View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Repositorio da Universidade da Coruña

XXXVIII Jornadas de Automática
https://doi.org/10.17979/spudc.9788497497749.1015

BY-NC-SA

EVALUACIÓN DE MÉTODOS PARA REALIZAR RESÚMENES AUTOMÁTICOS DE VÍDEOS

Pablo Rubio Fernández
Escuela de Ingenierías Industrial e Informática, Campus de Vegazana s/n 24071 León, España,
prubif00@estudiantes.unileon.es

Eduardo Fidalgo Fernández
Escuela de Ingenierías Industrial e Informática, Campus de Vegazana s/n 24071 León, España,
efidf@unileon.es

Enrique Alegre Gutiérrez
Escuela de Ingenierías Industrial e Informática, Campus de Vegazana s/n 24071 León, España,
ealeg@unileon.es

Víctor González Castro
Escuela de Ingenierías Industrial e Informática, Campus de Vegazana s/n 24071 León, España,
victor.gonzalez@unileon.es

Resumen

En este trabajo se estudian, presentan y evalúan tres métodos que permiten realizar resúmenes de vídeos de manera automática, manteniendo la información del vídeo que cada uno de los métodos presentados considera como esencial. Se han revisado los métodos Video2GIF, basado en una red neuronal convolucional de aprendizaje profundo, Move Detector, un algoritmo que detecta y almacena los fotogramas que contienen movimiento, y Peaks Volume, que resume en función de un análisis del espectro de audio del vídeo. La evaluación de los métodos Video2GIF y Peaks Volume se ha realizado utilizando el dataset VSUMM, y la evaluación del método Move Detector, utilizando el dataset VIRAT. Los resúmenes obtenidos se han evaluado utilizando CUS (Comparison of User Summaries). A partir de los mismos se puede concluir que los resultados obtenidos con Video2GIF contienen la información más relevante del vídeo original cuando este contiene escenas cortas que albergan acciones humanas, dado que este método utiliza una red entrenada con dicho propósito, mientras que Peaks Volume ha destacado en el resumen de documentales, pero también ha conseguido unos resultados superiores a 0.4 sobre 1 en el resto de categorías de vídeos reduciendo la duración del vídeo original a la mitad o menos.

Palabras Clave: Fotograma clave, resumen de vídeo, red neuronal convolucional, aprendizaje profundo, Python, CUS.

1 INTRODUCCIÓN

Actualmente, cada día se capturan y registran multitud de momentos de nuestra vida cotidiana a través de cámaras y teléfonos móviles. Para hacerse una idea, la plataforma Youtube recibe diariamente una media de 100 horas de vídeos por minuto, haciendo que escenas importantes pasen desapercibidas entre grandes cantidades de datos de vídeos. Por este motivo, el disponer de técnicas de resumen automáticos eficientes para extraer información de grandes cantidades de vídeo en el menor tiempo posible resulta fundamental para la labor diaria de determinadas personas. Por ejemplo, la extracción de inteligencia, de evidencias de tipo vídeo recibidas a diario por parte de los cuerpos y fuerzas de seguridad del estado se vería notablemente agilizada con el uso de dichas técnicas.

Un buen resumen tiene que capturar los momentos clave del vídeo original, manteniendo una coherencia temporal y reduciendo la duración del vídeo lo máximo posible, sin que se pierda información relevante.

En este trabajo se han analizado tres métodos diferentes, explicando su funcionamiento y realizando una pequeña evaluación contra conjuntos de datos disponibles públicamente para poder juzgar la calidad de los resúmenes obtenidos. En la sección 2 se realiza un repaso del estado del arte. La metodología de trabajo de los métodos Video2GIF, Peaks Volume y Move Detector se explica en la sección 3. En la sección 4 se muestra el detalle de la experimentación realizada y, por último, en la sección 5, se presentan brevemente las conclusiones a las que se ha llegado tras la experimentación realizada.

2 ESTADO DEL ARTE

En esta sección se revisa el estado del arte en torno a los tres métodos sobre los que se ha realizado la investigación.

2.1 RESÚMENES DE VÍDEOS EN FUNCIÓN DE SU AUDIO

Existen varias propuestas para resumir los vídeos automáticamente analizando el audio del mismo. Reconocer el género del audio (discurso, música, risas, gritos, etc.) y en función de este, detectar o no el plano en el que se graba a los protagonistas [8] es una de estas propuestas. Otra propuesta diferente consiste en cuantificar escenas de audio según la amplitud, frecuencia y energía instantánea de las modulaciones, ayudándose además de mapas de calor [11]. Jiang et al. [6] centraron su trabajo en vídeos de celebraciones y eventos. Primero, se segmenta el audio, utilizando BIC (Bayesian Information Criterion). Después, se calcula la modulación de la energía para distinguir entre discurso o música, o si una persona habla o canta. Una vez dividido el audio, se utiliza detección de rostros para intentar capturar a los protagonistas del evento, por ejemplo, los novios en el vídeo de una boda, y seleccionar así los fotogramas claves.

Furini et al. [2] propusieron un método para detectar los silencios del audio, y una vez detectados, realizaron tres experimentos. En el primero eliminaron todos los fotogramas asociados a los momentos de silencio, en el segundo reprodujeron esos fotogramas a una velocidad el doble de rápido de lo original, y en el tercero realizaron lo mismo que en el caso anterior, pero con una velocidad tres veces superior a la del vídeo original. Consideraron este tercer experimento como el mejor, pues reduce el tiempo del vídeo original y además mantiene todos los fotogramas para no perder detalle.

2.2 RESÚMENES DE VÍDEOS APLICANDO REDES NEURONALES

Gygli et al. [4] introdujeron un método que busca aprender la importancia de las características globales de un resumen de manera supervisada y optimizar estas para poder resumir una gran variedad de vídeos. La primera misión llevada a cabo fue el aprendizaje de la red convolucional. Después, se alimentó el sistema con unas funciones submodulares que le permiten captar los momentos clave del vídeo original en un resumen. Estas son: función de interés, que predice la importancia de un segmento del vídeo; función de representatividad, que evalúa la representación del vídeo original a través del resumen; función de uniformidad, que se encarga de que el resumen tenga coherencia temporal. Finalmente, se entrenó el sistema con un conjunto de vídeos originales y sus correspondientes resúmenes, para poder realizar el ajuste de cada función de optimización y deducir así la importancia de estas para cada vídeo.

Xu et al. [15] contribuyeron en la mejora de las técnicas de encriptación necesarias para generar resúmenes de vídeos con redes neuronales convolucionales. También propusieron el uso de un conjunto de descriptores latentes como descriptores de fotogramas, los cuales diversifican la salida agregando múltiples localizaciones especiales en la etapa más profunda de la red.

2.3 RESÚMENES DE VÍDEOS ESTÁTICOS

Godbehere et al. [3] realizaron una separación entre el fondo y el primer plano en vídeos estáticos para detectar cualquier variación que se produzca entre estos, y además analizaron el audio de la zona en la que se realiza la grabación de vídeo, para así poder saber la dirección que llevan las personas, la velocidad a la que se desplazan, etc.

Kaewtrakulpong et al. [7] compartieron una mejora del modelo de mezcla Gaussiana adaptable que añade detección de sombras y seguimiento en tiempo real.

Zivkovic [16] también propuso una mejora del modelo de mezcla Gaussiana adaptable para la substracción del fondo que selecciona el número de componentes por píxel necesarios para adaptar la escena observada a las necesidades del objetivo final.

3 MÉTODOS

En esta sección se analizan detalladamente tres métodos para resumir vídeos automáticamente. El primero trabaja con redes neuronales convolucionales, el segundo realiza un análisis del audio del vídeo para generar el resumen y el tercero es un detector de movimiento utilizado para resumir vídeos estáticos.

3.1 VIDEO2GIF

Video2GIF [5] es una aplicación creada por Michael Gygli que utiliza una red neuronal convolucional, entrenada con parejas GIF y no GIF sobre un dataset creado por los autores, para resumir vídeos generando GIFs de los momentos más destacables. Los GIFs son un formato de imagen de poca duración, sin sonido, que reproduce múltiples fotogramas en bucle.

3.1.1 Arquitectura y entrenamiento de la red neuronal

Video2GIF utiliza la arquitectura de red neuronal C3D propuesta por Tran et al. [14]. Para encontrar la mejor configuración, se realizaron pruebas con distintos modelos de red sobre el dataset UCF101, que contiene vídeos de 101 acciones humanas. La arquitectura de red de las primeras pruebas fue la siguiente: 5 capas convolucionales de 64, 128, 256, 256 y 256 filtros respectivamente, seguidas cada una de ellas de una capa de reducción (pooling), 2 capas totalmente conectadas y una capa softmax loss. De los primeros experimentos se concluyó que los mejores resultados se obtenían cuando todas las capas convolucionales tenían profundidad de kernel 3.

Para el aprendizaje de características espacio-temporales, diseñaron una arquitectura diferente basada en lo anterior, con 8 capas convolucionales de 64, 128, 256, 256, 512, 512, 512 y 512 filtros respectivamente, 5 capas de reducción, 2 capas totalmente conectadas de 4096 salidas cada una y una capa de salida softmax.

El entrenamiento de la red se realizó en un dataset creado por los autores de Video2GIF que reúne más de 100.000 GIFs animados y más de 80.000 de los vídeos originales de estos GIFs. Los GIFs se alinean con sus respectivos vídeos originales para así poder obtener los segmentos no seleccionados del vídeo, los cuáles sirven como muestras negativas en el entrenamiento (no GIFs). Para entrenar a la red, los autores realizaron comparaciones entre más de 500.000 parejas de GIFs y no GIFs.

3.1.2 Funcionamiento de Video2GIF

El primer paso que realiza la aplicación, es crear la red neuronal convolucional, con la arquitectura mencionada en el subapartado anterior. Después, divide el vídeo en segmentos no superpuestos. Para ello, utiliza el algoritmo de detección de límites de escena que detecta los cambios de toma, propuesto por Song et al. [12], en el que dada una matriz X y un número de puntos de cambio k, el objetivo se reduce a encontrar una aproximación constante de $H \in R^{dxn}$ en la que H minimice el error de reconstrucción $min_H \|X - H\|$. Los puntos de cambio se encuentran tomando la derivada discreta de primer orden de H y buscando las localizaciones de los valores distintos de 0. La fórmula es:

$$\min_{\boldsymbol{H}} \frac{1}{2} \|\mathbf{X} - \mathbf{H}\|_F^2 + \lambda \sum_{t=1}^{n-1} \|\mathbf{H}_{\cdot,t+1} - \mathbf{H}_{\cdot,t}\|_2 \quad (1)$$

donde $\|.\|_F$ es la norma de Frobenius. El primer término mide el error de reconstrucción, el segundo es la variación total, y λ>0 controla la relativa importancia entre ambos.

Una vez realizada la división de segmentos, el siguiente paso es puntuarlos. Durante el entrenamiento, la entrada es una pareja de segmentos GIF y no GIF. El conjunto de segmentos no GIF $\{s^-\}$ serán todos los segmentos del vídeo que no formen parte de los GIFs de ese vídeo incluidos en el dataset, y el conjunto de segmentos GIF $\{s^+\}$ serán los segmentos GIF de cada vídeo disponibles en el dataset. El modelo aprende una función h: $R^d \rightarrow R$ que mapea un segmento s para dar un valor $h(s)$ a su aptitud como GIF. El programa aprende la función comparando parejas de entrenamiento de forma que el conjunto de segmentos GIF $\{s^+\}$ obtenga una puntuación más alta que el conjunto de segmentos no GIF $\{s^-\}$. Durante la evaluación, la aplicación sólo recibe un segmento, y calcula su idoneidad como GIF mediante la función aprendida de puntuación. Por último, calcula el valor $h(s)$ para todos los segmentos $s \in S$ del vídeo y produce un ranking ordenado según las puntuaciones.

Finalmente, la función de pérdida Huber describe la penalización incurrida en los GIFs de menor calidad. La fórmula de la función de pérdida Huber utilizada por Video2GIF es la siguiente:

$$l_{Huber}(s^+, s^-) = \begin{cases} \frac{1}{2} l_2(s^+, s^-), & if\ u \leq \delta \\ \delta l_1(s^+, s^-) - \frac{1}{2}\delta^2, & otherwise \end{cases} \quad (2)$$

donde $u = 1 - h(s^+) + h(s^-)$. El parámetro δ define el punto en el cual la pérdida comienza a ser lineal, y tendrá un valor dependiente de la calidad del GIF, siendo más alto para los GIFs de mejor calidad. l_1 y l_2 vienen dados por la fórmula (3) para los valores p=1 y p=2 respectivamente.

$$l_p(s^+, s^-) = \max(0, 1 - h(s^+) + h(s^-))^p \quad (3)$$

Finalmente, definen el objetivo como la pérdida total sobre el dataset D y un término de regulación con la norma de Frobenius al cuadrado en el modelo de pesos W:

$$L(D, \mathbf{W}) = \sum_{S_i \in D} \sum_{(s^+, s^-) \in S_i} l_{Huber}(s^+, s^-) + \lambda \|\mathbf{W}\|_F^2 \quad (4)$$

donde λ es el parámetro de regulación.

3.1.3 Evaluación y resultados

Los autores de Video2GIF utilizan mean Average Precision (mAP) y Meaningful Summary Duration (MSD) para evaluar los resúmenes de vídeos.

El primer método, mean Average Precision, calcula la precisión con la que el programa captura los mejores momentos, es decir, los que compongan el ground truth. El segundo método, meaningful summary duration, se basa en agregar segmentos según su puntuación hasta cubrir los segmentos dados por el ground truth. Si alberga muchos segmentos de baja importancia en los primeros lugares obtendrá una puntuación de MSD baja.

Los resultados obtenidos por los autores se pueden observar en la Tabla 1.

Tabla 1: Resultados de la evaluación del método Video2GIF

Método	**nMSD ↓**	**mAP ↑**
Joint embedding	54.38%	12.36%
Category-spec. SVM	52.98%	13.46%
Domain-spec. RankSVM	46.40%	16.08%
Classification	61.37%	11.78%
Rank, video agnostic	53.71%	13.25%
Rank, l_1 loss	44.60%	16.09%
Rank, l_2 loss	44.43%	16.10%
Rank, Huber loss	44.43%	**16.22%**
Rank, adaptative Huber loss	44.58%	16.21%
Rank, adaptative Huber loss + context (Ours)	44.19%	16.18%
Ours + model averaging	**44.08%**	16.21%
Approx. Bounds	38.77%	21.30%

3.2 PEAKS VOLUME

En esta subsección se detalla el algoritmo Peaks Volume, el cual resume el contenido de un vídeo en base a los picos de volumen detectados en el audio del mismo.

3.2.1 Funcionamiento de Peaks Volume

El algoritmo analiza el espectro de audio en busca de picos de volumen, los cuales se encontrarán cuando la gráfica del audio sufra un incremento y un descenso del volumen en el mismo punto.

El primer paso es abrir el vídeo indicado y extraer toda su pista de audio, para posteriormente, calcular el volumen y dibujarlo en una gráfica. El segundo paso consiste en hallar los picos. Para ello, se recorre la gráfica multiplicando incrementos por descensos, y obteniendo un resultado distinto de 0 en los puntos donde incremento y descenso se produzcan a la vez. Tras un pico, habrá un tiempo determinado en el que no se podrán analizar otros picos, pues es evidente que dos picos de volumen muy seguidos seguramente pertenezcan a la misma acción. Finalmente, se recorta el vídeo en subclips asociados a los picos detectados, que se unirán dando lugar al resumen final.

3.3 MOVE DETECTOR

Para resumir vídeos estáticos grabados por cámaras fijas se utilizará el algoritmo Move Detector, que separa el fondo del primer plano para detectar movimiento.

3.3.1 Funcionamiento de Move Detector

Al iniciar el vídeo, el algoritmo captura el primer fotograma, que será el fondo inicial. Una vez capturado el fondo, el script empieza a analizar los fotogramas. Primero, redimensiona sus tamaños, y después, los convierte a escala de grises. Después, aplica un filtro Gaussiano para trabajar mejor con estos fotogramas. Tras el filtrado, se compara el fotograma de referencia con el resto de fotogramas extraídos mediante una substracción de imágenes, es decir, la diferencia en valor absoluto, píxel a píxel entre los diferentes fotogramas. El valor absoluto nos dará la intensidad de cada pixel diferente entre fondo y fotograma. Si este valor es superior a un umbral predeterminado y además el área ocupada por estos píxeles supera unas determinadas dimensiones, se considera que hay movimiento en el fotograma. El fotograma con movimiento detectado pasa a ser el nuevo fondo, sobre el que se realizarán estas mismas operaciones con el resto de fotogramas. Con el fondo cambiante se consigue que las variaciones de luz en el ambiente no sean detectadas continuamente como movimiento.

4 EXPERIMENTACIÓN Y RESULTADOS

Los experimentos han sido desarrollados con un ordenador portátil MSI con tarjeta gráfica NVIDIA GTX960M. Para utilizar la GPU, recomendada a la hora de crear la red neuronal artificial de Video2GIF, fue necesario instalar los programas NVIDIA Nsight HUD Launcher 5.2[1], NVIDIA GeForce Experience[2] y la biblioteca Cuda[3].

Los datasets utilizados durante este experimento han sido:

-VSUMM [1]: Compuesto por vídeos de usuarios que contienen gran variedad de géneros, desde una serie de

[1] https://developer.nvidia.com/gameworksdownload#?dn=nsight-visual-studio-edition-5-2-0

[2] http://www.nvidia.es/object/geforce-experience-es.html

[3] https://developer.nvidia.com/cuda-downloads

dibujos animados, hasta vídeos de fútbol, pasando por noticieros, documentales y por vídeoclips musicales. En la creación del ground truth de este dataset, que es el conjunto de fotogramas claves de cada vídeo, participaron varios usuarios que extrajeron los fotogramas claves de cada vídeo, permitiendo así evaluar cuantitativamente a través del método CUS la efectividad de los algoritmos de resúmenes automáticos.

-VIRAT [9]: Es un dataset de vídeos grabados con cámaras fijas, donde el fondo es estático y quedan registrados los movimientos de objetos móviles. En este dataset, el movimiento detectado de cada vídeo está incluido en documentos de texto donde aparecen registrados las coordenadas de cada fotograma que alberga movimiento. Los vídeos de este dataset no contienen audio.

-UCF101 [13]: Utilizado directamente por los autores de Video2GIF para realizar el entrenamiento de la red. Este dataset contiene vídeos de 101 acciones humanas, influyendo sobre los resultados finales que se obtienen en los resúmenes realizados con este método.

Se han elegido 6 vídeos aleatoriamente del dataset VSUMM para realizar los experimentos: los vídeos V11, V12, V79 y V107 de la carpeta *new_database* y los vídeos V21 y V42 de la carpeta *database*. V11 y V12 forman parte de una miniserie de dibujos animados. V79 es un vídeo corto de un partido de fútbol. V21 trata de un documental sobre el agua en inglés, similar a V42, y V107 es un vídeoclip musical.

4.1 MÉTODO DE EVALUACIÓN

Para evaluar los resúmenes de vídeos se ha utilizado el método CUS (*Comparison of User Summaries*) [1], el cuál realiza una comparación entre el resumen de usuario (ground truth) y el automático. Este método asigna una puntuación en función de los fotogramas claves que coincidan entre el resumen automático y el resumen presentado en el ground truth.

La fórmula del método CUS es la siguiente:

$$CUS_A = \frac{n_{mAS}}{n_{US}} \qquad (4)$$

donde n_{mAS} es el número de fotogramas claves del resumen automático que coinciden con fotogramas claves del ground truth, y n_{US} es el número de fotogramas clave del resumen manual. El valor oscilará entre 0 y 1, siendo 1 el mejor resultado.

Se ha añadido una modificación propia a este método para tener en cuenta el tiempo del vídeo y el del resumen y poder puntuar más alto los resúmenes que más reduzcan la duración de los vídeos originales. Para ello, se ha calculado el porcentaje de tiempo que el vídeo resumen dura en comparación con el vídeo original, y se ha dividido el resultado de la fórmula CUS_A entre este porcentaje. En las tablas nos referiremos a este método como CUS_A adaptado.

4.2 RESULTADOS

4.2.1 Resultados de los resúmenes realizados con Peaks Volume

En la Tabla 2 se puede observar que el vídeo V12 obtiene una mayor puntuación que el vídeo V21 a pesar de que tiene menor porcentaje de acierto de fotogramas claves, pero la duración de los resúmenes respecto a sus vídeos originales le otorgan una puntuación más alta en el método CUS adaptado.

Tabla 2: Puntuaciones CUS de los resúmenes realizados con Peaks Volume

Vídeos	% duración	Puntuación CUS_A	Puntuación de CUS_A adaptado
V11	39.1%	0.636	1.63
V12	42.1%	0.462	1.1
V21	56.9%	0.538	0.94
V79	61.7%	0.429	0.7
V107	60.9%	0.6	0.985
V42	**58.4%**	**1**	**1.71**

Algo importante, especialmente en los resúmenes de los vídeos V21, V42 y V107, y que no se evalúa en este trabajo, es la coherencia del audio. Pese a que tienen unos resultados próximos a 1 (muy superiores en el caso de V42, que obtiene el mejor resultado), la música o los discursos sufren cortes, muchas veces interrumpiendo una frase a la mitad.

La miniserie de los vídeos V11 y V12 no tiene conversaciones, sólo sonidos de movimiento, gritos, golpes, o risas. Este método capta muchas de las escenas clave y reduce el tiempo del vídeo original.
En V79, se muestra un gol en un partido de fútbol con su celebración y sus repeticiones desde diferentes tomas. Este método capta el gol perfectamente, pero no así la celebración del jugador, dado que el nivel de sonido es menor tras el grito del gol.

4.2.2 Resultados de los resúmenes realizados con Video2GIF

Para resumir los vídeos con el método Video2GIF, se han obtenido 5 GIFs de cada vídeo analizado, de 5 segundos cada uno, y se han extraído todos los fotogramas de cada GIF. Entre los fotogramas, se ha buscado la coincidencia con los fotogramas claves dados en el dataset VSUMM. Los resultados se muestran en la Tabla 3.

Tabla 3: Puntuaciones CUS de los resúmenes realizados con Video2GIF

Vídeos	% duración	Puntuación CUS_A	Puntuación de CUS_A adaptado
V11	54.3%	0.364	0.67
V12	43.9%	0.462	1.05
V21	No realizado	-	-
V79	**53.2%**	**0.571**	**1.074**
V107	12.1%	0.1	0.826
V42	28.1%	0.2	0.712

El método Video2GIF no ha podido resumir el vídeo V21, tras más de 12 horas de procesamiento. No se ha podido analizar la causa de este fallo, aunque se investigará en futuros experimentos con nuevos vídeos similares. Para el resto de los vídeos, el resumen se ha realizado en un tiempo medio de 4 minutos.

Como en este caso el usuario influye directamente en la duración de los resúmenes eligiendo cuantos GIFs quiere conseguir de cada vídeo, se ha seguido otro procedimiento en el que se analiza cuantos GIFs se necesitan para obtener el 50% y el 80% de los fotogramas claves del vídeo. De este modo, los resultados muestran los vídeos en los que Video2GIF ha capturado más fotogramas claves con menor número de GIFs, es decir, los vídeos en los que mejor captura las características clave. El procedimiento es el siguiente:

Inicialmente tenemos los 5 GIFs que tienen la puntuación más alta otorgada por el método Video2GIF. Seleccionamos el siguiente GIF del ranking de puntuación, extraemos todos sus fotogramas y comprobamos si captura alguno de los fotogramas claves del vídeo que aún no hubieran sido capturados. Repetimos este proceso siguiendo el orden de puntuación de Video2GIF hasta capturar el 50% y el 80% de los fotogramas claves de cada vídeo.

Tabla 4: Resultados cuando se han capturado más del 50% de los fotogramas claves

Vídeo	keyframes capturados (Puntuación CUS)	GIFs/ keyframes	% duración
V11	8/11 (0.727)	6/11 (0.545)	65.2%
V12	**7/13 (0.538)**	**6/13 (0.462)**	**52.6%**
V42	3/5 (0.6)	6/5 (1.2)	33.7%
V79	4/7 (0.571)	5/7 (0.71)	53.2%
V107	5/10 (0.5)	20/10 (2)	48.3%

La miniserie de los videos V11 y V12 requiere menos GIFs para obtener más del 50% de los fotogramas claves.

Tabla 5: Resultados cuando se han capturado más del 80% de los fotogramas claves

Vídeo	Keyframes capturados (Puntuación CUS)	GIFs/ keyframes	% duración
V11	**9/11 (0.818)**	**7/11 (0.636)**	**76.1%**
V12	11/13 (0.846)	9/13 (0.692)	78.9%
V42	4/5 (0.8)	12/5 (2.4)	67.4%
V79	6/7 (0.857)	8/7 (1.143)	85.1%
V107	8/10 (0.8)	33/10 (3.3)	79.7%

En este caso, vuelve a ser la miniserie de dibujos animados la que menos GIFs requiere para obtener más del 80% de los fotogramas claves (menos de un GIF por fotograma clave).

Con estos resultados podemos concluir que Video2GIF obtiene mejores resultados en vídeos cortos con mucha variedad de escenas distintas, y en los que estas escenas desarrollen toda su acción en unos pocos segundos. El ejemplo lo tenemos con V11 y V12, la miniserie de dibujos animados en la que los personajes sufren todo tipo de desgracias en un breve periodo de tiempo. En estos vídeos, Video2GIF necesitó menos de un GIF por escena clave para captar casi todos los fotogramas claves.

Por otra parte, en los vídeos con poca diversidad de escenas, como por ejemplo V79, un vídeo de fútbol que contiene un gol, varias repeticiones de este, y al goleador festejándolo, Video2GIF tiende a capturar el mismo tipo acciones, en este caso el gol y sus repeticiones. Por este motivo, se capturan más del 50% de los fotogramas claves utilizando aproximadamente 1 GIF por fotograma clave, pero la cantidad de GIFs necesarios para obtener casi todos los fotogramas aumenta cuando uno de estos fotogramas clave no está relacionado con el balón en juego. Para demostrar esto, escogimos el vídeo V80 del dataset VSUMM, que muestra imágenes de un partido de fútbol y que además también incorpora un logo de introducción a esta sección. Modificamos el vídeo de la siguiente forma:
Los primeros 5 segundos del vídeo son los mismos que los del vídeo original, el logo de introducción. Después, copiamos la primera escena del vídeo original (del segundo 5 al segundo 14) en nuestro vídeo 10 veces. En esta se muestra un rápido ataque que termina en penalti. Una vez realizada la modificación del vídeo, aplicamos sobre este el

método Video2GIF. Los resultados obtenidos mostraron que los 10 GIFs con mayor puntuación del ranking capturaban la misma escena, y el GIF que obtuvo menor puntuación fue el único que captó el logo de introducción.

Por último, se puede observar que para capturar los fotogramas claves de V107, el vídeoclip musical, necesitamos una gran cantidad de GIFs, debido a la similitud entre escenas del vídeo. Además, el hecho de que los GIFs no tengan sonido hace este método menos adecuado para vídeos con audio.

4.2.3 Resultados de los resúmenes realizados con Move Detector

Este método no se ha podido evaluar con CUS, dado que el dataset VIRAT no tiene fotogramas claves etiquetados manualmente. Los resúmenes capturan todo tipo de movimiento y eliminan los momentos en los que el fondo se encuentra desierto.

Para comentar brevemente la capacidad de este algoritmo, se han calculado los resúmenes sobre tres vídeos de VIRAT elegidos aleatoriamente: 0VIRAT_S_000200_00_000100_000171, VIRAT_S_010000_00_000000_000165 y VIRAT_S_000001.

Tabla 6: Resultados de Move detector sobre vídeos del dataset VIRAT

Vídeo	Resumen (mm:ss)	Vídeo original (mm:ss)
0VIRAT_S_000200_00_000100_000171	00:29	01:10
VIRAT_S_010000_00_000000_000165	01:41	02:45
VIRAT_S_000001	03:35	11:29

Move detector elimina imágenes muy similares en las que apenas se ha producido movimiento. Por ejemplo, si en el fondo aparece un paso de cebra, y una persona se detiene a esperar delante de este, el algoritmo sólo almacena unos pocos fotogramas de la espera, esto es debido a que el algoritmo no registra el tiempo de espera del peatón, dado que no hay movimiento, lo que se traduce en una reducción aún mayor del tiempo del resumen.

Uno de los inconvenientes de este método es el tamaño final del resumen, el cual puede llegar a ser superior al tamaño del vídeo original. Esto es debido a que el resultado final es la unión de todos los fotogramas que alberguen movimiento, y si estos han sido extraídos en un formato de mejor calidad que el original, el resumen final necesitará más espacio de memoria que el vídeo original.

5 CONCLUSIONES

En este artículo se han revisado tres métodos diferentes para resumir vídeos automáticamente. Video2GIF, que trabaja con redes neuronales artificiales, Move Detector, que se basa en un algoritmo de detección de movimiento con fondo cambiante y Peaks Volume, script que analiza el audio del vídeo con el objetivo de encontrar los picos de volumen y resumir el vídeo en torno a estos.

La evaluación de los métodos Video2GIF y Peaks Volume se ha realizado sobre el dataset VSUMM, mediante el método de evaluación CUS. La evaluación de Move Detector se ha realizado sobre el dataset VIRAT, el cual contiene vídeos grabados por cámaras fijas.

A pesar de que el dataset VIRAT, con el que se ha evaluado el método Move Detector, no tiene fotogramas claves etiquetados manualmente y no se ha podido aplicar el método de evaluación CUS, se han realizado tres resúmenes en los que se ha capturado todo el movimiento y se han eliminado los fotogramas que carecían de este. El inconveniente de este método es el tamaño final de los resúmenes, el cual, debido a la calidad de los fotogramas extraídos, ha sido superior al tamaño del vídeo original.

El método Peaks Volume ha obtenido una buena puntuación media CUS, siendo menos efectivo en vídeos que contengan mucho ruido. Ha obtenido una puntuación de 0.538 y 1 en vídeos de documentales cortos, pero aún hay margen de mejora en la coherencia del audio final.

Por otro lado, el método Video2GIF ha conseguido los mejores resultados en vídeos que contienen una gran variedad de escenas cortas y que albergan acciones humanas, pero su puntuación es bastante menor cuando se trabaja con vídeos que contienen muchas escenas similares, pues siempre tiende a captar la misma variedad de acciones. Esto es debido a la configuración de la red, la cual se realizó en función a un entrenamiento realizado sobre el dataset UCF101, que contiene vídeos de 101 acciones humanas, y al entrenamiento de la red sobre el dataset creado por los autores de Video2GIF que permitió ajustar los pesos de la red. Es importante aplicar un entrenamiento ligado al objetivo final que se pretende conseguir.

Agradecimientos

Esta investigación ha sido llevada a cabo en base al acuerdo entre la Universidad de León e INCIBE

(Instituto Nacional de Ciberseguridad de España) bajo la Adenda 22.

Referencias

[1] Eliza, S., Avila, F. De, Paula, A., Lopes, B., Jr, L., & Albuquerque, A. De. (2011). VSUMM : A mechanism designed to produce static vídeo summaries and a novel evaluation method. *Pattern Recognition Letters*, *32*(1), 56–68. https://doi.org/10.1016/j.patrec.2010.08.004

[2] Furini, M., & Ghini, V. (2006). An audio-vídeo summarization scheme based on audio and vídeo analysis. *2006 3rd IEEE Consumer Communications and Networking Conference, CCNC 2006*, *2*, 1209–1213. https://doi.org/10.1109/CCNC.2006.1593230

[3] Godbehere, A. B., & Goldberg, K. (2014). Algorithms for visual tracking of visitors under variable-lighting conditions for a responsive audio art installation. *Controls and Art: Inquiries at the Intersection of the Subjective and the Objective*, 181–204. https://doi.org/10.1007/978-3-319-03904-6_8

[4] Gygli, M., & Gool, L. Van. (2015). Vídeo Summarization by Learning Submodular Mixtures of Objectives, 3090–3098. https://doi.org/10.1109/CVPR.2015.7298928

[5] Gygli, M., Song, Y., & Cao, L. (2016). Video2GIF: Automatic Generation of Animated GIFs from Vídeo. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1001–1009. https://doi.org/10.1109/CVPR.2016.114

[6] Jiang, W., Cotton, C., & Loui, A. C. (2011). Automatic consumer vídeo summarization by audio and visual analysis. *Proceedings - IEEE International Conference on Multimedia and Expo*. https://doi.org/10.1109/ICME.2011.6011841

[7] Kaewtrakulpong, P., & Bowden, R. (2001). An Improved Adaptive Background Mixture Model for Real- time Tracking with Shadow Detection. *Advanced Vídeo Based Surveillance Systems*, 1–5. https://doi.org/10.1.1.12.3705

[8] Merialdo, Y. L. and B. (2012). Vídeo Summarization Based on Balanced AV-MMR. *MMM 2012, 18th International Conference on Multimedia Modeling*, *7131/2012*. https://doi.org/http://dx.doi.org/10.1007/978-3-642-27355-1_35

[9] Oh, S., Hoogs, A., Perera, A., Cuntoor, N., Chen, C. C., Lee, J. T., … Desai, M. (2011). AVSS 2011 demo session: A large-scale benchmark dataset for event recognition in surveillance vídeo. *2011 8th IEEE International Conference on Advanced Vídeo and Signal Based Surveillance, AVSS 2011*, (2), 527–528. https://doi.org/10.1109/AVSS.2011.6027400

[10] Radev, D. R. (2004). LexRank : Graph-based Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research*, *22*, 457–479. Retrieved from https://www.jair.org/media/1523/live-1523-2354-jair.pdf

[11] Rapantzikos, K., Evangelopoulos, G., Maragos, P., & Avrithis, Y. (2007). An audio-visual saliency model for movie summarization. *2007 IEEE 9Th International Workshop on Multimedia Signal Processing, MMSP 2007 - Proceedings*, 320–323. https://doi.org/10.1109/MMSP.2007.4412882

[12] Song, Y., Vallmitjana, J., Stent, A., & Jaimes, A. (2015). TVSum: Summarizing web vídeos using titles. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *07–12–June*, 5179–5187. https://doi.org/10.1109/CVPR.2015.7299154

[13] Soomro, K., Zamir, A. R., & Shah, M. (2012). UCF101: A Dataset of 101 human actions classes from vídeos in the wild. *CoRR*, *abs/1212.0*(November), 1–7. Retrieved from http://arxiv.org/abs/1212.0402

[14] Tran, D., Bourdev, L. D., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning Spatiotemporal Features with 3D Convoutional Networks. *CoRR*, *abs/1412.0*. https://doi.org/10.1007/s11263-012-0542-7

[15] Xu, Z., Yang, Y., & Hauptmann, A. G. (2015). A discriminative CNN vídeo representation for event detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *07–12–June*, 1798–1807. https://doi.org/10.1109/CVPR.2015.7298789

[16] Zivkovic, Z. (2004). Improved adaptive Gaussian mixture model for background subtraction. *Proceedings of the 17th International Conference on Pattern Recognition*, *2*(2), 28–31 Vol.2. https://doi.org/10.1109/ICPR.2004.1333992