# **OPTIMIZACION DEL PROCESO DE GANADERIA MEDIANTE EL USO DE ESTRUCTURAS DE DATOS**

|  |  |  |
| --- | --- | --- |
| Juan Andrés Gómez Universidad Eafit Colombia jagomezd@eafit.edu.co | Laura Ortiz  Universidad Eafit  Colombia  lortizc3@eafit.edu.co | Mauricio Toro  Universidad Eafit  Colombia  mtorobe@eafit.edu.co |

# **RESUMEN**

## El problema a tratar es el cómo optimizar el proceso de la ganadería mediante las tecnologías de la información diseñando algoritmos basados en compresión de imágenes y tablas de hash potenciando el consumo de energía en ámbitos de la ganadería de precisión, la cual es importante, ya que esta misma le da un mejor enfoque a la ganadería, y al mismo tiempo al cuidado bovino.

## Una problemática relacionada con el tópico de estudio es la aplicación de satélites de teledetección en ámbito agrícola puesto que en estas mismas se usan tecnología de la información las cuales permiten un estudio optimo de la superficie por la cual se puede deducir que tipo de fertilizante necesita cada sector del cultivo.

## **Palabras clave**

|  |
| --- |
| Algoritmos de compresión, aprendizaje de máquina,  aprendizaje profundo, ganadería de precisión, salud animal. |

# **1. INTRODUCCIÓN**

# Una de las principales necesidades del ser humano siempre ha sido la alimentación, lo cual ha llevado ya en el pasar del tiempo de la raza humana a la masificación de este tipo de necesidades por el aumento poblacional. Esto ha llevado a que en el ultimo milenio aumente de una manera exponencial la necesidad de buenas fuentes de alimento. Esto ha direccionado a industrias tales como son las ganaderas a un crecimiento de la mano con la demanda de sus productos. Debido a lo tedioso que ha sido mantener un modelo el cual parte de lo tradicional, se ha generado nuevas tecnologías las cuales permiten una gran mejora de los diferentes estudios los cuales proveen una manutención optima del sector agropecuario.

# Este mismo fenómeno ya descrito ha generado un nuevo movimiento de las tecnologías la cual se llama “Ganadería de precisión” (GdP), esto ha llevado al desarrollo de tecnologías similares las cuales ayudan a gran parte del sector agropecuario tal y como puede ser la aplicación de satélites de teledetección.

# **1.1. Problema**

El problema al cual nos enfrentamos es crear mediante una estructura de datos un sistema eficiente el cual permita identificar el estado de salud animal mediante imagen de madera eficiente en cuanto a memoria y tiempo. Esto significaría un avance gigante en la ganadería de precisión la cual podría impulsarse mas gracias a estos sistemas

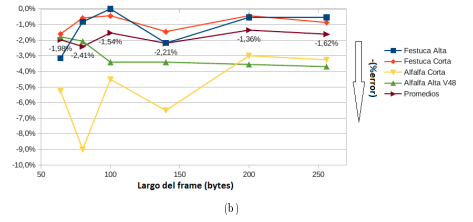
**3. TRABAJOS RELACIONADOS**

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.

3.1 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).*

**3.1.1 Almacenamiento de sonidos para ganadería de precisión**

Captar los sonidos producidos por los animales durante su alimentación, sin interferir en su comportamiento normal y sin intervención del operador, se eligió usar algoritmos de compresión con perdida ya que eran mas útiles que los algoritmos sin perdida y se termino usando el algoritmo CELT. A continuación la grafica de margen de error de compresión del algoritmo CELT.

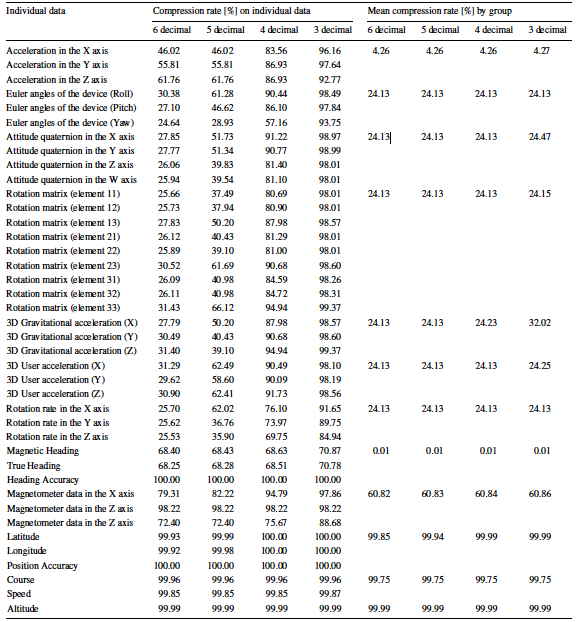


## **3.1.2 Control biológico de la garrapata mediante imagen**

## Contribuir con la escuela de veterinaria con el Desarrollo de un software con procesamiento de imágenes que reconozca la rapidez y eficacia de extracto de Neem como garrapaticida, sin la necesidad de la presencia de los doctores, se termino binarizando las imágenes para una mejor eficacia en el sistema

## **3.1.3 Cloud services integration for farm animals’ behavior studies based on smartphones as activity sensors**

Se plantea el uso de smartphones como sensores de actividad para la ganadería debido a que cada vez se ha venido implementando el uso de sensores en la industria, se proponen dos algoritmos de compresión complementarios: Un algoritmo que elimine información redundante y otro que trunca la información a 3 ,4 y 5 decimales. A continuación la grafica de efectividad de compresión.



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

## En lo que sigue, presentamos diferentes algoritmos usados para comprimir imágenes con pérdida. *(En este semestre, ejemplos de tales algoritmos son el tallado de costuras, el escalado de imágenes, la transformación de coseno discreto, la compresión con ondeletas y la compresión fractal).*

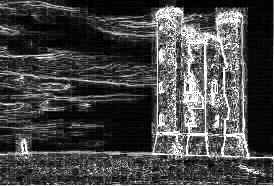
**3.2.1 Liquid Rescaling**

Es un algoritmo para el re-tallado de imágenes desarrollado por Shai Avidan y Ariel Shamir, que consiste en establecer una serie de costuras o caminos de menor importancia, una vez aplicado se pueden agregar o eliminar lo que amplia o reduce el tamaño de la imagen respectivamente.

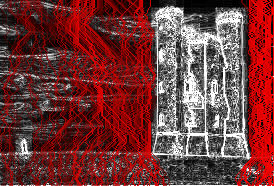
Permite definir de manera manual las áreas en las que no se desea que se modifiquen los pixeles y presenta la habilidad de remover objetos completos de las fotografías



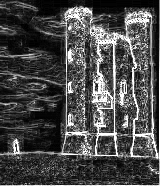
1.) Imagen Original



2.) Se calcula el peso/densidad/energía de cada pixel



3.) Se realiza una lista de costuras en base a la energía



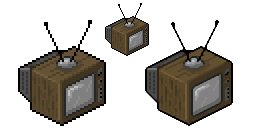
4.) Se eliminan todas las costuras de baja energía



5.) Imagen Final

**3.2.2 Image Scaling**

Se refiere al ajuste de tamaño de una imagen digital, que en el medio de tecnología de videos se conoce como “upscaling”; al momento de escalar un vector gráfico, los elementos primitivos que componen la imagen se pueden modificar usando transformaciones geométricas sin pérdida de la calidad de imagen, cuando se escalan gráficos ráster se debe generar otra imagen con un mayor o menor número de pixeles, en caso de que decrezcan (scaling down) resulta con una reducción de la calidad de imagen.



**3.2.3 Compresión JPEG**

Este algoritmo, pierde información y esto se refleja en una disminución de la calidad de la imagen; pero tal pérdida no es visible para el ojo humano. El funcionamiento del algoritmo se explica a continuación: se inicia con una imagen en donde cada píxel tiene 3 canales de color (rojo, azul y verde) y lo primero que hace el algoritmo es transformar estos 3 canales en 2 de color y uno de brillo; luego, a los canales de color les baja la resolución y allí es donde se generan las perdidas ya que elimina o modifica alguna información, ya que el ojo humano no nota estas modificaciones del color, pero si lo notase en el brillo. Luego divide cada canal en cuadros de 8x8 pixeles y a cada cuadro por separado le aplica la transformada discreta bidimensional del coseno y cuantificación perceptual que lo que realiza es suavizar las variaciones bruscas de brillo y color, y ya por último se aplica un algoritmo de compresión sin perdida el cual se explica líneas adelante, el algoritmo de Huffman.



**3.2.4 Compresión Fractal**

La palabra fractal es término reciente propuesto en el año de 1975, el cual se define como un patrón geométrico que se repite en varias escalas. Por tanto, como su nombre lo indica, la compresión fractal es un método de compresión basado en fractales, este es el más apropiado para imágenes o texturas naturales, porque ciertas partes de la imagen se parecen a otras. Como se deben buscar similitudes propias de la imagen, este método puede llegar a ser bastante costoso, aunque, por otro lado, la decodificación de la imagen se hace de una manera bastante rápida. Por tanto, la eficiencia del algoritmo va a depender mucho en la complejidad de la imagen y la profundidad del color. Una de las principales características de la compresión fractal, es que las imágenes se vuelven independientes de la resolución (escalado fractal), y esto hace posible, que la imagen se pueda pasar una mejor resolución (interpolación fractal).

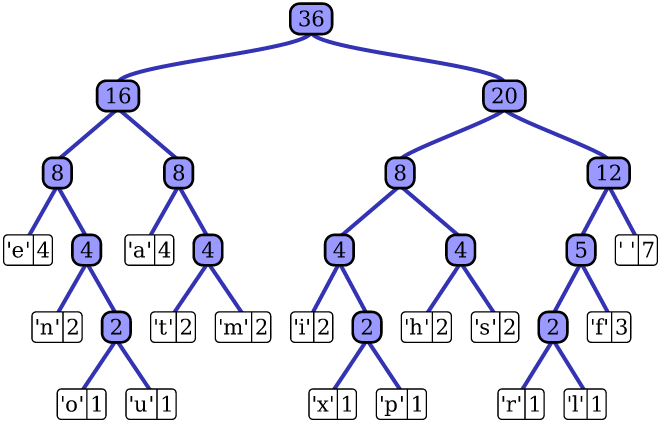


## **3.3 Alternativas de compresión de imágenes sin pérdida**

## En lo que sigue, presentamos diferentes algoritmos usados para comprimir imágenes sin pérdida. *(En este semestre, ejemplos de tales algoritmos son la transformada de Borrows y Wheeler, LZ77, LZ78, la codificación Huffman y LZS).*

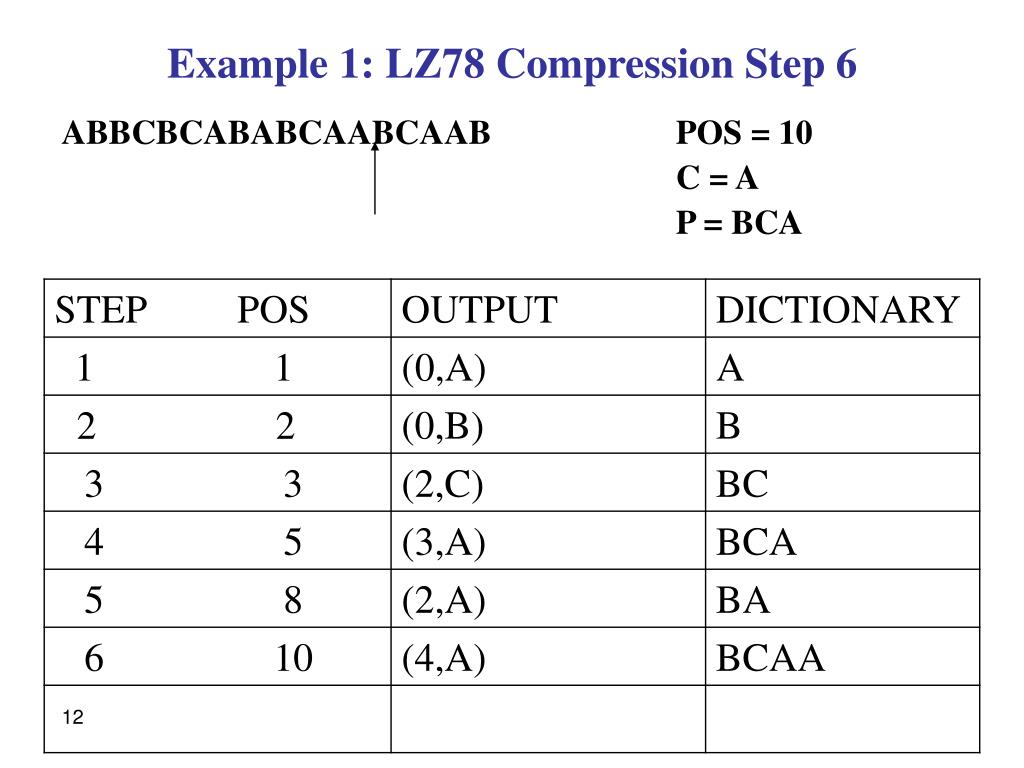
**3.3.1 Codificación de Huffman**

Es un procedimiento que permite asignar a los diferentes símbolos a comprimir, un código binario. Este algoritmo crea un árbol de nodos; primero se debe establecer un orden prioritario, el más importante es el símbolo que aparece con menor frecuencia en la cadena, luego, se eliminan los dos símbolos más prioritarios, construyendo así un nuevo “padre” que es el resultado de la suma de las frecuencias eliminadas y se ubica nuevamente en la cola de prioridad, iterativamente se realiza este proceso hasta que sólo quede un elemento, así queda construido el árbol. Luego, para asignar el código binario se contarán los pasos efectuados para llegar a cada símbolo del árbol (movimiento a la izquierda=0, a la derecha=1), obteniendo el valor de cada uno y por último reemplazándolos en la cadena.



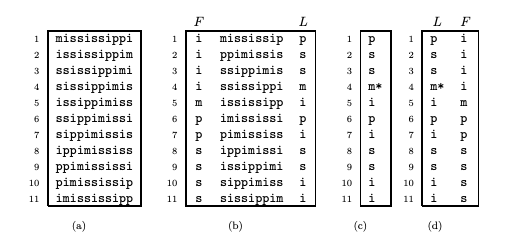
**3.3.2 LZ78**

El algoritmo LZ78 usa una estructura de datos, específicamente trie, basada en diccionario para comprimir los datos. Este nació para mejorar el rendimiento respecto al LZ77, es por así decirlo, una segunda versión mejorada. Respecto al funcionamiento para la compresión, se lee una cadena carácter por carácter y verificamos si el nodo actual (Cada nodo está marcado con el índice del diccionario) tiene algún borde de salida que contenga al carácter leído; en caso positivo configuramos el nodo actual como el nodo encontrado y repetimos el proceso, en caso negativo, se crea un borde de salida con el carácter leído se crea un nodo nuevo, el cual pasa a ser el nodo raíz e igualmente se repite el proceso



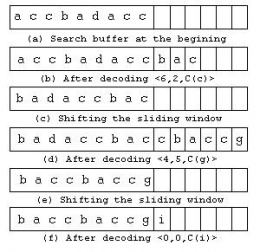
**3.3.3 Burrows Transform**

La transformación de Burrows-Wheeler reorganiza una cadena de caracteres en series similares. Es un algoritmo que prepara los datos para su posterior uso con técnicas de compresión, al ingresar una cadena de caracteres, la transformación conmuta su orden. El algoritmo toma la cadena de entrada y realiza todas sus rotaciones hasta que el carácter inicial vuelva a quedar en su misma posición, luego con los resultados de todas estas rotaciones los ordena por orden alfabético (los símbolos especiales como \*,|,”)se toman como últimos y ya una vez organizados, se toma el último carácter década cadena y así se obtiene la salida esperada.



**3.3.4 LZ77**

Es un compresor basado en algoritmo sin pérdida, este algoritmo es un tipo de codificador diccionario en el cual existen los literales, banderas y palabras claves; se empieza a recorrer la cadena y si se encuentra con un literal lo deja totalmente igual, si encuentra una bandera especifica si lo que sigue es un literal o un comprimido y si lo anterior es un comprimido (que es una especie de palabra clave) se lleva a una posición en un diccionario que arroja que bytes continúan. Por ejemplo, en una imagen las esquinas pueden ser iguales y al comprimir tales esquinas lo que hacemos es que se tenga que guardar una sola vez y por así decirlo evitar las repeticiones.

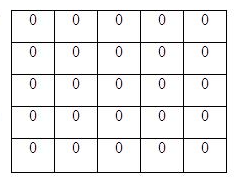


## **4. DISEÑO E IMPLEMENTACIÓN DE LOS ALGORITMOS**

## En lo que sigue, explicamos las estructuras de datos y los algoritmos utilizados en este trabajo. Las implementaciones de las estructuras de datos y los algoritmos están disponibles en GitHub.

## **4.1 Estructuras de datos**

Para almacenar los datos usamos una lista enlazada, y dentro de ella en cada posición tenemos un diccionario que contiene el nombre del archivo y una matriz que contiene los valores de los pixeles, utilizamos el diccionario ya que nos permitía conservar los nombres originales del archivo luego de la compresión y descompresión. Para el efecto del algoritmo usamos sólo la matriz de pixeles.

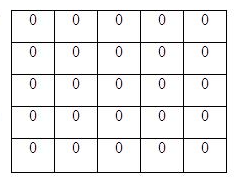


**4.2 Algoritmos**

En este trabajo, proponemos un algoritmo de compresión que es una combinación de un algoritmo de compresión de imágenes con pérdidas y un algoritmo de compresión de imágenes sin pérdidas. También explicamos cómo funciona la descompresión para el algoritmo propuesto.

**4.2.1 Algoritmo de compresión de imágenes con pérdida**

Para la compresión y descompresión de las imágenes con pérdidas usamos el escalado por medio del vecino más cercano. Este método es muy sencillo, lo único que se hace es dividir la matriz de pixeles de la imagen en submatrices de igual tamaño y escogemos cualquier pixel dentro de cada submatriz, pero siempre el de la misma posición, es decir si en la primera submatriz escogemos el pixel de la mitad en todas las submatrices también deberemos de seleccionar el de la mitad

 --------> 

**REFERENCIAS**

1. Jose Chelloti O. 2014. Desarrollo e implementación de un dispositivo de adquisición y almacenamiento de sonidos para ganadería de precisión**.** (November 2014). Retrieved August 16, 2021 from http://sedici.unlp.edu.ar/handle/10915/42007

2. Zenteno Aguilar, Cristian del Carmen. 2012. Control biológico de la garrapata en el sector ganadero con procesamiento de imágenes**.** (January 2012). Retrieved August 16, 2021 from http://repositoriodigital.tuxtla.tecnm.mx/xmlui/handle/123456789/171

3. Debauche, O., Mahmoudi, S., Andriamandroso, A.L.H., Manneback, P., Bindelle, J., Lebeau, F., 2019. Cloud services integration for farm animals’ behavior studies based on smartphones as activity sensors. J. Ambient Intell. Humanized Comput. 10 (12), 4651–4662.

4. Explicación detallada del algoritmo de interpolación bilineal de imágenes Python realiza el algoritmo de interpolación bilineal – programador clicl.(s.f.) Programador clic. Recuperado 11 de abril de 2021, de https://programmerclick.com/article/ 9212974781