Técnica de Detección de Manipulación en Vídeos Digitales Basada en los Algoritmos de Compresión

Edgar González Fernández, Ana Lucila Sandoval Orozco, and Luis Javier García Villalba, *Member, IEEE*

***Resumen*—Las imágenes y vídeos digitales juegan un papel muy importante en la vida cotidiana. A día de hoy, la mayor parte de la población es poseedora de cámaras fotográficas de última generación integradas en su dispositivo móvil. El desarrollo tecnológico no sólo facilita la generación de contenido multimedia, sino también la manipulación intencionada de éste, y es aquí donde las técnicas forenses de detección de manipulación sobre imágenes y vídeos cobran gran importancia. En este trabajo se proponen dos metodologías forenses basadas en algoritmos de compresión: La primera de ellas trata de detectar la presencia de recompresión en un vídeo digital mediante el análisis de sus macrobloques, característica propia del estándar H.264-MPEG4. Posteriormente, se utiliza la máquina de soporte vectorial para crear el modelo que permita la verificación del número de recompresiones de un vídeo. La segunda metodología que se explica en este trabajo tiene por objetivo detectar alteraciones de tipo 'empalme', es decir, regiones que no pertenecen al contenido original de una imagen digital, técnica que está basada en la tasa de error que introduce el algoritmo de compresión JPEG cada vez que recomprime una imagen.**

***Palabras claves*—Análisis Forense, Clasificación, Compresión, Macrobloques, Manipulación, Máquinas de Vector Soporte, Vídeos Digitales.**

I. Introducción

Desde siglos atrás, el ser humano siempre ha utilizado la imagen para plasmar la realidad que le rodeaba, o modificarla, en función del mensaje que se quisiera transmitir. Aunque esta evolución, sin duda, tiene un antes y un después con la creación de la fotografía en el siglo XIX.

> "La excitación que acompañó a la invención de la fotografía fue la sensación de que el hombre por primera vez podía ver el mundo como realmente era"(Collier 1986: 3) [1].

Esta afirmación que hace Collier acerca de la fotografía podría no ajustarse al pie de la letra en la actual era digital. Actualmente existe un significante número de delitos informáticos relacionados con la posesión ilícita, distribución o modificación de contenido multimedia. El uso de dispositivos móviles para este propósito hace de estos una importante fuente de evidencia, hecho por el cual los análisis forenses deben ser capaces de autenticar el contenido y examinar si es original o fue manipulado.

La facilidad para manipular imágenes y vídeos digitales se ha incrementado vertiginosamente en los últimos tiempos, y está al alcance del usuario convencional mediante programas como Adobe Photoshop, GIMP, Adobe Premiere, etc. Inclusive, estas manipulaciones son realizadas de manera automática en dispositivos móviles mediante nuevas herramientas que hacen uso de la inteligencia artificial, como pueden ser los embellecedores de rostros, cambios de la expresión facial, mejora de la iluminación de la escena, etc.

En Julio del año 2017 los investigadores de la revista Cognitive-Research [2] utilizaron un dataset de 40 escenas, 30 de las cuales fueron sometidas a cinco tipos diferentes de manipulación, incluyendo manipulaciones físicamente plausibles y no plausibles. Se mostraron a 707 participantes con el fin de evaluar la capacidad de las personas para detectar escenas manipuladas del mundo real. El estudio encontró que sólo el 60 % de las personas fue capaz de detectar las escenas falsas, e incluso entonces, sólo un 45 % de ellos fueron capaces de decir dónde exactamente se encontraba la alteración del contenido (ver Figura 1).

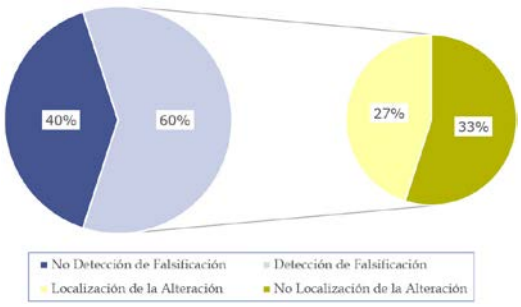


Figura 1: Resultados del estudio de la revista Cognitive-Research.

Es por estos motivos que deben desarrollarse técnicas de detección de manipulación en imágenes y vídeos, imprescindibles para dicho fin. Se hace necesaria la revisión y mejora de los métodos de verificación de la autenticidad e integridad del contenido de una imagen o vídeo, así como el desarrollo de nuevos métodos dirigidos a las técnicas que en el futuro puedan plantearse.

El presente trabajo se desarrolla como sigue: En la Sección II se da una breve introducción a las técnicas de manipulación en vídeos digitales. En la Sección III se presenta el estado del arte sobre detección de manipulaciones en vídeos. Los algoritmos propuestos así como los conceptos necesarios para comprenderlos son explicados en la Sección IV. Posteriormente, en la Sección V se muestran los experimentos realizados y los resultados obtenidos. Finalmente, en las Sección VI se recogen las conclusiones y se proponen trabajos futuros en este campo respectivamente.

E. González Fernández, A. L. Sandoval Orozco and L. J. García Villalba son miembros del Grupo de Análisis, Seguridad y Sistemas (GASS), Departamento de Ingeniería del Software e Inteligencia Artificial (DISIA), Facultad de Informática, Despacho 431, Universidad Complutense de Madrid (UCM), Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid e-mail: edggonza@ucm.es, {asandoval, javiergv}@fdi.ucm.es.

https://doi.org/10.12804/si9789587844337.06

II. Técnicas de Manipulación en Vídeos

II-A. Inter-Fotograma

Un vídeo digital se compone de una secuencia de imágenes llamadas fotogramas. Las manipulaciones de tipo inter-fotograma (inter-frame) se centran en la modificación de la correlación temporal entre ellos. Para modificar la correlación temporal del vídeo es posible insertar, duplicar, intercambiar o eliminar cualquiera de los fotogramas que lo conforman (Figura 2).

Otra forma de manipular un vídeo inter-fotograma es mediante el empalme de dos o más vídeos, es decir, interpolando fotogramas de ambos para generar uno nuevo. Además, es posible que los vídeos originales no compartan los mismos fotogramas por segundo (fps), por lo que será necesario también manipular esta característica para ajustar los fps de uno al otro.

El principal objetivo de esta manipulación es el de eliminar de la escena grabada un evento indeseado. También es posible incriminar en la escena a otros objetos con la adición de un fotograma externo. Si se toma como ejemplo la secuencia de las imágenes de vigilancia de una cámara de tráfico, como las de la Figura 3, es sencillo hacer que el vehículo blanco de la Figura 3d desaparezca de la escena eliminando ese fotograma.

En general, el ojo humano no puede detectar diferencias entre el vídeo original y el vídeo con manipulación inter-fotograma pero las operaciones de procesamiento de la manipulación dejan una huella en la información del contenido.

II-B. Intra-Fotograma

La manipulación intra-fotograma se centra en la alteración de cada fotograma individualmente. Estas manipulaciones pueden clasificarse en:

- **Manipulación a nivel de píxel**: La cual consiste en tratar al fotograma como una imagen individual y aplicar técnicas de manipulación en imágenes como las vistas en la sección anterior, por ejemplo, copia-pega o empalmes.
- **Manipulación a nivel de fotograma**: Mediante la cual se cambia de tamaño o se recortan las extremidades de un fotograma con el objetivo de ocultar cierto contenido del vídeo que se ubique en los bordes del fotograma. Por ejemplo marcas de Hora y lugar de grabación.

A diferencia del ejemplo expuesto con las técnicas inter-fotograma, si se tiene como objetivo ocultar el paso de un vehículo de la cámara de vigilancia de la Figura 3 con técnicas intra-fotograma, en lugar de eliminar el fotograma en el que aparece, se podría hacer desaparecer con técnicas de copia-pega o incluso se podría re-escalar el fotograma y recortar la zona en la que aparece.

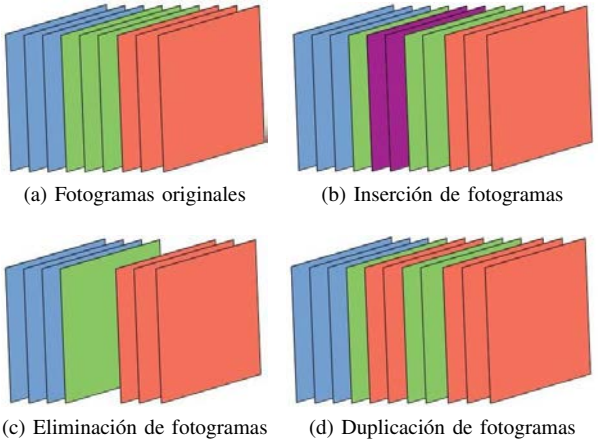
III. Trabajos previos

La Figura 4 Muestra una clasificación de las técnicas de detección de manipulaciones en vídeos digitales.

III-A. Detección de Manipulación Inter-Fotograma

Los dispositivos introducen un ruido en cada fotograma cuando graban un vídeo. Dado que este ruido sigue un patrón particular en una secuencia de fotogramas consecutivos, es posible que, investigando estos rastros, se detecten cambios entre los fotogramas.

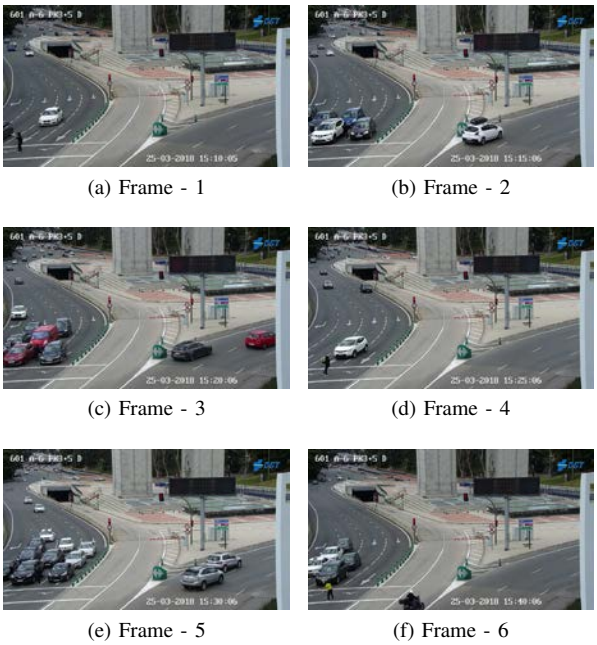
En [3] los autores utilizaron la varianza entre el ruido promedio de los fotogramas y uno en particular. Los fotogramas



(a) Fotogramas originales (b) Inserción de fotogramas

(c) Eliminación de fotogramas (d) Duplicación de fotogramas

Figura 2: Ejemplo de Manipulación Inter-Fotograma.



(a) Frame - 1 (b) Frame - 2

(c) Frame - 3 (d) Frame - 4

(e) Frame - 5 (f) Frame - 6

Figura 3: Ejemplo de fotogramas de una cámara de vigilancia de la Dirección General de Tráfico (DGT).

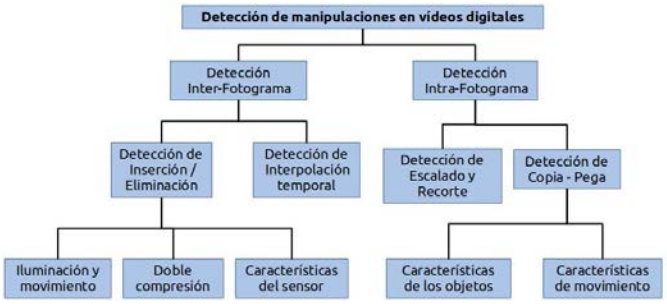


Figura 4: Esquema de Detecciones de Manipulación en Vídeos.

con varianzas más altas serían marcados como inserciones. No se demostró su eficacia sobre vídeos comprimidos, y además, su eficacia fue probada sobre vídeos auto-grabados y no es suficiente para determinar su aplicabilidad.

Los autores de [4] propusieron un algoritmo de movimiento adaptativo que fue capaz de detectar y localizar falsificaciones en vídeos entrelazados y descentrelazados. Se basaban en la detección de las perturbaciones en la correlación para los entrelazados y en los disturbios de movimiento entre fotogramas para los descentrelazados. Este método, no obstante, resultaba ineficaz para vídeos de baja calidad.

En [5] utilizan el concepto del Patrón de Ruido del Sensor (SPN) de la cámara para determinar si todos los fotogramas del vídeo habían sido grabados con el mismo dispositivo. Los resultados obtenidos indicaron que el algoritmo era fiable para vídeos no comprimidos, pero el rendimiento se deterioraba para vídeos comprimidos.

Otra propuesta basada en la cámara es presentada en [6], que proporcionó autenticación a nivel de píxel para todos los fotogramas del vídeo. Se basaba en las inconsistencias en los fotones del ruido de disparo que introduce la cámara durante el proceso de adquisición. También lograrían encontrar regiones sospechosas en los fotogramas.

Otra forma de manipular un vídeo es mediante un corte temporal, intercalando fotogramas de dos vídeos diferentes. Cuando se intercalan dos fotogramas hay que tener en cuenta que es necesario sincronizar sus velocidades (frame-rate).

El método sugerido en [7] se basa en la propiedad de interpolación compensada por movimiento ya que deja huellas detectables en los fotogramas. Los autores pudieron sugerir un sistema que funcionaba para vídeos no comprimidos y ligeramente comprimidos (por ejemplo, H.264, o vídeos de transmisión de televisión) y lograba resultados prometedores, incluso cuando se usaba sólo en un subconjunto de fotogramas. Además, el sistema funcionaba bien en ventanas espaciales de pequeño tamaño, lo que permitió que este detector se usara como una posible herramienta para detectar ataques de falsificación de copiar y pegar. Sin embargo, el número de cuadros interpolados observados tenía que ser lo suficientemente grande para que el sistema detectara las falsificaciones con éxito.

En [8] se detecta la conversión de velocidad ascendente de fotogramas basándose en la intensidad de los bordes. Utilizan un umbral determinado para distinguir las zonas originales de las convertidas al alza, y en base a ello, estiman la velocidad teórica de los fotogramas originales. Para un total de 300 secuencias de prueba, consiguieron un promedio de detección de 95 %.

Los autores en [9] desarrollaron un método de detección ciego basado en el análisis a nivel de fotograma de una característica llamada "variación media de la textura" (ATV). Cada curva ATV generada se procesaba en el vídeo candidato como evidencia de la conversión de velocidad ascendente. Esta técnica podría localizar la posición de la interpolación de los fotogramas y ayudar a estimar su velocidad original.

III-B. Detección de Manipulación Intra-Fotograma

Las técnicas de detección de copia-pega proceden buscando similitudes entre regiones de fotogramas sucesivos o dentro del mismo fotograma.

Para detectar estas falsificaciones, los autores en [4] calcularon coeficientes de correlación espacial y temporal para identificar y localizar semejanza entre partes separadas del vídeo. Este método, obtuvo muy buenos resultados para vídeos con compresión MPEG debido a que los artefactos de compresión son más pronunciados en presencia de movimiento en el vídeo.

Otra técnica propuesta en [10] y [11] se basó en la hipótesis de que los atributos de correlación de sub-bloques de píxeles intra e inter fotograma están obligados a ser desorganizados debido a alteraciones como doble compresión, retoque o remuestreo. Los autores extrajeron los residuos de ruido y la cuantificación de características de fotogramas adyacentes para luego realizar un análisis de correlación usando el análisis de correlación canónico, análisis factorial intermodal, y análisis semántico latente. Tales perturbaciones ayudaron a la técnica para diferenciar las huellas de un vídeo original de las de uno manipulado.

En [12], se propone detectar alteraciones con la conversión del vídeo a una secuencia de fotogramas, seguido de un proceso de emparejamiento de bloques dentro de la región sospechosa. Al trabajar en una parte del fotograma en lugar de todo el fotograma, la técnica es capaz de mantener un buen equilibrio entre rendimiento y complejidad.

El método de detección y localización de falsificación de [13] era similar en funcionalidad a [14] pero de manera completamente automática. Es un algoritmo de dos pasos, en el que primero se detectan manipulaciones a nivel de fotograma y se analiza el vídeo residual que se obtiene al restar píxeles que ocupan la misma posición espacial en fotogramas consecutivos. Entonces, para detectar el contenido duplicado, los autores ponen en relación los bloques 3D de los fotogramas. Así, la presencia de alta correlación indica la ubicación del contenido idéntico.

En [15], los autores propusieron un enfoque para detectar y localizar falsificaciones a nivel de región en vídeos. El método detecta irregularidades en la coherencia espacio-temporal entre fotogramas consecutivos. El vídeo primero se divide en conjuntos de fotogramas y luego se calcula la coherencia entre cada una de estos conjuntos. Dicho así, los conjuntos con coherencia antinaturalmente alta o coherencia anormalmente baja se clasificarían como fotogramas manipulados.

Una técnica de localización y detección de eliminación de objetos es la que se nos presenta en [16]. Se utilizó aquí Scale-Invariant Feature Transform (SIFT) junto con la coincidencia k-NN y correlación cruzada ruido-residuo para detectar falsificaciones copia-pega. Aunque la técnica funciona bien para la prueba vídeos, esta sufre una degradación significativa a medida que aumenta la resistencia a la compresión.

Otra forma de manipular el contenido del vídeo es ampliando un fotograma y después eliminando el evento incriminatorio recortando la parte externa de éste. Es importante saber que cuando se hace un recorte se produce un remuestreo para mantener una resolución constante en todos los fotogramas del vídeo.

En [17] los autores observaron que el remuestreo introduce ciertas correlaciones estadísticas sobre el contenido dado. Explotaron el SPN como característica forense y analizaron las variaciones en las propiedades de correlación de referencia SPN y el de re-escalado. Este método es bastante firme en

cualquier tipo de vídeo, pero también resulta excesivamente dependiente de una gran cantidad de parámetros y umbrales dependientes del contenido, lo cual requiere un ajuste empírico extremadamente cuidadoso.

III-C. Detección de Doble Compresión

La recompresión o doble compresión es, una consecuencia inevitable de la falsificación, y su detección podría ayudar a detectar la presencia manipulaciones.

Los primeros pasos en esta dirección se pueden atribuir a los autores de [18]. Su algoritmo se basaba en la suposición simple de que cuando se manipulaba un vídeo MPEG, se producían dos compresiones: primero, cuando se creaba el vídeo y, segundo, cuando se volvían a guardar después de dicha alteración. También explotaron el hecho de que dentro de un Grupo de Imágenes (GOP), los fotogramas muestran una gran correlación entre ellos, de manera que al agregar o eliminar un fotograma en un GOP aumenta el error de estimación de movimiento, lo que también da como resultado picos periódicos detectables.

En [19], los autores presentaron una técnica para detectar la cuantización doble, que resultó de la recompresión de un vídeo comprimido Moving Picture Experts Group (MPEG) o de la combinación de vídeos de características diferentes. La técnica podría detectar una manipulación si los coeficientes Transformada Discreta del Coseno (DCT) de los fotogramas del vídeo se sometieron a doble compresión en cualquier punto. Los resultados empíricos indicaron que la tasa de detección fue altamente dependiente de la relación de la primera y la segunda escala de cuantificación. Evidentemente, la técnica fue efectiva siempre que el segundo factor de calidad de compresión fuera más alto que el primero.

En [20], las falsificaciones en vídeos codificados en MPEG-2 se detectaron mediante el examen de la distribución del coeficiente DCT. Este algoritmo se basó en la observación de que el histograma de coeficientes de DCT cuantificados de un vídeo que había experimentado una doble compresión exhibía un patrón convexo. A diferencia de [19], que dependía en gran medida de las escalas de cuantificación, los autores en este caso sugirieron controlar la tasa de bits de salida, lo que hizo que este algoritmo se adaptara a las necesidades de diferentes tipos de sistemas de codificación de vídeo pero no pudo localizar la falsificación en el vídeo. Tampoco pudo funcionar bien para vídeos de cámara lenta.

El trabajo en [21] también se centró en la detección de alteraciones basadas en fotogramas al detectar la compresión doble en vídeos MPEG-2. En lugar de basar el proceso de detección de agregación/eliminación de trama en las características temporales, los autores sugirieron utilizar las características de frecuencia. Se observó que cuando se vuelve a comprimir un vídeo MPEG después de agregar/eliminar el fotograma, se pierden algunos componentes de alta frecuencia en los fotogramas recomprimidos debido a la descincronización de los GOP y la cuantificación no lineal realizada en el proceso de codificación. Estas variaciones no solo ayudan a detectar la falsificación sino también a localizarla.

Otra técnica de detección de falsificación basada en doble compresión MPEG es la propuesta en [22], donde las anormalidades en los patrones de coeficientes DCT son tratadas como indicativo de inserción/eliminación de fotogramas. Los autores extrajeron características de los GOP, que luego son utilizados por una Máquina de Soporte Vectorial (SVM) para determinar la velocidad de bits original del vídeo doblemente comprimido dado, y se observa que el rendimiento de detección de la técnica era relativamente inferior para los vídeos con menor tasa de bits, porque una escala de cuantificación mayor requiere un proceso de cuantificación más robusto, que la técnica no estaba preparada para manejar.

En el mismo año, se pre-planteó una técnica similar en [23] aunque con una novedad: su capacidad para detectar vídeos transcodificados, es decir, vídeos que habían sido doblemente comprimidos utilizando dos estándares de compresión diferentes. Los autores observaron además que después de que un vídeo MPEG-2 se transformara en vídeo MPEG-4, las trazas de compresión MEPG-2 anteriores, estos generan nuevas periodicidades que se observaron claramente en los histogramas de los coeficientes DCT reconstruidos. Los autores presentaron los resultados en forma de curvas de características operativas del receptor y declararon que se habían obtenido resultados perfectos en caso de bajas tasas de bits. Estas curvas también demostraron que a medida que aumentaba la velocidad de bits de salida objetivo, el rendimiento de detección disminuía. También asumió que la transcodificación siempre sugería manipulación.

Por otra parte, en [24] se propone detectar la codificación doble incluso si el conjunto de fotogramas principales hubiera sido eliminado. Este método tiene la ventaja adicional de poder ubicar efectivamente la falsificación, además de resultar adecuado también para vídeos codificados H.264, a diferencia de [23] que funcionaba solo para vídeos MPEG. La metodología modificada también fue capaz de estimar el número de fotogramas borrados.

Los autores en [25] declararon que las compresiones múltiples eran un tema poco explorado y que era arriesgado hacer suposiciones con respecto a la autenticidad del contenido digital simplemente sobre la base de la presencia de doble compresión. Su afirmación fue respaldada por el simple hecho de que el contenido digital disponible en Internet, generalmente, sufre más de una compresión.

En [26] utilizan las estadísticas de Markov para detectar doble compresión. Se basan en que la cuantización doble con diferentes parámetros inevitablemente introducirá errores de redondeo, dejando artefactos detectables. El proceso aleatorio de Markov podría capturar dichos artefactos para la detección.

En [27] se basan en características estadísticas de los macrobloques de los P-fotogramas. Proponen detectar la compresión doble de MPEG con el mismo QS. La extracción de características se produce durante la compresión repetida del vídeo en el mismo factor de calidad.

En [28] analizan la degradación que se produce durante una recompresión encontrando que la variación de las características de un vídeo tienden a estabilizarse tras múltiples recompresiones.

En [29] estudian los efectos de la recompresión en los fotogramas predictivos para generar un vector de características con el cual detectar doble compresión a nivel GOP.

Para un vídeo ordinario (descargado de Internet o grabado con ciertos dispositivos móviles), la presencia de signos de doble compresión puede no ser sospechosa pero tampoco

debe considerarse inocua. Si se supone que un vídeo ha sido inalterado, la doble compresión no debería aparecer en dicho vídeo. Por otro lado, si un vídeo dado muestra signos de doble compresión, indicaría la presencia de algún tipo de modificación no autorizada.

Por lo tanto, la presencia de signos de doble compresión serviría como primera, y posiblemente, más importante evidencia de alteración en vídeos.

IV. Detección de Doble Compresión en Vídeos

IV-A. Conceptos Generales

Para una correcta comprensión de la explicación del algoritmo de detección de doble compresión primero es importante conocer las características del formato de vídeo H.264 y de las herramientas FFMPEG y LIBSVM, utilizadas ambas por dicho algoritmo.

IV-A1. El Formato H.264/MPEG4: Este estándar de codificación fue desarrollado con el objetivo de mejorar la calidad de la imagen, mejorar la eficiencia de codificación y mejorar la robustez de errores en comparación con normas anteriores como MPEG-2, H.263, etc...

El diseño de codificación de este estándar está basado en bloques, es decir, cada fotograma codificado se representa como una unidad de bloques llamados macrobloques. El algoritmo de codificación es el conjunto que se forma al predecir fotogramas por medio de esos macrobloques para explotar dependencias estadísticas temporales, y al transformar la predicción residual para explotar las dependencias estadísticas espaciales.

Algunas de las características más destacadas del diseño, y que además permiten una mayor eficacia de codificación, incluyen las siguientes mejoras en la capacidad para predecir los valores del contenido del fotograma que se va a codificar:

- Por una parte, el tamaño de bloque de compensación de movimiento variable: Este estándar admite más flexibilidad en la selección de tamaños y formas de los bloques, con un tamaño de bloque mínimo de 4:4.
- Por otra, una referencia múltiple para la compensación de movimiento de un fotograma: Los fotogramas con codificación predictiva, llamados P-fotogramas, en estándares anteriores usan sólo el fotograma previo para predecir los valores del fotograma entrante. Este modelo extiende la codificación eficiente al permitir que un codificador seleccione, para fines de compensación de movimiento, entre un mayor número de fotogramas que se han decodificado y almacenado en el decodificador.

El ojo humano percibe el contenido de una escena en términos de información de brillo y color por separado, y con mayor sensibilidad a la de brillo que la de color. El formato H.264 separa una representación de color en tres componentes llamados Y, Cb y Cr. El componente "Y" representa el brillo, mientras que los dos componentes de color "Cb" y "Cr" representan la medida en la que el color se desvía del gris hacia azul y rojo, respectivamente. Debido a que el sistema visual humano es más sensible al brillo que al color, H.264 utiliza una estructura de muestreo en la que el componente cromático tiene un cuarto del número de muestras que el componente lumínico.

Cada fotograma se divide en macrobloques de tamaño fijo que cubren un área rectangular de 16:16 muestras de la componente de brillo y 8:8 muestras de cada uno de los dos componentes de color. Los macrobloques son los componentes básicos para el que se especifica el proceso de decodificación. Todas las muestras de luminancia y croma de un macrobloque se predicen espacial y temporalmente. La señal de vídeo de entrada se divide en macrobloques, cuya asociación se realiza en base a los tipos de fotogramas a los que pertenecen, y luego se procesa cada macrobloque de cada tipo. Es posible un procesamiento en paralelo eficiente [30].

En la compresión .H264 se puede seleccionar la predicción de los macrobloques de manera individual, en lugar de ser los mismos para todo el fotograma, como se observa en la Figura 5):

- **Fotogramas - I**: Todos los macrobloques del fotograma son codificados usando intra-predicción, es decir, no utiliza la información codificada de otros fotogramas.
- **Fotogramas - P**: Además de la codificación de intra-predicción. También se pueden codificar usando inter-predicción con como máximo una señal de predicción de compensación de movimiento por bloque de predicción, es decir, su información proviene del fotograma previo.
- **Fotogramas - B**: Además de los tipos de codificación disponibles en un fotograma P, algunos macrobloques del fotograma B también se pueden codificar utilizando la inter-predicción con dos señales de predicción de compensación de movimiento por bloque de predicción, es decir, su información proviene del fotograma previo y del siguiente.

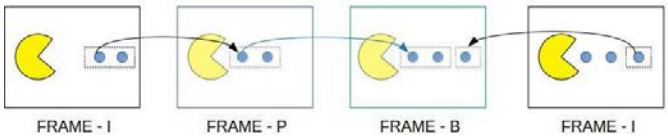


Figura 5: Secuencia de predicción de fotograma.

La señal de predicción para cada macrobloque de N×M codificado predictivamente se obtiene desplazando un área de la imagen de referencia correspondiente, que se especifica mediante un vector de movimiento de traslación y un índice de referencia de imagen. Los componentes del vector de movimiento se codifican de forma diferencial usando predicción mediana o direccional de bloques vecinos. Ninguna predicción del componente del vector de movimiento (o cualquier otra forma de predicción) tiene lugar a lo largo de los límites del fotograma. En la Figura 6 se muestra un ejemplo de cómo actúa el vector de movimiento en la predicción [30].

IV-A2. La Herramienta FFMPEG: FFMPEG es una plataforma de software libre multimedia capaz de decodificar, codificar, transcodificar, transmitir, filtrar y reproducir la mayoría de formatos de audio y vídeo. Está desarrollado en GNU/Linux pero también compila y ejecuta en la mayoría de sistemas operativos, entornos de desarrollo, arquitecturas y configuraciones. Tiene una licencia GNU LGPL, la cual garantiza una cierta libertad a la hora de compartir y modificar el software, asegurando que el software es libre para todos sus usuarios [31]. Es posible utilizar FFMPEG para analizar los macrobloques y vectores de movimiento de cualquier archivo

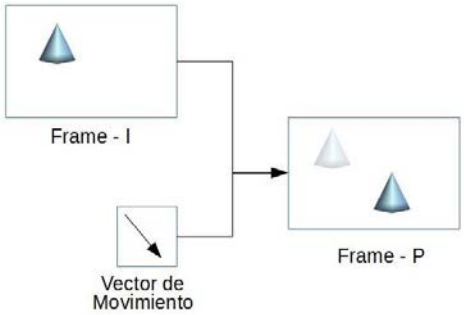


Figura 6: Secuencia de predicción de un fotograma.

de vídeo MP4. En la Figura 7 se puede ver un ejemplo de un fotograma con los vectores de movimiento analizados impresos en forma de flechas [32].

Figura 7: Análisis de los vectores de movimiento.

IV-A3. La Máquina de Soporte Vectorial: Las SVM son técnicas supervisadas de aprendizaje automático, muy útiles para la resolución de problemas de reconocimiento de patrones y para el análisis de regresión. A partir de un conjunto de muestras la SVM construye un modelo que se utiliza para predecir la clase a la que pertenece una nueva muestra. El objetivo de la SVM es encontrar el mejor hiper-plano que divida los datos de todas las muestras del entrenamiento en dos o más clases bien diferenciadas, es decir, determinar el hiper-plano con la máxima distancia con el punto de cada clase que esta más cercano a éste. En la Figura 8 el hiper-plano H2 sería óptimo.

No obstante, cuando se utiliza SVM surgen dos problemáticas:

- Los universos que se estudian utilizan más de dos dimensiones y no tienen una representación lineal. Este problema se soluciona con la representación por funciones kernel, que proyectan la información a un espacio de características multidimensional mediante un mapeo no lineal.
- Seleccionar los parámetros apropiados del kernel. Hay dos parámetros en la función Función de Base Radial (RBF) del kernel (C y γ). Para encontrar los mejores parámetros de clasificación de prueba y entrenamiento se utiliza el método de optimización de parámetros.

IV-B. Algoritmo de Detección de Doble Compresión en Vídeos

Este algoritmo se utilizará con el fin forense de determinar si un vídeo ha sufrido más de una compresión, evidencia primera de que ese vídeo haya podido sufrir cualquier tipo de manipulación. La detección de recompresiones está basada

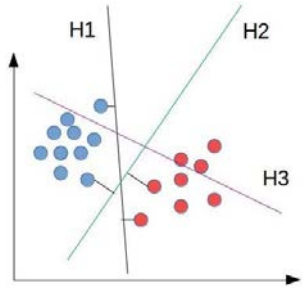


Figura 8: Muestras e hiperplanos.

en el estudio de las características estadísticas del Modo de Macrobloque (MBM) [27].

El MBM es una característica que consta del tipo de macrobloque y vector de movimiento. Para extraer esta característica, un vídeo es recomprimido repetidamente en la misma escala de calidad para luego calcular el número de MBM diferentes entre dos compresiones secuenciales. Finalmente, estas estadísticas extraídas son utilizadas por la SVM para determinar si el vídeo es original o si ha sido recomprimido.

Este método viene inspirado de la convergencia de los coeficientes Joint Photographic Experts Group (JPEG) cuando se recomprime una imagen. Ambos métodos para cada recompresión varían en la forma de la Figura 9.

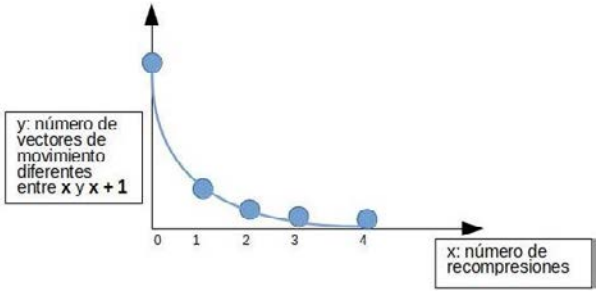


Figura 9: Número de MBM diferentes entre recompresiones.

A parte de los 3 tipos de fotogramas (I, P, B) que tiene el estándar MPEG4, también existen 3 tipos de macrobloques:

- I-MB: Macrobloques con intra-codificación.
- P-MB: Macrobloques con inter-codificación.
- S-MB: Macrobloques saltados.

Un MBM está compuesto de las dos propiedades de la siguiente manera:

$$[MBM(M) = M_{type}, M_{mv}]$$
$$\left.\begin{aligned} M_{type} \in \{I-MB, P-MB, S-MB\} \\ M_{mv} = \{(u,v)|u, vcZ\} \end{aligned}\right\}$$

M es el macrobloque, M_{TYPE} el tipo del macrobloque M, y M-MV el vector de movimiento del macrobloque M.

Dos macrobloques se consideran que tienen el mismo MBM si y sólo si tienen el mismo M-TYPE y M-MV. Hay que tener en cuenta que cuando el M-TYPE es un I-MB su vector de movimiento es {0,0}, es decir, sólo es necesario evaluar los MBM diferentes de los P-fotogramas y por tanto sólo hay que comparar el Vector de Movimiento (VM).

Para una secuencia de compresiones sobre un vídeo, si el macrobloque del mismo fotograma y misma posición de

la compresión (n) y de la compresión (n + 1) tienen la característica MBM igual, se considera que ese macrobloque es estable. De otro modo se considera inestable. Ver Figura 10.

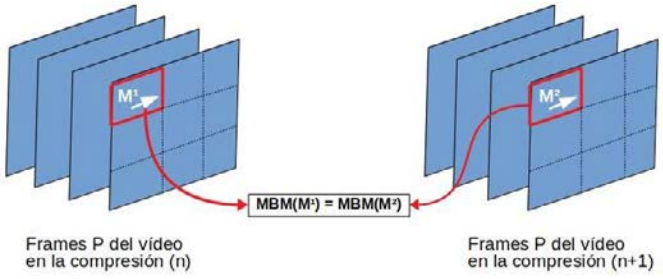


Figura 10: MBM estable.

A continuación se explica más detalladamente el algoritmo:

- Entrada: Vídeo o Vídeos en formato H.264-MP4
- Salida: Vector de características extraídas de los vídeos

Para un vídeo dado, este algoritmo devuelve un vector de características. En este caso el algoritmo se ha desarrollado para extraer tres características con el objetivo de que la SVM las pueda clasificar hasta la triple compresión. También es posible extraer sólo dos y así discernir sólo entre vídeos originales y doblemente comprimidos, o incluso, para detectar más allá de la triple compresión, aunque, a partir de ésta, la tasa de confianza de la máquina de soporte vectorial es demasiado bajo.

Vector de características:

- Número promedio de macrobloques inestables por P-fotograma encontrados entre el vídeo de entrada y su recompresión.
- Número promedio de macrobloques inestables por P-fotograma encontrados entre el vídeo recomprimido y su re-recompresión.
- Número promedio de macrobloques inestables por P-fotograma encontrados entre el vídeo de re-recomprimido y su re-re-recompresión.

En primer lugar, se evalúa el MBM de los macrobloques. Para ello, con apoyo de la herramienta FFMPEG, se extraen los VM de los P-fotogramas. Es importante tener en cuenta que para poder extraer información el vídeo debe estar primero en un formato crudo (.YUV). Estos vectores de movimiento contienen la información del fotograma al que pertenecen, la posición del macrobloque y la posición del eje X, Y de origen y destino del vector. Todos los vectores son almacenados en un archivo de texto para su posterior procesamiento.

Una vez realizado el proceso de extracción de VM, se debe recomprimir el vídeo de entrada con el formato H264-MP4. Esta recompresión tiene que tener la misma escala de calidad que el original, es decir la recompresión se realiza utilizando las mismas características del vídeo de entrada (qs, ancho, alto, rate, etc).

El vídeo recomprimido es almacenado para volver a ejecutar los pasos anteriores: Extracción de vectores de movimiento y, a partir de ahí, una nueva recompresión. Estas acciones pueden realizarse el número de veces que se quiera según el nivel de precisión de detección que se pretenda alcanzar. Para este caso se han realizado hasta tres recompresiones.

Cuando se alcance el número de recompresiones indicado, se va a disponer de tres archivos de texto que contienen los vectores de movimiento del vídeo de entrada y de sus recompresiones posteriores. Con toda esta información se puede proceder a calcular el número promedio de macrobloques inestables por P-fotograma (C).

La siguiente fórmula muestra cómo se realiza ese cálculo:

$$C_n = \frac{1}{N} \sum_{i,x,y} I(M_n(i,x,y), M_{n+1}(i,x,y))$$

N es el número total de P-fotogramas y M muestra el macrbloque de la recompresión enésima localizado en (x, y) del P-fotograma i-ésimo.

I se define como:

$$\left.\begin{array}{l} 1 \rightarrow MBM(M_1) \neq MBM(M_2) \\ 0 \rightarrow MBM(M_1) = MBM(M_2) \end{array}\right\} \quad I(M_1, M_2)$$

Para realizar el cálculo de M(i,x,y), los vectores de movimiento contenidos en los ficheros de texto son tratados en forma de matrices N×M dónde N es VM y M el P-fotograma al que pertenece con el fin de facilitar la tarea de la comparación.

Una vez encontrado I, es decir, el número de MBM diferentes de los ficheros de texto correspondientes a la compresión (n) y a la compresión (n + 1), se divide entre el número de P-fotogramas. El resultado se almacena en la posición (n) del vector de características del vídeo en cuestión.

Cuando el vector de características está completo se formatea para que la máquina de soporte vectorial lo pueda utilizar para realizar las tareas de clasificación. El diagrama de flujo de este algoritmo se muestra en la Figura 11.

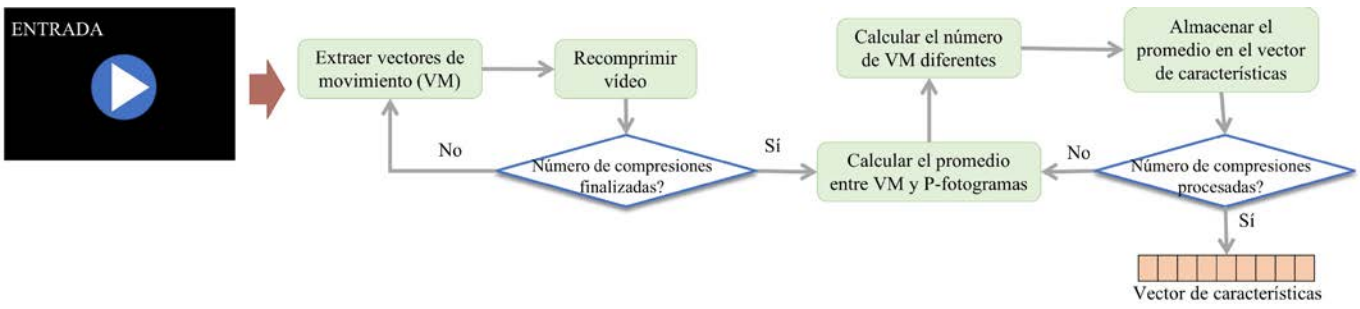


Figura 11: Diagrama del Algoritmo de Detección de Doble Compresión.

V. Evaluación del Algoritmo Propuesto

Para evaluar el algoritmo propuesto se ha utilizado un dataset que contiene vídeos digitales procedentes de distintos modelos de dispositivos móviles con tamaños de resolución diferentes. Se han seleccionado vídeos en formato .MP4 con la resoluciones más comunes: 720x480, 720x1280, 1920x1080 y 3840x2160 (4K). La mayor parte de los vídeos seleccionados se han utilizado para el entrenamiento de la máquina de soporte vectorial con el fin de tener una base de conocimiento más amplia. El resto se han utilizado para la predicción. La Tabla I muestra un resumen de las características del dataset utilizado para el entrenamiento y predicción.

Tabla I: Dataset utilizado en la evaluación

Resolución	Formato	No. vídeos pruebas	No. vídeos analizados
720x480	MP4	20	20
720x1280	MP4	40	35
1920x1080	MP4	40	35
4K	MP4	20	17

Las características del equipo en el cual se han realizado los experimentos se presentan en la Tabla II. Es un factor importante a tener en cuenta ya que los tiempos de ejecución de las diferentes pruebas varían según los recursos computaciones disponibles.

Tabla II: Características del equipo de experimentación

Recursos	Características
Sistema operativo	Ubuntu 18.04
Memoria	4 GB
Procesador	Intel® Core™ 2 Quad CPU Q8200 @ 2.33GHz x 4
Gráficos	NV96
Tipo de SO	64 bits
Disco	100 GB

Para la realización de los experimentos se ha utilizado un componente llamado **MPEGflow** [33] bajo licencia Massachusetts Institute of Technology (MIT) que se apoya en FFMPEG para facilitar las tareas de extracción de los vectores de movimiento de los fotogramas de un vídeo y de almacenamiento en un fichero de texto.

También se ha utilizado el módulo LIBSVM, un software integrado que desempeña las funciones de una máquina de soporte vectorial, (C-SVC, nu-SVC), regresión (epsilon-SVR, nu-SVR) y estimación de distribución (SVM de una clase) compatible con la clasificación de clases múltiples.

A lo largo de esta sección se muestran todos los experimentos realizados para evaluar la efectividad del algoritmo de detección de manipulaciones basados en entrenamiento. Con ellos se pretende comprobar la variación de la precisión al aplicar el algoritmo sobre distintas resoluciones. Se estudia la capacidad de detectar si el vídeo es original, ha tenido doble compresión o triple compresión.

En primer lugar, se creó la base de conocimiento para el entrenamiento de la máquina de soporte vectorial. Los vídeos seleccionados como dataset de entrenamiento para cada resolución se utilizan como entrada del algoritmo de detección de recompresiones. Una vez entrenada la máquina con los vectores de características generados por el algoritmo, se puede comenzar la predicción. Esta prueba consiste en extraer las características de los vídeos a testear para que la máquina SVM, una vez entrenada, los clasifique en función de sus recompresiones.

En el primer grupo de experimentos tiene como objetivo detectar si un vídeo es original o ha tenido al menos una recompresión. En este caso, el algoritmo extrae sólo dos características de los vídeos del dataset. Por tanto, se entrena la máquina de soporte vectorial con vídeos originales y vídeos recomprimidos, el modelo resultante consta de dos clases para discernir si el vídeo es original o no. Se ejecutó un experimento para los vídeos con las diferentes resoluciones del dataset y adicionalmente se hizo un experimento mezclando todas las resoluciones para evaluar la tolerancia del algoritmo al tamaño del vídeo. Para cada uno de estos experimentos se han tomado los vectores de características escalados y sin escalar. Los resultados obtenidos para cada uno de los experimentos con diferentes resoluciones se presentan en la Tabla III.

Tabla III: Tasa de acierto para la detección de vídeos originales y manipulados con recompresión.

Tipos de datos	Resolución				
	720x480	720x1280	1920x1080	4K	MIX
Escalados	90 %	94,28 %	97,14 %	100 %	91,59 %
No escalados	90 %	100 %	100 %	100 %	92,52 %

Como se observa en la Tabla III, el algoritmo de detección propuesto presenta una tasa de acierto superior al 90 % incluso cuando la resolución es baja (720x480). En todas las resoluciones los resultados son superiores cuando los datos del vector de características no son escalados. Incluso en el caso de que el sistema sea entrenado con vídeos de diferentes resoluciones.

Las Tablas IV y V muestran las matrices de confusión resultantes para cada una de las resoluciones analizadas con los vectores de características escalados y sin escalar, respectivamente.

Tabla IV: Matriz de confusión por resolución con datos escalados.

Resolución	Matriz de Confusión				Acierto Prom.
	Clases	Original	Doble Comp.	Total Vídeos	
720x480	Original	**10**	0	10	90 %
	Doble Comp.	2	**8**	10	
720x1280	Original	**20**	0	20	94,28 %
	Doble Comp.	2	**13**	15	
1920x1080	Original	**19**	1	20	97,14 %
	Doble Comp.	0	**15**	15	
4K	Original	**9**	0	9	100 %
	Doble Comp.	0	**8**	8	
MIX	Original	**44**	9	53	91,59 %
	Doble Comp.	0	**54**	54	

Tabla V: Matriz de confusión por resolución con datos no escalados.

Resolución	Matriz de Confusión				Acierto Prom.
	Clases	Original	Doble Comp.	Total Vídeos	
720x480	Original	**9**	1	10	90 %
	Doble Comp.	1	**9**	10	
720x1280	Original	**20**	0	20	100 %
	Doble Comp.	2	**0**	15	
1920x1080	Original	**20**	1	20	100 %
	Doble Comp.	0	**15**	15	
4K	Original	**9**	0	9	100 %
	Doble Comp.	0	**8**	8	
MIX	Original	**46**	7	53	92,52 %
	Doble Comp.	1	**53**	54	

En el segundo grupo de experimentos se extrae una característica más que en los experimentos anteriores con el

objetivo de determinar si los vídeos analizados han sido recomprimidos más de una vez, y en tal caso, saber sí ha sido recomprimido una o dos veces adicionales. Como en el caso anterior, para los experimentos se han tomado los vectores de características escalados y sin escalar. Los resultados obtenidos para cada uno de los experimentos con diferentes resoluciones se presentan en la Tabla VI.

Tabla VI: Tasa de acierto para la detección de vídeos con más de una recompresión.

Tipos de datos	Resolución				
	720x480	720x1280	1920x1080	4K	MIX
Escalados	83,33 %	70 %	68 %	80 %	49,67 %
No escalados	66,67 %	88 %	70 %	60 %	63,23 %

Como se observa en la Tabla VI, el algoritmo de detección propuesto presenta la mejor tasa de acierto (88 %) cuando el vídeo tiene una resolución de 720x1280 y los datos no escalados. Las Tablas VII y VIII muestran las matrices de confusión resultantes para cada una de las resoluciones analizadas con los vectores de características escalados y sin escalar, respectivamente.

Tabla VII: Matriz de confusión por resolución con tres clases con datos escalados.

Res.	Matriz de Confusión					Acierto Prom.
	Clases	Original	Doble	Triple	Total Vídeos	
720x480	Original	**10**	0	0	10	83,33 %
	Doble	0	**5**	5	10	
	Triple	0	0	**10**	10	
720x1280	Original	**20**	0	0	20	70 %
	Doble	6	**9**	0	15	
	Triple	1	8	**6**	15	
1920x1080	Original	**20**	0	0	20	68 %
	Doble	0	**0**	15	15	
	Triple	0	1	**14**	15	
4K	Original	**9**	0	0	9	80 %
	Doble	0	**6**	2	8	
	Triple	0	3	**5**	8	
MIX	Original	**25**	2	20	47	49,67 %
	Doble	0	**0**	54	54	
	Triple	1	1	**52**	54	

Tabla VIII: Matriz de confusión por resolución con tres clases con datos sin escalar.

Res.	Matriz de Confusión					Acierto Prom.
	Clases	Original	Doble	Triple	Total Vídeos	
720x480	Original	**10**	0	0	10	66,67 %
	Doble	1	**5**	4	10	
	Triple	0	5	**5**	10	
720x1280	Original	**19**	1	0	20	88 %
	Doble	0	**12**	3	15	
	Triple	0	2	**13**	15	
1920x1080	Original	**20**	0	0	20	70 %
	Doble	0	**8**	7	15	
	Triple	0	8	**7**	15	
4K	Original	**7**	2	0	9	60 %
	Doble	0	**0**	8	8	
	Triple	0	0	**8**	8	
MIX	Original	**44**	3	0	47	63,23 %
	Doble	6	**29**	19	54	
	Triple	7	22	**25**	54	

La Tabla IX muestra el tiempo de ejecución del algoritmo para cada una de las resoluciones.

La Tabla X muestra una comparativa entre el método propuesto y la investigación más relacionada en la literatura. En la tabla se observa que el resultado obtenido con el método propuesto para detectar doble compresión es superior al del trabajo comparado.

Tabla IX: Rendimiento del algoritmo propuesto.

Compresión detectada	720x480	720x1280	1920x1080	4K
Doble	00:00:07.03s	00:00:24.49s	00:01:16.32s	00:05:44.13s
Triple	00:00:16.23s	00:00:55.11s	00:02:37.32s	00:11:41.02s

Tabla X: Comparativa con la literatura.

Referencias	Características utilizadas	Dataset	Precisión	
			Doble Compresión	Triple Compresión
[27]	Macrobloques	YUV	94,10 %	–
Método propuesto	Vectores de Movimiento	Propio	95,27 %	75.33 %

VI. Conclusiones

El contenido de imágenes y vídeos digitales posee información que va más allá de la visual. Esta información es de gran valor forense, pues su correcta explotación puede garantizar la autenticidad e integridad del contenido. Debido a esto, las imágenes y vídeos digitales son una excepcional fuente de evidencias a la hora de resolver procesos judiciales. El desarrollo y mejora continua de las nuevas tecnologías propicia que usuarios convencionales sean capaces de alterar el contenido de imágenes y vídeos con resultados profesionales, imperceptibles para el ojo humano. Ello se suma al hecho de que la detección de manipulaciones es una tarea compleja y también requiere de una mejora continua para adaptarse a tal escenario por lo que resulta imprescindible desarrollo de herramientas forenses capaces de detectar estas manipulaciones, cada vez más profesionales y habituales.

La línea de investigación que se ha seguido en este trabajo comienza realizando un estudio de las técnicas existentes de detección de manipulación sobre imágenes y vídeos digitales dedicando más esfuerzo a técnicas de detección de empalme en imágenes y en detección de doble compresión en vídeos.

Se ha diseñado e implementado una técnica de detección de manipulaciones basado en el estándar de vídeo H.264/MPEG4 para la detección de recompresiones en vídeos MP4 que compara los vectores de movimiento de los macrobloques de dos compresiones secuenciales del mismo vídeo para, a continuación, hacer uso de una SVM que clasifique el vídeo.

Se ha creado un dataset para evaluar la técnica de detección de recompresiones propuesta y se ha utilizado un dataset público para comparar los resultados con otras investigaciones relacionadas. La evaluación ha constado de dos experimentos divididos en grupos según la resolución de cada vídeo:

- Detección de vídeo original o doblemente comprimido, el algoritmo ha conseguido una precisión máxima con datos escalados del 100 % para vídeos de resolución 4K, para el resto de resoluciones no baja del 90 %.
- Detección de vídeo original, doble compresión, o triple compresión donde la precisión disminuye ligeramente respecto a la detección de original o doble compresión, tiene un promedio de precisión del orden del 80 %. No obstante, el mejor resultado lo sigue teniendo una alta resolución.

Los experimentos se han realizado tanto para datos sin escalar como para datos escalados, obteniendo unos resultados muy similares entre ellos. Por tanto no es relevante hacer un escalado de los mismos.

También se han realizado pruebas mezclando todas las resoluciones obteniendo unos resultados menos precisos que en aquellas pruebas donde sí se han separando las resoluciones. El rendimiento del algoritmo es directamente proporcional a la resolución del vídeo que se quiera procesar y a la cantidad de recompresiones que se quieran detectar.

En base a los resultados obtenidos en esta investigación, las líneas futuras de investigación que se proponen en el presente trabajo son las siguientes:

- Extender el algoritmo de detección de recompresiones para utilizarlo con otros códecs de vídeo a parte del H264.
- Utilizar técnicas de aprendizaje profundo y aumentar el número de características extraídas para mejorar la precisión de detección de recompresiones.
- Optimizar el algoritmo de detección de recompresiones para reducir el tiempo de procesado en vídeos de alta resolución.

Aknowledgment

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 700326. Website: http://ramses2020.eu. This paper has also received funding from THEIA (Techniques for Integrity and authentication of multimedia files of mobile devices) UCM project (FEI-EU-19-04).

Referencias

[1] E. M. Nieto, "The value of photography: Anthropology and image."

[2] S. J. Nightingale, K. A. Wade, and D. G. Watson, "Can people identify original and manipulated photos of real-world scenes?" *Cognitive research: principles and implications*, vol. 2, no. 1, p. 30, 2017.

[3] A. De, H. Chadha, and S. Gupta, "Detection of forgery in digital video," in *The 10th World Multi Conference on Systemics Cybernetics and Informatics*, vol. 5, 2006, pp. 229–233.

[4] W. Wang and H. Farid, "Exposing digital forgeries in interlaced and deinterlaced video," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 438–449, 2007.

[5] N. Mondaini, R. Caldelli, A. Piva, M. Barni, and V. Cappellini, "Detection of malevolent changes in digital video for forensic applications," in *Security, steganography, and watermarking of multimedia contents IX*, vol. 6505. International Society for Optics and Photonics, 2007, p. 65050T.

[6] M. Kobayashi, T. Okabe, and Y. Sato, "Detecting forgery from static-scene video based on inconsistency in noise level functions," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 883–892, 2010.

[7] P. Bestagini, S. Battaglia, S. Milani, M. Tagliasacchi, and S. Tubaro, "Detection of temporal interpolation in video sequences," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3033–3037.

[8] Y. Yao, G. Yang, X. Sun, and L. Li, "Detecting video frame-rate up-conversion based on periodic properties of edge-intensity," *Journal of Information Security and Applications*, vol. 26, pp. 39–50, 2016.

[9] M. Xia, G. Yang, L. Li, R. Li, and X. Sun, "Detecting video frame rate up-conversion based on frame-level analysis of average texture variation," *Multimedia Tools and Applications*, vol. 76, no. 6, pp. 8399–8421, 2017.

[10] G. Chetty, "Blind and passive digital video tamper detection based on multimodal fusion," in *Proc. of the 14th WSEAS International Conference on Communications*, 2010, pp. 109–117.

[11] J. Goodwin and G. Chetty, "Blind video tamper detection based on fusion of source features," in *Digital Image Computing Techniques and Applications (DICTA), 2011 International Conference on*. IEEE, 2011, pp. 608–613.

[12] S. Das, G. Darsan, L. Shreyas, and D. Devan, "Blind detection method for video inpainting forgery," *International Journal of Computer Applications*, vol. 60, no. 11, 2012.

[13] P. Bestagini, S. Milani, M. Tagliasacchi, and S. Tubaro, "Local tampering detection in video sequences," in *Multimedia Signal Processing (MMSP), 2013 IEEE 15th International Workshop on*. IEEE, 2013, pp. 488–493.

[14] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting duplication," in *Proceedings of the 9th workshop on Multimedia & security*. ACM, 2007, pp. 35–42.

[15] C.-S. Lin and J.-J. Tsay, "A passive approach for effective detection and localization of region-level video forgery with spatio-temporal coherence analysis," *Digital Investigation*, vol. 11, no. 2, pp. 120–140, 2014.

[16] R. C. Pandey, S. K. Singh, K. Shukla, and R. Agrawal, "Fast and robust passive copy-move forgery detection using surf and sift image features," in *Industrial and Information Systems (ICIIS), 2014 9th International Conference on*. IEEE, 2014, pp. 1–6.

[17] D.-K. Hyun, S.-J. Ryu, H.-Y. Lee, and H.-K. Lee, "Detection of upscale-crop and partial manipulation in surveillance video based on sensor pattern noise," *Sensors*, vol. 13, no. 9, pp. 12 605–12 631, 2013.

[18] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting double mpeg compression," in *Proceedings of the 8th workshop on Multimedia and security*. ACM, 2006, pp. 37–47.

[19] ——, "Exposing digital forgeries in video by detecting double quantization," in *Proceedings of the 11th ACM workshop on Multimedia and security*. ACM, 2009, pp. 39–48.

[20] Y. Su and J. Xu, "Detection of double-compression in mpeg-2 videos," in *Intelligent Systems and Applications (ISA), 2010 2nd International Workshop on*. IEEE, 2010, pp. 1–4.

[21] Y. Su, W. Nie, and C. Zhang, "A frame tampering detection algorithm for mpeg videos," in *Information Technology and Artificial Intelligence Conference (ITAIC), 2011 6th IEEE Joint International*, vol. 2. IEEE, 2011, pp. 461–464.

[22] T. Sun, W. Wang, and X. Jiang, "Exposing video forgeries by detecting mpeg double compression," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1389–1392.

[23] J. Xu, Y. Su, and X. You, "Detection of video transcoding for digital forensics," in *Audio, Language and Image Processing (ICALIP), 2012 International Conference on*. IEEE, 2012, pp. 160–164.

[24] A. Gironi, M. Fontani, T. Bianchi, A. Piva, and M. Barni, "A video forensic technique for detecting frame deletion and insertion," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6226–6230.

[25] A. W. A. Wahab, M. A. Bagiwa, M. Y. I. Idris, S. Khan, Z. Razak, and M. R. K. Ariffin, "Passive video forgery detection techniques: a survey," in *Information assurance and security (IAS), 2014 10th International Conference on*. IEEE, 2014, pp. 29–34.

[26] X. Jiang, W. Wang, T. Sun, Y. Shi, and S. Wang, "Detection of double compression in mpeg-4 videos based on markov statistics," vol. 20, pp. 447–450, 05 2013.

[27] J. Chen, X. Jiang, T. Sun, P. He, and S. Wang, "Detecting double mpeg compression with the same quantiser scale based on mbm feature," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2064–2068.

[28] X. Jiang, P. He, T. Sun, F. Xie, and S. Wang, "Detection of double compression with the same coding parameters based on quality degradation mechanism analysis," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 1, pp. 170–185, 2018.

[29] J. A. Aghamaleki and A. Behrad, "Detecting double compressed mpeg videos with the same quantization matrix and synchronized group of pictures structure," *Journal of Electronic Imaging*, vol. 27, no. 1, p. 013031, 2018.

[30] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h. 264/avc video coding standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 560–576, 2003.

[31] "Ffmpeg," https://www.ffmpeg.org.

[32] "Debug/macroblocksandmotionvectors," https://trac.ffmpeg.org/attachment/wiki/Debug/MacroblocksAndMotionVectors/vismv_pf.png.

[33] "github: vadimkantorov/mpegflow," https://github.com/vadimkantorov/mpegflow.

Edgar González Fernández was born in Mexico City. He received a Degree in Applied Mathematics from the Universidad Autónoma del Estado de Hidalgo in 2010, and a Master in Science with Specialization in Mathematics from the Center for Reasearch and Advanced Studies of the National Polytechnic Institute (CINVESTAV-IPN). He is currently a Ph.D. student in the Computer Science Department at CINVESTAV-IPN. Currently he is Member in the Research Group GASS (Group of Analysis, Security and Systems, http://gass.ucm.es) at the Universidad Complutense de Madrid (UCM). His research interests are Cryptography, Information Security and Data Science.

Ana Lucila Sandoval Orozco was born in Chivolo, Magdalena, Colombia in 1976. She received a Computer Science Engineering degree from the Universidad Autónoma del Caribe (Colombia) in 2001. She holds a Specialization Course in Computer Networks (2006) from the Universidad del Norte (Colombia), and holds a M.Sc. in Research in Computer Science (2009) and a Ph.D. in Computer Science (2014), both from the Universidad Complutense de Madrid (Spain). She is currently a postdoctoral researcher at Universidad Complutense de Madrid (Spain). Her main research interests are coding theory, information security and its applications.

Luis Javier García Villalba received a Telecommunication Engineering degree from the Universidad de Málaga (Spain) in 1993 and holds a M.Sc. in Computer Networks (1996) and a Ph.D. in Computer Science (1999), both from the Universidad Politécnica de Madrid (Spain). Visiting Scholar at COSIC (Computer Security and Industrial Cryptography, Department of Electrical Engineering, Faculty of Engineering, Katholieke Universiteit Leuven, Belgium) in 2000 and Visiting Scientist at IBM Research Division (IBM Almaden Research Center, San Jose, CA, USA) in 2001 and 2002, he is currently Associate Professor of the Department of Software Engineering and Artificial Intelligence at the Universidad Complutense de Madrid (UCM) and Head of Complutense Research Group GASS (Group of Analysis, Security and Systems) which is located in the School of Computer Science at the UCM Campus. His professional experience includes research projects with Hitachi, IBM, Nokia and Safelayer Secure Communications.