# **ANALISIS DE DATOS Y COMPRESIÓN**

|  |  |  |  |
| --- | --- | --- | --- |
| Sebastián Diaz Gómez  Universidad Eafit  Colombia  Sdiazg1@eafit.edu.co | Jeronimo Gonzalez Gonzalez  Universidad Eafit  Colombia  Jgonzalez7@eafit.edu.co | Simón Marín Universidad Eafit Colombia smaring1@eafit.edu.co | Mauricio Toro  Universidad Eafit  Colombia  mtorobe@eafit.edu.co |

# **RESUMEN**

El problema principal que se nos presenta es el análisis de la salud de el animal por medio de un algoritmo. Pero por otro lado tenemos la compresión de estos datos, ya que la red en las zonas donde seria utilizado este algoritmo es muy inestable y limitada, por lo tanto, debemos encontrar la forma mas eficiente de realizar todos estos procesos.

Este proyecto puede muy importante para el avance tecnológico en lo que tiene que ver con la ganadería. Gracias a esto se podría automatizar este proceso y ser mucho más sencillo.

Un problema relacionado con este mismo puede ser la compresión de datos en los videojuegos, ya que se busca lograr optimizar estos de la forma mas eficiente posible, sin llegar a pedir mucha calidad en estos.

**1. INTRODUCCIÓN**

Actualmente la tecnología lleva a el hombre a buscar nuevas soluciones hacia problemas que requieren mayor esfuerzo, usando la tecnología como un método para facilitar tanto la vida de usuario comunes y corrientes como la de los ingenieros y los alrededores.

Hablando de lo anteriormente mencionado un gran problema que se maneja constantemente en el mundo de la informática son los sistemas de archivos. Las problemáticas de esto son las constante limitaciones de el sistema para almacenar una gran cantidad de los mismos.

Sin embargo, Muchos de los avances que se lograron fueron las capacidades de un sistema operativo el cual de manera eficiente logra administrar el espacio libre y acceder de manera rápida a los archivos guardados de la actualización más reciente.

# **1.1. Problema**

Como mencioné anteriormente el problema principal es lograr analizar los datos del ganado para determinar si están bien, o enfermos. Y a su vez lograr comprimir estos datos para hacer mas eficiente el proceso debido a la mala calidad de red en las zonas ganaderas.

Esto puede aportar mucho a la ganadería en cuanto a eficiencia, ya que ayuda a que estos procesos se hagan mucho más rápidos y sencillos, por lo que es un avance muy importante.

**1.2 Solución**

En este trabajo, utilizamos una red neuronal convolucional para clasificar la salud animal, en el ganado vacuno, en el contexto de la ganadería de precisión (GdP). Un problema común en la GdP es que la infraestructura de la red es muy limitada, por lo que se requiere la compresión de los datos.

**1.3 Estructura del artículo**

En lo que sigue, en la Sección 2, presentamos trabajos relacional con el problema. Más adelante, en la Ssección 3, presentamos los conjuntos de datos y los métodos utilizados en esta investigación. En la Sección 4, presentamos el diseño del algoritmo. Después, en la Sección 5, presentamos los resultados. Finalmente, en la Sección 6, discutimos los resultados y proponemos algunas direcciones de trabajo futuras.

**2. TRABAJOS RELACIONADOS**

## En lo que sigue, explicamos cuatro trabajos relacionados. en el dominio de la clasificación de la salud animal y la compresión de datos. en el contexto del PLF.

## Explique cuatro (4) artículos relacionados con el problema descrito en la sección 1.1. Puede encontrar los problemas relacionados en las revistas científicas, en lo posible, en inglés. Considere Google Scholar para su búsqueda. *(En este semestre, el trabajo relacionado es la investigación sobre la clasificación de la salud animal y la compresión de datos, en el contexto de la GdP).*

## **3.1 Escriba un título para el primer problema**

Deberían mencionar el problema que resolvieron, el algoritmo que usaron, la métrica que obtuvieron y la cita de ACM.

## **3.2 Escriba un título para el segundo problema**

Deberían mencionar el problema que resolvieron, el algoritmo que usaron, la métrica que obtuvieron y la cita de ACM.

## **3.3 Escriba un título para el tercer problema**

Deberían mencionar el problema que resolvieron, el algoritmo que usaron, la métrica que obtuvieron y la cita de ACM.

## **3.4 Escriba un título para el cuarto problema**

Deberían mencionar el problema que resolvieron, el algoritmo que usaron, la métrica que obtuvieron y la cita de ACM.

## **3. MATERIALES Y MÉTODOS**

En esta sección, explicamos cómo se recogieron y procesaron los datos y, después, diferentes alternativas de algoritmos de compresión de imágenes para mejorar la clasificación de la salud animal.

## **3.1 Recopilación y procesamiento de datos**

Recogimos datos de *Google Images* y *Bing Images* divididos en dos grupos: ganado sano y ganado enfermo. Para el ganado sano, la cadena de búsqueda era "cow". Para el ganado enfermo, la cadena de búsqueda era "cow + sick".

En el siguiente paso, ambos grupos de imágenes fueron transformadas a escala de grises usando Python OpenCV y fueron transformadas en archivos de valores separados por comas (en inglés, CSV). Los conjuntos de datos estaban equilibrados.

El conjunto de datos se dividió en un 70% para entrenamiento y un 30% para pruebas. Los conjuntos de datos están disponibles en https://github.com/mauriciotoro/ST0245-Eafit/tree/master/proyecto/datasets .

Por último, utilizando el conjunto de datos de entrenamiento, entrenamos una red neuronal convolucional para la clasificación binaria de imágenes utilizando *Teachable Machine* de Google disponible en https://teachablemachine.withgoogle.com/train/image.

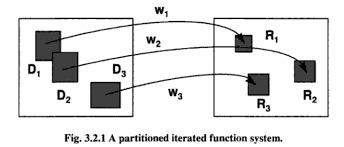
## **3.2 Alternativas de compresión de imágenes con pérdida**

## En lo que sigue, presentamos diferentes algoritmos usados para comprimir imágenes con pérdida.

**3.2.1 Compresión fractal**

La compresión fractal es un método de compresión con pérdida para imágenes digitales, basado en fractales. El método es el más apropiado para texturas e imágenes naturales, basándose en el hecho de que partes de una imagen, a menudo, se parecen a otras partes de la misma imagen. Los algoritmos fractales convierten estas partes en datos matemáticos llamados «códigos fractales» los cuales se usan para recrear la imagen codificada.

Con la compresión fractal, la codificación es extremadamente cara a nivel computacional debido a la búsqueda de similitudes propias. Sin embargo, la decodificación es bastante rápida. Mientras que esta asimetría lo hace poco práctico para aplicaciones en tiempo real, cuando el vídeo es guardado para distribución desde un disco, la compresión fractal llega a ser más competitiva.



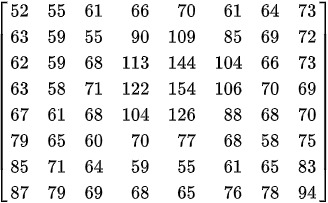
**3.2.2 Joint Photographic Experts Group (JEPG)**

JPEG utiliza una forma de compresión con pérdida basada en la transformada de coseno discreta (DCT). Esta operación matemática convierte cada cuadro / campo de la fuente de video del dominio espacial (2D) al dominio de frecuencia (también conocido como dominio de transformación).

Un modelo perceptivo basado libremente en el sistema psicovisual humano descarta información de alta frecuencia, es decir, transiciones bruscas en intensidad y tono de color. En el dominio de la transformación, el proceso de reducción de información se denomina cuantificación. En términos más simples, la cuantificación es un método para reducir de manera óptima una escala de números grandes (con diferentes ocurrencias de cada número) en una más pequeña.

El método de compresión suele tener pérdidas, lo que significa que parte de la información de la imagen original se pierde y no se puede restaurar, lo que posiblemente afecte a la calidad de la imagen.

A continuación, cada bloque de 8 × 8 de cada componente (Y, Cb, Cr) se convierte en una representación de dominio de frecuencia, utilizando una transformada de coseno discreta (DCT) de tipo II bidimensional normalizada:



**3.2.3 codificación por transformación (NTSC, PAL, SECAM)**

Codificación por transformación: Es un tipo de compresión para datos naturales como: Audio, video e imagen, la transformación conlleva perdida de información, resultando una copia de menor calidad a la que entra originalmente.

Un ejemplo de esto es la televisión, que utiliza sistemas de compresión por transformación como:

NTSC: Este es uno de los sistemas de codificación por transformación con mayor éxito – se enfocó en el formato de televisión a color. Este conocimiento permitió desarrollar in sistema que descartara mayor parte de la señal entrante, el resultado fue una señal con menos contenido que encajo en los 6MHz de señal a blanco y negro con una diferencia modulada.

PAL y SECAM: Estos sistemas utilizan métodos muy parecidos para transmitir color.

Ejemplo de codificación por compresión:

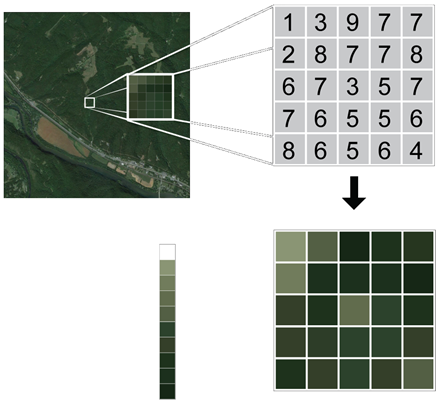


**3.2.4 MrSid**

Es un algoritmo de compresión de imágenes con perdida y un estándar abierto de compresión de imágenes raster (Estructuras de pixeles), permite mostrar archivos digitales de gran tamaño con un tiempo de carga mínimo gracias a tecnologías ondículas (Transforma las señales en gráficas y algoritmos matemáticos).

La característica predominante del formato es el teselado (Patrón de figureas) que logra una alta compresión de imágenes digitales con la pérdida mínima de detalle. El teselado da la capacidad de descomprimirse solamente aquella porción de imagen solicitada por el usuario, extrayendo y entregando únicamente los bitplanes y datos raster necesarios para construir la vista requerida.

Datos raster: También conocido como mapa de bits o plano de bits, son datos representados por medio de pixeles o puntos de color, que se pueden visualizar por medio de monitores u otros dispositivos (Tiene cierta relación con el PNG).



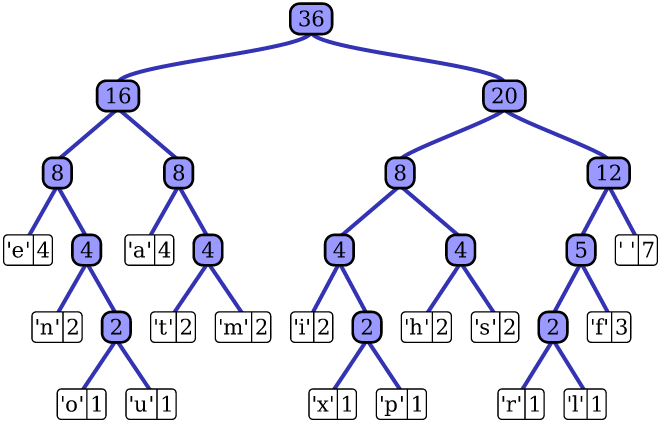
## **3.3 Alternativas de compresión de imágenes sin pérdida**

## En lo que sigue, presentamos diferentes algoritmos usados para comprimir imágenes sin pérdida.

**3.3.1 codificación de Huffman**

Es un algoritmo usado para la compresión de datos. Se refiere al uso de una tabla de códigos de longitud variable para codificar determinado símbolo, donde la tabla ha sido rellenada de tal forma que se basa en la probabilidad estimada de aparición de cada de cada posible valor de dicho símbolo.

La codificación de Huffman usa un método especifico para elegir la representación de cada símbolo, que representa los caracteres mas comunes usando las cadenas de bits mas cortas, y viceversa.

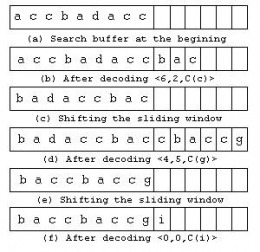


Un ejemplo de un árbol dibujado es el siguiente donde los nodos hijos finales son a los que les corresponde un símbolo y una frecuencia, y los nodos padres de estos últimos se conectan como pequeños árboles a otros nodos. Los nodos con puros números indican la suma de las frecuencias de los nodos hijos correspondientes.

**3.3.2 LZ77**

Los algoritmos LZ77 logran la compresión reemplazando las apariciones repetidas de datos con referencias a una sola copia de esos datos existente anteriormente en el flujo de datos sin comprimir.

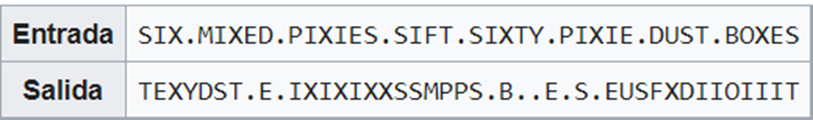
Para detectar coincidencias, el codificador debe realizar un seguimiento de cierta cantidad de los datos más recientes, como los últimos 2 kB, 4 kB o 32 kB. La estructura en la que se guardan estos datos se denomina ventana deslizante, por lo que LZ77 a veces se denomina compresión de ventana deslizante. El codificador necesita mantener estos datos para buscar coincidencias, y el decodificador necesita mantener estos datos para interpretar las coincidencias a las que se refiere el codificador. Cuanto más grande sea la ventana deslizante, más atrás podrá buscar el codificador para crear referencias.



**3.3.3 Burros Wheller**

Es un algoritmo sin perdidas, es utilizado para la compresión de datos como en bzip.2. Cuando se transforma una cadena de caracteres mediante la BTW, ningún carácter cambia de valor. Si la cadena contiene muchas sub cadenas, entonces la cadena transformada contendrá múltiples opciones en las que el mismo carácter este repetido varias veces.

Es muy útil para la compresión de archivos, ya que tiende a ser fácil comprimir una cadena que tiene una secuencia de caracteres repetidos con técnicas como: move-to-front-transform y run-length encording.



La salida es más fácil de comprimir ya que tiene muchos caracteres repetidos. De hecho, en la cadena transformada.

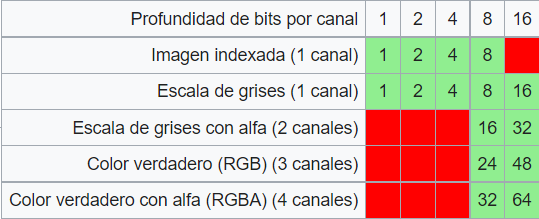
aparece un total de seis secuencias de caracteres idénticos:

XX, SS, PP, ..., II, y III, que juntos representan 13 de los 44 caracteres.

**3.3.4 PNG**

Es un algoritmo de comprensión sin perdidas para bitmaps. Este formato fue desarrollado para solventar las deficiencias del formato gif y permita almacenas imágenes con mayo profundidad de contraste y otros datos importantes.

El método de compresión utilizado por el PNG es conocido como deflación (Algoritmo sin perdidas). También tiene métodos de filtrado de información en pixeles, el mas utilizado es prediciendo el valor aproximado de los pixeles lo cual mejora la comprensión para cada línea de la imagen a comprimir. En otras palabras, este método de filtrado predice el color de cada pixel basándose en los colores de los pixeles previos a comprimir, restándolos con los actuales, Lo cual reduce el tamaño del archivo de imagen.



# **REFERENCIAS**

Wikipedia, Edit 2021, Codificación Huffman, <https://es.wikipedia.org/wiki/Codificaci%C3%B3n_Huffman>

Wikipedia, Edit 2021, LZ77, <https://en.wikipedia.org/wiki/LZ77_and_LZ78>

Wikipedia, Edit 2021, Compresión fractal <https://es.wikipedia.org/wiki/Compresi%C3%B3n_fractal>

Wikipedia, Edit 2021, JEPG https://en.wikipedia.org/wiki/JPEG