Para cada valor de sumaHistogramas
 - a) Dividir valor entre 3
- 5) Devolver sumaHistogramas

Pseudocódigo 3. Función media_histograma

concatena_histogramas: Dada una imagen o fotograma y el parámetro H

- 1) listaConcatenada \leftarrow Lista vacía
- 2) Para cada histograma de cada color:
 - a) Para cada posición del histograma
 - i) Añadir esa posición a listaConcatenada
- 3) Devolver listaConcatenada

Pseudocódigo 4. Función concatena_histogramas

aplicaKMedias: Dada una ListaFrames y el parámetro K

- 1) kmeans \leftarrow KMeans(n_clusters=K).fit(ListaFrames)
- 2) listaCentroides \leftarrow kmeans.clusters_centers_
- 3) Devolver listaCentroides

Pseudocódigo 5. Función aplicaKMedias

CalculaCentroidesClases: Dada una ListaCentroides, una ListaFramesBueno, el parámetro K y el tipoDistancia

- 1) ListaKeyFrames \leftarrow Lista vacía
- 2) Si tipoDistancia es euclídea:
 - a) Para cada centroide k de ListaCentroides:
 - i) Añadir a lista ListaKeyFrames *distancia_euclidea*(ListaCentroides, ListaFramesBuenos, k)
- 3) Si tipoDistancia es hamming:
 - a) Para cada centroide k de ListaCentroides:
 - i) Añadir a lista ListaKeyFrames *distancia_hamming*(ListaCentroides, ListaFramesBuenos, k)
- 4) Si tipoDistancia es manhattan:
 - a) Para cada centroide k de ListaCentroides:
 - i) Añadir a lista ListaKeyFrames *distancia_manhattan*(ListaCentroides, ListaFramesBuenos, k)
- 5) Devolver ListaKeyFrames

Pseudocódigo 6. Función CalculaCentroidesClases

creaVideo: Dada una lista de imágenes o fotogramas y una carpeta de salida OUTPUTPATH

- 1) img1 \leftarrow imágenes[0]
- 2) height, width, layers \leftarrow img1.shape
- 3) video \leftarrow cv2.VideoWriter(OUTPUTPATH + "\solucionVideo.avi",-1,1,(width,height))
- 4) Para cada imagen de imágenes:
 - a) Añadir 3 veces la imagen a video
- 5) cv2.destroyAllWindows
- 6) video.release()

Pseudocódigo 7. Función creaVideo

IV. RESULTADOS

Prueba 1:

Para experimentar hemos usado el llamado v21 del DataBase[5] del cual hemos sacado mediante el programa FreeVideo (Free Video to JPG Converter) el total de sus frames (3282).

Nuestra experimentación tiene principalmente 3 objetivos determinar unos buenos parámetros de K, T y H; encontrar la mejor técnica para sacar información de los histogramas (media o concatenación) y encontrar la mejor distancia (euclídea, haming o manhattan) para sacar los frames correctos a partir de los centroides.

Para ello vamos a usar como benchmark la función time() del módulo time [6] de Python con la cual calcularemos el tiempo de ejecución para cada una de las pruebas realizadas.

Para resumir los resultados obtenidos para cada prueba usaremos una tabla, a partir de la cual posteriormente sacaremos unas conclusiones.

Este método y análisis de experimentación que acabamos de explicar va a ser el que vamos a usar en las próximas pruebas

Objetivo 1: El primer objetivo es determinar unos buenos parámetros para K, T y H por eso vamos a dejar como distancia y técnica para los histogramas, distancia euclídea y media, respectivamente. Posteriormente experimentaremos con estos parámetros.

| N.º Experimento | T | K | H | Tiempo de ejecución |
|-----------------|----|----|-----|------------------------|
| 1 | 30 | 20 | 256 | 1.26627397 53723145 |
| 2 | 20 | 20 | 256 | 1.65159177 78015137 |
| 3 | 40 | 20 | 256 | 0.89849138 2598877 |
| 4 | 10 | 20 | 256 | 3.82014346 1227417 |
| 5 | 50 | 20 | 256 | 0.72083258 62884521 |
| 6 | 40 | 10 | 256 | 0.66929745 6741333 |
| 7 | 40 | 15 | 256 | 0.77049779 89196777 |
| 8 | 40 | 30 | 256 | 1.11536765 09857178 |
| 9 | 40 | 40 | 256 | 1.36125469 20776367 |
| 10 | 40 | 20 | 200 | 0.78489017 48657227 |

| | | | | |
|----|----|----|-----|------------------------|
| 11 | 40 | 20 | 150 | 0.71207451 82037354 |
| 12 | 40 | 20 | 100 | 0.63110280 03692627 |
| 13 | 40 | 20 | 50 | 0.55768013 00048828 |
| 14 | 40 | 20 | 10 | 0.49599075 31738281 |

Conclusiones:

Para determinar el parámetro T hemos fijado aleatoriamente los parámetros K y H en 20 y 256 respectivamente. Si nos fijamos en los primeros 4 ejemplos podemos ver que cuanto mayor es la T, menor es el tiempo de ejecución puesto que el número de fotogramas es menor, pero también hemos observado que si la T es muy grande se pierde información en el video, es decir hay partes importantes que no aparecen en la solución, en consecuencia a esto hemos observado que el valor más apropiado es 40 puesto que no tiene un tiempo de ejecución excesivamente alto, y hemos observado que apenas se pierde información. El parámetro T también depende de la cantidad de fotogramas que hallamos sacado del video, y la longitud de este. Este valor lo hemos determinado para nuestro video, el cual tiene 3282 fotogramas.

Por último, para el parámetro H hemos utilizado los valores anteriores de T y K que son 40 y 20 respectivamente. El significado del parámetro H es la longitud que tendrá cada histograma de cada canal de color de cada frame, por lo tanto, como el máximo de valores de un color es 256, hemos limitado el valor H a 256. Podemos observar que cuanto menor es el valor de H, menor es el tiempo de ejecución, puesto que la longitud de los arrays con los que operar es menor. Pero también cuanto menor es el valor, menor es la precisión a la hora de elegir los frames más importantes puesto que divide los valores de cada color en H, por lo tanto, perdemos mucha información. En consecuencia a esto, hemos determinado que un buen valor de H para este video es 256 porque es el que mejor resultado nos da en cuanto a calidad de los frames seleccionados y el tiempo de ejecución no es excesivamente alto.

Objetivo 2 y 3: A partir de los valores K, T y H anteriores que son 20, 40 y 256 respectivamente, debemos probar cual es la mejor técnica de recopilación de información del histograma y la mejor distancia para sacar los frames.

| N.º Experimento | Técnica | Distancia | Tiempo de ejecución |
|-----------------|---------------|-----------|----------------------------|
| 1 | media | Euclídea | 0.898491 38259887 7 |
| 2 | concatenación | Euclídea | 3.849737 40577697 75 |
| 3 | media | Hamming | 0.750463 72413635 25 |
| 4 | media | Manhattan | 0.884790 18211364 75 |

Conclusiones:

Para experimentar con la técnica hemos decidido fijar la distancia Euclídea. Observando los experimentos 1 y 2 podemos ver que el tiempo de ejecución de una técnica con respecto a otra es el cuádruple. Aunque el resultado de la concatenación es mejor, el tiempo de ejecución es excesivamente alto. Esto se debe a que al concatenar los 3 histogramas la longitud de los arrays es el triple. Así que determinamos que la mejor técnica es hacer la media puesto que la concatenación no tiene un gran impacto en el resultado, pero si en el rendimiento.

Para decidir que distancia es la mejor, partimos de elegir, como técnica, la media de los 3 histogramas de cada imagen. En el experimento 3 podemos ver que el uso de la distancia Hamming reduce el tiempo de ejecución, pero el resultado obtenido no es demasiado bueno puesto que las imágenes de la solución son muy parecidas entre si y esto hace que se pierda información. Con respecto al experimento 4 podemos ver que el tiempo de ejecución de la distancia Manhattan es prácticamente el mismo, pero el resultado obtenido es un poco peor que el de la distancia Euclídea, por lo tanto, nos quedaremos con esta última.

Resultado Final Prueba 1:

K=20, T=40, H=256, Técnica=media,
Distancia= Euclídea

Prueba 2:

Como hemos podido ver en el experimento 1, los atributos K, T y H guardan relaciones en cuanto a la naturaleza del video y con ellos mismos:

K: Cuanto mayor es su valor, mayor es el tiempo de ejecución. Depende del vídeo debido a que, por un lado, un número pequeño de K en un vídeo extenso produce pérdida de información, y por otro lado un número grande en un vídeo corto produce fotogramas clave muy parecidos y redundantes.

T: Depende de la cantidad de fotogramas de vídeo y el aumento de su valor reduce el tiempo de ejecución. Si T es muy grande puede producir pérdida de información. Existe una particularidad con T respecto a K en el sentido de, si tenemos N fotogramas de los cuales vamos a usar N/T fotogramas, se debe cumplir que

$N/T \geq K$.

H: Este parámetro es el más importante a la hora de evaluar la calidad visual de vídeo ya que representa el tamaño del histograma, por lo cual debe ser acorde a nuestro video inicial. Un mayor valor en H supondrá un mayor tiempo de ejecución debido a que se procesarán más datos, los cuales pueden ser obviados o no.

Esta vez vamos a experimentar con un vídeo bastante más corto con un total de 290 frames correspondiente al vídeo v54 del DataBase[5]. Empezamos como en la prueba 1, primero, con la elección correcta de los valores de T, K y H con técnicas predefinidas: media y distancia euclídea.

Objetivo 1:

| N.º Experimento | T | K | H | Tiempo de ejecución |
|-----------------|----|----|-----|-------------------------|
| 1 | 10 | 29 | 256 | 0.671267986 2976074 |
| 2 | 20 | 14 | 256 | 0.303365468 97888184 |
| 3 | 20 | 10 | 256 | 0.241067409 51538086 |
| 4 | 20 | 7 | 256 | 0.193928956 98547363 |
| 5 | 15 | 9 | 256 | 0.303633689 8803711 |
| 6 | 15 | 8 | 256 | 0.272756814 95666504 |
| 7 | 12 | 8 | 256 | 0.295920610 42785645 |
| 8 | 15 | 9 | 128 | 0.244835615 15808105 |
| 9 | 15 | 9 | 68 | 0.244322299 9572754 |
| 10 | 40 | 20 | 10 | 0.495990753 1738281 |

Conclusiones:

Concluimos que los valores óptimos en cuanto a tiempo de ejecución y calidad de los resultados son los correspondientes a el experimento 5:

T=15, K=9 y H=256. Nótese que al disminuir la H disminuye el tiempo de ejecución, pero el vídeo final posee fotogramas muy parecidos y por tanto no vale la pena.

Otra comparación que cabe destacar son las del experimento 4 con el 5, finalmente escogido este último. Si bien el experimento 4 su tiempo de ejecución es mucho menor que el del 5, El valor de T de este primero provoca saltos de fotogramas que son importantes, esto seguramente es debido a partes cortas del vídeo, lo que supone menos fotogramas de estas.

Objetivo 2 y 3:

Vamos a evaluar ahora cuál sería la mejor técnica de preparación de datos y distancia.

| N.º Experimento | Técnica | Distancia | Tiempo de ejecución |
|-----------------|---------------|-----------|------------------------|
| 1 | media | Euclídea | 0.3036336 898803711 |
| 2 | concatenación | Euclídea | 0.7683568 000793457 |
| 3 | media | Hamming | 0.2643363 475799560 |
| 4 | concatenación | Hamming | 0.5447552 20413208 |
| 5 | media | Manhattan | 0.2843534 946441650 |
| 6 | concatenación | Manhattan | 0.6660721 302032471 |

Conclusiones:

Los mejores valores son técnica: media y distancia: euclídea. Esto es así porque, aunque el tiempo no sea el mejor, la calidad final es mucho mejor a la obtenida en los demás experimentos.

Resultado Final Prueba 2:

K=9, T=15, H=256, Técnica=media, Distancia=Euclídea

Prueba 3:

Proseguimos con el video v22 del DataBase[5].

Objetivo 1:

| N.º Experimento | T | K | H | Tiempo de ejecución |
|-----------------|---|-----|-----|-------------------------|
| 1 | 1 | 183 | 256 | 10.45605635 6430054 |
| 2 | 5 | 30 | 256 | 0.834032058 7158203 |
| 3 | 5 | 15 | 256 | 0.534448623 6572266 |
| 4 | 5 | 6 | 256 | 0.343614339 8284912 |
| 5 | 5 | 6 | 64 | 0.281977176 66625977 |
| 6 | 5 | 6 | 1 | 0.989299774 1699219 |

En el primer experimento estamos procesando el vídeo entero y sacando una copia de él sólo para ver cuál es el tiempo máximo que podría durar esta prueba.

En comparación al segundo se nota bastante la diferencia en cuanto a que es más rápido y el vídeo ya es resumido, pero se pueden observar fotogramas que dicen lo mismo. Vamos a probar a disminuir la mitad el número de clusters que queremos.

3. hemos disminuido los fotogramas repetidos, pero consideramos que no es suficiente.

4. queda perfectamente resumido el vídeo. Veamos ahora la posibilidad de disminuir el valor de H para hacer aún más rápida la ejecución.

5. No se aprecia variación en cuanto a la calidad y hemos mejorado la calidad, vamos a probar a disminuir un poco más.

6. Cuando la H toma un valor muy pequeño como 1 el tiempo de ejecución es mucho mayor y el vídeo resumen se reduce a un solo fotograma repetido.

Conclusiones:

Por lo visto anteriormente establecemos como valores correctos para este vídeo: T=5, K=6, y H=64. En comparación a otras pruebas no hay mucho que decir, mantiene el mismo patrón que ya explicamos detalladamente en la Prueba 2.

Objetivo 2 y 3:

| N.º Experimento | Técnica | Distancia | Tiempo de ejecución |
|-----------------|---------------|-----------|-------------------------|
| 1 | media | Euclídea | 0.28197717 666625977 |
| 2 | concatenación | Euclídea | 0.409262657 16552734 |
| 3 | media | Hamming | 0.255826711 6546631 |
| 4 | concatenación | Hamming | 0.336610078 8116455 |
| 5 | media | Manhattan | 0.270396471 02355957 |
| 6 | concatenación | Manhattan | 0.395735502 243042 |

Conclusiones:

En cuanto a si elegir media o concatenación escogemos la media debido a que la concatenación únicamente nos aporta más valores para procesar y por tanto mayor tiempo de ejecución, cosa que si no nos da ningún beneficio no nos interesa.

En cuanto a la distancia, con Hamming obtenemos buenos tiempos, pero la calidad de vídeo es muy mala, faltan fotogramas representativos y sobran repetidos. Por otro lado; las distancias Euclídea y Manhattan varían muy poco en tiempo y son iguales en calidad, escogemos la distancia euclídea.

Resultado Final Prueba 3:

K=6, T=5, H=64, Técnica=media, Distancia=Euclídea

Prueba 4:

La última prueba con el video v66 del DataBase[5]

Objetivo 1:

| N.º Experimento | T | K | H | Tiempo de ejecución |
|-----------------|----|-----|-----|-------------------------|
| 1 | 1 | 219 | 256 | 122.1669766 9029236 |
| 2 | 10 | 21 | 256 | 1.462972879 40979 |
| 3 | 10 | 11 | 256 | 0.808196067 8100586 |
| 4 | 10 | 9 | 256 | 0.849412202 835083 |
| 5 | 10 | 6 | 128 | 0.238606214 52331543 |
| 6 | 10 | 6 | 1 | 0.877357959 7473145 |

1. Cota Superior

2. Ha quedado bastante resumido, pero aún podemos precisar más eliminando algunos fotogramas repetidos, entonces, disminuimos el valor de K lo cual va a dar además velocidad.

3. Volveremos a disminuir un poco más el valor de K.

4. Queda muy bien resumido y ahora vamos a ver los valores H si cambian en cuánto a la calidad del vídeo resultado

5. Ha mejorado el tiempo y sin cambios en la calidad.

6. Si el valor de H es 1 se queda.

Conclusiones:

Por lo visto anteriormente establecemos como valores correctos para este vídeo: T=10, K=6, y H=128. En comparación a otras pruebas no hay mucho que decir, mantiene el mismo patrón que ya explicamos detalladamente en la Prueba 2.

Objetivo 2 y 3:

| N.º Experimento | Técnica | Distancia | Tiempo de ejecución |
|-----------------|---------------|-----------|---------------------|
| 1 | media | Euclídea | 0.23860621452331543 |
| 2 | concatenación | Euclídea | 0.41271400451660156 |
| 3 | media | Hamming | 0.21354007720947266 |
| 4 | concatenación | Hamming | 0.3343179225921631 |
| 5 | media | Manhattan | 0.22660517692565918 |
| 6 | concatenación | Manhattan | 0.37082791328430176 |

Conclusiones:

La conclusión que podemos dar es la misma que la prueba anterior, es decir, Hamming nos aporta un mal resultado y en cuanto a euclídea y manhattan los resultados son casi los mismos. Se mantiene que media es más rápido que concatenación.

Resultado Final Prueba 3:

K=6, T=10, H=128, Técnica=media, Distancia=Euclídea

En vista a los resultados de los experimentos hemos determinado que los parámetros K, T y H van a variar en función de la longitud del video, si queremos que el tiempo de ejecución sea alto o bajo y la calidad que queramos para el resultado.

Partiendo de que dado un video sacamos todos sus fotogramas que serán N, el parámetro T debe ser en un rango aproximado entre 10 y 60 para que ni seleccionemos demasiados frames iguales, ni nos dejemos frames sin seleccionar, por eso nosotros lo hemos determinado en 20, así el algoritmo KMedias recibirá N/20 frames.

Después el parámetro K el cual será el número de imágenes que queremos como solución lo hemos determinado que debe ser aproximadamente $T/(2 \text{ ó } 3)$ para que por cada 2 ó 3 frames saquemos 1 como solución.

Por último, en vista a los experimentos y los resultados el parámetro H lo determinamos como 256 por defecto.

V. CONCLUSIONES

El objetivo final del trabajo era conseguir resumir un video de la manera más eficiente y con la máxima calidad posible. En vista a la experimentación que hemos llevado a cabo en el apartado anterior y a otra serie de experimentos realizados durante el desarrollo del trabajo, podemos decir que hemos conseguido el objetivo principal del trabajo, pudiendo además de experimentar con los parámetros K, T y H, también con otras técnicas como son la media y la concatenación de histogramas y con otras distancias como son la distancia Euclídea, Hamming y Manhattan.

Hemos llegado a la conclusión de que quizás se podrían usar otros algoritmos como el K-NN y comprobar si se obtienen mejores o peores resultados, ya que con el algoritmo K-Medias los resultados varían mucho dependiendo de los valores que le demos a los parámetros de entrada.

También pensamos que este campo de investigación, de extraer los datos más importantes de un conjunto de datos mucho

mayor, es muy interesante y podría tener muchas aplicaciones de utilidad en la vida real.

Además, se le podrían aplicar algunas mejoras al trabajo como por ejemplo perfeccionar el código para hacerlo mucho más eficiente de cara a utilizarlo con grandes cantidades de datos y que siga resultando útil. También sería interesante buscar la forma de que parezca el video resultante parezca una especie de “tráiler” en vez de una sucesión de fotogramas. Por último, añadir que una posible mejora sería trabajar con el sonido para que el video resultante también tenga sonido, pero para ello habría que trabajar antes con la mejora anterior.

REFERENCIAS

- [1] <https://www.cs.us.es/cursos/iais-2017/trabajos/Clustering.pdf>
- [2] <https://es.wikipedia.org/wiki/K-means>
- [3] <https://www.cs.us.es/cursos/iais-2017/temas/Aprendizaje.pdf>
- [4] <https://www.cs.us.es/cursos/iais-2017/temas/Aprendizaje.pdf>
- [5] <https://sites.google.com/site/vsummsite/download>
- [6] <https://julioecheverri.wordpress.com/2015/12/25/medir-tiempo-de-ejecucion-en-python/>