# **ESTRUCTURAS DE DATOS PARA EL ANALISIS Y LA COMPRESIÓN DE IMAGENES**

|  |  |  |  |
| --- | --- | --- | --- |
| Daniel Vélez  Universidad Eafit  Colombia  dvelezd2@eafit.edu.co | Juan José Villada  Universidad Eafit  Colombia  jjvilladac@eafit.edu.co | Simón Marín Universidad Eafit Colombia smaring1@eafit.edu.co | Mauricio Toro  Universidad Eafit  Colombia  mtorobe@eafit.edu.co |

# **RESUMEN:**

El objetivo de este informe es analizar y principalmente solucionar el problema que se tiene, el cual es comprimir imágenes de manera eficiente para determinar el estado de salud de un ganado definido.

Lo que se busca con este trabajo será desarrollar un algoritmo que pueda identificar a través de una estructura de datos si el ganado bobino está en condiciones óptimas de salud o se encuentra enfermo. Darle una solución a esta problemática sería de enorme ayuda, pues en la industria de la ganadería no se cuenta la tecnología suficiente para desarrollar este tipo de problemas que hace el estudio de este ganado se dificulte en la mayoría de los casos.

## **Palabras clave:**

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

# **1. INTRODUCCIÓN***:*

Hoy en día los algoritmos de compresión están tomando un rol muy importante dentro la ganadería de precisión, gracias a esto se generan programas que ayudan a saber los posibles estados de salud que se presentan dentro del ganado.

Por esto se van a usar algoritmos de compresión, ya que están diseñados para representar cierta cantidad de información utilizando u ocupando un espacio menor, y luego al descomprimirse no se pierda información en el proceso, esto es muy eficiente dentro del mundo de la ganadería de precisión ya que facilitará el estudio para la recolección de datos de manera general.

# **Problema:**

El problema al cual nos enfrentamos es básicamente crear a través de una estructura de datos, una red de imágenes csv, para que, mediante estas, de manera eficiente, se pueda encontrar dentro de estas que animal puede estar enfermo o en buenas condiciones.

Es importante resolver este problema debido a que al desarrollar un programa como estos ayudará en gran cantidad a los usuarios de la ganadería de precisión como a los ganaderos y empresas ligadas a este mundo.

# **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 relacionados con el problema. Más adelante, en la Sección 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.

## **2.1 Cálculo automatizado del vector eléctrico integral de la actividad ventricular cardíaca en equinos:**

El problema que se resolvió aquí fue el desarrollo de un algoritmo que hallara la determinación automatizada de la posición del VEI medio del complejo QRS en equinos.

En el algoritmo se usó inicialmente un seudocódigo en java el cual tenía como datos principales relacionados con las variaciones en los ventrículos de cada animal estudiado y se llevaron todos los datos anteriores al resultado final para así calcular la eficiencia y la veracidad del programa.

Para hallar el resultado hicieron primero cálculos manuales sobre la posición del eje eléctrico y después insertaron el mismo problema en el programa para comparar y al final determinar que los dos datos coincidían

## **2.2 Caracterización de variables utilizando inteligencia computacional para identificar alteraciones en la salud:**

Deberían mencionar el problema que resolvieron, el algoritmo que usaron, la métrica que obtuvieron y la cita de ACM. El problema que se presenta en este caso es el de desarrollar un programa para hallar las alteraciones en la salud de los bovinos, determinando a FP-GROWTH como mejor algoritmo para la solución del problema.

Las conclusiones que se obtuvieron en el estudio es que primero la cantidad de variables presentes es muy extensa, haciendo que sea muy minuciosa la investigación con cada dato y también el algoritmo FP-GROWTH identifica patrones de irregularidad a través de pruebas de homagrama.

## **2.3 DESARROLLO DE UNA BASE DE DATOS CON IMÁGENES TERMOGRÁFICAS PARA USO EN ALGORITMOS DE VISIÓN E INTELIGENCIA ARTIFICIAL:**

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

Lo que se presenta como problema es que se está realizando un estudio para desarrollar un programa que, a través de inteligencia artificial, pueda identificar los indicadores térmicos para así visualizar la realidad desde distintos rangos del espectro electromagnético.

Lo que se concluyó es que el número de imágenes gestantes no esta tan alejado del número de imágenes de ovejas no gestantes, tanto como imágenes digitales en el espectro visible de termogramas.

## **2.4 Desarrollo e implementación de un dispositivo de adquisición y almacenamiento de sonidos para ganadería de precisión:**

El problema aquí presente es que se desea encontrar la implementación de un dispositivo de adquisición y almacenamiento de señales para el monitoreo de

actividades alimentarias en ganado bovino que pueda captar variables como el sonido durante su alimentación sin molestar al animal.

Como solución el programa fue exitoso, pero aun así decidieron implementar una mejora en cuanto a el manejo de datos, la medida que toaron fue que dividieron las tareas de acuerdo a la carga computacional respectiva.

## **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.

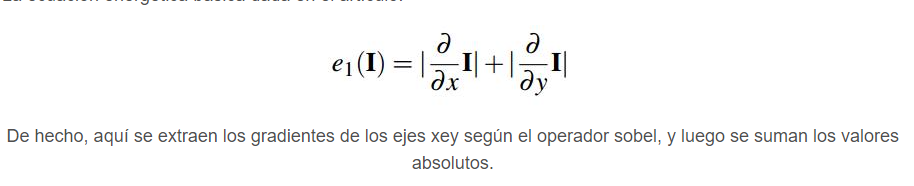
Por último, utilizando el conjunto de datos de entrenamiento, entrenamos una red neuronal convolucional para la clasificación binaria de imágenes utilizando *Teachable Machine* de Google disponible en https://teachablemachine.withgoogle.com/train/image.

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

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

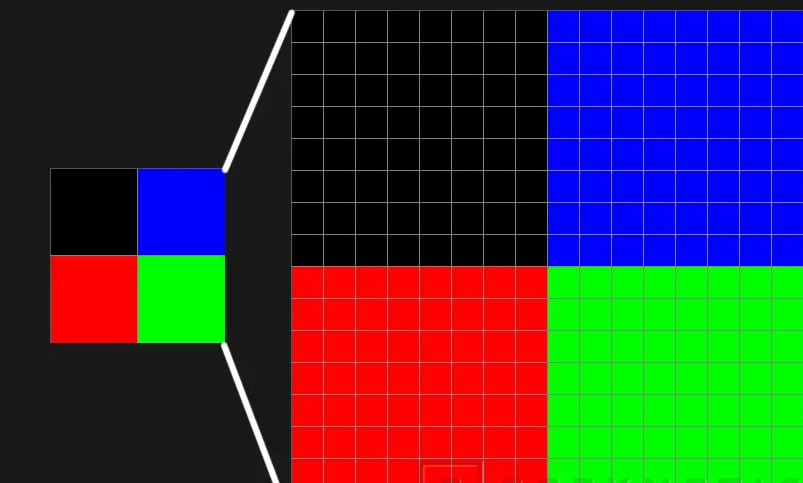
**3.2.1 Algoritmo de Talla de Costura:**

En pocas palabras, a cada valor de píxel se le asigna un valor de energía y, en función de ese valor de píxel, se utilizan ocho dominios conectados en la programación dinámica para obtener el valor más pequeño. Luego aplique este algoritmo fila por fila o columna por columna a: Una línea de alimentación es en realidad una fase. Las filas de píxeles con el valor de píxel más pequeño en dos filas (columnas) adyacentes se eliminarán de la imagen original. El número de estas filas que se pueden eliminar depende de la escala que esté cortando. La energía anterior es en realidad la misma que el gradiente de píxeles.



**3.2.2 Integer Scalling:**

La escala de enteros, definida por Intel, es una técnica de escala que simplemente escala los píxeles existentes usando un multiplicador como un número entero, un número que se puede describir sin tener un componente decimal. Por ejemplo, 1 o 3son números enteros que se pueden usar con esta escala, pero los números negativos o decimales no son completamente válidos. Este cambio de tamaño utiliza un algoritmo de interpolación de "vecino cercano", también conocido como interpolación cercana, para cambiar el tamaño de la imagen a partir de la información original y cambiar el tamaño de la imagen en función de los colores relacionados con el gusto. El píxel original es el más cercano al píxel de destino.



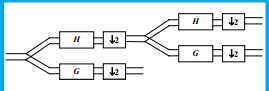
**3.2.3 Compresión fractal:**

Este algoritmo se basa en que analiza la imagen y busca patrones en los que haya similitud y dependiendo de esta los clasifica en fractales, los cuales al final se convertirán en datos matemáticos como funciones y así recrear el total de la imagen.



**3.2.4 Compresión por wavelets:**

Este método de compresión comienza utilizando una wavelet transformada a una imagen de prueba, esto se correlaciona con los valores de pixeles en la imagen original llevando así esta información a coeficientes relativamente pequeños.



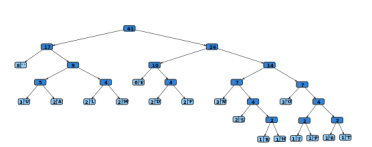
## **3.3 Alternativas de compresión de imágenes sin pérdida:**

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

**3.3.1 Codificación Huffman:**

Este tipo de codificación se refiere a la codificación de un símbolo específico usando la codificación de longitud variable, compilada de una manera específica basada en la probabilidad estimada de ocurrencia de todos los valores es posible para la notación anterior.

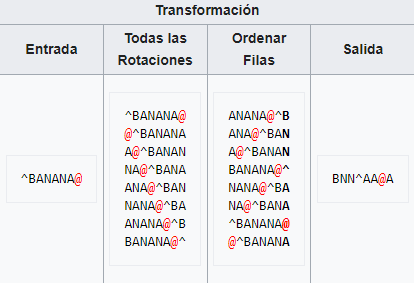
La codificación de Huffman utiliza métodos específicos para seleccionar una representación para cada símbolo. Esto hace que sea un código de prefijo, utilizando la cadena de bits más corta. Representa el carácter más común de bits más cortas y viceversa

****

**3.3.2 Transformación Burrows-Wheeler:**

La conversión de Burrows-Wheeler es un transformador de texto que acepta una cadena de entrada y produce una cadena de salida que contiene una gran cantidad de caracteres repetidos. Por ejemplo, la palabra "banana" se convierte en "annb $ aa”. Esta salida se puede comprimir de manera eficiente.

La transformación se realiza ordenando todas las rotaciones del texto en orden lexicográfico, y seleccionando la última columna.



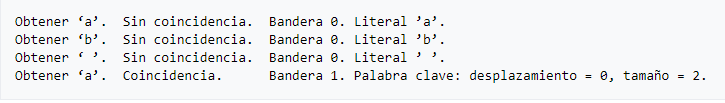
**3.3.3 LZ77:**

Lo que diferencia a este algoritmo de lo demás es que en este no se presenta omisión de datos al comprimirlo.

Este algoritmo se conforma de los siguientes tipos de datos, literales, banderas y palabras clave.

Primero, antes de usar estos datos se hace uso de un bit que es el que actuará como bandera que ayudará al programa a saber que tipo de dato viene a continuación, Si el prefijo es 0, entonces lo que viene es un byte sin comprimir. Si, por el contrario, el prefijo es 1, entonces lo que sigue a continuación es un par desplazamiento/tamaño. En el siguiente paso están las palabras clave, que son grupos de bits o bytes que contienen información requerida por el compresor y el descompresor, y por último estará el literal que será un byte sin comprimir

* **Literales**: Son simplemente bytes sin comprimir.
* **Palabras Clave:** En nuestro caso son pares tamaño/desplazamiento.
* **Banderas:** Simplemente nos indican si los datos que hay a continuación son literales o palabras clave.

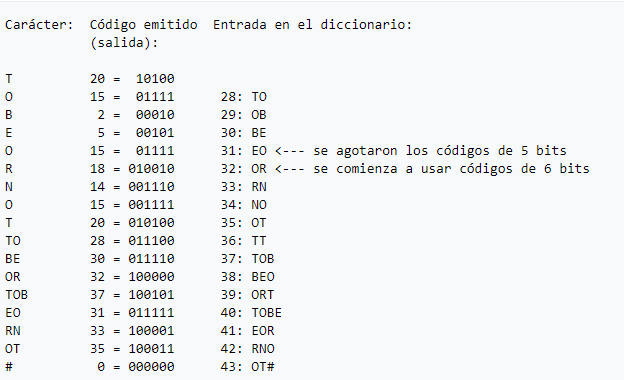


**3.3.4 LZW:**

Este algoritmo es por así decirlo una versión mejorada del LZ78, sirve para comprimir secuencias de bits sin tener en cuenta el tipo de información, su principio consiste en sustituir patrones con un código de índice y construir progresivamente un diccionario.

Para la construcción del diccionario pasa que este comienza con los 256 valores de la tabla ASCII. El archivo para comprimir se divide en cadenas de bytes cada una de estas cadenas se compara con el diccionario y se agrega si no se encuentra ahí, lo cual sería el proceso total de compresión.

Durante la descompresión, el algoritmo reconstruye el diccionario en la dirección opuesta; por lo tanto, no necesita almacenarse.



# **REFERENCIAS:**

1. Revista de Salud Animal. Cálculo automatizado del vector eléctrico integral de la actividad ventricular cardíaca en equinos. Retrieved August 13, 2021.

<http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0253-570X2016000100007>.

1. Ciencia y Agricultura de la Universidad Pedagógica y Tecnológica de Colombia. Caracterización de variables utilizando inteligencia computacional para identificar alteraciones en la salud. Retrieved August 13, 2021.

<https://www.redalyc.org/pdf/5600/560058660005.pdf>

1. Pérez, S. DESARROLLO DE UNA BASE DE DATOS CON IMÁGENES TERMOGRÁFICAS PARA USO EN ALGORITMOS DE VISIÓN E INTELIGENCIA ARTIFICIAL. Retrieved August 13, 2021.

<https://www.redalyc.org/pdf/5600/560058660005.pdf>

1. Chelotti, J. Desarrollo e implementación de un dispositivo de adquisición y almacenamiento de sonidos para ganadería de precisión. Retrieved August 13, 2021.

<http://sedici.unlp.edu.ar/handle/10915/42007>

1. Programador Click. algoritmo de talla de costura. Retrieved August 14, 2021.

<https://programmerclick.com/article/4510360417/>

1. Bercial, J. Integer Scalling: Qué es y como funciona. Retrieved August 14, 2021.

<https://www.geeknetic.es/Guia/1662/Integer-Scaling-Que-es-y-como-funciona.html>

1. Osuna, J. Introducción a lo Fractales y a la compresión Fractal. Retrieved August 14, 2021.

<http://www.iespravia.com/carpetas/recursos/mates/recursos_2005/fotografia/intro/intro.htm>

1. Ingeniería de la Universidad Distrital Francisco José de Caldas. Compresión de imágenes con Wavelets y Multiwavelets. Retrieved August 14, 2021.

<https://www.redalyc.org/pdf/4988/498850160008.pdf>

1. Wikipedia. Codificación Huffman. Retrieved August 15, 2021.

<https://es.wikipedia.org/wiki/Codificaci%C3%B3n_Huffman>

1. Wikipedia. Compresión de Burrows-Wheeler. Retrieved August 15, 2021.

<https://es.wikipedia.org/wiki/Compresi%C3%B3n_de_Burrows-Wheeler>

1. Wikipedia. LZSS. Retrieved August 15, 2021.

<https://es.wikipedia.org/wiki/LZSS#Caracter%C3%ADsticas>

1. Wikipedia. LZW. Retrieved August 15, 2021.

https://es.wikipedia.org/wiki/LZW