Método General para la Detección de Imágenes Alteradas Utilizando Técnicas de Compresión

A General Method Based on Data Compression for Manipulated Image Detection

Avid Roman-Gonzalez[1, 2], Camilo J. Reynaga-Cardenas[2] y Cristhian Ganvini-Valcarcel[2]

[1] TELECOM ParisTech, 46 rue Barrault, 75013 – Paris, France
[2] Universidad Andina del Cusco

Resumen

En la actualidad, con el incremento desmedido de la cantidad de información digital, ya sea texto, audio, video o imágenes y el avance acelerado de la tecnología; es muy fácil poder realizar algunas alteraciones en este tipo de datos, alteraciones que pueden ser o no visibles. Estas alteraciones puede tener diferentes objetivos, como por ejemplo el de introducir una marca de agua para proteger los derechos de autor, o para introducir un mensaje oculto que se desea enviar en secreto, también están las alteraciones en adicionar o remover algún tipo de información para alterar la información original con algún propósito ya sea bueno o malo. Frente a todas estas diversas posibilidades de alteración que muchas veces no pueden ser detectadas a simple vista; pues se hace necesario el desarrollo e implementación de métodos automáticos que nos permitan identificar si un dato fue alterado o no. En el presente trabajo nos enfocaremos netamente en la detección de imágenes alteradas o manipuladas.

Descriptores: *Análisis de imágenes, watermarking, esteganalisis, falsificación de imágenes, rate-distortion, complejidad de Kolmogorov, detección.*

Abstract

Currently, with the excessive increase in the amount of digital information, whether text, audio, video or images and the acceleration of technology, it is very easy to make some alterations in this type of data that can be visible or not. These alterations may have different objectives, such as introducing a watermark to protect the copyright, or to enter a hidden message to be sent in secret, also alterations to add or remove any information for alter the original information for any purpose, whether good or bad. Faced with all these different possibilities of alterations that often cannot be detected by a simple visual examination; that is why it is necessary to develop and implement automated methods that allow us to identify when a data is altered. In this paper we focus only on the detection of altered or manipulated images.

Keywords: *Image analysis, watermarking, steganalysis, image fakery, rate-distortion, Kolmogorov Complexity, detection*

1. Introducción

El desarrollo de las tecnologías de información ha permitido la integración de muchos dispositivos electrónicos entre ellos las cámaras fotográficas, que inicialmente funcionaban con rollos de película, en la actualidad es usual compartir la información de una cámara digital, es decir las fotografías, copiarlas en el disco duro de la computadora, enviarlas por correo electrónico o subirlas en la red social preferida.

Existen además aplicaciones de software que permiten alterar fotografías, en muchos casos sin que el usuario necesite ser experto, se tiene software para todo tipo de usuarios. Es posible usar aplicaciones sencillas como el Paint de Microsoft Windows, o más complejas como: Adobe Photoshop, Corel Draw o GIMP, si se prefiere una

solución GNU, para cambiar la esencia o forma de la fotografía, pudiendo ser el fin de la alteración bueno o malo.

Las alteraciones de imagen pueden ser detectadas a simple vista o no, resultando en la mayoría de casos un proceso complicado, dado que no es fácil determinar el tipo de alteración que se ha podido realizar. La imagen ha podido ser alterada para enviar información secreta, técnica conocida como esteganografía, o se ha podido quitar alguna parte de la imagen sobreponiendo otras partes de la misma imagen haciendo parecer que existe menos personas o que el paisaje es diferente, también es posible añadir una marca de agua "Watermarking" a la fotografía para resguardar los derechos de autor y prevenir el plagio. Sea cual sea el método de alteración utilizado la imagen termina siendo distinta a la original.

Actualmente existe varios trabajos de investigación los cuales han desarrollado métodos para realizar la detección automática de imágenes manipuladas, pero estos métodos son específicos; vale decir que estos métodos son orientados hacia la detección de un caso específico de manipulación como por ejemplo: Cambios en la luminosidad de la imagen, remuestreo de partes de la imagen, duplicación de zonas, inconsistencias en las sombras, inconsistencias en el ruido, etc. Por ello la utilización de estos métodos no asegura una detección correcta de una imagen manipulada cuando no se conoce el tipo de alteración. Por esa razón se hace necesaria la implementación de un método más general para la detección de imágenes alteradas sin importar el tipo de manipulación, para lograr este propósito, utilizaremos técnicas de compresión de datos.

Como antecedentes tenemos muchos trabajos de investigación dedicados a la detección de algún tipo de manipulación en una imagen, así como también trabajos de investigación relacionados con la detección de marcas de agua y esteganografia. Pero como ya se mencionó anteriormente, estos trabajos tienen la peculiaridad de estar diseñados para identificar algún tipo específico de manipulación o alteración; podemos mencionar algunos trabajos como: Babak y Stabislav en 2008 [1] presentaron un método para la detección de partes remuestreadas e inconsistencia en el ruido para imágenes alteradas; los mismos autores en el 2010 [2] presentan otros métodos para la detección de regiones duplicadas; en el 2009 Hany Farid [3] presenta un método para la detección de inconsistencias en la iluminación basado en el reflejo de la luz en los ojos de los protagonistas. También existe un método más general presentado por Ismail Avcibas en 2003 [4] que usa métricas de la calidad de imagen para detectar información oculta en imágenes.

El presente estudio tiene como objetivo implementar un método general que permita detectar si una imagen fue alterada, tomando como basé la compresión de datos.

2. Marco Teórico

Calidad de la Imagen

Las imágenes digitales están siempre sujetas a una variedad de distorsiones y modificaciones durante los procesos de compresión, transmisión, reproducción, etc.

Para tener el control y alguna posibilidad de mejorar la calidad de la imagen, es importante poder medir e identificar la calidad y la degradación de calidad en nuestros datos.

Los trabajos de investigación relacionados con la evaluación de la calidad de imágenes, tienen como objetivo el desarrollo de métodos y algoritmos que puedan evaluar de manera automática la calidad de una imagen; por ejemplo en [5] se muestra un interesante método basado en el enfoque "reduce-reference" para la evaluación automática de la calidad de una imagen; en [6] y [7] se muestran otros métodos para medir la calidad visual de una imagen; en [8] los autores presentan una evaluación de las diferentes métricas de calidad de imágenes. Para lograr este propósito, algunos métodos utilizan medidas de comparación frente a una referencia. En ese sentido tenemos 3 enfoques: el enfoque "*full-reference*" (*FR*), el enfoque "*non-reference*" (*NR*) y el enfoque "*reduced reference*" (*RR*) [5].

Watermarking

Es una técnica cuyo objetivo principal es poner de manifiesto el uso ilícito de un cierto servicio digital por parte de un usuario no autorizado. Concretamente, esta técnica consiste en insertar un mensaje (oculto o no) en el interior de un objeto digital, como podrían ser imágenes, audio, vídeo, texto, software, etc. Dicho mensaje es un grupo de bits que contiene información sobre el autor o propietario intelectual del objeto digital tratado.

Existen varias técnicas de watermarking, se puede considerar marcas visibles y no visibles.

Para insertar una marca visible podemos seguir los siguientes pasos: denotamos a la imagen original como f, la marca como w, y la imagen marcada como f_w; finalmente aplicamos el siguiente proceso:

$$f_w = (1-\alpha) f + \alpha w$$

Dónde: α es una constante de visibilidad de la marca.

Si deseamos introducir una marca no visible, pues esta no será distinguible de manera visual, pero será posible detectarla o recuperarla utilizando códigos y algoritmos orientados a dicho fin. La invisibilidad es asegurada por la inserción de información redundante.

Por ejemplo podemos insertar la marca en los 2 últimos bits menos significativos de la imagen de acuerdo con:

$$f_w = 4(\frac{f}{4}) + \frac{w}{64}$$

Esteganografia

La esteganografia consiste en técnicas de ocultamiento de información, existen muchos trabajos relacionados al ocultamiento de la información así como a la detección de información oculta, en [4] los autores muestran un método de esteganalisis usando métricas de calidad para imágenes, en [9] y [10] se presenta una introducción de modelos teóricos para esteganalisis y ocultamiento de información, en los trabajos [11] y [12] los autores presentan otros métodos de esteganalisis. El problema de la información oculta se traduce en lo siguiente: Tener un mensaje M que puede ser embebido en un dato S y como resultado dar X, este X puede ser sujeto a diversos procesos e intentos de ataque.

Un sistema de ocultamiento de información debe cumplir 2 requerimientos: X debe ser muy similar a S; y el mensaje M oculto debe sobrevivir a distintos procesos (compresión, redimensionamiento, etc.).

Manipulación de Imágenes

El arte de falsificar imágenes tiene una historia larga, y hoy en día que es la era digital es posible realizar cambios en la información representada de manera muy fácil sin dejar rastros de la manipulación.

Existen métodos para la detección de imágenes falsificadas como los descritos en [3] y [13]; en [14] y [2] los autores presentan un método a ciegas para la detección de falsificaciones en imágenes; en [1] se muestra un método para la detección de secciones remuestreadas. Algunos de estos métodos se basan en los siguientes principios:

- Regiones duplicadas.
- Interpolación y remuestreo.
- Inconsistencias en el color.
- Inconsistencias en el ruido.
- Inconsistencias en el Color Filter Array (CFA).
- Inconsistencias en la iluminación.

La detección de inconsistencias en la iluminación es muy importante y permite detectar la manipulación de imágenes.

La Función Rate-Distortion

La función Rate-Distortion (*RD*) está dada por el mínimo valor de información mutual (mutual information) entre la fuente y el receptor bajo ciertas restricciones de distorsión.

$$R(D) = \min_{Q \in Q_D} I(p,Q)$$

Dónde: $I(p,Q)$ es la información mutual entre p y Q.

La función *RD* muestra el error de compresión dado para diferentes factores de compresión.
La función *RD* es el límite aceptable de distorsión para un factor de compresión dado. La función *RD* mide de manera indirecta la complejidad visual de una imagen.

La Complejidad de Kolmogorov

La complejidad de Kolmogorov $K(x)$ de una cadena x se define como la longitud del programa más corto capaz de producir x en una maquina universal, como una máquina de Turing. Los diferentes lenguajes de programación darán lugar a distintos valores de $K(x)$, pero se puede probar que las diferencias son sólo hasta una constante aditiva fija. Intuitivamente, $K(x)$ es la cantidad mínima de información necesaria para generar x a través de un algoritmo.

$$K(x) = \min_{q \in Q_x} |q|$$

Donde: Q_x es el conjunto de códigos que generan instantáneamente x. Dado que los programas pueden ser escritos en diferentes lenguajes de programación, $K(x)$ se mide a una constante aditiva no en función de los objetos sino en la máquina de Turing empleada. Una interpretación es la cantidad de información necesaria para recuperar x desde el principio, las cadenas que presentan patrones recurrentes son de baja complejidad, mientras que la

complejidad de cadenas al azar es alta y es casi igual a su longitud. La propiedad principal de $K(x)$ es que no puede ser calculable.

Función Estructura de Kolmogorov

Una aproximación de la curva *RD* usando la teoría de complejidad de Kolmogorov podría ser la Función Estructura de Kolmogorov (*KSF*).

En [15] los autores presentan un análisis con respecto a la Función Estructura de Kolmogorov.

La relación entre un dato individual y su explicación (modelo) esta expresado por la Función Estructura de Kolmogorov.

La Función Estructura de Kolmogorov original para un dato x es definida por:

$$h_x(\alpha) = \min_S \{\log|S| : S \ni x, K(S) \leq \alpha\}$$

Dónde: S es un conjunto que contempla los modelos para x.
α es un valor entero no negativo que representa el límite de la complejidad de S.

La Función Estructura de Kolmogorov $h_x(\alpha)$ nos proporciona las propiedades estocásticas de un dato x. [15].

La Función Estructura de Kolmogorov es una función no calculable ya que la complejidad de Kolmogorov también es una función no calculable; es por ello que usamos el factor de compresión como una aproximación de la complejidad. La idea es usar la *KSF* como una aproximación del análisis *RD* para la describir el comportamiento de las imágenes que contienen manipulaciones y así desarrollar un método general para su detección.

La teoría de la *KSF* también presenta la función "Best Fit" (Mejor Ajuste).

$$\beta_x(\alpha) = \min_S \{\delta(x \mid S) : S \ni x, K(S) \leq \alpha\}$$

Dónde: $\delta(x \mid S)$ es la deficiencia aleatoria de x en S: $\delta(x \mid S) = \log \mid S \mid -K(x \mid S)$

El estimador para la Longitud de Descripción Mínima (*MDL*) está definida por:

$$\lambda_x(\alpha) = \min_S \{\Lambda(S) : S \ni x, K(S) \leq \alpha\}$$

Dónde: $\Lambda(S) = \log \mid S \mid + K(s)$ es la longitud de ambos códigos de x con la ayuda del modelo S.

Compresor ZIP

Este es un solo paso de codificación basado en la combinación del código LZW y el código de Huffman. El archivo de entrada es dividido en secuencia de bloques donde cada bloque es comprimido usando dicha combinación de códigos.

Compresor JPEG

"JPEG" significa "Joint Photographic Experts Group" (Grupo conjunto de expertos en fotografía), nombre de la comisión que creó la norma, la cual fue integrada desde sus inicios por la fusión de varias agrupaciones en un intento de compartir y desarrollar su experiencia en la digitalización de imágenes.

El JPEG-LS es una forma de codificación JPEG sin pérdidas. Aunque ésta no es muy utilizada por la comunidad de procesamiento de datos en general, se utiliza especialmente para la transmisión de imágenes médicas a fin de evitar que se produzcan artefactos en la imagen (exclusivamente dependientes de la imagen y su digitalización) y confundirlos con signos patológicos reales. De esta manera, la compresión resulta mucho menos efectiva.

JPEG – BaseLine Es un algoritmo de compresión con pérdida, esto significa que al descomprimir la imagen no obtenemos exactamente la misma imagen que teníamos antes de la compresión. El compresor JPEG con pérdidas toma una imagen y primero lo divide en bloques de 8x8 pixeles, a cada bloque le aplica la Transformada Discreta Coseno (DCT) para luego aplicar un cuantificador y finalmente un codificador de entropía y así obtener la imagen comprimida; vale resaltar que la perdida de la información se encuentra en la parte del cuantificador.

3. Método para la Detección de Imágenes Manipuladas

Como ya se mencionó anteriormente, en la actualidad se cuenta con métodos específicos para la identificación de una imagen alterada dependiendo del tipo de alteración. En esta sección presentamos nuestra propuesta de método general para la identificación de imágenes alteradas sin importar el tipo de alteración.

Análisis de Imágenes Utilizando la Curva Rate-Distortion

Se puede decir que la función Rate-Distortion (*RD*) mide de manera indirecta la complejidad visual de las imágenes, por ejemplo, graficando la curva experimental *RD* donde el eje horizontal representa el factor de compresión (tamaño del archive de la imagen comprimida / tamaño del archive de la imagen original) y el eje vertical representa la distorsión calculada utilizando el Mean Square Error (*MSE*); podemos hacer un análisis de la imagen.

Podemos decir que la falsificación de una imagen puede alterar la curva experimental *RD* de dicha imagen, por ejemplo podemos observar la Figura 1 que presenta en (a) una imagen original y en (b) la misma imagen pero con una alteración donde se ha eliminado a la persona.

(a) Fotografía Original (b) Fotografía Manipulada

Fig. 1: (a) Fotografía de un Salón de Autos Original. (b) es la misma imagen (a) pero se borró a la persona y fue sustituida por duplicación de regiones.

Al realizar el análisis de las imágenes de la Figura 1 (a) y (b) utilizando la gráfica de las curvas experimentales *RD*, se puede observar que existe una variación en dichas curvas debido a la manipulación de la imagen. La Figura 2 muestra dicha variación, siendo la curva azul para la Figura 1 (b) y la curva en verde para la Figura 1 (a).

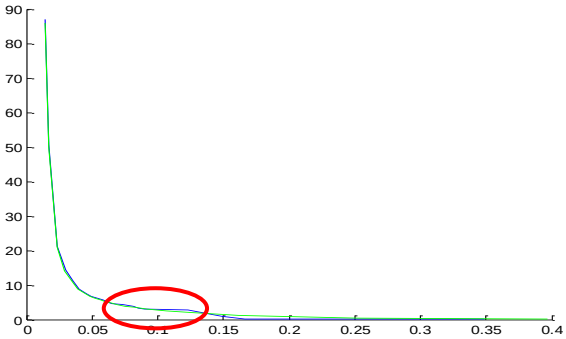

Fig. 2 Curva Experimental Rate-Distortion, el eje horizontal representa el factor de compresión (tamaño del archive de la imagen comprimida / tamaño del archive de la imagen original), el eje vertical representa la distorsión calculada utilizando el MSE (mean square error). Azul para la Figura 1 (a) y verde para la Figura 1 (b), la curva experimental RD de la imagen original es diferente a la curva experimental RD de la imagen que contiene manipulaciones.

Otro ejemplo se puede ver en la Figura 3 (a) y (b) y sus respectivas curvas experimentales *RD* que se muestran en la Figura 4. Se puede observar la misma variación del ejemplo anterior, nosotros utilizaremos esta variación para poder determinar si una imagen fue sujeta a alteraciones o no.

(a) Fotografía Original (b) Fotografía Manipulada

Fig. 3: (a) Fotografía de un Rodadero Original. (b) es la misma imagen (a) pero se borró a 2 personas y fue sustituida por duplicación de regiones

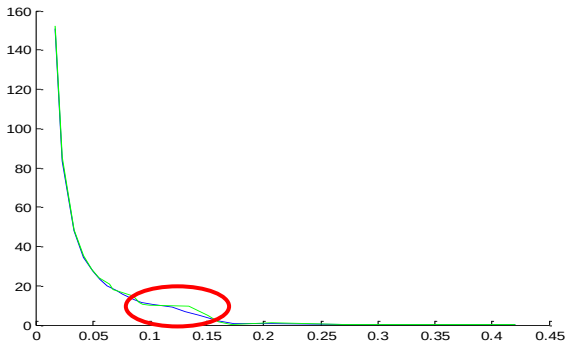

Fig. 4 Curva Experimental Rate-Distortion, el eje horizontal representa el factor de compresión (tamaño del archive de la imagen comprimida / tamaño del archive de la imagen original), el eje vertical representa la distorsión calculada utilizando el MSE (mean square error). Azul para la Figura 3 (a) y verde para la Figura 3 (b), la curva experimental RD de la imagen original es diferente a la curva experimental RD de la imagen que contiene manipulaciones.

Análisis de Imágenes Utilizando la Función Estructura de Kolmogorov

Como ya se vio anteriormente, pues la Función Estructura de Kolmogorv (*KSF*) es una aproximación de la Función Rate-Distoricion utilizando la Teoría de Complejidad de Kolmogorov.

En ese sentido, utilizaremos dicha función para ver los cambios en la curva experimental de la *KSF*

cuando se producen manipulación en una imagen, para ello utilizaremos las mismas imágenes de prueba que se utilizó para el análisis con la curva *RD*. En la Figura 5 podemos observar las curvas *KSF* para las imágenes de la Figura 1, siendo la curva azul para la imagen (a) y la curva verde para la imagen (b). Así mismo en la Figura 6 se muestran las curvas para las imágenes de la Figura 3, azul para la imagen (a) y verde para la imagen (b). Se puede observar que al igual que las curvas *RD*, en las curvas *KFS* se puede observar una variación cuando existe una manipulación de las imágenes; nosotros utilizaremos esta variación para determinar si una imagen fue alterada o no.

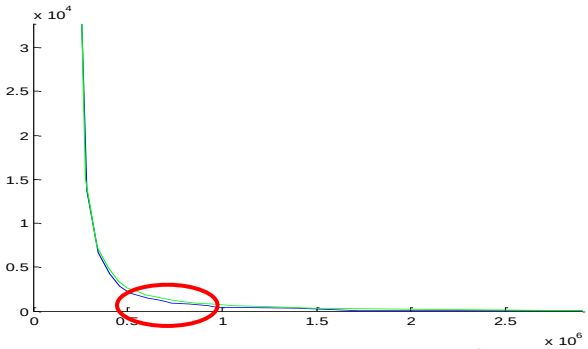

Fig. 5 Curva Experimental para la Función Estructura de Kolmogorov, el eje horizontal representa la aproximación de la Complejidad de Kolmogorov como el tamaño del archivo de la imagen comprimida en bytes, el eje vertical representa la cantidad de bits necesarios para representar un modelo de la imagen original. Curva azul para la Figura 1 (a) y verde para la Figura 1 (b), la curva experimental KFS de la imagen original es diferente a la curva experimental KFS de la imagen que contiene manipulaciones.

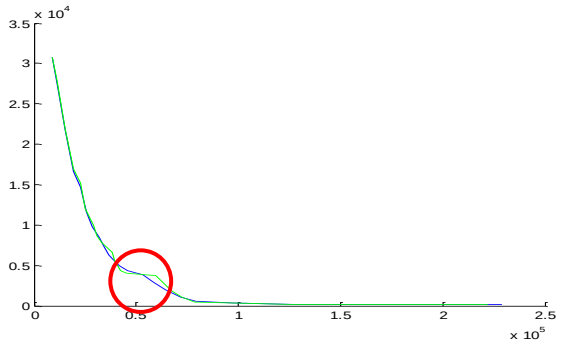

Fig. 6 Curva Experimental para la Función Estructura de Kolmogorov, el eje horizontal representa la aproximación de la Complejidad de Kolmogorov como el tamaño del archivo de la imagen comprimida en bytes, el eje vertical representa la cantidad de bits necesarios para representar un modelo de la imagen original. Curva azul para la Figura 3 (a) y verde para la Figura 3 (b), la curva experimental KFS de la imagen original es diferente a la curva experimental KFS de la imagen que contiene manipulaciones.

Proceso del Método Propuesto

Para la detección de imágenes manipuladas o falsificadas, proponemos la utilización del análisis de la Función Rate-Distortion obteniendo su curva experimental a través de la compresión con pérdidas a diferentes factores de compresión. La idea es examinar cómo se es el comportamiento de la una imagen manipulada y una imagen original frente a esta curva experimental *RD*, como es que una manipulación en la imagen puede alterar la curva experimental *RD*. Para la aplicación de este método se procede como se indica en la Figura 7, primero tomamos la imagen bajo prueba y la comprimimos con diferentes factores de compresión, luego lo descomprimimos y para cada factor de compresión sacamos un error de la comparación con la imagen original; con este conjunto de errores formamos el vector de características para luego aplicarlos a un método de clasificación y así determinar si la imagen fue manipulada o es original. Es necesario recalcar que se necesitará un entrenamiento previo para los métodos de clasificación en base de ejemplos conocidos.

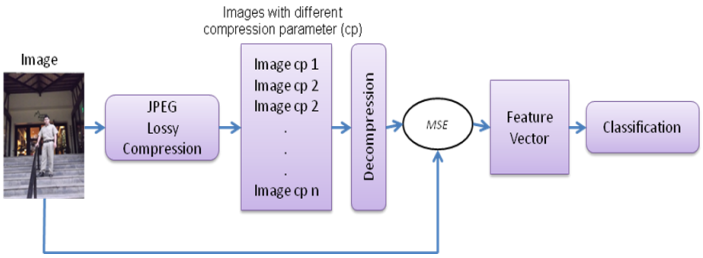

Fig. 7 Diagrama de Bloques del Método Propuesto: tomamos la imagen bajo prueba, lo comprimimos con diferentes factores de compresión, luego lo descomprimimos y calculamos el error con los cuales formamos el vector de características para finalmente aplicarlo a un método de clasificación.

Métodos de Clasificación

Para el presente trabajo, hemos utilizado los siguientes métodos de clasificación:

KNN (K Nearest Neighbors): Algoritmo de clasificación supervisado. Procede a la clasificación de un dato determinado en función a los K (número entero y positivo) vecinos más cercanos al dicho dato en consulta.

SVM (Support Vector Machine): Algoritmo de clasificación supervisado. Toma la mitad de los

datos para entrenamiento dividiéndolos en dos grupos, después crea entre ellos un separador que puede obedecer a propiedades lineales o no lineales. Un dato en consulta es clasificado según de que lado del separador se encuentre.

K-Means: Algoritmo de clasificación no supervisado. Clasifica los datos según los K (número entero y positivo) grupos o centroides que se le asigne. Calcula la mínima distancia de estos centroides a todos lo demás datos ya grupa según la mínima distancia.

Dendrograma: Algoritmo de clasificación no supervisado. Agrupa los datos en función a la mínima distancia euclidiana, formando jerarquías de clasificación.

4. Análisis de Pruebas y Resultados

Base de Datos de Imágenes

Para poder realizar la validación del método propuesto y descrito en la sección anterior, lo aplicamos a una base de datos que contiene 50 imágenes entre originales y alteradas; estas alteraciones pueden ser de supresión de objetos, adición de personas, ambas operaciones, etc. Las imágenes pertenecientes a la nuestra base de datos tienen diferentes tamaños, algunos ejemplos de nuestra base de datos se muestran en las Figura 8 y Figura 9.

Fig. 8 Muestra de algunas imágenes alteradas de nuestra base de datos

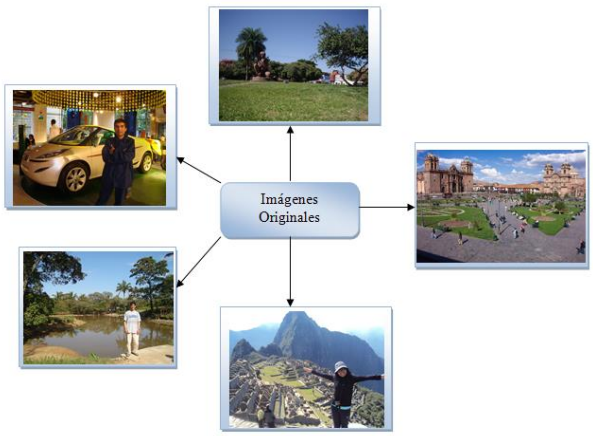

Fig. 9 Muestra de algunas imágenes originales de nuestra base de datos

Proceso del Experimento

Cada uno de las imágenes de nuestra base de datos, ya sean alteradas u originales pasan por un proceso experimental detallado en el diagrama de la Figura 10; es decir, cada imagen es comprimida con diferentes factores de compresión utilizando el compresor con pérdidas JPEG para después realizar la descompresión y compararla con la imagen original; el error resultante de dicha comparación pertenecerá al vector de características que finalmente ingresará al método de clasificación para realizar la detección de imágenes alteradas.

Resultados

Para la realización del experimento se ha utilizado distintos métodos de clasificación como ya se mencionó en el capítulo anterior. Para los métodos supervisados (*KNN* y *SVM*) se ha realizado una validación cruzada, eso quiere decir que se ha utilizado la mitad de las imágenes de la base de datos para el entrenamiento y la clasificación se ha realizado sobre las imágenes restantes, reduciéndose de esta manera a 24 imágenes de prueba, por lo que las matrices de confusión para los métodos supervisados estarán en función de 24 elementos. Para el método de clasificación no supervisada (*kmeans*) se ha trabajado con toda la base de datos competa ya que no se necesita un entrenamiento, por ello la matriz de confusión para este método está basada en 50 elementos.

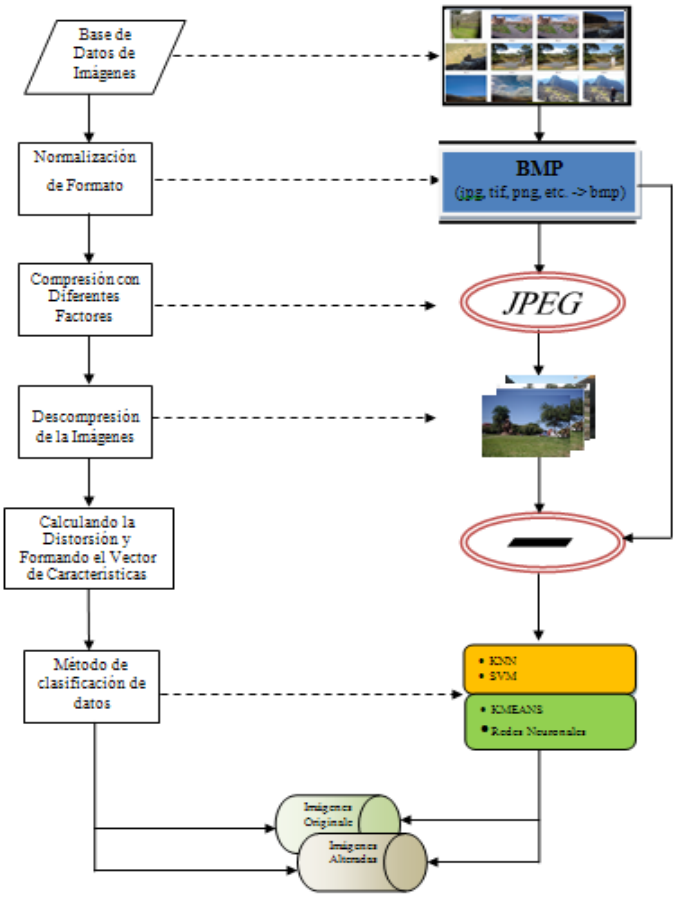

Fig. 10 Diagrama del Proceso del Experimento

Utilizando el Análisis de la Función Rate-Distortion:

Los resultados obtenidos mediante la utilización de la Función Rate-Distortion para cada método de clasificación son los siguientes:

Tabla 1 Matriz de Confusión para el Método de Clasificación KNN Utilizando el Análisis de la Función Rate-Distortion

CLASIFICACION KNN		
	Imágenes Originales	Imágenes Manipuladas
Imágenes Originales	8	3
Imágenes Manipuladas	3	10

Tabla 2 Matriz de Confusión para el Método de Clasificación SVM Utilizando el Análisis de la Función Rate-Distortion

CLASIFICACION SVM		
	Imágenes Originales	Imágenes Manipuladas
Imágenes Originales	7	4
Imágenes Manipuladas	3	10

Tabla 3 Matriz de Confusión para el Método de Clasificación KMEANS Utilizando el Análisis de la Función Rate-Distortion

CLASIFICACION KMEANS		
	Imágenes Originales	Imágenes Manipuladas
Imágenes Originales	17	6
Imágenes Manipuladas	5	22

Un resumen en función de porcentajes de error al momento de hacer la clasificación, se puede observar en la siguiente tabla:

Tabla 4 Matriz de Confusión para el Método de Clasificación KMEANS Utilizando el Análisis de la Función Rate-Distortion

ERROR RATE	
KNN	25 %
SVM	29.17 %
KMEANS	22 %

Utilizando la Función Estructura de Kolmogorov

Los resultados obtenidos mediante la utilización de la Función Estructura de Kolmogorv para cada método de clasificación son los siguientes:

Tabla 5 Matriz de Confusión para el Método de Clasificación KNN Utilizando el Análisis de la Función Estructura de Kolmogorov

CLASIFICACION KNN		
	Imágenes Originales	Imágenes Manipuladas
Imágenes Originales	6	5
Imágenes Manipuladas	3	10

Tabla 6 Matriz de Confusión para el Método de Clasificación SVM Utilizando el Análisis de la Función Estructura de Kolmogorov

CLASIFICACION SVM		
	Imágenes Originales	Imágenes Manipuladas
Imágenes Originales	6	5
Imágenes Manipuladas	3	10

Tabla 7 Matriz de Confusión para el Método de Clasificación KMEANS Utilizando el Análisis de la Función Estructura de Kolmogorov

CLASIFICACION KMEANS		
	Imágenes Originales	Imágenes Manipuladas
Imágenes Originales	14	9
Imágenes Manipuladas	5	22

Un resumen en función de porcentajes de error al momento de hacer la clasificación, se puede observar en la siguiente tabla:

Tabla 8 Matriz de Confusión para el Método de Clasificación KMEANS Utilizando el Análisis de la Función Rate-Distortion

ERROR RATE	
KNN	33.33 %
SVM	33.33 %
KMEANS	28 %

Para poder hacer un análisis de los resultados obtenidos recurriremos a un diagrama de barras que nos muestre el porcentaje de éxito en la clasificación con los diferentes métodos y utilizando ambos análisis propuestos. Este grafico de barras se muestra en la Figura 11.

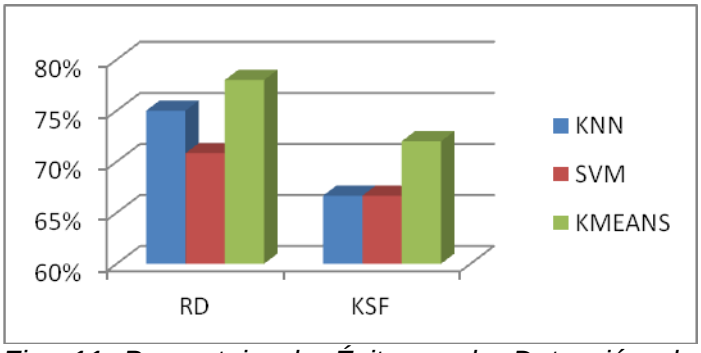

Fig. 11 Porcentaje de Éxito en la Detección de Imágenes Manipuladas Utilizando los Diferentes Métodos de Clasificación y las Diferentes Funciones a Analizar

En el grafico se puede observar que los mejores resultados en cuanto a los métodos supervisados se dan cuando utilizamos la clasificación KNN y analizamos la Función Rate-Distortion. Así mismo, en las matrices de confusión se puede observar que la tendencia es que la clasificación pone a la mayor parte de las imágenes en la parte de imágenes manipuladas, esto puede deberse a que las imágenes que hoy en día se manipulan, todas ya están comprimidas por que vienen en formato JPEG y pueden tener algunos artefactos propios de la compresión que cambian las estadísticas de la imagen y ello provoca una clasificación hacia las imágenes manipuladas.

En cuanto al método de clasificación no supervisado, pues muestra mejores resultados frente a los demás y esto puede deberse ya que al no necesitar de entrenamiento actúa sobre toda la base de datos, mientras los métodos supervisados utilizan solo la mitad de los datos; esto reflejaría una necesidad de mayor cantidad de imágenes en nuestra base de datos para poder realizar un mejor entrenamiento y así obtener mejores resultados.

5. Conclusiones y Recomendaciones

Se desarrolló un método general para la detección de imágenes manipuladas o alteradas basado en técnicas de compresión de datos, básicamente basado en el compresor con pérdidas JPEG y complementado con métodos de clasificación supervisados como KNN y SVM, así como el método de clasificación no supervisado KMEANS. Dicho método fue validado con una base de datos elaborado por nosotros mismos.

Ya sea utilizando una clasificación supervisada o no supervisada, pues los mejores resultados se obtienen al realizar un análisis sobre la Función Rate-Distortion, que finalmente nos muestra una mejor caracterización de lo que sucede en el error al momento de comprimir las imágenes con cierta perdida, estas manipulaciones pueden saltar a la luz durante este proceso.

En cuanto a los métodos supervisados de clasificación, se obtuvo un porcentaje de éxito mayor al utilizar el KNN, siendo este de 75 % de éxito.

En cuanto al método de clasificación no supervisada, se obtuvo un 78% de éxito en la detección.

Estos resultados de 75% y 78% de éxito en la detección, si bien es cierto no son tan buenos, pero teniendo en cuenta lo difícil de esta tarea y que además realizan la detección sin importar el tipo de manipulación o alteración como lo hacen los métodos actuales existentes, pues demuestran que es viable su utilización para una primera etapa de detección general.
El análisis del cambio de la información y la estadística de una imagen conlleva a poder comprenderla mejor, analizar los procesos por los cuales ha pasado y finalmente determinar alguna de sus características.

El hecho de que se haya obtenido mejores resultados con el método de clasificación no supervisado, puede deberse a que dicho método al no necesitar entrenamiento previo, pues utiliza toda la base de datos para hacer la clasificación; mientras los métodos supervisados dividen la base de datos en 2, una para entrenamiento y la otra para la clasificación. Esto nos conlleva a la necesidad de una base de datos más grande para hacer un mejor entrenamiento y así obtener mejores resultados, por lo que se recomienda para próximas investigaciones

aumentar el número de imágenes dentro de la base de datos.

En próximas investigaciones, también se podría realizar mayor experimentación con otros tipos de compresores con pérdidas y también sin perdidas analizando otro tipo de propiedades.

Con la ampliación de la base de datos, pues se recomienda que incluso se pueda armar diferentes bases de datos dependiendo del tipo de imagen y del tipo de contenido en la imagen, pues esto podría ayudar a un entrenamiento mejor y por ende a mejores resultados en la detección, ya que manipular una imagen de paisaje es diferente a manipular una imagen de personas y así con distintos entornos.

Referencias

[1] B. Mahdian, S. Saic; "Detection of Resampling Supplement with Noise Inconsistencies Analysis for Image Forensics", IEEE International Conference on Computational Sciences and Its Applications ICCSA, (2008), pp. 546-556.

[2] B. Mahdian, S. Saic; "Blind Methods for Detecting Image Fakery", IEEE Aerospace and Electronic Systems Magazine, **25** (2010) 18-24.

[3] H. Farid; "Image Forgery Detection", IEEE Signal Processing Magazine, **26** (2009) 16-25.

[4] I. Avcibas, N. Memon, B. Sankur; "Steganalysis Using Image Quality Metrics"; IEEE Transaction on Image Processing, **12** (2003) 221-229.

[5] Z. Wang, G. Wu, H. R. Sheikh, E. P. Simoncelli, E. Yang, A. C. Bovik; "Quality-Aware Images"; IEEE Transaction on Image Processing, **15** (2006) 1680-1689.

[6] H. R. Sheikh, A.C. Bovik; "Image Information and Visual Quality"; IEEE Transaction on Image Processing, **15** (2006) 430-444.

[7] U. Rajashekar, A. C. Bovik, L. K. Cormack; "Visual Search in Noise: Revealing the Influence of Structural Cues by Gaze-contingent Classification Image Analysis"; Journal of Vision, **6** (2006).

[8] H. R. Sheikh, M. F. Sabir, A. C. Bovik; "A Statistical Evaluation of Recent Full Reference Image Quality assessment Algorithms"; IEEE Transaction on Image Processing, **15** (2006) 3441-3452.

[9] P. Moulin, J. A. O'Sullivan; "Information-Theoretic Analysis of Information Hiding"; IEEE Transaction on Information Theory, **49** (2003) 563-593.

[10] C. Cachin; "An Information-Theoretic Model for Steganography"; Information and Computation, **192** (2004) 41-56.

[11] S. Lyu, H. Farid; "Steganalysis Using Higher-Order Image Statistics"; IEEE Transaction on Image Forensics and Security, **1** (2006) 111-119.

[12] K. Suvillan, U. Madhow, S. Chandrasekaran, B. S. Manjunath; "Steganalysis for Markov Cover Data with Applications to Images"; IEEE Transaction on Information Forensics and Security, **1** (2006) 275-287.

[13] J. Fridrich; "Digital Image Forensics", IEEE Signal Processing Magazine, **26** (2009) 26-37.

[14] B. Mahdian, S. Saic; "Blind Authentication Using Periodic Properties of Interpolation", IEEE Transaction on Information Forensics and Security, **3** (2008) 529-538.

[15] N. K. Vereshchagin, P. M. B. Vitanyi; "Kolmogorov's Structure Functions and Model Selection"; IEEE Transaction on Information Theory, **50** (2004) 3265-3290.

.

E-mail: avid.roman-gonzalez@ieee.org