Análisis forense aplicado a sistemas multimedia

Detección de Manipulaciones Copy-Move en Ficheros Multimedia mediante la Transformada Discreta del Coseno

Esteban Alejandro Armas Vega, Ana Lucila Sandoval Orozco, and Luis Javier García Villalba, *Member, IEEE*

***Resumen*—Las imágenes digitales tienen un papel muy importante en la vida cotidiana. La mayoría de la población tiene una cámara fotográfica de última generación integrada 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 manipulaciones sobre imágenes cobran gran importancia. En este trabajo se propone una técnica forense basada en el algoritmo de compresión para detectar alteraciones de tipo *copy-move* en una imagen, utilizando para ello la transformada discreta del coseno. Las características obtenidas de estos coeficientes permite obtener vectores de transferencia, los cuales se agrupan y mediante el uso de un umbral de tolerancia permite determinar si existe o no regiones duplicadas dentro de la imagen analizada. Los resultados obtenidos de los experimentos llevados acabo en este trabajo demuestran la eficacia del método propuesto. Para la evaluación del método propuesto se realizaron experimentos con bases de datos públicas de imágenes falsificadas que son ampliamente utilizadas en la literatura.**

***Palabras Clave*—Análisis Forense, Copia-pega, Imágenes Digitales, Manipulación, Transformada Discreta del Coseno.**

I. Introducción

El uso de dispositivos móviles ha aumentado considerablemente convirtiéndose en una herramienta que forma parte de la vida cotidiana de la sociedad actual. En 2017, un informe de Cisco Systems [1] indica que el tráfico de datos móviles se ha multiplicado por 18 en los últimos 5 años y se espera que este tráfico continúe aumentando. Estos datos fueron confirmados en 2018 por Ericsson [2] que estima que para el año 2023 el tráfico de datos móviles se multiplicará por 7 y casi tres cuartos del tráfico de datos móviles del mundo se utilizará para transferencia de ficheros multimedia y redes sociales. Como consecuencia, el proceso de compartir datos de forma masiva es fácil y casi inmediato. Las imágenes y vídeos digitales son, gracias a las redes sociales y a las aplicaciones de mensajería instantánea, uno de los recursos que más tráfico de datos genera actualmente.

Por otro lado, la continua mejora de las prestaciones de las cámaras incorporadas en los dispositivos móviles junto a la evolución de las herramientas de edición de imágenes hacen más sencillo manipular una imagen con excelentes resultados. Para enfrentar este tráfico masivo de imágenes manipuladas el área de análisis forense investiga nuevas técnicas de detección de manipulaciones, para evaluar la integridad de una imagen.

Las imágenes manipuladas llevan existiendo desde hace muchas décadas y están presentes en muchos sectores (política, cine, prensa, rama judicial, etc.).

La manipulación de contenido visual no ha sido algo exclusivo de la era digital actual. A lo largo del tiempo la manipulación siempre ha estado presente. Una de las primeras imágenes manipuladas de la historia [3], es la del fotógrafo Hippolyte Bayard, quien creó una imagen falsa suya suicidándose. Posteriormente, se descubrió que la fotografía fue hecha por el sentimiento de frustración del autor al perder la oportunidad de convertirse en "el inventor" de la fotografía, en lugar de Louis Daguerre que patentó el proceso fotográfico. En el cuadro "El Juicio Final", el pintor Miguel Ángel cubrió la desnudez de algunas figuras a posteriori por orden del Papa. En fotografía convencional, era posible la manipulación mediante empalme de los negativos de las fotografías, por ejemplo, en la Figura 1 se muestra la manipulación realizada a la famosa foto del dictador soviético Iósif Stalin con su comisario para Asuntos Internos Nikolai Yezhov (Figura 1(a)) para eliminarlo de la foto por orden de Stalin tras ser ejecutado en 1940 (Figura 1(b)).

(a) Imagen Original (b) Imagen Manipulada

Figura 1: Ejemplo de Manipulación en Fotografía [4]

La facilidad para manipular imágenes y vídeos digitales se ha incrementado en los últimos tiempos y está al alcance del usuario convencional mediante programas como Adobe Photoshop, GIMP, Adobe Premiere, etc. Manipulaciones como los embellecedores de rostros, cambios de expresión facial, mejora de iluminación de la escena, etc., ya las hace de manera automática nuestro dispositivo móvil mediante nuevas herramientas que hacen uso de inteligencia artificial. Por tanto, detectar imágenes digitales manipuladas es de gran importancia en muchas áreas y con diferentes objetivos. Una de las áreas en donde la verificación de la legitimidad de una imagen es fundamental es en lo judicial, donde las imágenes o vídeos pueden suponer evidencia de gran valor para la

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

https://doi.org/10.12804/si9789587844337.04

resolución de la demanda. Un ejemplo de esto fue el arresto de un conductor [5] que conducía su automóvil a más de 200 Km/h y la evidencia utilizada por la fiscalía fue el vídeo grabado por un peatón, a través del cual se demostró que el imputado circulaba a dicha velocidad.

Sin embargo, para que una imagen pueda ser usada como prueba válida o evidencia en un juicio, se debe asegurar su integridad y demostrar que no ha sido objeto de manipulación. Para llevar a cabo este tipo de autenticación es necesario hacer uso de técnicas robustas de identificación de manipulaciones que puedan garantizar con gran fiabilidad que la imagen es original. Una imagen puede ser manipulada mediante el uso de una variedad de técnicas de manipulación, como *copy-move*, empalme, retoque, filtrado, etc.

En Julio del año 2017 los investigadores de la revista Cognitive-Research [6] 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. Por todo lo anterior, se deben estudiar y proponer técnicas forenses que permitan hacer frente al gran número de imágenes manipuladas que existen hoy en día.

El resto del trabajo está organizado como sigue: La Sección II detalla las características de las manipulaciones comúnmente utilizadas. En la Sección III-D se describen las principales técnicas de detección de imágenes manipuladas, haciendo énfasis en las técnicas con enfoque pasivo más relevantes de la literatura. Los detalles de la técnica de detección propuesta en este trabajo se presenta en la Sección IV. En la Sección V se analizan los resultados de los experimentos realizados y, finalmente, las conclusiones del trabajo se recogen en la Sección VI.

II. Manipulación de Imágenes

Entre los tipos de manipulación de imágenes, destacan los siguientes: retoque, *Copy – Move*, empalme de imágenes y falsificación de huellas digitales [7].

II-A. Copy – Move

La manipulación "*Copy-Move*" típicamente se realiza con el objetivo de hacer que un objeto "desaparezca" de la imagen original cubriéndolo con un pequeño fragmento copiado de otra parte de la misma imagen. Este método también se usa para duplicar objetos existentes en la imagen. Como estos bloques copiados provienen de la misma imagen todas sus características serán compatibles con el resto del contenido por lo cual hace muy difícil que el ojo humano lo detecte. Cuando se pega la región copiada se suele acompañar del efecto "blurring" en los bordes del área modificada para disminuir las irregularidades entre la región original y la modificada. Las técnicas de detección de este tipo de manipulación se centran en la búsqueda de áreas duplicadas. Sin embargo, si la manipulación se combina con otras técnicas de post-procesamiento, como la aplicación de filtros de color, el reconocimiento del área modificada es mucho mas difícil para este tipo de técnicas[8][9]. Un ejemplo de este tipo de manipulación se muestra en la Figura 2. En la Figura manipulada 2(b) se han duplicado los dos animales que aparecían en la Figura 2(a).

(a) Imagen original (b) Imagen manipulada

Figura 2: Ejemplo de Manipulación con la Técnica Copia-Pega [10]

II-B. Empalme

Esta técnica es similar a la técnica "*Copy-Move*", con la diferencia de que el fragmento que se copia no pertenece a la misma imagen, es decir, la imagen manipulada es el resultado de la mezcla de dos o más imágenes. El objetivo de esta técnica es insertar elementos que no estaban en la escena que fue capturada originalmente. Por regla general, el bloque de imagen "donante" ha podido ser adquirido por otro dispositivo móvil y por tanto sus características y rastros serán diferentes al resto de la imagen. Es muy usada en fotomontajes donde se combinan dos imágenes dando la sensación de ser una sola. Detectar el área exacta que se ha falsificado en la imagen, mediante la técnica de empalme, es de gran complejidad en comparación con la técnica de manipulación anterior. Esto se debe a que no es posible buscar áreas duplicadas ya que la región manipulada proviene de una imagen diferente [11].

En la Figura 3 se muestra un ejemplo de esta técnica. La Figura 3(a) es la imagen donante, el faro es copiado y pegado en la imagen receptora (Figura 3(b)) el resultado del empalme se muestra en la Figura 3(c).

Las técnicas de detección de empalme se centran en hallar la región de la imagen que contenga estas variaciones de características y rastros con respecto al resto del contenido de la imagen original.

II-C. Aplicación de Filtros

Esta técnica de manipulación es de las más utilizadas por su sencillez. Casi todos los programas de edición de imágenes digitales incorporan una selección de filtros ya predefinidos para aplicarlos automáticamente sobre la imagen e incluso los dispositivos móviles con cámaras integradas pueden aplicar este tipo de "mejoras" a la imagen al momento de capturar la escena. La aplicación de filtros tiene como objetivo mejorar el acabado final de la imagen modificando aspectos como tonos, saturaciones, brillos, contrastes, etc. No tienen porqué conllevar un cambio "malicioso" en el contenido de la imagen pero se toma en cuenta porque puede aplicarse cualquiera de estos filtros en combinación con otras técnicas de manipulación de imágenes y es muy probable que esto afecte la precisión de los algoritmos de detección de manipulación en imágenes. En

(a) *Imagen donante original*

(b) *Imagen receptora original*

(c) *Imagen manipulada*

Figura 3: Ejemplo de Manipulación con la técnica de Empalme [12]

(a) Imagen original

(b) Imagen manipulada

Figura 4: Ejemplo de Manipulación mediante la aplicación de un filtro.

(a) Imagen original

(b) Imagen manipulada

Figura 5: Portada manipulada de la revista Nitro[13]

la Figura 4 se muestra un ejemplo de esta técnica, donde se ha aplicado el filtro predefinido “Sepia” del software de edición GIMP.

II-D. Retoque

Esta manipulación consiste en aplicar pequeñas modificaciones sobre la imagen original sin copiar ningún área del resto de la imagen o tomarla de una diferente. Tiene como objetivo perfeccionar acabados u ocultar imperfecciones con fines estéticos manteniendo siempre unas características similares a las de la imagen original. Para ello se copian y pegan regiones de la imagen de la misma área. Los retoques que se realizan suelen estar enfocados a perfeccionar la escena [14].

El acabado de las imágenes varía dependiendo del contenido y de los fines con los que se realiza la alteración. Las herramientas más utilizadas en este tipo de manipulaciones suelen ser el saneado, perfilado, emborronado, difuminado y realce. Esta técnica de manipulación es muy común en los sectores de la publicidad, cine y comunicación [15]. La Figura 5 muestra un ejemplo de retoque fotográfico en el que la apariencia de la modelo se modificó digitalmente. La Figura 5(a) muestra la imagen original sin retoque y la Figura 5(b) muestra el resultado de retocar la imagen para la portada de la revista *Nitro*.

Las portadas y los anuncios de las revistas de moda generalmente utilizan algún tipo de retoque para ocultar las imperfecciones y así aumentar los niveles de belleza en las fotografías.

II-E. Manipulación de la Huella Digital

La manipulación de la huella digital de una imagen no está centrado en la parte visual de la imagen si no en la información que ésta contiene [16].

La huella digital es un rastro que dejan todas las cámaras de los dispositivos móviles sobre la imagen que toman durante el proceso de captura. Cuando se genera una imagen digital se introduce este rastro, también llamado ruido.

Extraer el ruido de una imagen proporciona una información valiosa acerca de la fuente (modelo y marca del dispositivo) que generó dicha imagen ya que el tipo de ruido que contiene es intrínseco y único al modelo de la cámara que lo generó. El objetivo de manipular la huella digital de una imagen es el de poder modificar su origen. Si se sustituye la huella digital de la imagen por otra, es posible incriminar a otro dispositivo móvil en la escena en cuestión. También es posible eliminar la huella y así anonimizarla. Este tipo de técnica se subdivide en: Anonimización de la imagen, que consiste en eliminar la información del origen de la imagen y, la falsificación de la imagen, que elimina la huella de la cámara que generó la imagen y coloca una huella de otra cámara de un dispositivo diferente. Estas técnicas no implican la alteración de la imagen en sí, sino la modificación de la información asociada (huella digital) que proviene del sensor que capturó la imagen.

Existen varias fuentes de imperfecciones y ruido introducidas durante el proceso de adquisición de imágenes. Esas imperfecciones aparecen principalmente por dos razones; primero hay componentes aleatorios como el ruido de lectura o el ruido de disparo y segundo debido al ruido del patrón, que es un componente determinista del sensor y permanece aproximadamente igual si varias fotos de la misma escena están tomados. Este patrón es útil para detectar la fuente de origen de una imagen, ya que cada dispositivo tendrá un patrón de ruido específico [16] [17] .

III. Técnicas de Detección de Manipulaciones

Existen dos enfoques forenses de detección de imágenes manipuladas: Intrusivo o activo y no intrusivo o pasivo [18].

- **Enfoque Activo**: Analiza las marcas de agua o señales que deja un dispositivo al momento de generar una imagen digital. El mayor inconveniente de este tipo de enfoque es que muchas cámaras no tienen la capacidad de incorporar este tipo de marcas o firmas, por lo que su alcance es limitado.
- **Enfoque Pasivo**: Analiza el contenido y las características de la imagen digital. A su vez este enfoque puede clasificarse en: métodos basados en aprendizaje y métodos basados en bloques.

El enfoque pasivo tiene un alcance más amplio que el enfoque activo ya que no necesita información previa sobre las imágenes. A continuación se presentan las propuestas de enfoque pasivo más relevantes. La Figura 6 Muestra una clasificación de las técnicas de detección de manipulaciones en imágenes digitales

III-A. Detección de Copy-Move

La detección de falsificaciones "*copy-move*" son las técnicas más utilizadas en el campo forense debido a su simplicidad y eficacia. La principal evidencia que se explota para detectar este tipo de manipulación es la existencia de dos áreas iguales basándose en las propiedades de los bloques en los que se divide la imagen.

La primera aproximación que se realizó para identificar áreas copiadas fue realizada en el año 2003, en [19] los autores propusieron un método que hacia uso Transformada Discreta del Coseno (DCT) para localizar coincidencias entre bloques de una manera más eficiente que la de realizar una búsqueda por fuerza bruta.

En [20] se propone un método que utiliza el Análisis de Componentes Principales (PCA) para representar una imagen como una representación de bloques superpuestos. Obtuvieron unos resultados más eficientes que los obtenidos en la técnica anterior debido a que consiguieron reducir el coste computacional al rebajar a la mitad el número de cálculos requeridos para procesar los bloques con PCA. Aún así, el coste computacional seguiría siendo grande y por ello, en [21] se propone encontrar las coincidencias entre los bloques buscando patrones de intensidad similares.

Los autores de [22] proponen reducir aún más el tamaño de los bloques superpuestos en los que se subdivide la imagen para mejorar la precisión al comparar similitudes entre ellos. Las posibles áreas duplicadas tendrán unas propiedades de intensidad similares. Este método es más efectivo frente a pérdidas por compresión que los métodos anteriores.

En [23] se propone detectar las regiones duplicadas estudiando todas las invariantes de desenfoque de una imagen. Los resultados de éste método fueron correctos pero con la desventaja de obtener un tiempo de computación demasiado grande (un promedio de 30 minutos para una imagen RGB de tamaño medio).

En [24] los autores desarrollaron un método que descomponía la imagen en registros de coordenadas polares y, haciendo uso de la Transformada Wavelet, detectar las regiones copiadas. Se reducía así la dimensión de la imagen de entrada debido a la aplicación de Wavelet. Para encontrar los bloques similares se realizaba una búsqueda por fuerza bruta mapeando cada uno de los bloques con las coordenadas polares y la correlación entre ellos como criterio. Otros autores en [25] hicieron algo similar pero se basaron en la Transformada de Fourier (FFT).

En [26] utilizan Transformada Wavelet (WT) para reducir la dimensión de la imagen y Descomposición en Valores Singulares (SVD) para generar el vector de características de cada región con el fin de buscar las similitudes con mayor eficacia. Las regiones duplicadas eran localizadas por clasificación lexicográfica y vecindad detectando todos los bloques, incluso cuando la imagen había sido muy comprimida.

Los métodos descritos hasta ahora no producían resultados óptimos cuando las imágenes sufrían cierta transformación geométrica. En [27] los autores proponen una nueva metodología basada en el algoritmo *Scale-invariant Feature Transform* (SIFT) para estimar los parámetros de la transformación geométrica aplicada sobre la imagen (traslación horizontal o vertical, escalados o rotación del ángulo) con alta fiabilidad, pudiendo así, detectar falsificaciones en imágenes que han sufrido alguna de estas transformaciones. Los autores de [28] mejoran la robustez de SIFT proponiendo un método basado en el algoritmo *Speeded Up Robust Features* (SURF). Este método es además capaz de detectar también áreas copiadas a las que se le ha aplicado modificaciones en brillo o contraste.

III-B. Detección de Empalme

El empalme de imágenes es uno de los esquemas de manipulación más simples y comúnmente utilizados. La detección de este tipo de manipulación es una tarea fundamental durante la verificación de la integridad de imágenes.

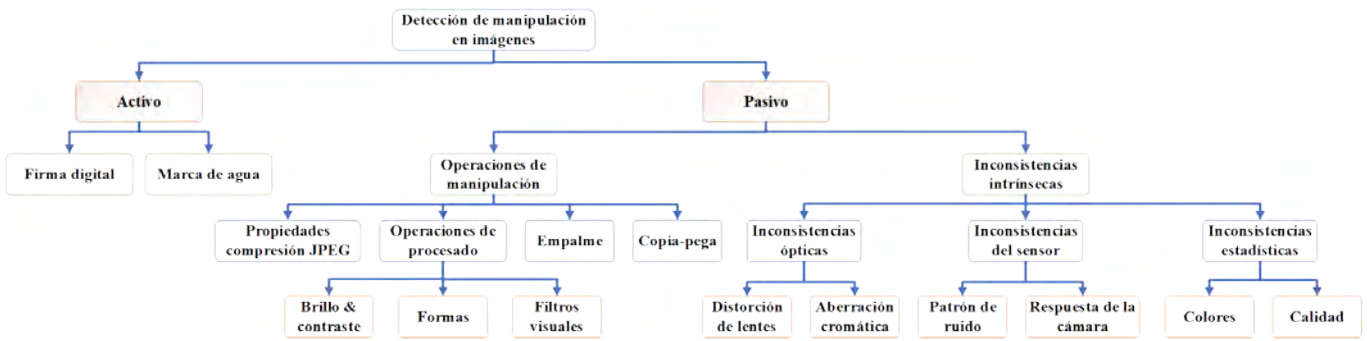


Figura 6: Esquema de detecciones de manipulación en imágenes.

Por lo general, todas las técnicas se basan en las variaciones que se encuentran en el patrón de características del área pegada respecto del contenido de la imagen original.

El primer método propuesto fue presentado en [29]. En este trabajo se propone una técnica basada en el análisis de la señal de la imagen para detectar las correlaciones no naturales que se introducen durante el proceso de falsificación. Obtuvo buenos resultados cuando la detección se llevaba a cabo sobre empalmes realizados por personas y no por máquinas.

En [30] los autores presentaron un modelo de detección de empalme de imágenes basado en el uso de características de magnitud y fase de la propia imagen. Los resultados de la precisión de detección fueron de aproximadamente del 70 %. Posteriormente, los mismos autores propusieron un método para detectar empalmes abruptos utilizando las mismas características.

Los autores de [31], por su parte, se apoyaron en la transformada de Hilbert-Huang para generar estadísticas con el fin de utilizarlas para la clasificación de un modelo de imagen natural. Este modelo estaba basado en los momentos característicos obtenidos con ayuda de la descomposición de Wavelet y así conseguir distinguir las imágenes empalmadas de las imágenes auténticas.

Los autores de [32] propusieron extraer las invariantes de geometría de los píxeles de cada región de una imagen para estimar la Función de Respuesta de la Cámara (CRF) y estudiar las variaciones entre las distintas zonas de la imagen para, así, detectar las áreas que han sufrido un empalme.

En [33] los autores investigaron las características estadísticas de los bloques de una imagen para detectar empalmes. Estas características son extraídas de matrices 2D, las cuales son generadas al aplicar al bloque de imágenes de varios tamaños la DCT. Los experimentos tuvieron una precisión del 91 %.

En [34] se presentó un método de detección basado en características de momento extraídas de la DCT y en métricas de calidad de imagen extraídas de la propia imagen. Descubrieron que ambas características sufrían variaciones cuando una imagen había sufrido un empalme y explotaron dichas variaciones.

Los autores de [35] sugieren un método basado en dividir la imagen por áreas para después extraer las características de densidad de los coeficientes DCT vecinos de cada área. Todas las variaciones en las densidades se clasifican mediante un Máquina de Soporte Vectorial (SVM) para identificar si esas áreas son diferentes.

En [36] se construye un método de detección de empalmes basado en el estudio de las sombras de la imagen. Mediante combinaciones y estimaciones de las zonas de sombra logran encontrar bloques empalmados. Los mismos autores en [37] utilizan la teoría de la homografía plana para localizar la región manipulada y aparte, desarrollaron un método de extracción automático que segmentaba el objeto falso de la imagen manipulada.

En [38] los autores proponen un método basado en la comparación de los espacios cromáticos que forman la imagen. Se utilizan cuatro vectores *Run-length Run-number* (RLRN) con diferentes direcciones extraídos de los canales de crominancia que tienen correlación con características utilizadas en la detección de empalmes.

Los autores de [39] investigaron una técnica basada en el estudio de la iluminación de los objetos de una imagen. Se basan en la consistencia que debe existir en las sombras en función del grado de iluminación y en las características de color del valor de dicha sombra.

En [40] se evaluó la técnica Error Level Analysis (ELA) en imágenes manipuladas con distintos métodos, demostrando una que sólo era efectiva para detectar empalmes.

En [41] se propuso un algoritmo basado en el uso de ELA que demostró detectar con éxito la imagen modificada y el punto exacto de la modificación mediante el uso de histogramas.

En [42] utilizan la Transformada de Wavelet para filtrar los resultados de aplicar ELA con el objetivo de resaltar las alteraciones sobre la imagen.

En [43] se propone un método que combina el descriptor de textura *Local Binary Patterns* (LBP) junto a DCT para detectar cambios producidos por las manipulaciones de empalme y también de copia-pega. En los experimentos se utiliza SVM obteniendo una tasa de acierto entre el 97.50 % y el 97.77 % sobre el conjunto de datos "*CASIA TIDE v2.0*".

El método de detección de empalme propuesto en [44] modela los cambios de manipulación utilizando características estadísticas extraídas de matrices 2D generadas al aplicar la transformada discreta de coseno de bloques de varios tamaños (*Multi-size Block Discrete Cosine Transform* (MBDCT)). En los experimentos se obtuvo un 91.40 % de acierto sobre el conjunto de datos "*Columbia*" usando SVM.

En [45] exploraron el efecto de diferentes modelos de color en la detección de falsificación de empalme. En este trabajo, se hace una comparación de los modelos cromáticos frente a los modelos *Red-Green-Blue* (RGB) y de luminancia utilizados comúnmente. Se emplean cuatro vectores RLRN con diferentes direcciones extraídas de canales de crominancia correlacionados como características para la detección de empalme en imágenes. Finalmente, se usa SVM como algoritmo clasificador. El conjunto de datos utilizado en los experimentos son "*CASIA TIDE v1.0*" y "*Columbia*" con una precisión de 94.7 %.

III-C. Detección de Manipulación de Huella Digital

La huella digital de una imagen identifica su origen y garantiza su integridad. Las técnicas para detectarla se encargarán pues, de estudiar si dicha huella ha sufrido modificaciones, pues en tal caso, es una evidencia de que la imagen en cuestión ha sido manipulada. Las técnicas se basan principalmente en el estudio de los patrones del ruido del sensor que introduce cada cámara en las imágenes que genera durante el proceso de captura de una fotografía.

En [46] se propone un método basado en la extracción de características del ruido de foto-respuesta no uniforme (Patrón de Ruido de Respuesta no Uniforme (PRNU)), junto con un SVM para su clasificación. Este trabajo se utilizó únicamente en dispositivos móviles y se consiguió mostrar que este método consigue buenos resultados cuando se tiene que clasificar una gran cantidad de cámaras fuente.

En [47] se propone combinar dos métodos de detección: el estudio de las imperfecciones del sensor y las *Transformada Discreta de Wavelet* (DWT). Los resultados confirman que

estas dos técnicas juntas ayudan a rastrear con precisión el dispositivo fuente que tomó la imagen, además del modelo y marca de dicho dispositivo.

En [48] se estudian investigaciones recientes en el campo y proponen la mezcla de dos técnicas (imperfecciones del sensor y transformadas wavelet) para obtener una mejor identificación de fuentes de imágenes generadas con dispositivos móviles. Los resultados muestran que las imperfecciones del sensor y las transformadas wavelet pueden servir conjuntamente como buenas características forenses para ayudar a rastrear la cámara fuente de las imágenes producidas por teléfonos móviles. Además, este modelo también permite determinar con gran precisión la marca y el modelo del dispositivo.

Como se puede observar la mayoría de técnicas emplean los patrones de ruido para identificar y extraer el ruido del sensor. Para poder comprobar si se ha llevado a cabo una modificación o eliminación bastaría con comparar la huella digital de la imagen original y de la imagen manipulada.Sin embargo, nuestra propuesta combina el análisis de los patrones de textura locales junto con las características obtenidas de aplicar la transformada discreta de Wavelets junto con la del Coseno a la imagen.

III-D. Detección Manipulación Retoque

En [15] se propuso un algoritmo eficiente diseñado específicamente para predecir la presencia de retoques en imágenes de portadas de revistas. El conjunto de datos de 468 fotos (originales y retocadas) se valoraron entre 1 (muy similar) y 5 (muy diferente) dependiendo de la cantidad de alteración fotográfica. Se calcularon las modificaciones geométricas y fotométricas de cada foto original y retocada y, posteriormente, se extrajeron ocho estadísticas de resumen que incorporan el grado de retoque fotográfico para calcular la correlación con la valoración de cada foto. Se utilizó el algoritmo de Máquina de Soporte Vectorial SVM para determinar el grado de modificación de la imagen. La precisión máxima obtenida con los experimentos fue de 98,75 %.

El algoritmo propuesto en [49], utiliza una red neuronal para extraer características y SVM para clasificar las imágenes en una clase sin retoques o retocada. En los experimentos se utilizó el conjunto de datos "*ND-IIITD retouched faces*" de 325 de caras retocadas y se obtuvo un 87 % de acierto.

En [50] se propone la extracción de las características de color, forma y textura de tres regiones faciales predefinidas. Se utilizaron los conjuntos de datos *YouTube Makeup* (YMU) y *Makeup in the Wild* (MIW) [51] para entrenar y predecir, respectivamente, un sistema SVM con núcleo *Radial Basis Function* (RBF) para clasificarlas. La precisión que se obtuvo fue de un 93 %. Posteriormente, en [52] se propuso un algoritmo más preciso para la detección de maquillaje en los mismos conjuntos de datos utilizando características de textura y forma. La técnica propuesta extrae un vector de características que captura las características de forma y textura de la cara usada como entrada del algoritmo. Se consiguió aumentar la precisión a un 98.5 % usando un clasificador SVM.

IV. Método Propuesto

En esta sección se propone un esquema mejorado de detección de falsificación de copia-pega basado en el esquema presentado por primera vez por Fridrich [19]. A continuación se especifican los parámetros de entrada y los resultados que genera el algoritmo tras su ejecución.

- Entrada: Imagen a analizar.
- Salida: Imagen con la región duplicada marcada de un color determinado, de esta forma se puede visualizar claramente el área sobre el que se ha realizado la copia y la región exacta donde se ha pegado.

El primer paso es convertir la imagen que se desea analizar a escala de grises. Para ello se extraen los componentes del canal de luminancia y se representa la imagen con ellos.

A continuación, se establece un tamaño de bloque $B = 8$ para dividir la imagen desde la esquina superior izquierda a la esquina inferior derecha.

Los bloques se superponen con un desplazamiento de un píxel hasta obtener $(M–B+1)(N–B+1)$ bloques superpuestos, siendo M y N las dimensiones de la imagen. El tamaño de bloque B se ha establecido en 8 para conseguir resultados más precisos con un nivel óptimo de ruido.

Seguidamente se extraen las características DCT de cada uno de los bloques. DCT puede eliminar la redundancia entre píxeles adyacentes de manera rápida y efectiva, y tiene propiedades de compactación de energía [53], por ello es razonable adoptar los coeficientes DCT como características de los bloques de imagen. Una propiedad de DCT es que la energía solo se enfoca en los coeficientes de baja frecuencia, es decir, no todos los elementos son igual de importantes, por ello se descartan los coeficientes de alta frecuencia por que solo introducen ruido y pueden dar lugar a errores en procesos posteriores.

Para llevar a cabo este proceso se establece un valor de truncamiento y se realiza a su vez un escaneado en zigzag. El valor de truncamiento k se calcula mediante la ecuación 1 y corresponde a la longitud del vector de características de un bloque. Para establecerlo se fija un factor de truncamiento $f_t(0 < f_t < 1)$.

$$k = \left[f_t.B^2\right] \tag{1}$$

A su vez se realiza un escaneado en *zigzag* sobre el bloque de coeficientes DCT como se muestra en la Figura 7. Este tipo de escaneo permite realizar un recorrido ascendente por los coeficientes de menor a mayor frecuencia y gracias al valor k antes mencionado accede solo a los coeficientes más importantes. La zona verde de la Figura 7 representa el área de coeficientes DCT más significativos.

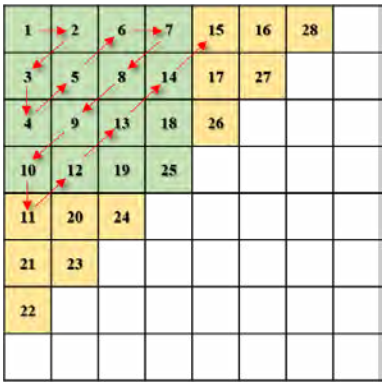


Figura 7: Escaneo en zig-zag de coeficientes DCT

Por último, para reducir las dimensiones y mejorar la eficiencia del proceso de adaptación, los coeficientes DCT se cuantifican mediante un factor de cuantificación f_q y se redondean al entero más cercano usando la ecuación 2.

$$\vec{a_i} = \left(\left[\frac{a_{i1}}{f_q} \right], \left[\frac{a_{i2}}{f_q} \right], ..., \left[\frac{a_{ik}}{f_q} \right] \right) \quad (2)$$

Este proceso proporciona una secuencia con los coeficientes de baja frecuencia agrupados y truncados para cada bloque de la imagen.

$$\#Bloques = (M - B + 1)(N - B + 1)(f_t.B^2) \quad (3)$$

Finalmente, se crea una matriz A de una sola columna para guardar cada secuencia en una fila diferente, a la vez que se añaden las coordenadas x e y de la esquina superior izquierda del bloque al final de la secuencia de coeficientes.

El siguiente paso es ordenar lexicográficamente la matriz A con todos los vectores de características, de esta manera las filas de características similares quedarán juntas y así se podrá determinar qué bloques de la imagen están relacionados. Por lo tanto, se requieren algunos métodos para juzgar si los vectores de características correspondientes de los bloques de imagen son los mismos.

Si los componentes correspondientes de los dos vectores de bloques de imagen son casi iguales, los dos bloques de imagen pueden considerarse estrechamente relacionados. Posteriormente, estos bloques se estudiarán para determinar si uno de ellos o ambos son objetos de manipulación. Para juzgar la similitud entre dos bloques se realizan las siguientes comprobaciones para cada fila de la matriz A:

1. Cada vector de fila $\vec{a_i} = (a_i^1, a_i^2, ..., a_i^k)$ debe ser comparado con sus vectores de fila adyacentes $\vec{a_j} = (a_j^1, a_j^2, ..., a_j^k)$.
2. Se define un parámetro N_a que corresponde al número de máximo de filas que van a ser comparadas con a_i, por lo que debe satisfacer $(j–i < N_a)$.
3. Se definen los umbrales S_t y T_t que serán usados más adelante y se inicializa la variable r_{max} a un valor suficientemente pequeño y la variable r_{min} a un valor suficientemente grande.
4. Se crea un contador c inicializado a 0.
5. Para cada $\vec{a_i}$ y $\vec{a_j}$ dentro del intervalo $(1 <= l <= k)$ se comprueba que:
   - 4.1. Si $a_j^l = 0$ se comprueba si se cumple $|a_i^l – a_j^l| < S_t$, en caso afirmativo se incrementa el valor de c en 1.
   - 4.2. Si $a_j^l \neq 0$ se calcula $r_i = a_i^l / a_j^l$ y se cambian los valores r_{min} y r_{max} según corresponda:
     - Si $r_{max} < r_l$ entonces $r_{max} = r_l$
     - Si $r_{min} > r_l$ entonces $r_{min} = r_l$.
   - 4.3. Si $r_{max} – r_{min} > T_t$ entonces c se incrementa en 1.
6. Finalmente si $c < C_t$ entonces a_i y a_j son similares.

Después de comprobar si el vector de fila a_i (la coordenada superior izquierda del bloque de imagen es (x_1, y_1)) y el vector de fila a_j (la coordenada superior izquierda del bloque de imagen es (x_2, y_2)) son similares, se calculan los vectores de transferencia entre los dos vectores.

$$\vec{s} = (s_1, s_2) = (x_1 - x_2, y_1 - y_2) \quad (4)$$

A continuación, se comprueba si la distancia excede de un parámetro T_d, con la ecuación 5. Si se cumple, la frecuencia existente del vector de transferencia se incrementa en 1, en caso contrario, no se modifica dicha frecuencia.

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} > T_d \quad (5)$$

Una vez obtenidas las frecuencias para los vectores de transferencia se procede a buscar el vector principal de transferencia cuyas frecuencias exceden un umbral T_f. Los bloques de la imagen correspondientes al vector principal se pueden considerar como regiones copiadas y pegadas. Estas regiones se marcan respectivamente en un color que las haga distinguirse del resto de la imagen. La Figura 8 presenta el diagrama de los procesos más característicos del algoritmo.

V. Experimentos y Resultados

En los experimentos realizados se ha utilizado *Python* como lenguaje de programación, debido a su gran flexibilidad para poder realizar el análisis de datos y su alta velocidad en gestionar la entrada y salida. Para la evaluación del algoritmo se ha hecho uso de varios datasets públicos ([12], [54]) para realizar experimentos con varios formatos y tamaños. La Tabla I muestra las características de los datasets utilizados en los experimentos.

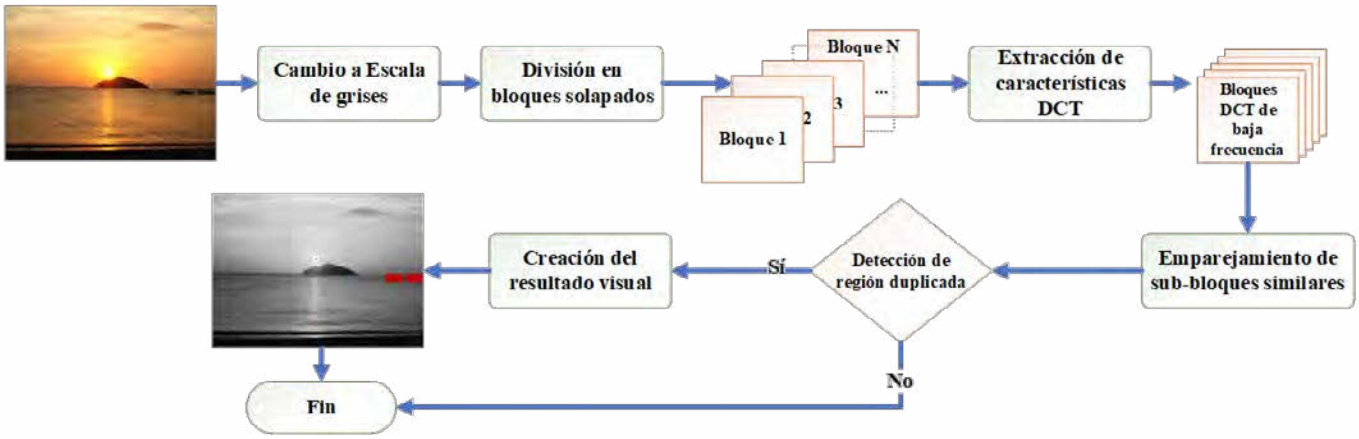


Figura 8: Diagrama de flujo del algoritmo de identificación de Copia-Pega propuesto

Tabla I: Características de los datasets utilizados

Datasets	Formato	Resolución	Número de Imágenes		
			Originales	Manipuladas	Total
CASIA v1.0 [12]	JPEG	384x256	800	921	1721
CASIA v2.0 [12]	JPEG, BMP, JTIFF	240x160 900x600	7491	5123	12614
IFS-TC [54]	PNG	1024x768 3648x2736	424	451	875

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 17.04
Memoria	12 GB
Procesador	Intel® Core™ i5-6200U CPU @ 2.30GHz x 4
Gráficos	Intel® HD Graphics 520 (Skylake GT2)
Tipo de SO	64 bits
Disco	64 GB

V-A. Experimento 1

El primer conjunto de experimentos se basó en comprobar la efectividad del algoritmo propuesto en la Sección IV.

Este algoritmo hace uso de diferentes parámetros configurables, dependiendo del valor asignado los resultados pueden variar notablemente. En [55] se propone un algoritmo que da excelentes resultados en identificación de manipulaciones copia-pega. Para realizar sus experimentos hacen comparaciones entre los parámetros usados por otras investigaciones. Los valores que han establecido han servido como referencia para inicializar los parámetros del algoritmo que se ha propuesto en este trabajo. En la siguiente tabla se exponen cada uno de los parámetros utilizados y sus valores correspondientes.

Tabla III: Parámetros configurables del algoritmo copia-pega

Parámetro	Nombre	Valor asignado
ft	Factor de truncamiento	0,25
fq	Factor de cuantificación	4
Na	Filas vecinas comparables	3
St	Umbral S	4
Tt	Umbral T	0,06
Ct	Umbral de similitud	3
Tf	Umbral de frecuencia	50
Td	Distancia de los vectores	20

El parámetro que mejora los resultados ha sido el *umbral de frecuencia* o *Tf*. Este parámetro establece el valor con el cual un bloque de la imagen puede considerarse una manipulación válida. Si un bloque aparece varias veces en la imagen como duplicado y dicha frecuencia de aparición supera a la establecida por el umbral *Tf* se considerará que forma parte de la manipulación. Estudiar la frecuencia de aparición de los bloques es posible gracias a la superposición con la que se extraen de la imagen. Cuando este parámetro es

(a) Imágenes originales

(b) Imágenes manipuladas

(c) Resultados con Tf=50

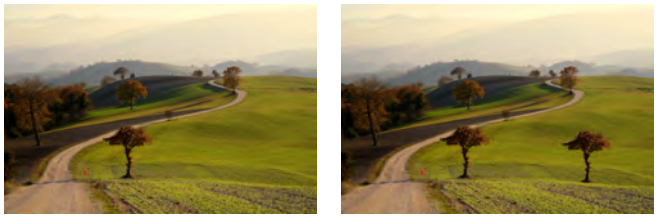
(d) Resultados con Tf=100

(e) Resultados con Tf=150

Figura 9: Identificación de la manipulación copia-pega

alto los resultados finales son más refinados, eliminando las áreas identificadas como manipuladas pero que en realidad son falsos positivos. En el experimento se ha ajustado el parámetro *Tf* en tres valores: 50, 100 y 150. En la Figura 9 se muestran 3 imágenes manipuladas de ejemplo. En la Figura 9(a) se muestran las imágenes manipuladas, en la Figura 9(b) su respectiva imagen sin manipulación y en la Figura 9(c) la zona manipulada.

En la Figura 10 se muestran los resultados de la detección con diferentes valores del parámetro *Tf*. Como se puede observar en la figura a mayor valor del parámetro *Tf* los resultados presentan menos ruido, es decir, se eliminan las zonas negras que no forman parte de la manipulación. En la primera imagen la manipulación se identifica con el *Tf*=50, con un valor más alto el algoritmo no encuentra ningún bloque duplicado que cumpla la frecuencia de aparición establecida por *Tf*. En cambio puede apreciarse en las otras dos imágenes que a mayor valor del parámetro *Tf* se elimina el ruido producido por los falsos positivos. Esto se debe a que son manipulaciones de gran tamaño en proporción a la imagen por lo que la frecuencia de aparición de los bloques manipulados será muy superior al existir la superposición.Sin embargo, el

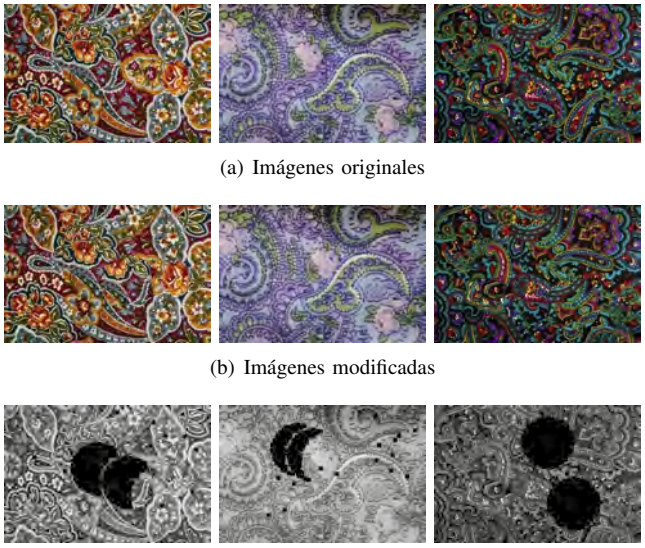


(a) *Imagen original* (b) *Imagen manipulada*

(c) Tf=50 (d) Tf=100 (e) Tf=150

Figura 10: Área duplicada con detalles de la imagen real

algoritmo falla con un tipo concreto de manipulaciones. Estas manipulaciones consisten en duplicar determinadas áreas que presentan zonas de la imagen real. Esto se puede observar en el ejemplo de la Figura 11.



(a) Imágenes originales

(b) Imágenes modificadas

(c) Resultados de la detección

Figura 11: Detección de copia-pega en regiones de texturas similares

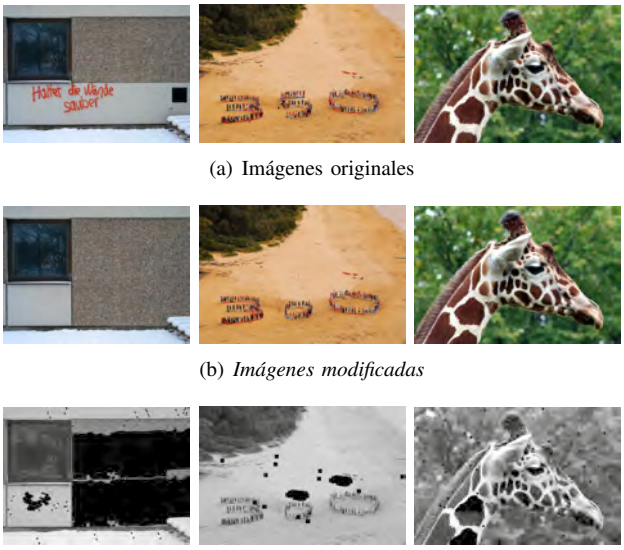
En esta imagen se ha duplicado el árbol situado en la parte central. Este árbol presenta huecos entre las ramas que han sido editados en la duplicación para que se integre perfectamente con el fondo, es por ello que el algoritmo trata ambos árboles como objetos diferentes y no es capaz de dar un resultado correcto.

V-B. Experimento 2

En el segundo experimento se comprobó la precisión del algoritmo de identificación de la región de copia-pega en imágenes de texturas con patrones similares. En este tipo de imágenes la manipulación pasa inadvertida debido a su excelente integración con el fondo original. Esto se debe a que se usa un mismo patrón de colores sin áreas que resalten por encima de otras. En este experimento se realizaron dos pruebas con este tipo de imágenes, se ajustó el parámetro *Tf* al valor 150 para disminuir el ruido de puntos negros en los resultados.

En la primera prueba se usaron imágenes con muchos detalles de múltiples colores pero a su vez siguen un mismo patrón, esto hace que el área duplicada sea difícil de detectar. En la Figura 12 se muestran tres ejemplos de identificación en este tipo de imágenes. Como se observa, el algoritmo consigue una precisión destacable.

Para la segunda prueba se usaron imágenes donde la región duplicada se encontraba en un área del mismo color que otras regiones de la imagen. En este tipo de imágenes también es difícil detectar la región duplicada ya que puede confundirse con otra región original que tenga el mismo color. En la Figura 13 se muestran tres ejemplos donde puede apreciar que el algoritmo presenta un buen funcionamiento ante este tipo de manipulaciones.

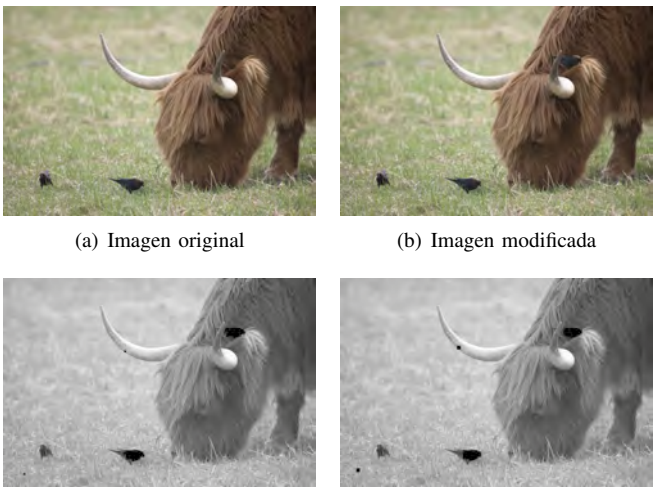


(a) Imágenes originales

(b) *Imágenes modificadas*

(c) *Resultados de la detección*

Figura 12: Detección de copia-pega en imágenes con áreas del mismo color



(a) Imagen original (b) Imagen modificada

(c) Imagen de 1280×854 (d) Imagen de 640×427

Figura 13: Detección de copia-pega en imágenes escaladas

V-C. Experimento 3

En este experimento se analizó la eficiencia del algoritmo en imágenes de gran tamaño y resolución. Se observó que al escalar una imagen a un tamaño más pequeño la precisión del algoritmo sigue manteniéndose alta sin sufrir cambios significativos. Esta observación permite realizar un escalado de las imágenes grandes antes de que el algoritmo las procese, esto aumenta la eficiencia sin perder calidad en los resultados.

En la Figura 13 se muestra un ejemplo de una imagen manipulada por la técnica copia-pega, en la cual se ha copiado el pájaro situado encima del césped y se ha pegado sobre la cabeza de la vaca. El tamaño original de la imagen es de 1080x854 píxeles, también se muestra el resultado obtenido al escalar la imagen a un tamaño de 640x427 píxeles. El tiempo de ejecución que ha tardado el algoritmo en procesar la imagen original ha sido de 160 segundos, en cambio en la imagen escalada ha tardado 48 segundos. Como puede observarse se ha detectado la manipulación perfectamente en ambas imágenes, por lo que es posible realizar el escalado sin afectar a la precisión del algoritmo y mejorando considerablemente el tiempo de ejecución. A lo largo de las pruebas realizadas se ha podido comprobar que el algoritmo funciona con cualquier tipo de formato, como *Joint Photographic Experts Group* (JPEG), *Portable Network Graphics* (PNG), Mapa de Bits (BMP), entre otros. También hay que destacar que el tamaño de la imagen no influye en la precisión de los resultados, solo produce variaciones en el tiempo de procesamiento como se muestra en el Experimento 3.

VI. Conclusiones

Las imágenes digitales contienen una gran cantidad de información relevante. Debido a esto, son un elemento muy importante en el ámbito legal y se han convertido en evidencias que aportan gran valor en la resolución de un juicio. Para que estas evidencias lleguen a ser válidas se debe poder garantizar su autenticidad e integridad de forma fiable. Existen numerosas aplicaciones que consiguen editar imágenes con resultados altamente profesionales y detectar si una imagen ha sido modificada mediante alguna técnica de manipulación es una tarea complicada. Para poder garantizar la integridad de una imagen es de mucho interés tener herramientas forenses que puedan detectar estas falsificaciones. En este trabajo se ha realizado un estudio exhaustivo sobre las técnicas existentes de detección de manipulaciones haciendo énfasis en las técnicas de detección de *Copy – Move*. Se han estudiado en profundidad las técnicas que dan los mejores resultados, analizando el proceso que realiza para la detección. Se ha diseñado una técnica para la detección de la región exacta duplicada en técnicas *Copy – Move*. Para evaluar la técnica diseñada en este trabajo, se han realizado pruebas con un conjunto numeroso de imágenes de diferentes texturas, dimensiones y formatos. Los resultados muestran que el algoritmo detecta de forma precisa las áreas duplicadas en imágenes de texturas similares. De igual forma, muestra excelentes resultados cuando la imagen contiene un alto nivel de detalles similares representados por un mismo patrón de color. Este algoritmo presenta un tiempo de ejecución superior a los demás ya que hace uso de numerosos cálculos para determinar la zona duplicada. Pero como se ha observado en los experimentos es posible escalar una imagen y así mejorar la eficiencia del mismo sin perder calidad en los resultados. Sin embargo, el algoritmo presenta dificultades en imágenes cuya área duplicada ha sido modificada posteriormente con partes de la imagen original.

Agradecimientos

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 700326. 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] CISCO, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021," https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html, February 2017.

[2] ERICSSON, "Ericsson Mobility Report," ERICSSON, Tech. Rep., 06 2018. [Online]. Available: \url{https://www.ericsson.com/assets/local/mobility-report/documents/2018/ericsson-mobility-report-june-2018.pdf}

[3] M. Sapir, "The impossible photograph: Hippolyte bayard's self-portrait as a drowned man," *MFS Modern Fiction Studies*, vol. 40, no. 3, pp. 619–629, 1994.

[4] H. Farid, "Creating and Detecting Doctored and Virtual Images: Implications to the Child Pornography Prevention Act." UnDartmouth College, Technical Report, September 2004.

[5] E. Mundo, "Detenido por Circular a 200 Kilómetros por Hora tras Subir un Vídeo a Redes Sociales," http://www.elmundo.es/madrid/2017/08/30/59a68f0a468aeb7a658b4607.html, August 2017.

[6] 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.

[7] M. A. Qureshi and M. Deriche, "A Bibliography of Pixel-Based Blind Image Forgery Detection Techniques," *Signal Processing: Image Communication*, vol. 39, pp. 46–74, 2015.

[8] M. Boutell and J. Luo, "Beyond pixels: Exploiting camera metadata for photo classification," *Pattern Recognition*, vol. 38, no. 6, pp. 935–946, 2005.

[9] H. Huang, W. Guo, and Y. Zhang, "Detection of copy-move forgery in digital images using sift algorithm," in *Proceedings of the IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, vol. 2, Wuhan, China, December 2008, pp. 272–276.

[10] I. Fourandsix Technologies, "Photo Tampering Throughout History," https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html, December 2017.

[11] X. Zhao, S. Wang, S. Li, J. Li, and Q. Yuan, "Image splicing detection based on noncausal markov model," in *Proceedings of the IEEE International Conference on Image Processing*, Melbourne, VIC, Australia, September 2013, pp. 4462–4466.

[12] J. Dong and W. Wang, "CASIA TIDE v1.0 - v2.0," http://forensics.idealtest.org/.

[13] V. Sun, "Photos: 20 More Stars and Celebrities Before and After Photoshop," http://www.vancouversun.com/life/fashion-beauty/Photos+more+stars+celebrities+before+after+Photoshop/7841314/story.html, July 2014.

[14] I. T. Young, J. J. Gerbrands, and L. J. Van Vliet, *Fundamentals of image processing*. Delft University of Technology Delft, 1998.

[15] E. Kee and H. Farid, "A Perceptual Metric for Photo Retouching," *National Academy of Sciences*, vol. 108, no. 50, pp. 19 907–19 912, November 2011.

[16] L. J. García Villalba, A. L. Sandoval Orozco, J. Rosales Corripio, and J. Hernández Castro, "A PRNU-based Counter-forensic Method to Manipulate Smartphone Image Source Identification Techniques," *Future Generation Computer Systems*, vol. 76, pp. 418–427, November 2017.

[17] N. Khanna, A. K. Mikkilineni, G. Chiu, J. P. Allebach, and E. Delp, "Forensic Classification of Imaging Sensor Types," in *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 6505, no. 65050U, February 2007.
[18] B. Mahdian and S. Saic, "A Bibliography on Blind Methods for Identifying Image Forgery," *Signal Processing: Image Communication*, vol. 25, no. 6, pp. 389–399, July 2010.
[19] J. Fridrich, D. Soukal, and J. Lukas, "Detection of Copy Move Forgery in Digital Images," in *Proceedings of the Digital Forensic Research Workshop*, Binghamton, New York, August 2003, pp. 5–8.
[20] A. C Popescu and H. Farid, "Exposing Digital Forgeries by Detecting Duplicated Image Regions," *Department of Computer Science*, vol. 646, January 2004.
[21] A. Langille and M. Gong, "An efficient match-based duplication detection algorithm," in *Computer and Robot Vision, 2006. The 3rd Canadian Conference on*. IEEE, 2006, pp. 64–64.
[22] W. Luo, J. Huang, and G. Qiu, "Robust detection of region-duplication forgery in digital image," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 4. IEEE, 2006, pp. 746–749.
[23] A. D. Warbhe, R. V. Dharaskar, and V. M. Thakare, "International journal of engineering sciences & research technology block based image forgery detection techniques."
[24] A. Myna, M. Venkateshmurthy, and C. Patil, "Detection of region duplication forgery in digital images using wavelets and log-polar mapping," in *Conference on Computational Intelligence and Multimedia Applications, 2007. International Conference on*, vol. 3. IEEE, 2007, pp. 371–377.
[25] Q. Wu, S. Wang, and X. Zhang, "Log-polar based scheme for revealing duplicated regions in digital images," *IEEE Signal Processing Letters*, vol. 18, no. 10, pp. 559–562, 2011.
[26] G. Li, Q. Wu, D. Tu, and S. Sun, "A sorted neighborhood approach for detecting duplicated regions in image forgeries based on dwt and svd," in *Multimedia and Expo, 2007 IEEE International Conference on*. IEEE, 2007, pp. 1750–1753.
[27] H. Huang, W. Guo, and Y. Zhang, "Detection of Copy-Move Forgery in Digital Images Using SIFT Algorithm," in *2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, vol. 2, December 2008, pp. 272–276.
[28] X. Bo, W. Junwen, L. Guangjie, and D. Yuewei, "Image Copy-Move Forgery Detection Based on SURF," in *In proceedings of the International Conference on Multimedia Information Networking and Security*, Nanjing, China, November 2010, pp. 889–892.
[29] H. Farid, "Detecting digital forgeries using bispectral analysis," 1999.
[30] T.-T. Ng and S.-F. Chang, "A model for image splicing," in *Proceedings of the International Conference on Image Processing, 2004*, vol. 2. IEEE, 2004, pp. 1169–1172.
[31] D. Fu, Y. Q. Shi, and W. Su, "Detection of image splicing based on hilbert-huang transform and moments of characteristic functions with wavelet decomposition," in *International workshop on digital watermarking*. Springer, 2006, pp. 177–187.
[32] Y.-F. Hsu and S.-F. Chang, "Detecting image splicing using geometry invariants and camera characteristics consistency," in *Multimedia and Expo, 2006 IEEE International Conference on*. IEEE, 2006, pp. 549–552.
[33] Y. Q. Shi, C. Chen, and W. Chen, "A natural image model approach to splicing detection," in *Proceedings of the 9th workshop on Multimedia & security*. ACM, 2007, pp. 51–62.
[34] Z. Zhang, J. Kang, and Y. Ren, "An effective algorithm of image splicing detection," in *Computer Science and Software Engineering, 2008 International Conference on*, vol. 1. IEEE, 2008, pp. 1035–1039.
[35] Q. Liu and A. H. Sung, "A new approach for jpeg resize and image splicing detection," in *Proceedings of the First ACM workshop on Multimedia in forensics*. ACM, 2009, pp. 43–48.
[36] W. Zhang, X. Cao, J. Zhang, J. Zhu, and P. Wang, "Detecting photographic composites using shadows," in *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*. IEEE, 2009, pp. 1042–1045.
[37] W. Zhang, X. Cao, Y. Qu, Y. Hou, H. Zhao, and C. Zhang, "Detecting and extracting the photo composites using planar homography and graph cut," *IEEE transactions on information forensics and security*, vol. 5, no. 3, pp. 544–555, 2010.
[38] X. Zhao, J. Li, S. Li, and S. Wang, "Detecting digital image splicing in chroma spaces," in *International Workshop on Digital Watermarking*. Springer, 2010, pp. 12–22.
[39] Q. Liu, X. Cao, C. Deng, and X. Guo, "Identifying image composites through shadow matte consistency," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1111–1122, 2011.
[40] N. B. A. Warif, M. Y. I. Idris, A. W. A. Wahab, and R. Salleh, "An evaluation of error level analysis in image forensics," in *System Engineering and Technology (ICSET), 2015 5th IEEE International Conference on*. IEEE, 2015, pp. 23–28.
[41] T. S. Gunawan, S. A. M. Hanafiah, M. Kartiwi, N. Ismail, N. F. Za'bah, and A. N. Nordin, "Development of photo forensics algorithm by detecting photoshop manipulation using error level analysis," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 7, no. 1, pp. 131–137, 2017.
[42] D. C. Jeronymo, Y. C. C. Borges, and L. dos Santos Coelho, "Image forgery detection by semi-automatic wavelet soft-thresholding with error level analysis," *Expert Systems with Applications*, vol. 85, pp. 348–356, 2017.
[43] A. Alahmadi and M. Hussain, "Passive Detection of Image Forgery Using DCT and Local Binary Pattern," *Signal, Image and Video Processing*, vol. 11, no. 1, pp. 81–88, January 2017.
[44] Y. Q. Shi, C. Chen, and W. Chen, "A Natural Image Model Approach to Splicing Detection," in *Proceedings of the 9th workshop on Multimedia & security*, Dallas, Texas, September 2007, pp. 51–62.
[45] X. Zhao and J. Li, "Detecting Digital Image Splicing in Chroma Spaces," in *Digital Watermarking*, vol. 6526. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 12–22.
[46] J. R. Corripio, D. M. Arenas González, A. L. Sandoval Orozco, L. J. García Villalba, J. Hernandez-Castro, and S. J. Gibson, "Source smartphone identification using sensor pattern noise and wavelet transform," 2013.
[47] A. L. Sandoval Orozco, D. M. Arenas González, J. R. Corripio, L. G. Villalba, and J. C. Hernandez-Castro, "Source identification for mobile devices, based on wavelet transforms combined with sensor imperfections," *Computing*, vol. 96, no. 9, pp. 829–841, 2014.
[48] A. L. Sandoval Orozco, D. M. Arenas González, J. Rosales Corripio, L. J. García Villalba, and J. C. Hernandez-Castro, "Source Identification for Mobile Devices, Based on Wavelet Transforms Combined with Sensor Imperfections," *Computing*, vol. 96, no. 9, pp. 829–841, September 2014.
[49] A. Bharati, R. Singh, M. Vatsa, and K. W. Bowyer, "Detecting Facial Retouching Using Supervised Deep Learning," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 1903–1913, September 2016.
[50] C. Chen, A. Dantcheva, and A. Ross, "Automatic Facial Makeup Detection with Application in Face Recognition," in *Proceedings of the International Conference on Biometrics (ICB)*, Madrid, Spain, June 2013, pp. 1–8.
[51] A. Dantcheva, C. Chen, and A. Ross, "Can Facial Cosmetics Affect the Matching Accuracy of Face Recognition Systems?" in *Proceedings of the IEEE 5th International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. Washington DC, USA: IEEE, September 2012, pp. 391–398.
[52] N. Kose, L. Apvrille, and J. L. Dugelay, "Facial Makeup Detection Technique Based on Texture and Shape Analysis," in *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, vol. 1, Ljubljana, Slovenia, May 2015, pp. 1–7.
[53] Q. Fu, X. Zhou, C. Wang, and B. Jiang, "Mathematical relation between APBT-based and DCT-based JPEG image compression schemes," *Journal of Communications*, vol. 11, pp. 84–92, January 2016.
[54] I. IFS-TC, "IFS-TC Image Forensics Challenge," http://ifc.recod.ic.unicamp.br/, January 2014.
[55] Z. Zhang, D. Wang, C. Wang, and X. Zhou, "Detecting Copy-move Forgeries in Images Based on DCT and Main Transfer Vectors," *KSII Transactions on Internet and Information Systems*, vol. 11, pp. 4567–4587, September 2017.

Esteban Alejandro Armas Vega received his Computer Science degree in 2009 at the Polytechnic Institute "José Antonio Echeverría" in Havana (Cuba) and a M.Sc. degree in Computer Science in 2016 from the Universidad Complutense de Madrid (Spain). He is currently a Ph.D. student in the Department of Software Engineering and Artificial Intelligence of the Faculty of Computer Science and Engineering at the Universidad Complutense de Madrid (UCM) and Member of the Complutense Research Group GASS (Group of Analysis, Security and Systems, http://gass.ucm.es). His research interests include computer networks and computer security.

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 and member of the Research Group GASS (Group of Analysis, Security and Systems, http://gass.ucm.es) 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 Ph.D. in Computer Science (1999) 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 Faculty of Computer Science and Engineering at the UCM Campus. His professional experience includes the management of both national and international research projects and both public (Spanish Ministry of R&D, Spanish Ministry of Defence, Horizon 2020 - European Commission, . . .) and private financing (Hitachi, IBM, Nokia, Safelayer Secure Communications, TB Solutions Security, . . .). Author or co-author of numerous international publications is editor or guest editor of numerous journals such as Entropy MPDI, Future Generation Computer Systems (FGCS), Future Internet MDPI, IEEE Latin America Transactions, IET Communications (IET-COM), IET Networks (IET-NET), IET Wireless Sensor Systems (IET-WSS), International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC), International Journal of Multimedia and Ubiquitous Engineering (IJMUE), Journal of Supercomputing, Sensors MDPI, etc.