Clasificación del estado de salud del ganado en base a imágenes comprimidas para el ahorro de batería.

Miguel Angel Calvache Giraldo
Universidad Eafit
Colombia
macalvachg@eafit.edu.co

Pablo Arango Castaño
Universidad EAFIT
Colombia
parangoc2@eafit.edu.co

Simón Marín
Universidad Eafit
Colombia
smaring1@eafit.edu.co

Mauricio Toro
Universidad Eafit
Colombia
mtorobe@eafit.edu.co

RESUMEN

El manejo óptimo de los datos ha sido una necesidad fundamental para el desarrollo y la mejora de nuevas tecnologías. Desde hace unos cuantos años, se han usado diversos algoritmos para optimizar el consumo de energía, el cual representa un componente vital para la creación de mejores y nuevos sistemas. Incluyendo el de la ganadería de precisión y otras actividades como los videojuegos, identificación visual, fotografía y muchos más.

Este informe tiene como objetivo analizar algoritmos de compresión para la optimización del consumo de batería en ganadería de precisión, además de presentar el diseño y resultados de un algoritmo basado en tablas de hash y árboles, para la compresión y descompresión de imágenes. Con el fin último de hacer uso de estas imágenes comprimidas poniéndolas a prueba al determinar con la ayuda de otro algoritmo de clasificación el estado de salud del ganado.

Palabras clave

Algoritmos de compresión, aprendizaje de máquina,

aprendizaje profundo, ganadería de precisión, salud animal.

1. INTRODUCCIÓN

Cada año, el ser humano consume más de 252 millones de toneladas de carne proveniente del ganado, que combinado con el crecimiento de la población y de este tipo de mercados se estima que anualmente la ganadería presenta un crecimiento aproximado del 1.15% a nivel mundial, trayendo a su vez un incremento en la población animal explotada y muchas necesidades consigo [1].

Así comienza a surgir el concepto de Ganadería de Precisión (GdP) un sistema de producción sostenible que ayuda a reducir costos de inversión e impacto ambiental y contribuye a incrementar la producción pecuaria y el bienestar animal. [2]

Uno de los factores claves y el tratado en este informe, es la clasificación del estado de salud del ganado, que muchas veces representa un gran desafío para los ganaderos por la cantidad de cabezas de ganado de las que están a cargo o por simple desconocimiento al momento de clasificar. De esta manera, en el contexto de Ganadería de Precisión, surge la necesidad de comprimir imágenes tomadas al ganado para clasificar por medio de un algoritmo la salud del animal de una manera óptima y rápida. [3]

1.1. Problema

En pocas palabras, el problema con el cual nos enfrentamos en este proyecto es la creación y diseño de un algoritmo de compresión de las imágenes del ganado para alcanzar la mejor tasa de compresión y tiempo posible, sin que la exactitud de la clasificación se afecte en más del 5% al momento de usar el algoritmo que determina la salud del animal.

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 comprensión de los datos.

Para dar solución a esta problemática, inicialmente se usó una estructura de datos con matrices de la biblioteca NumPy de python para guardar la gran cantidad de datos de los píxeles de las imágenes. Para la compresión de dichos datos se planteó el uso de dos métodos de compresión diferentes. Seam carving o “tallado de costuras” el cual es un método de compresión con pérdidas y la codificación huffman, el cual es implementado para compresión de datos sin pérdidas. Se decidió utilizar estos algoritmos debido a que son ampliamente conocidos, cumplen con los requerimientos que se buscaron y no afectan al algoritmo de clasificación

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 An Animal Welfare Platform for Extensive Livestock Production Systems

En el año 2018 dos universidades de Grecia, de manera conjunta realizaron un prototipo, el cual consistía en un collar idealizado para el monitoreo del ganado, tanto en características como localización como de aceleración al moverse. Cabe destacar que estos datos que se obtenían se almacenaban en tiempo real en la nube y podían ser observados desde una aplicación móvil. [4]

2.2 Técnicas automatizadas para el seguimiento del comportamiento y bienestar de pollos de engorde y gallinas ponedoras

La calidad del bienestar animal y la salud de los consumidores está estrechamente relacionada con el bienestar animal, por este motivo, el informe publicado por la universidad de Cambridge, se da a la tarea de analizar el monitoreo del comportamiento de pollos de engorde y gallinas ponedoras en base de una gran lista de artículos relacionados, además de proporcionar herramientas potenciales para llevar a cabo un monitoreo correcto por medio de sensores

inalámbricos portátiles con dispositivos de identificación por radiofrecuencia, que pueden identificar automáticamente pollos individuales, rastrear la ubicación y el movimiento de los individuos en tiempo real y cuantificar algunos rasgos de comportamiento en consecuencia y tecnología de procesamiento de imágenes, que puede considerarse una herramienta directa para medir comportamientos, especialmente los comportamientos de actividad y la alerta temprana de enfermedades. [5]

En lo que procesamiento de imágenes respecta se presenta una tabla con materiales y métodos típicos de procesamiento de imágenes automatizado utilizados en el monitoreo del comportamiento de los pollos: (Figura)

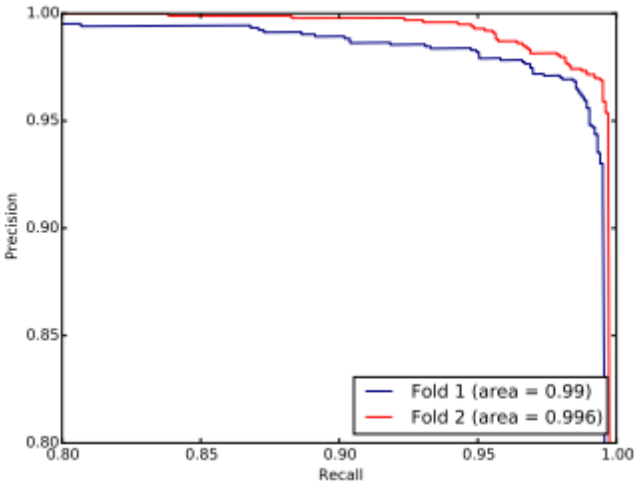
Reference	Camera type	Image pixel and frame rate	Main processing method	Detecting number	Main software
Zaninelli *et al.*, 2018	Thermografic camera	320 × 240; 1 fps	Threshold method and image pattern recognitions	10	LabVIEW, NI Vision Acquisition Software
Li *et al.*, 2018	IR camera	–	Threshold method	8	MATLAB
Pu *et al.*, 2018	Kinect camera	512 × 424; 30 fps	CNN	Commercial flocks	Open CV
Zhuang *et al.*, 2018	CCD camera	640 × 480	K-means clustering and the ellipse model	20	Open CV
Aydin *et al.*, 2017b	Camera	1024 × 768; 5 fps	Background subtraction and feature variables	1	MATLAB
Dawkins *et al.*, 2017	Camera	320 × 240	Optical flow	Commercial flocks	–
Fraess *et al.*, 2016	IR camera	320 × 240; 30 fps	The proportion of pixel changes	34	Etho Vision XT 10
Colles *et al.*, 2016	Web camera	320 × 240; 1 fps	Optical flow	Commercial flocks	MATLAB Python
Youssef *et al.*, 2015	CCD camera	640 × 480; 1 fps	The proportion of pixel changes	45	–
Kashiha *et al.*, 2014	Camera	1280 × 960; 0.5 fps	Binary image and count the pixel proportion	Commercial flocks	eYeNamic

(Figura)

2.3 Localización visual e identificación individual del ganado a través del Deep learning

En 2018 la universidad británica de Bristol se dio a la tarea de solucionar el problema al etiquetar el ganado bovino. En el que la identificación y la trazabilidad del ganado no solo es necesario para las demandas de los consumidores, sino, de hecho, muchos países han introducido marcos legales obligatorios para la protección del bienestar animal y evitar pérdidas y daños físicos que frecuentemente ocurren al marcar al ganado, ya sea con marcas auriculares, tatuajes u otras soluciones electrónicas. Así pues, los investigadores plantean proporcionar una prueba de concepto de que la identificación robusta de un tipo de ganado individual específico (Holstein Friesian) puede ocurrir de forma automática y no intrusiva, utilizando computer vision pipelines alimentadas por arquitecturas estándar que utilizan deep neural networks.

La arquitectura central consta de 5 capas convolucionales apiladas, más dos capas completamente conectadas. Se inicializaron los pesos con un modelo entrenado en la base de datos ImageNet suministrado con la implementación de Python Faster-RCNN para no iniciar el proceso desde cero. Tras la tarea de detección y localización de objetos que se producen por las imágenes, se crean regiones de interés (RoI) de imágenes en forma de cuadros delimitadores, para posteriormente ser pasadas a un proceso posterior, como un algoritmo de seguimiento o un clasificador. De una forma similar se trabajó con videos del ganado, pero las imágenes es lo que nos compete con este informe.[6]



2.4 Integración de servicios en la nube para estudios en el comportamiento de animales de granja usando teléfonos inteligentes como sensores de actividad

En este proyecto se buscó dar solución a la falta de información esencial sobre el comportamiento del ganado que es fundamental para conocer el estado reproductivo o de salud del animal. Para dar solución a este problema se usaron sensores de teléfonos inteligentes y los siguientes tres componentes claves fueron analizados: i) La ubicación obtenida por radiofrecuencia

triangulación o por sistema de posicionamiento global. ii) el componente de baja frecuencia del comportamiento como postura del animal (por ejemplo: posición de la cabeza, inclinación del cuello, etc.) y iii) El componente de comportamiento de alta frecuencia (por ejemplo, movimiento de las mandíbulas).

Por la gran cantidad de datos recolectados que debían ser enviados a la nube, una aplicación en Xaramin2 fue desarrollada para medir el compresibilidad de los datos al reducir la precisión con un algoritmo que actúa de la siguiente manera: (1) eliminando redundancias, es decir, reemplazo de datos redundantes por un intervalo de tiempo durante el cual los valores permanece constantes para preservar la integridad de los datos, y (2) truncando los datos a 3, 4 y 5 dígitos decimales, de los 6 dígitos decimales originales. Las métricas obtenidas tras la comprensión fueron las siguientes (Figura). [7]

Type of data	Variable number	Data treated	Data size (Mb)	Comp rate [%]
Acceleration in the X, Y, Z axis	3	25,920,000	98.87	4.26
Euler angles of the device	3	25,920,000	98.87	24.13
Attitude quaternion	4	34,560,000	131.84	24.13
Rotation matrix (3×3)	9	77,760,000	296.63	24.13
Gravitational component of 3D acceleration	3	25,920,000	98.87	24.13
User acceleration component of 3D acceleration	3	25,920,000	98.87	24.13
Rotation rate	3	25,920,000	98.87	24.13
Magnetic and true heading	3	25,920,000	98.87	0.01
Magnetometer data	3	25,920,000	98.87	60.79
Position (latitude, longitude)	3	25,920,000	98.87	0.01
Course and speed	2	17,280,000	95.92	99.75
Altitude	1	8,640,000	32.96	99.99

(Figura : Tasa de compresión obtenido para cada categoría de datos recogidos en 24h)

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

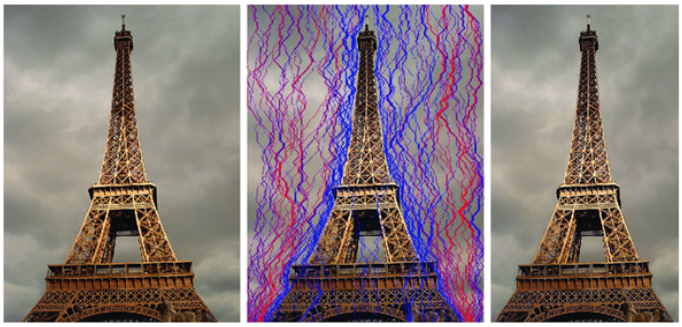
En lo que sigue, presentamos diferentes algoritmos usados para comprimir imágenes con pérdida.

3.2.1 Seam Carving

El algoritmo de Seam Carving o en español “Tallado de costuras” es un método de compresión con pérdidas desarrollado por Shai Avidan y Ariel Shamir que funciona principalmente para la reorientación de imágenes para evitar la distorsión del contenido en diferentes dispositivos de varios tamaños y que permite el cambio de tamaño de imagen de forma consciente de su contenido.

El algoritmo funciona estableciendo uniones entre píxeles que definen una ruta (Por eso el nombre de costura) en función de la energía de los pixeles comparados con los píxeles vecinos para determinar la ruta de píxeles que se debe borrar (La de menor energía), todo según el contenido de

la imagen. La complejidad está dada por el orden o(m n), ancho y largo de la imagen en pixeles. [8]



3.2.2 Transformada de coseno discreta

La transformada de coseno discreto, al igual que el método de compresión de Burrows Wheeler no es un método de compresión como tal, sino uno que facilita el proceso de compresión que está basado en la transformada de Fourier discreta, pero utilizando únicamente los números reales. La transformada de coseno discreta expresa una secuencia finita de varios puntos como resultado de la suma de distintas señales sinusoidales que trabaja con una serie de números finitos consiguiendo concentrar la mayor parte de la información en pocos coeficientes que son realmente relevantes para la compresión.

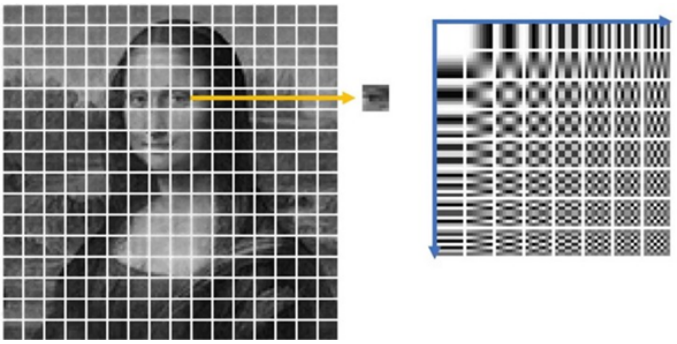
La complejidad del algoritmo dependerá del tipo de DCT que se esté empleando. Siendo las mas comunes: i) DCT-I que se usa empleando la fórmula:

$$X_k = \frac{1}{2}(x_0 + (-1)^k x_{N-1}) + \sum_{n=1}^{N-2} x_n \cos\left[\frac{\pi}{N-1} nk\right] \qquad k = 0, \ldots, N-1.$$

Con una complejidad de O(N log2 N) y ii) DCT-II que se usa empleando la fórmula:

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right] \qquad k = 0, \ldots, N-1.$$

Con una complejidad de O(N2 log2 N). [9]

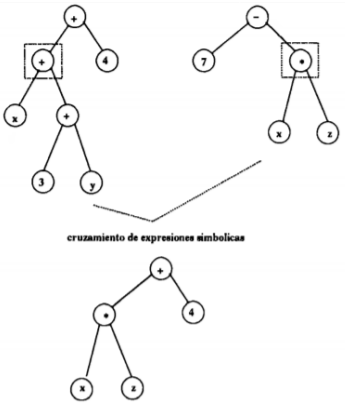


3.2.3 Programación Genética

La programación genética es una técnica relativamente reciente que ha demostrado ser una herramienta efectiva en la solución de algunas aplicaciones concretas de Generación Automática de Programas. La programación genética sigue el camino de los algoritmos genéticos (los cuales describiremos en el capítulo 3) para buscar en un espacio de posibles programas el que mejor puede realizar una tarea dada. Para la programación genética, un programa se representa como un árbol de parseo cuyos nodos son procedimientos, funciones, variables y constantes. Los subárboles de un nodo representan los argumentos del procedimiento o funciones de éstos.

En la programación genética, el operador básico es el de cruzamiento. Este operador trabaja de la siguiente forma: Primero, selecciona aleatoriamente un nodo de cada uno de los dos árboles sobre los que va a operar. Entonces, toma, del primer árbol, el subárbol que tiene como raíz el nodo seleccionado y lo reemplaza en el segundo árbol por el subárbol con raíz en su nodo seleccionado.

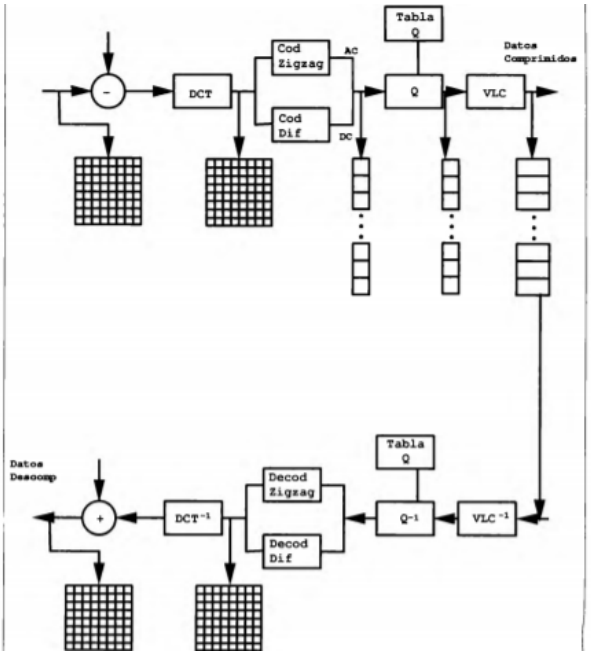
Para la compresión de imágenes usando programación genética, se realiza una regresión simbólica sobre un arreglo bidimensional de datos que representan el color de los píxeles de la imagen en cuestión. El objetivo de usar la regresión simbólica es encontrar un programa, o expresión simbólica de Lisp, que represente una imagen dada, de forma exacta o aproximada; de manera que, esta representación resulte ser más compacta que la imagen original.



La complejidad del algoritmo está dada por O(m,n) ancho y largo de la imagen en píxeles. [10]

3.2.4 JPEG

El sistema basado en líneas es la parte más simple del algoritmo propuesto por .JPEG. Este sistema consta de técnicas que son muy conocidas como la Transformada Discreta del Coseno, la cuantización uniforme y la codificación de Huffman. Además, hace uso también de algoritmos que proporcionan mayor compresión en imágenes. El sistema está construido de tal forma que permite que una imagen codificada sea transmitida secuencialmente, y sus píxeles sean reconstruidos por el decodificador siguiendo el orden de un cuadriculado. El algoritmo de compresión propuesto por JPEG consiste primeramente en dividir la imagen en bloques no superpuestos de 8 x 8 píxeles. A cada uno de estos bloques se le aplica la transformada discreta del coseno, de donde se obtienen 64 coeficientes representan el contenido de frecuencia del bloque. El primero de ellos mide la energía de la frecuencia cero o el término de corriente directa, los 63 restantes son los coeficientes de corriente alterna. Estos coeficientes son cuantizados a un conjunto finito de valores, ordenados en un arreglo unidimensional siguiendo un camino en zigzag y codificados usando una representación aritmética. [11]



La complejidad del algoritmo es igual a la del algoritmo DTC, es decir O(n^2 * log2 n)

3.3 Alternativas de compresión de imágenes sin pérdida

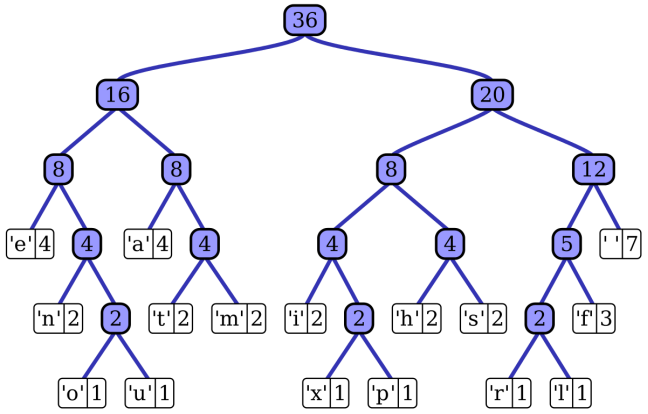
En lo que sigue, presentamos diferentes algoritmos usados para comprimir imágenes sin pérdida.

3.3.1 *codificación Huffman*

El código Huffman es un algoritmo desarrollado por David A. Huffman que es mundialmente usado para la compresión de datos sin pérdidas. El algoritmo consiste en la creación de un árbol binario en el que los nodos hoja son etiquetados por los caracteres junto a su frecuencia y de forma consecutiva se van uniendo cada pareja de nodos que menos frecuencia sumen, pasando a crear un nuevo nodo intermedio etiquetado con dicha suma. Repitiendo esta acción hasta que no quedan nodos hoja por unir a ningún otro nodo superior.

En términos simples la creación del árbol se realiza de la siguiente manera. Inicialmente, se crea un nodo hoja para cada símbolo, asociando un peso según su frecuencia de aparición que es insertado en la lista ordenada ascendentemente.

Posteriormente, mientras haya nodos en la lista realizar los siguientes pasos: I) Eliminar los dos nodos con menor probabilidad de la lista. II) Crear un nuevo nodo interno que enlace a los nodos anteriores, asignándole como peso la suma de los pesos de los nodos hijos. III) Insertar el nuevo nodo en la lista, (en el lugar que le corresponda según el peso). Y por último el nodo que queda al final sera el encargado de crear la raíz del árbol binario. La complejidad de dicho algoritmo está dada por el número de símbolos que se codifican “n” y esta dada por el orden O(nlogn) que es el peor de los casos al ordenar las frecuencias de los símbolos. [12]



3.3.2 Transformación de Burrows-Wheeler

La compresión de Burrows-Wheeler es un algoritmo inventado por Michael Burrows y David Wheeler en 1994 que es usado en técnicas de compresión de datos, pero que en realidad no es una técnica de compresión como tal sino una transformación permutada del orden de caracteres que no altera el valor de los caracteres y que puede ser revertido con facilidad. Al usar dicho algoritmo se creará una nueva cadena transformada que contendrá múltiples posiciones en las que un mismo carácter esté repetido varias veces en una fila, lo que permite una compresión fácil ya que contiene secuencias de caracteres repetidos. La transformación se realiza ordenando todas las rotaciones del texto en orden lexicográfico, y seleccionando la última columna de letras de todas las cadenas rotadas. Generando una complejidad de orden O(n) en la que la “n” representa la cantidad de caracteres de la cadena. [13]

		F	L
mississippi#		#	mississipp i
ississippi#m		i	#mississip p
ssissippi#mi		i	ppi#missis s
sissippi#mis		i	ssippi#mis s
issippi#miss		i	ssissippi# m
ssippi#missi	$\Longrightarrow$	m	ississippi #
sippi#missis		p	i#mississi p
ippi#mississ		p	pi#mississ i
ppi#mississi		s	ippi#missi s
pi#mississip		s	issippi#mi s
i#mississipp		s	sippi#miss i
#mississippi		s	sissippi#m i

3.3.3 RLE

El principio fundamental consiste en codificar un primer elemento al dar el número de repeticiones de un valor y después el valor que va a repetirse. Por lo tanto, según este principio, la cadena “AAAAAHHHHHHHHHHHHHH” cuando está comprimida da como resultado "5A14H". La ganancia de compresión es (19-5) / 19, es decir, aproximadamente 73,7%. Por otro lado, para la cadena "CORRECTLY", donde hay poca repetición de caracteres, el resultado de la compresión es “1C1O2R1E1C1T1L1Y”. Por lo tanto, la compresión genera un costo muy elevado y una ganancia de compresión negativa de (9-16) / 9, es decir, -78%

En realidad, la compresión RLE está regida por reglas particulares que permiten que se ejecute la compresión cuando sea necesario y que se deje la cadena como está cuando la compresión genere pérdida. Las reglas son las siguientes:

- Si se repiten tres o más elementos consecutivamente, se utiliza el método de compresión RLE.
- De lo contrario, se inserta un carácter de control (*00*) seguido del número de elementos de la cadena no comprimida y después la última.

- Si el número de elementos de la cadena es extraño, se agrega el carácter de control (*00*) al final.
- Finalmente, se definen los caracteres de control específicos según el código:
    - un final de línea (*00 01*)
    - el final de la imagen (*00 00*)
    - un desplazamiento de puntero sobre la imagen de XX columnas e YY filas en la dirección de lectura (*00 02 XX YY*).

Por lo tanto, no tiene sentido utilizar la compresión RLE excepto para datos con diversos elementos repetidos de forma consecutiva, en imágenes particulares con áreas grandes y uniformes. Sin embargo, la ventaja de este método es que es de fácil implementación. Existen alternativas en las que la imagen está codificada en bloques de píxeles, en filas o incluso en zigzag, siendo su complejidad de O(n), donde n es la cantidad de elementos, específicamente píxeles. [14]

3.3.4 LZW

LZW es un algoritmo muy rápido tanto para la compresión como para la descompresión, basado en la multiplicidad de aparición de secuencias de caracteres en la cadena que se debe codificar. Su principio consiste en sustituir patrones con un código de índice y construir progresivamente un diccionario.

Además, funciona en bits y no en bytes, por lo tanto, no depende de la manera en que el procesador codifica información. Es uno de los algoritmos más populares y se utiliza particularmente en formatos TIFF y GIF.

El diccionario comienza con los 256 valores de la tabla ASCII. El archivo a comprimir se divide en cadenas de bytes (por lo tanto, para las imágenes monocromáticas codificadas en 1 bit, esta compresión no es muy eficaz), cada una de estas cadenas se compara con el diccionario y se agrega si no se encuentra ahí.

Su complejidad es de O(m,n) donde se representan los píxeles como un arreglo bidimensional. [15]

4. DISEÑO E IMPLEMENTACIÓN DE LOS ALGORITMOS

En lo que sigue, explicaremos 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

La estructura de datos utilizada para hacer la compresión de las imágenes son matrices de la biblioteca NumPy de Python. Por ejemplo, el siguiente conjunto de datos entregado en formato csv separado por comas:

```
255,0,120,255,255,255,255,255,255,255
255,120,0,120,120,255,255,80,255,255
255,255,120,0,120,120,255,255,255,255
255,120,0,0,0,120,120,255,255,255
255,255,120,0,120,120,255,255,0,255
255,255,255,120,0,120,255,255,255,0
```

sería organizado en una matriz numpy de la siguiente manera:

```
[[255,0,120,255,255,255,255,255,255,255],
[255,120,0,120,120,255,255,80,255,255],
[255,255,120,0,120,120,255,255,255,255],
[255,120,0,0,0,120,120,255,255,255],
[255,255,120,0,120,120,255,255,0,255],
[255,255,255,120,0,120,255,255,255,0]]
```

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.

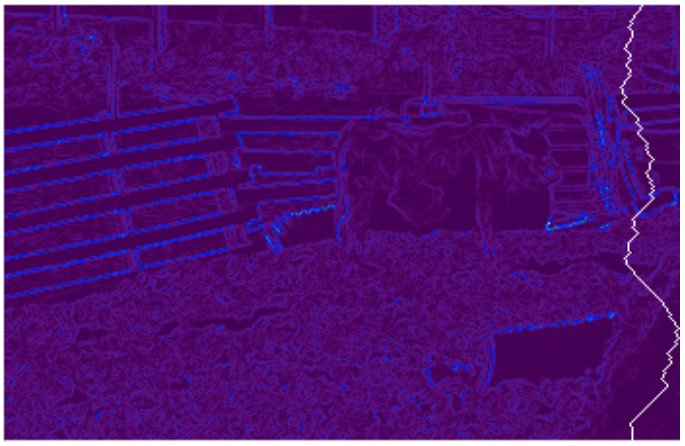
4.2.1 Algoritmo de compresión de imágenes con pérdida

En el algoritmo de compresión con pérdidas utilizado, hay una reducción en la cantidad de información que no podrá ser reconstruido exactamente igual a la original, pero que no representa un problema gracias a que dicho cambio no es significativo al momento de ser clasificado por el algoritmo que decide el estado de salud del animal.

En este caso, se implementó el método de Seam Carving o “tallado de costuras” el cual funciona de la siguiente manera:

Imagen original (Sacada del repertorio de las imágenes por clasificar)

Se asigna un valor de energía a cada píxel y se crea una ruta de píxeles con la menor energía posible, los cuales serán borrados.



La línea blanca representa la ruta de pixeles con menor energía próxima a ser borrada

Y se eliminan la cantidad de píxeles que sean necesarios.

A la izquierda se muestra la imagen comprimida a la que se le eliminaron varios pixeles, y a la derecha la imagen original.

REFERENCIAS

1. Valeria Errecat, Mariana Lucero, and Maria Alejandra Sosa. 2015.(May 2015). ANÁLISIS DEL MERCADO MUNDIAL DE CARNES. Retrieved August 2021 from http://www.unsam.edu.ar/escuelas/economia/economia_regional/CERE%20-%20Mayo%20-%202015.pdf
2. Orlando Santillan. Ganadería de Precisión. Retrieved August 2021 from https://foroconsultivo.org.mx/INCyTU/index.php/notas/102-23-ganaderia-de-precision
3. Anon. 2013.(November 2013). IMPLEMENTACIÓN DE BUENAS PRÁCTICAS PARA EL MANEJO ADAPTATIVO DEL SISTEMA PECUARIO Y LA CONSERVACIÓN DEL ECOSISTEMA PÁRAMO EN LA MICROCUENCA DE PAPALLACTA Retrieved August 2021 from https://www.ambiente.gob.ec/wp-content/uploads/downloads/2014/07/Gu%C3%ADa-Sanitaria-Ganado.pdf
4. Doulgerakis, V., Kalyvas, D., Bocaj, E., Giannousis, C., Feidakis, M., Laliotis, G. P., Patrikakis, C. and Bizelis, I. (2020). An Animal Welfare Platform for Extensive Livestock Production Systems.
5. N. Li, Z. Ren, D. Li, and L. Zeng. 2019. Review: Automated techniques for monitoring the behaviour and welfare of broilers and laying hens: Towards the goal of precision livestock farming: Animal. (September 2019). Retrieved August 15, 2021 from https://www.cambridge.org/core/journals/animal/article/review-automated-techniques-for-monitoring-the-behaviour-and-welfare-of-broilers-and-laying-hens-towards-the-goal-of-precision-livestock-farming/7D334A718C877E8E8F8DDB660EC98A4F
6. Andrew, W., Greatwood, C., & Burghardt, T. (2018). Visual Localisation and Individual Identification of Holstein Friesian Cattle via Deep Learning. In 2017 IEEE International Conference of Computer Vision Workshop (ICCVW 2017) (pp. 2850-2859). Institute of Electrical and Electronics Engineers (IEEE). https://doi.org/10.1109/ICCVW.2017.336
7. Olivier Debauche, Saïd Mahmoudi, Andriamasinoro Lalaina Andriamandroso, Pierre Manneback, Jérôme Bindelle, and Frédéric Lebeau. 2018. Cloud services integration for farm animals' behavior studies based on smartphones as activity sensors. *Journal of Ambient Intelligence and Humanized Computing* 10, 12 (2018), 4651–4662. DOI:http://dx.doi.org/10.1007/s12652-018-0845-9
8. Shai Avidan Mitsubishi Electric Research Labs et al. 2007. Seam carving for content-aware image resizing. (August 2007). Retrieved September 29, 2021 from https://dl.acm.org/doi/10.1145/1275808.1276390
9. Chen, Weng-Hsiung; Smith, C. Harrison; Fralick, Stanley C. (Septiembre de 1977). «A Fast Computational Algorithm for the Discrete Cosine Transform». *IEEE TRANSACTIONS ON COMMUNICATIONS* **25**.
10. Programación genetica
11. JPEG
12. D.A. Huffman, "A method for the construction of minimum-redundancy codes", Proceedings of the I.R.E., sept 1952, pp 1098-1102
13. Burrows M and Wheeler D (1994), *A block sorting lossless data compression algorithm*, Technical Report 124, Digital Equipment Corporation
14. Sandoval, M. (2008). ALGORITMO DE COMPRESIÓN DE IMÁGENES DE ALTA RESOLUCIÓN SIN PÉRDIDAS. [Tesis Pregrado, Instituto Politécnico Nacional]. https://tesis.ipn.mx/jspui/bitstream/123456789/5745/1/ALGORITMODECOMPR.pdf
15. RLE
16. LZW

---

[1] https://github.com/acpablox/ST0245-002