

Gestión eficiente de la ganadería en tiempos modernos

Rafael Villegas
Universidad Eafit
Colombia

rvillegasr@eafit.edu.co

Federico Velez
Universidad Eafit
Colombia

fvelezq@eafit.edu.co

Simón Marín
Universidad Eafit
Colombia

smaringl@eafit.edu.co

Mauricio Toro
Universidad Eafit
Colombia

mtorobe@eafit.edu.co

José David Gómez

Universidad Eafit

Colombia

jdgomezz@eafit.edu.co

Manuela Franco Orozco

Universidad Eafit

Colombia

mfrancool@eafit.edu.co

Texto negro = Contribución de Simón y Mauricio

Texto en verde = Completar para el 1er entregable

Texto en azul = Completar para el 2º entregable

Texto en violeta = Completar para el tercer entregable

RESUMEN

Para escribir un resumen, deben responder a las siguientes preguntas en un solo párrafo: ¿Cuál es el problema? ¿Por qué es importante el problema? ¿Cuáles son los problemas relacionados?

En este trabajo el problema sería identificar cuál y cómo implementar el algoritmo de compresión más útil para que este método de automatización de identificación de la salud animal trabaje eficientemente y con la mayor eficacia, y cómo se logra identificar que hay muchos problemas relacionados con “Precision Livestock farming” como por ejemplo la automatización de la monitorización en las vacas lecheras que ya en muchas partes del mundo se está tomando esta herramienta como solución y como este ejemplo hay muchos reales que están en uso actualmente por medio de la compresión de datos y es por esto que es de suma importancia el uso de compresión de datos a la hora de utilizar GdP.

Se utilizaron dos algoritmos de compresión de imágenes, el algoritmo de interpolación bilineal el cual es con pérdidas de datos que la forma en la que comprime la imagen ocurre en un espacio continuo, ósea se pueden tener valores decimales de los píxeles y encontrar ese píxel intermedio que hay entre dos píxeles vecinos, se dan pérdida de datos y el tiempo de ejecución será mucho más lenta. Y por último el $O(n \log n)$

Palabras clave

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

1. INTRODUCCIÓN

Explica la motivación, en el mundo real, que lleva al problema. Incluyan la historia de este problema. *(En este semestre, la motivación es la razón por la que necesitamos comprimir imágenes para clasificar la salud animal en el contexto de la ganadería de precisión).*

Actualmente el uso de la ganadería de precisión (GdP) está siendo implementado por muchos de los ganaderos en todo el mundo. Un problema general de la ganadería actual es el control de salud del ganado, para solucionar este problema se tiene que diseñar una forma de optimizar la memoria y el tiempo requerido basándose en los algoritmos de compresión para optimizar el control de salud del ganado basándose en fotos.

Problemas relacionados con la compresión de datos existen y estos serán analizados con la intención de hallar una solución efectiva.

1.1. Problema

En pocas palabras, expliquen el problema, el impacto que tiene en la sociedad y por qué es importante resolver el problema. *(En este semestre, el problema es comprimir las imágenes para clasificar la salud animal en el contexto de la ganadería de precisión).*

El problema existente es crear un algoritmo que por medio de una estructura de datos haga una búsqueda de archivos guardados de la información adecuada a lo que nos brinden los clientes (ganaderos).

la cual será implementada en una búsqueda rápida para encontrar de manera eficiente estos archivos y a su misma vez la implementación de la herramienta “Precision Livestock farming” como objetivo de ofrecer al ganadero un monitoreo de sanidad animal

Esto es de suma importancia y necesidad para aquellas empresas donde tienen una cantidad masiva de archivos e información importante.

01.2 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.

Expliquen, brevemente, su solución al problema (En este semestre, la solución es una implementación de algoritmos de compresión. ¿Qué algoritmos han elegido? ¿Por qué?)

1.3 Estructura del artículo

En lo que sigue, en la Sección 2, presentamos trabajos relacionales 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

“Unsupervised automated monitoring of dairy cows’ behavior based on Inertial Measurement Unit attached to their back”

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

El problema a resolver es La monitorización automatizada del comportamiento de las vacas lecheras basada en sensores no invasivos ofrece un gran potencial para mejorar los procesos de monitorización de su bienestar y salud en el contexto de la granja inteligente. Puede detectar cualquier cambio antes de la aparición de los signos clínicos, permitiendo al ganadero tomar las medidas necesarias lo antes posible. este método utilizado tiene como objetivo:

“The objective of this study is to develop an effective unsupervised classification model of data collected by Inertial Measurement Units (IMU) attached to the back of dairy cows housed in free-stall. “

(panelBrahimAchour, MalikaBelkadi, RachidaAoudjit, & MouradLaghrouche, 13 March 2019, Revised 18 July 2019, Accepted 20 October 2019, Available online 28 October 2019.)

3.2 Escriba un título para el segundo problema

“Real-Time Extensive Livestock Monitoring Using LPWAN Smart Wearable and Infrastructure”

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

La suelta de animales puede hacerse durante meses, en grandes áreas y con diferentes especies que se empacan y se comportan de manera muy diferente. Sin embargo, las necesidades del ganadero son similares: dónde está el ganado (y dónde ha estado) y cómo está de sano. Las zonas geográficas implicadas suelen ser de difícil acceso, con una orografía dura y falta de infraestructuras de comunicación.

“Our proposal is based in a wearable equipped with inertial sensors, global positioning system and wireless communications; and a Low-Power Wide Area Network infrastructure that can run with and without internet connection. Using adaptive analysis and data compression, we provide real-time monitoring and logging of cattle’s position and activities. Hardware and firmware design achieve very low energy consumption allowing months of battery life”

<https://www.mdpi.com/2076-3417/11/3/1240/htm>

(Casas, Hermosa, Marco, Blanco, & Zarazaga-Soria, Received: 31 December 2020 / Revised: 22 January 2021 / Accepted: 25 January 2021 / Published: 29 January 2021)

3.3 Escriba un título para el tercer problema

“Automatic cough detection for bovine respiratory disease in a calf house”

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

la enfermedad respiratoria bovina (ERB) es un importante problema de salud animal. Los ganaderos sufren graves pérdidas económicas a causa de esta enfermedad. Además de los costes económicos, los brotes de ERB perjudican el bienestar de los animales y se necesitan conocimientos y mano de obra adicionales para tratar y cuidar a los animales infectados. La tos se reconoce como una manifestación clínica de la ERB. Por lo tanto, la monitorización de la tos en un establo de terneros tiene el potencial de detectar los casos de infección respiratoria antes de que sean demasiado graves, y así limitar el impacto de la ERB tanto en el ganadero como en el animal. el objetivo es:

“The objective of this study was to develop an algorithm for detection of coughing sounds in a calf house. Sounds were recorded in four adjacent compartments of one calf house over two time periods (82 and 96 days). There were approximately 21 and 14 calves in each compartment over the two time-periods, respectively. The algorithm was developed using 445 min of sound data. These data contained 664 different cough references, which were labelled by a human expert. It was found that, during the first time period in all 3 of the compartments and during the second period in 2 out of 4 compartments, the algorithm worked very well”

(Lenn Carpentier, Tullo, Guarino, Vranken, & Norton, 2018)

3.4 Escriba un título para el cuarto problema

“Precision agriculture on grassland: Applications, perspectives and constraints”

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

La comprensión de los flujos de datos reales y potenciales en la práctica de la agricultura de precisión es un requisito sustancial para la mejora y automatización de la administración de la información en este entorno. Como contribución a este proceso, este artículo muestra un estudio de los flujos de datos en el ámbito de los trabajos de un plan de indagación en participación que se encarga de situar a prueba los procedimientos desarrollados en el marco del plan en 2 campos de demostración. Este trabajo da un óptimo análisis de caso para la modelización de una secuencia de flujos de datos que cubren un extenso espectro de técnicas.

“Using the notation and software tools for the Unified Modeling Language (UML), a complete model of all identified data-flows was created. Individual data-streams relating to particular source or product datasets were then extracted from this model. These data-streams present a practical application of the model in identifying the benefit that may be obtained from a particular gathered dataset (e.g. yield data) or in identifying the data that must be gathered to generate a particular product dataset (e.g. sustainability indicators).”

<https://www.sciencedirect.com/science/article/abs/pii/S1161030108000646>

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

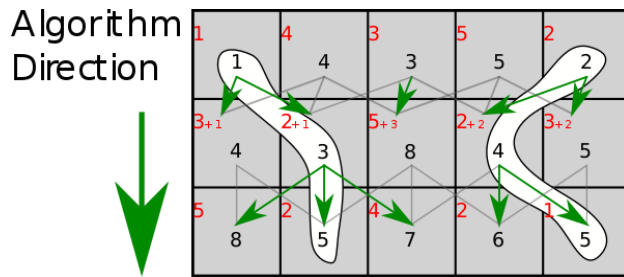
En lo que sigue, presentamos diferentes algoritmos usados para comprimir imágenes con pérdida. (*En este semestre, ejemplos de tales algoritmos son el tallado de costuras, el escalado de imágenes, la transformación de coseno discreto, la compresión con ondeletas y la compresión fractal*).

3.2.1 Tallado de Costuras (Seam Carving)

Por favor, explique el algoritmo, su complejidad e incluye una figura vectorial.

Algoritmo usado para cambiar el tamaño de una imagen recortando píxeles para no distorsionar la imagen.

El recorte de los píxeles se da mediante las costuras. Las costuras son rutas de píxeles conectadas en una imagen, estas se dan de arriba a abajo de cada fila de píxeles de la imagen. Para poder utilizar este algoritmo se tiene que identificar la imagen por lo que es útil pasarla a una escala de grises. Con la imagen ya identificada lo que se tiene que hacer es identificar los píxeles de baja energía para eliminarlos y así dejar la imagen sin ninguna distorsión, pero de distinto tamaño. (Wikipedia, 2020)

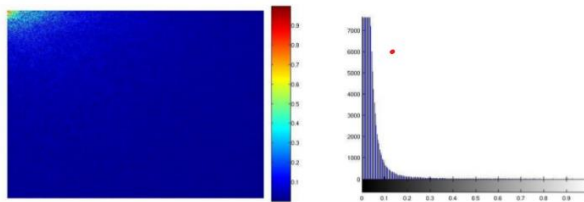


3.2.2 Transformación del Coseno Discreto (DCT)

La transformada del coseno discreta trabaja con una serie de números finitos que se ven como como puntos que sumados resultan en señales con distintas frecuencias y amplitudes. La DCT trabaja con cosenos principalmente. Además de esto existe también la DCT multidimensional que se puede considerar como la multiplicación de varias transformada del coseno discreto, por ejemplo, la DCT de dos dimensiones es una transformada normal para calcular las filas y las columnas.

Con lo anterior mencionado la DCT logra compactar la energía por lo que es útil a la hora de compactar imágenes y uno de sus beneficios es que no hay casi errores a la hora de los bordes de una imagen comprimida. (Wikipedia, 2021)

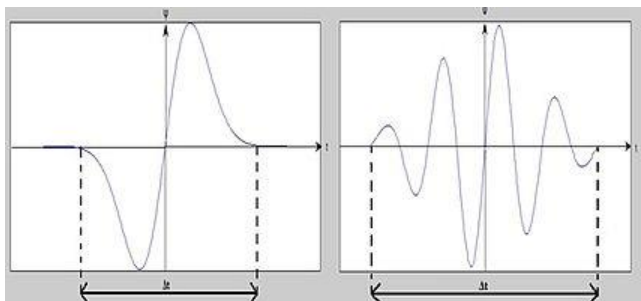
DCT



Por favor, explique el algoritmo, su complejidad e incluya una figura vectorial.

3.2.3 La compresión con ondeletas (Wavelets)

Las wavelets u ondelettes son funciones que satisfacen ciertos requerimientos matemáticos y son utilizados para la representación de datos. La idea fundamental de las wavelets es analizar funciones de acuerdo con unas escalas. Por este motivo las wavelets son utilizados a la hora de comprimir imágenes ya que éstas son datos y estos datos son representados siempre como señales digitales bidimensionales. Por eso este algoritmo sirve para comprimir las imágenes mediante ondas. (Guerrero & Escobar, 2007)

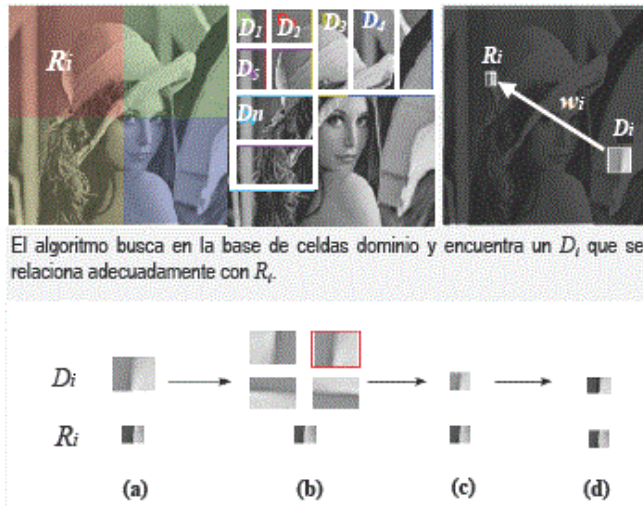


Por favor, explique el algoritmo, su complejidad e incluya una figura vectorial.

3.2.4 Comprensión fractal

La compresión fractal es un método de compresión de imágenes mediante fractales como su nombre lo indica. Este es uno de los algoritmos más adecuados para las imágenes naturales o que tienen texturas.

Este algoritmo funciona partiendo de una parte de la imagen y este lo convierte en datos que se llaman códigos fractales y estos datos se utilizan para recrear la imagen codificada. (Wikipedia, 2019)



Por favor, explique el algoritmo, su complejidad e incluya una figura vectorial.

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 Burrows y Wheeler, LZ77, LZ78, la codificación Huffman y LZS).

3.3.1 Compresión de Burrows-Wheeler (BWT)

Por favor, explique el algoritmo, su complejidad e incluya una figura vectorial.

Es un algoritmo usado en técnicas de compresión de datos como en bzip2 (es un programa libre que se encarga de comprimir y descomprimir archivos).

Este algoritmo funciona tomando un bloque de datos y ordenándolos con un algoritmo. El bloque de salida resultante va a contener los mismos elementos de datos con los que inició, cambiando en su ordenamiento. La transformación puede revertirse, lo que significa que puede volver a su estado original. (Wikipedia, 2021)

Transformación			
Entrada	Todas las Rotaciones	Ordenar Filas	Salida
\wedge BANANA@	\wedge BANANA@ @ \wedge BANANA A@ \wedge BANAN NA@ \wedge BANA ANA@ \wedge BAN NANA@ \wedge BA ANANA@ \wedge B BANANA@ \wedge	ANANA@ \wedge B ANA@ \wedge BAN A@ \wedge BANAN BANANA@ \wedge NANA@ \wedge BA NA@ \wedge BANA \wedge BANANA@ @ \wedge BANANA	BNN \wedge AA@A

3.3.2 Codificación Huffman

Por favor, explique el algoritmo, su complejidad e incluya una figura vectorial.

Es un algoritmo usado para comprimir datos, esta usa un método específico para elegir la representación de cada símbolo, que da lugar a un código prefijo (“la cadena de bits que representa a un símbolo en particular nunca es prefijo de la cadena de bits de un símbolo distinto”).

La asignación de la cadena es de la siguiente manera:

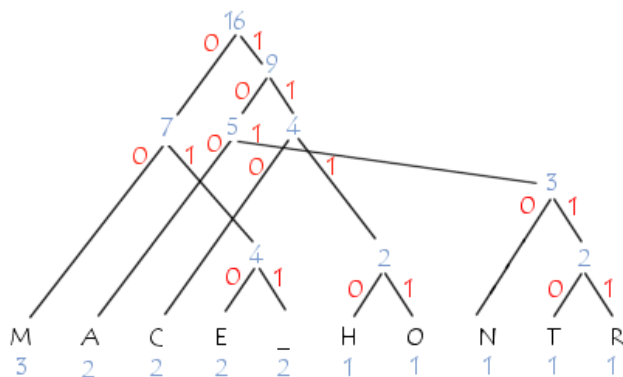
“Se calcula la probabilidad que tiene cada carácter en aparecer”.

“Se toman los dos caracteres menos probables y a uno de ellos se le asigna un 0 y al otro un 1 como inicio de su código”.

“Se suman las probabilidades de estos dos caracteres y esta suma es asignada a un carácter virtual, que no posee representación, pero participa en las elecciones”.

“Los caracteres anteriormente tomados ya no participarán de las elecciones. Sin embargo, si el carácter virtual es eventualmente elegido, el código asignado al mismo irá directamente seguido a los códigos del par de caracteres que lo crearon”.

“Se repite el proceso hasta que se cree el carácter virtual de probabilidad 1”. (Wikipedia, 2021)



3.3.3 Codificación Lempel-Ziv

Por favor, explique el algoritmo, su complejidad e incluya una figura vectorial.

Es una codificación de tamaño variable a fijo basado en un diccionario. La idea es separar la entrada en bloques no superpuestos de diferentes longitudes mientras se construye un diccionario de bloques ya vistos.

se deben seguir los siguientes pasos

“Inicializar el diccionario para que contenga todos los bloques de longitud 1, esto para todos los posibles bloques de una letra.”

“Buscar en la entrada el bloque más largo que aparece en el diccionario”.

“Codificar el bloque por su índice en el diccionario”.

“Agregar el bloque encontrado seguido del siguiente símbolo del siguiente bloque en el diccionario. Si se ha establecido un límite en la cantidad de entradas, no agregar nuevos bloques si se ha llegado al mismo”.

“Seguir con este procedimiento hasta terminar la entrada”. (Gimenez, 2004)

a	b	b	a	a	b	b	a	a	b	a	b	b	a	a	a	b	a	a	b	b	a
0	1	1	0	2	4	2	6	5	5	7	3	0									

Índice	Entrada	Índice	Entrada	Índice	Entrada
0	a	5	aa	10	aaa
1	b	6	abb	11	aab
2	ab	7	baa	12	baab
3	bb	8	aba	13	bba
4	ba	9	abba		

3.3.4 La codificación Run-Length

Por favor, explique el algoritmo, su complejidad e incluya una figura vectorial.

Es la más simple de todas las técnicas de compresión. Esta forma de compresión de datos sin pérdidas se encarga de almacenar series de datos como un único valor de datos y un recuento, en lugar de como la serie original.

Esto es muy útil para los datos que contienen muchas ejecuciones de este tipo. Por ejemplo, en el caso de imágenes gráficas sencillas, como iconos, dibujos lineales, el Juego de la Vida de Conway y animaciones, lo cual la convierte en una perfecta opción para nuestro problema de GdP.

Esta forma de codificación no es realizable de manera directa ya que no hay manera de diferenciar un número de un código de repetición. Lo más común es utilizar bytes con signo para indicar la presencia o ausencia de repeticiones. Si el byte encontrado es negativo, significa que los siguientes n bytes (indicado por el byte) deben ser escritos de manera normal. Si el byte es positivo, significa que el siguiente byte debe ser repetido n veces (indicado por el byte). (Wikipedia, 2020)

Character	Code	Pattern received	Distance between received pattern and code
A	000000	010100	2
B	001111	010100	4
C	010011	010100	3
D	011100	010100	1
E	100110	010100	3
F	101001	010100	5
G	110101	010100	2
H	111010	010100	4

Smallest distance

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.

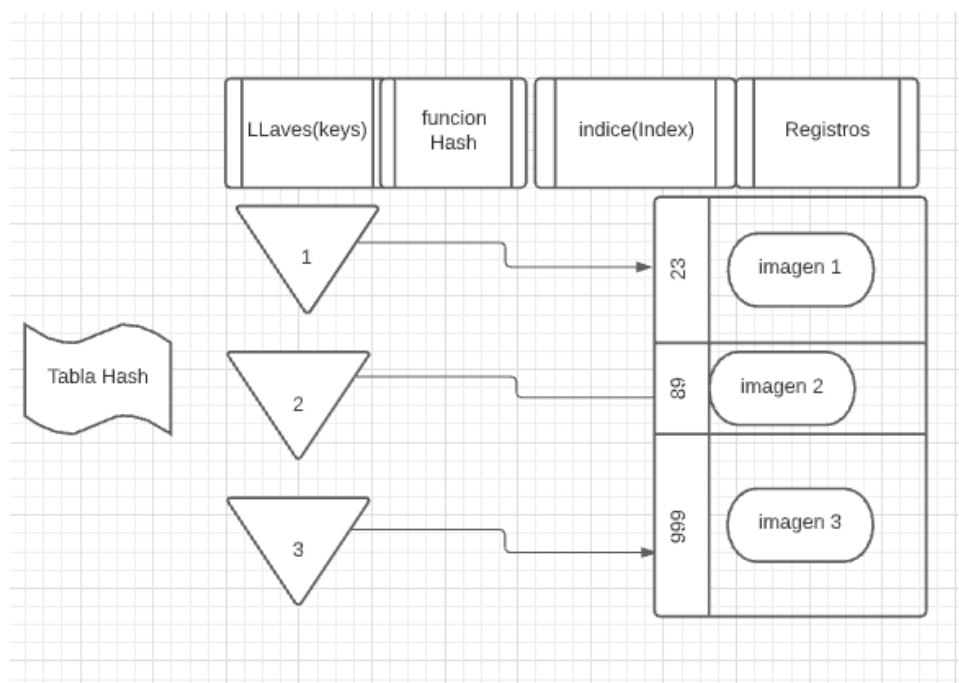
4.1 Estructuras de datos

Explique la estructura de datos utilizada para hacer la compresión de las imágenes y haga una figura que la explique. No utilice figuras de Internet. (En este semestre, ejemplo de las estructuras de datos son los árboles y las tablas hash)

Tabla de HASH

La tabla hash es una estructura de datos que almacena datos de forma asociativa. En una tabla hash, los datos se almacenan en un formato de matriz, donde cada valor de los datos tiene su propio valor de índice único. El acceso a los datos es muy rápido si conocemos el índice de los datos deseados.

Así se convierte en una estructura de datos en la que las operaciones de inserción y búsqueda son muy rápidas independientemente del tamaño de los datos. La tabla Hash utiliza un array como medio de almacenamiento y utiliza la técnica hash para generar un índice en el que se inserta o se localiza un elemento.



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 de imágenes con pérdida usamos el algoritmo de interpolación bilineal. Este consiste en calcular los valores de una ubicación de una malla basada en los cuatro centros de celda más cercanos. (Martínez, 2020)

Utilizando estas cuatro celdas, la interpolación bilineal asigna el valor de la celda de salida haciendo un promedio ponderado. Este algoritmo sacrifica resolución por tamaño. Y para la descompresión de este algoritmo duplica los píxeles $n \times n$. Esto significa que los archivos incrementan el tamaño, pero se queda con los mismos datos que tenía el archivo cuando estuvo comprimido.

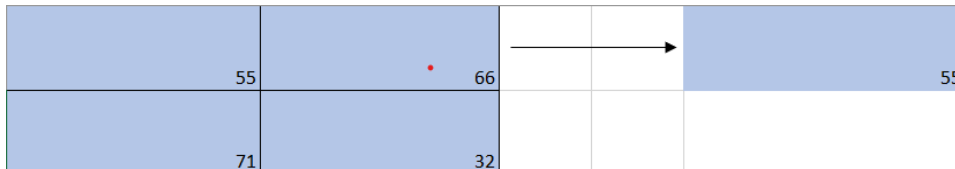
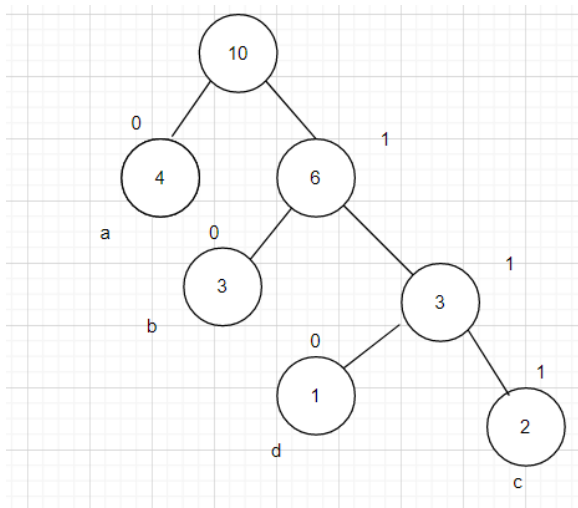


Figura 2: Compresión y descompresión de imagen usando interpolación bilineal

4.2.2 Algoritmo de compresión de imágenes sin pérdida

Para la compresión de imágenes sin pérdida decidimos utilizar el algoritmo de Huffman. Este algoritmo consiste en crear un árbol binario en el que se etiquetan los nodos hoja con los caracteres, junto a sus frecuencias. De forma consecutiva se unen los nodos que menos frecuencia sumando, generando así otro nodo que contenga la suma. Esto genera el árbol binario y con este se pueden disminuir los bits necesarios que se utilizan en el archivo.

El proceso anterior se puede revertir para que quede el archivo con la misma información y no haya pérdida de información.



4.3 Análisis de la complejidad de los algoritmos

Explique, con sus propias palabras, el análisis del peor caso usando la notación O . ¿Cómo calculó tales complejidades? Por favor, explique brevemente.

Algoritmo	La complejidad del tiempo
Algoritmo de descompresión sin pérdida(huffman)	$O(N \log N)$
Compresión sin pérdidas(huffman)	$O(N \log N)$

Tabla 2: Complejidad temporal de los algoritmos de compresión y descompresión de imágenes.

N es el número de nodos que recorre del árbol que se crea con el algoritmo de Huffman.

Algoritmo	Complejidad de la memoria
Algoritmo de descompresión sin pérdida(huffman)	$O(N \log N)$
Compresión sin pérdidas(huffman)	$O(N \log N)$

Tabla 3: Complejidad de memoria de los algoritmos de compresión y descompresión de imágenes.

N es el numero de nodos que recorre del arbol que se crea con el algortimo de Huffman.

4.4 Criterios de diseño del algoritmo

Se diseño y utilizo este algoritmo porque es eficaz y poco complejo, además de que uno de los algoritmos ocupa más memoria y es más veloz y el otro ocupa menos memoria y es menos veloz por lo cual uno se complementa con el otro y se genera un balance entre ellos.

5. RESULTADOS

5.2 Tiempos de ejecución

En lo que sigue explicamos la relación entre el tiempo promedio de ejecución y el tamaño promedio de las imágenes del conjunto de datos completo, en la Tabla 6.

Calcular el tiempo de ejecución de cada imagen en Github. Informar del tiempo medio de ejecución vs. el tamaño medio del archivo.

Huffman	<i>Tiempo promedio de ejecución (s)</i>	<i>Tamaño promedio del archivo (MB)</i>
<i>Compresión sin pérdidas</i>	0.69333 s	0.0621111111 MB
<i>Descompresión sin pérdidas</i>	0.6051 s	0.0621111111 MB

Tabla 6: Tiempo de ejecución del algoritmo de huffman para diferentes imágenes en el conjunto de datos.

6. DISCUSIÓN DE LOS RESULTADOS

6.1 Trabajos futuros

Responda ¿qué le gustaría mejorar en el futuro? ¿Cómo le gustaría mejorar su algoritmo y su implementación? ¿Qué tal usar la transformación de coseno discreto o la compresión con ondeletas a futuro?

Para trabajos futuros nos gustaría poder lograr todos los objetivos propuestos en el trabajo, sin embargo, nos gustaría encontrar una manera de que al utilizar la codificación Huffman, hubiera alguna manera de que esta fuese más rápida al momento de leer o escribir archivos.

RECONOCIMIENTOS

Gracias al monitor Simón por responder las preguntas siempre que se le hicieron, ya que no solo nos ayudó en el proyecto, también lo hizo cuando lo necesitamos en los laboratorios y talleres.

REFERENCIAS

3^{il}.3

Referencias

- Carpentier, L., Berkmans, D., Youssef, A., Berkmans, D., van Waterschoot, T., Johnston, D., . . . Norton, T. (2018, Julio 6). *Automatic cough detection for bovine respiratory disease in a calf house*. Retrieved from Science Direc:
<https://www.sciencedirect.com/science/article/abs/pii/S1537511017304762>
- Casas, R., Hermosa, A., Marco, Á., Blanco, T., & Zarazaga-Soria, F. J. (Received: 31 December 2020 / Revised: 22 January 2021 / Accepted: 25 January 2021 / Published: 29 January 2021). Real-Time Extensive Livestock Monitoring Using LPWAN Smart Wearable and Infrastructure.
- Gimenez, R. (2004). *Algoritmos de Compresión Sin Pérdida de*. Retrieved from
<http://jeuazarru.com/wp-content/uploads/2014/10/Compresion.pdf>
- Guerrero, G. P., & Escobar, H. S. (2007). *Compresion de imagenes usando wavelets*. Retrieved from Eafit:
https://repository.eafit.edu.co/bitstream/handle/10784/155/Gloria_PuetamanGuerrero_2007.pdf?sequence=1
- Lenn Carpentier, D. B., Tullo, E., Guarino, M., Vranken, E., & Norton, T. (2018). Automatic cough detection for bovine respiratory disease in a calf house.
- Martínez, J. (2020). *DataSmarts*. Retrieved from K-Nearest Neighbors para Clasificación de Imágenes: <https://datasmarts.net/es/k-nearest-neighbors-para-clasificacion-de-imagenes/>
- panelBrahimAchour, A. I., MalikaBelkadi, RachidaAoudjit, & MouradLaghrouche. (13 March 2019, Revised 18 July 2019, Accepted 20 October 2019, Available online 28 October 2019.). *Unsupervised automated monitoring of dairy cows' behavior based on Inertial Measurement Unit attached to their back*. Retrieved from
<https://www.sciencedirect.com/science/article/abs/pii/S0168169919304648>
- Schellberg, J., Hill, M. J., Gerhards, R., Rothmund, M., & Braun, M. (2007, Noviembre 20). *Precision agriculture on grassland: Applications, perspectives and constraints*. Retrieved from Science Direc:
<https://www.sciencedirect.com/science/article/abs/pii/S1161030108000646>
- Wikipedia. (2021, Febrero 2). *Transformada de coseno discreta*. Retrieved from Wikipedia:
https://es.wikipedia.org/wiki/Transformada_de_coseno_discreta#:~:text=La%20transformada%20de%20coseno%20discreta%20expresa%20una%20secuencia%20finita%20de,con%20distintas%20frecuencias%20y%20amplitudes
- Wikipedia. (2019, Septiembre 1). *Compresion Fractal*. Retrieved from Wikipedia:
https://es.wikipedia.org/wiki/Compresión_fractal

Wikipedia. (2020, Diciembre 26). *Run-length encoding*. Retrieved from Wikipedia:

[https://en.wikipedia.org/wiki/Run-length_encoding#:~:text=Run-length%20encoding%20\(RLE\),than%20as%20the%20original%20run](https://en.wikipedia.org/wiki/Run-length_encoding#:~:text=Run-length%20encoding%20(RLE),than%20as%20the%20original%20run).

Wikipedia. (2020, Noviembre 7). *Tallado de costuras - Seam carving* . Retrieved from Wikipedia:

https://es.qaz.wiki/wiki/Seam_carving

Wikipedia. (2021, Febrero 12). *Codificación Huffman*. Retrieved from Wikipedia:

https://es.wikipedia.org/wiki/Codificación_Huffman#:~:text=La%20codificación%20Huffman%20usa%20un,caracteres%20más%20comunes%20usando%20las

Wikipedia. (2021, Febrero 12). *Compresión de Burrows-Wheeler*. Retrieved from Wikipedia:

https://es.wikipedia.org/wiki/Compresión_de_Burrows-Wheeler