

Marcado de audio mediante scripts de Python usando SVD-STFT

JULIÁN A. LUONGO¹

¹Universidad Nacional de Tres de Febrero, Buenos Aires, Argentina. julianluongo@outlook.com

Resumen – Este trabajo de investigación presenta el estudio basado en el marcado de agua y posterior detección en archivos de audio. Mediante la teoría de SVD y un archivo de imagen en escala blanco y negro puro (Binaria), se genera un script con el lenguaje Python para el marcado y la posterior detección por correlación normalizada, obteniendo gran efectividad en el marcado y detección del audio, y relación señal-ruido dependiente del audio original.

1. Introducción.

El presente trabajo tiene como finalidad el estudio y puesta a prueba, en el área de ingeniería de sonido, de un método de watermarking para señales de audio basado en SVD (Singular Value Decomposition) y su posterior recuperación. Utilizando conceptos de álgebra lineal y procesamiento de señales dentro del área de la programación, se pretende diseñar con Python un algoritmo capaz de introducir y detectar, de manera eficaz, marcas de agua de imágenes en blanco y negro dentro de un archivo de audio de tipo musical y voz hablada.

2. Conceptos relevantes.

Para una completa comprensión del método de Watermarking presentado en este trabajo, resulta relevante la explicación de SVD y STFT.

Singular Value Decomposition o “SVD” es una herramienta escalable muy útil para reducción de datos donde, a partir de un archivo con una gran cantidad de datos (imágenes, videos, audio de alta calidad) y utilizando los principios de factorización de matrices cuadradas y no cuadradas, se reducen estos archivos a sus valores más significativos, denominados “valores singulares” (Figura (1)). Este nuevo conjunto de datos, de tamaño mucho menor al archivo de origen, permitirá tener una reconstrucción muy fiel al archivo original (dependiendo de la cantidad de valores singulares que tomemos del archivo original mediante SVD) [1].

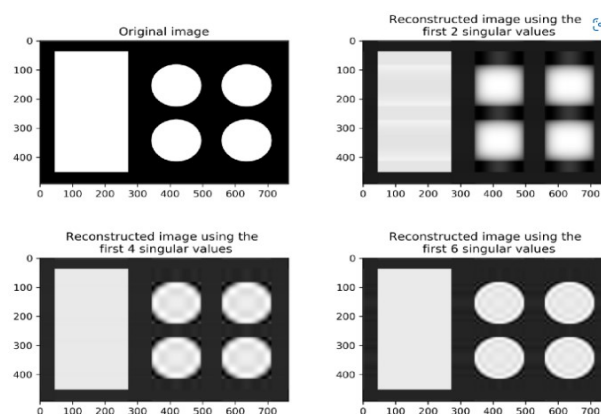


Figura 1. Reconstrucción de imagen en blanco y negro mediante SVD

Por otro lado, la transformada de Fourier de tiempo corto (STFT) es una sucesión de transformadas de Fourier de una señal a intervalos especificados por una ventana, la cual divide la señal en múltiples segmentos, como se aprecia en la Figura (2). El resultado de la STFT estará determinado por los parámetros de la ventana elegida, a saber; tipo de ventana, ancho de ventana y solapamiento de segmentos (“overlap”).

Mientras que la transformada de Fourier estándar proporciona la información de frecuencia promediada durante todo el intervalo de tiempo de la señal, la STFT permite visualizar cómo la información frecuencial varía en el tiempo de duración de la señal [2].

La visualización de STFT, para señales de audio, se realiza a través de su espectrograma, que es un gráfico de frecuencia vs. tiempo. Además, se debe tener en cuenta que existe una compensación entre la resolución de tiempo y frecuencia, dependiendo del ancho elegido para la ventana: una ventana estrecha dará como resultado una mejor resolución en el dominio del tiempo, pero una pobre resolución frecuencial, y viceversa.

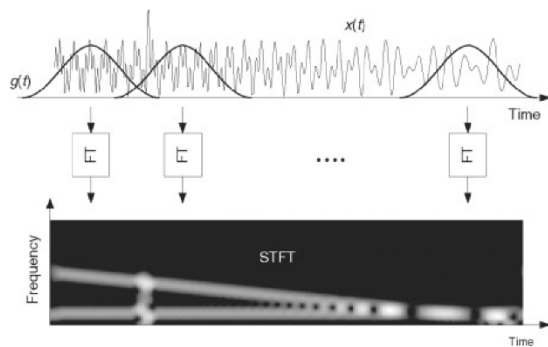


Figura 2. Proceso del algoritmo basado en STFT.

3. Desarrollo experimental: Procedimiento.

El siguiente método se basa principalmente en el procedimiento aplicado en [3]; por lo que el algoritmo de marcado, y en consecuencia el código programado en Python, seguirán los pasos mencionados en dicho documento, aunque con ligeras variaciones.

Todo el procedimiento detallado a continuación se encuentra en el diagrama de bloques adjunto en la Figura (8).

Se presentan los audios a marcar, uno de tipo voz hablada (Figura (3)) y otro de tipo musical (Figura (4)). También se presenta la imagen que servirá como polaridad de bits del pseudo ruido (Figura (5)), la cual es de 8x8 pixeles, dando una totalidad de 64 bits de watermark.

Ambos archivos sonoros poseen una duración de 12 segundos, frecuencia de muestreo de 44.100 KHz y profundidad de 16 bits.

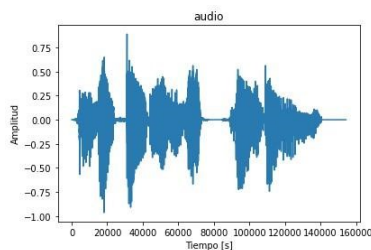


Figura 3. Waveform del audio "speech".

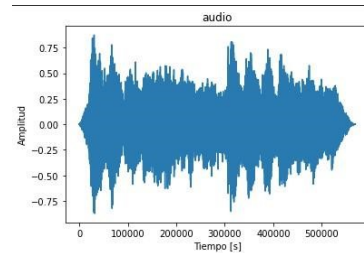


Figura 4. Waveform del audio musical

El audio a marcar es convertido a una matriz de dos dimensiones (tiempo vs frecuencia) mediante STFT (Figura (6) y (7)); luego, esta matriz se divide en múltiples fragmentos -determinados por la cantidad de bits a marcar de la imagen- y posteriormente se aplica SVD a cada fragmento de STFT, obteniendo 3 matrices de interés: la matriz con los valores singulares "D", y las matrices Left-Right "U" y "Vt" respectivamente.

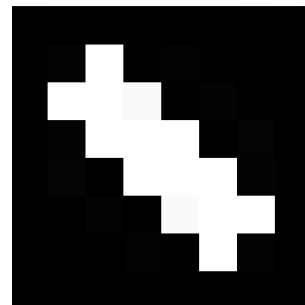


Figura 5. Imagen watermark de 8x8 bits.

Cabe destacar que, al realizar STFT a los audios, para iguales parámetros (Ventana, overlap, ancho de ventana) se aprecia reducción de energía en frecuencias altas para el audio musical. Esto es apreciable también en la escucha, ya que es una característica propia del audio y, como se analiza más adelante en el informe, influye en el comportamiento del audio al marcarse.

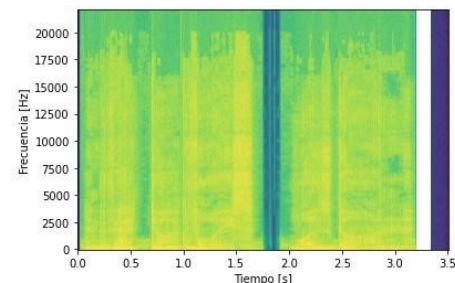


Figura 6. STFT del audio tipo "speech".

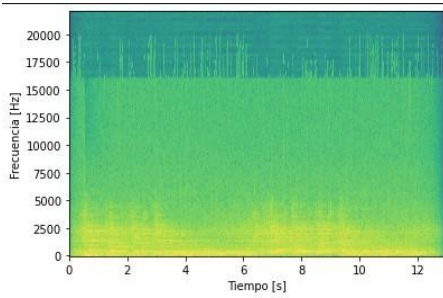


Figura 7. STFT del audio tipo musical.

En paralelo, la imagen establecida para marcar es convertida a matriz binaria de ceros y unos, y cada bit de dicha matriz será embebido en la matriz "D" obtenida en cada fragmento de STFT. Es por ello que la fragmentación de la STFT dependerá del tamaño de la imagen. La fórmula de embebido consta de la adición, a cada ítem de "D", de un bit correspondiente de la imagen, atenuado u amplificado según un factor de fuerza denominado "a" y por un conjunto de datos aleatorio tipo pseudo – ruido "w", generado de forma gaussiana.

Ya que cada valor del pseudo ruido es aleatorio, y modula (junto con el factor de fuerza) la intensidad del bit marcado a lo largo de cada matriz "D", se considera a

dicho ruido como la "llave" que permite detectar la marca de agua y recomponer la imagen. Por ello, al final del procedimiento este ruido es guardado para la posterior detección.

La intensidad de cada valor aleatorio del pseudo ruido influye tanto en la robustez del marcado como a la distorsión producida por el mismo, ya que, a mayores magnitudes de los valores aleatorios, mas intensa será la adición de bits a "D", lo cual provoca una mayor distorsión a los valores singulares y, por lo tanto, mayor distorsión en el audio. La fórmula (1) representa el embebido de cada bit;

$$w_D(i, j) = \delta_{(i, j)} + a \cdot w(i, j) \text{ for } \begin{cases} i = 1, 2, \dots, F \\ j = 1, 2, \dots, M \end{cases} \quad (1)$$

Como resultado del embebido, se genera una matriz a la que se denominará "Matriz marcada W_e ". Ahora bien, a dicha matriz se le realiza un último SVD para obtener una matriz con valores singulares " D_e " (las matrices " U_e " y " V_e " se guardan para la detección). Finalmente, la matriz con los valores singulares " D_e " se multiplica con las matrices Left-Right " U " y " V " obtenidas al principio para conseguir el fragmento de STFT del audio ya marcado. Luego se unen todos los fragmentos marcados y, a través de una STFT inversa, se obtiene el audio marcado.

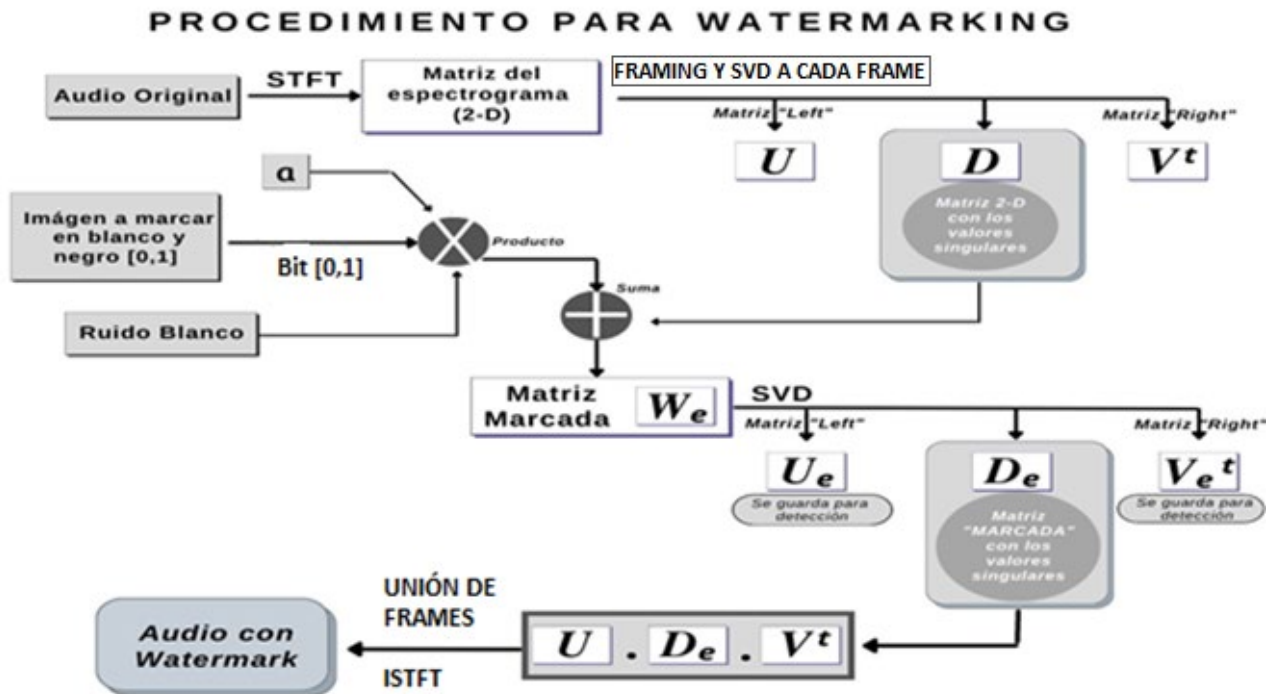


Figura 8. Procedimiento de embebido del watermark.

Detección del Watermarking.

Para la detección del watermark y la posterior reconstrucción de la imagen, basta con aplicar un algoritmo que realice el trabajo inverso al marcado. Todo el procedimiento detallado a continuación se encuentra en el diagrama de bloques adjunto en la Figura (9).

Suponiendo que las matrices " U_e ", " V_e^t ", " D " de cada fragmento, la cantidad de bits del watermark y la llave del watermarking (el pseudo ruido) son conocidos, el procedimiento para detectar la marca resulta el siguiente;

- STFT, fragmentación de la STFT según las dimensiones de la imagen y SVD a cada fragmento.
- Con la matriz de valores singulares obtenida, se realiza el producto matricial;

$$W'_e = U_e \cdot D'_e \cdot V_e^t \quad (2)$$

Así obtenemos la matriz originalmente marcada, y mediante despeje de la fórmula del marcado, obtenemos una matriz con el bit que fue marcado en cada fragmento.

$$\text{BIT} = \frac{(W'_e - D)}{a.\text{ruido}[w]} \quad (3)$$

Los bits obtenidos se unen en un array, y se redimensiona el array según las dimensiones conocidas de la imagen. Finalmente, para mejor resolución de la imagen, se convierte según su valor absoluto a cada valor de bit en ceros y unos (Figura 10).

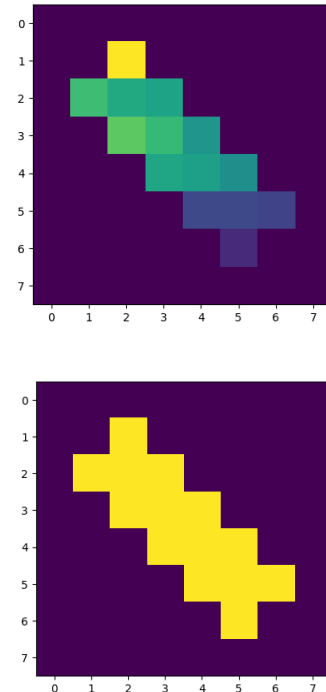


Figura 10. Reconstrucción de la imagen watermark.

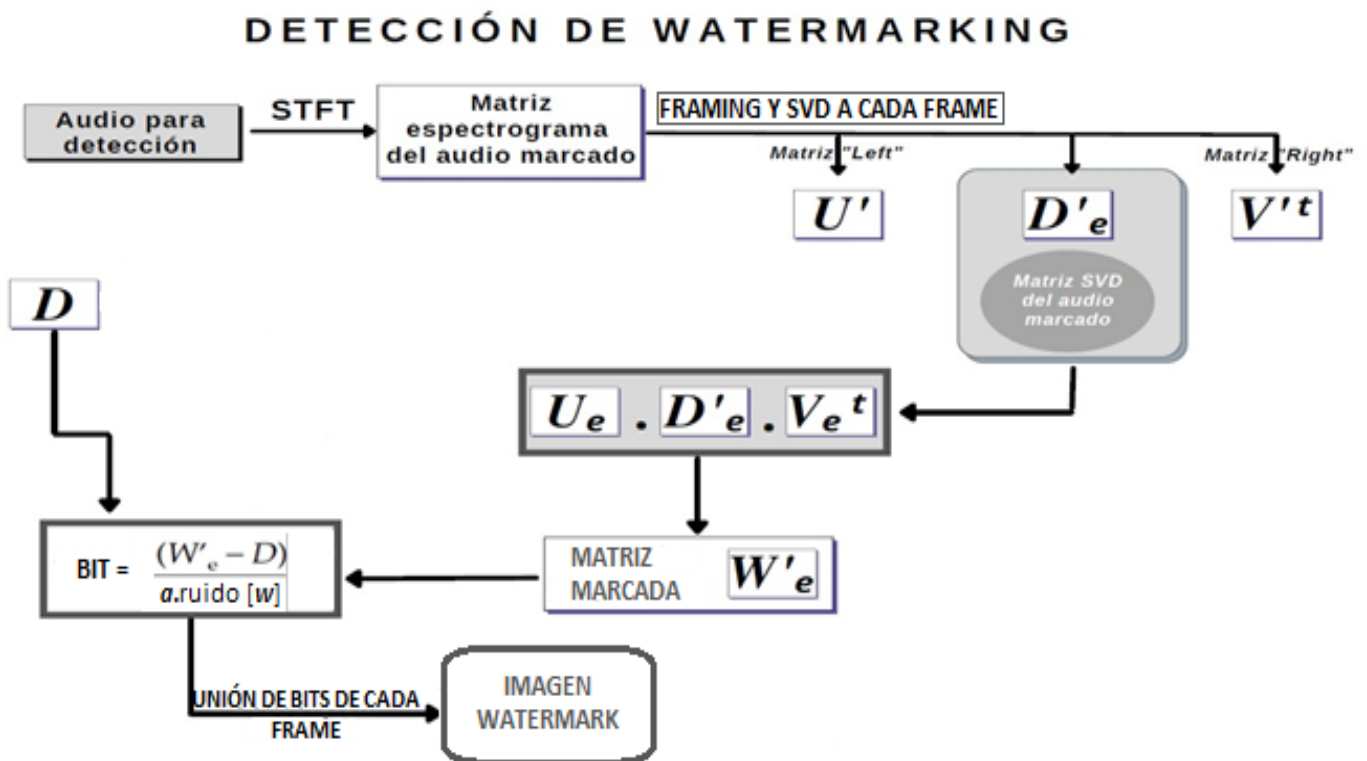


Figura 9. Procedimiento de detección del watermark.

4. Resultados y Análisis.

A continuación, se detallan los resultados tanto para el audio de voz hablada como para el audio musical. Las variaciones de parámetros en el embebido se basan en el uso de distintos tipos de ventanas para la STFT y en la variación del factor de fuerza “ α ”. Por otro lado, se determinará la eficacia e imperceptibilidad del watermark mediante correlación normalizada y relación señal-ruido (SNR).

- La **relación señal-ruido “SNR”** (Signal to Noise Ratio) se define como la proporción existente entre la potencia de salida de la señal y el ruido que posee la misma [4]. Este parámetro objetivo resulta sencillo de implementar y útil para cuantificar la cantidad de distorsión añadida al marcar el audio, por lo que se utiliza en este informe para analizar la imperceptibilidad auditiva del watermark.
- Para el estudio de la eficacia del marcado se valdrá de una herramienta denominada **Correlación cruzada**, esto es, una medida de rastrear los movimientos de dos o más conjuntos de datos en serie entre sí. Se utiliza para comparar múltiples series de tiempo y para determinar objetivamente qué tan bien coinciden ambos conjuntos de datos, además de en qué punto se produce la mejor coincidencia [5].

Se estudia el efecto de tres tipos de ventanas en las dos señales de audio diferentes: “Boxcar”, “Hamming” y “Blackman”; además, para cada tipo de audio y ventana, se imprime la marca de agua con tres magnitudes de fuerza: 0.15, 0.015 y 0.0015. Finalmente, se intenta detectar la imagen marcada utilizando de forma aleatoria claves falsas y la verdadera, a fin de obtener información sobre la seguridad de embebido.

Los resultados de audio marcado e imagen recuperada son sometidos a SNR (audio marcado vs. audio original) y Correlación Cruzada (imagen original vs. Imagen recuperada).



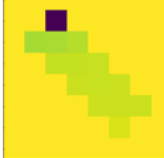
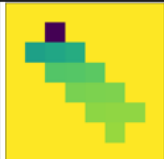





		Music		
		Ventana		
		Boxcar	Hamming	Blackman
Factor de fuerza	0,15	SNR: -0.00014166363253308878 Correlation: 1.0  FALSE KEY	SNR: -0.000311759379552681 Correlation: 1.0  FALSE KEY	SNR: -0.00024678209975112535 Correlation: 1.0  FALSE KEY
	0,015	SNR: -1.4192164076053488e-05 Correlation: 1.0  TRUE KEY	SNR: -1.4443208879618679e-05 Correlation: 1.0  TRUE KEY	SNR: -1.1040756201735984e-05 Correlation: 1.0  FALSE KEY
	0,0015	SNR: -1.4318469905471258e-05 Correlation: 1.0  FALSE KEY	SNR: -1.5200739193216957e-05 Correlation: 1.0  FALSE KEY	SNR: -1.236599960210846e-05 Correlation: 1.0  TRUE KEY

Tabla 1. Resultados audio Music.


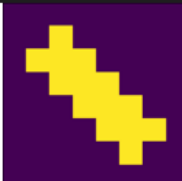
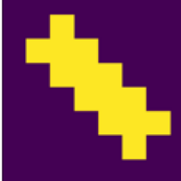
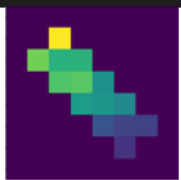





		Speech		
		Ventana		
		Boxcar	Hamming	Blackman
Factor de fuerza	0,15	SNR: 29.884620521301258 Correlation: 1.0  FALSE KEY	SNR: 29.882565497568535 Correlation: 1.0  TRUE KEY	SNR: 29.87934806940985 Correlation: 1.0  TRUE KEY
	0,015	SNR: 29.884215554067694 Correlation: 1.0  FALSE KEY	SNR: 29.87938362395036 Correlation: 1.0  FALSE KEY	SNR: 29.880074686580222 Correlation: 1.0  FALSE KEY
	0,0015	SNR: 29.884143176101663 Correlation: 1.0  TRUE KEY	SNR: 29.87938021493858 Correlation: 1.0  FALSE KEY	SNR: 29.879455961362524 Correlation: 1.0  FALSE KEY

Tabla 2. Resultados audio Speech.

Como primeras impresiones, resulta clara la diferencia en el marcado entre los dos audios. La muestra de audio de tipo musical posee mucho ruido y distorsión desde el audio original, sumado a poca energía en frecuencias altas. Esto parece influir en el marcado del audio, ya que, no importa si se utiliza el pseudo ruido original o uno falso, la calidad de bits recuperados resulta ser más pobre que en el caso del audio tipo “speech”.

Además, en el audio de tipo musical se podía apreciar mayor distorsión percibida al marcarse; en cambio, para el audio “speech”, el ruido añadido en el proceso de watermarking resulta casi imperceptible, y la imagen recuperada se obtiene con mayor claridad. Esto puede corroborarse observando las imágenes correspondientes a la Tabla (2); observándose que, al utilizar la clave verdadera, los bits finales poseen una calidad mucho mayor que al utilizar la clave falsa.

La diferencia de ruido entre los dos tipos de señales también puede detectarse objetivamente con SNR, y los resultados entre ambas lo confirman; para el audio musical, el SNR resulta ser muy pobre sin importar las ventanas usadas ni el factor de fuerza. En cambio, con el audio “speech” se obtienen valores de SNR mucho más elevados para todas las ventanas y factores de fuerza estudiados. Aquí cabe mencionar la posibilidad de que el audio musical resulte tan distorsionado desde el principio,

que influya en las medidas de SNR, arrojando valores alejados de la percepción real.

Observando los resultados de los análisis de correlación, sus valores resultan compatibles con la imagen final recuperada, ya que en todos los casos se pudo obtener el watermark entero y, por lo tanto, el resultado de correlación tendrá su valor pico en 1, tal y como sucede en todos los casos estudiados para cada audio. Un factor que valdría la pena estudiar posteriormente sería la efectividad de detección de la marca de agua en audios que hayan sido modificados por diversos ataques como filtros, resampling, flipping, etc. De todos modos, la efectividad de la detección en los casos analizados resultó ser máxima, tal y como muestran los resultados de correlación.

Finalmente, para distintos tipos de ventana estudiados, se obtienen diferencias muy variadas respecto a la calidad de bits obtenidos, mas no en la efectividad de la detección. En cambio, las diferencias en SNR para los distintos tipos de ventanas resultan mínimas y, aunque parece que la ventana rectangular “boxcar” produce mejor calidad de bits, todas sirvieron para la detección del marcado.

5. Conclusión.

El procedimiento de watermarking, descrito y estudiado en este documento, resulta eficaz para el marcado y detección de señales de audio. Hay que tener en cuenta que un audio con mucho ruido y distorsión de base puede afectar al proceso de marcado de audio, tanto en la percepción de ruido añadido como en la posterior detección de los bits marcados. El marcado de audio mejora cuanto mejor sea la claridad y respuesta frecuencia del audio original.

Queda por estudiar el efecto de los ataques a la señal de audio en relación a la detección de bits de la imagen marcada, ya que diversos ataques podrían atenuar e incluso evitar la detección de la marca de agua.

6. Referencias.

[1] <https://towardsdatascience.com/understanding-singular-value-decomposition-and-its-application-in-data-science-388a54be95d>.

[2] https://ccrma.stanford.edu/~jos/sasp/Short_Time_Fourier_Transform.html

[3] An SVD-Based Audio Watermarking Technique - Hamza Ozer, Bulent Sankur, Nasir Memon - MMSec '05: Proceedings of the 7th workshop on Multimedia and security August 2005 Pages 51–56.

[4] www.sciencedirect.com/topics/engineering/signal-to-noise-ratio

[5] <https://traders.studio/correlacion-cruzada/>