

ESTRUCTURAS DE DATOS PARA LA EFICIENCIA EN LA GANADERIA DE PRECISION

Manuel Arango Gomez
Universidad EAFIT
Colombia
Marangog3@eafit.edu.co

Camilo Monsalve Montes
Universidad EAFIT
Colombia
cmonsalmem@eafit.edu.co

Simón
Universidad
Colombia
smaring1@eafit.edu.co

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

RESUMEN

Actualmente existe una gran cantidad de circulación de información y de datos en la humanidad con los que cada vez se hace más compleja y difícil la manipulación de estos y afecta al cumplimiento de ciertas tareas de nuestra vida diaria por eso a medida del tiempo el desarrollamiento de la tecnología ha avanzado considerablemente ayudando así a la solución de problemas de una manera rápida y eficaz. Por eso al momento de hablar de rendimiento en el trabajo ganadero, reconocer rápidamente que animales están enfermos y que tipo de enfermedades tiene se ha vuelto algo muy importante para todas las personas ya que si no se trata estos animales de manera rápida podría ocasionar severos problemas para las empresas ganaderas.

Los algoritmos que se usaron para la realización de este trabajo consistieron en dos, el primero el cual es un algoritmo de comprensión con pérdida que es el tallado de costuras y el segundo que se uso fue un algoritmo de comprensión sin pérdida que fue el de Huffman.

Se concluye en este trabajo con la realización de un algoritmo de compresión capaz de ayudar a el desarrollamiento de la ganadería de precisión, ayudando en la eficacia y rapidez para identificar el estado de salud del ganado mejorando así la economía de la ganadería.

Palabras clave

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

1. INTRODUCCIÓN

La ganadería bovina es una de las actividades económicas de más importancia en la actualidad, en la que resulta altamente primordial aumentar su producción, para esto debemos de reconocer de forma eficaz los diferentes factores que podrían afectar a la vacada.

Para poder mejorar la producción de carne y leche, debemos de tener muy en cuenta la salud del animal ya que las enfermedades en los animales generan daños al comercio internacional, controlar

la enfermedad de las vacas permite que la economía crezca y ayuda a la salud de la población más vulnerable.

Dado lo anterior nuestra motivación es aumentar la producción y ahorro de energía con el uso de datos en la ganadería de precisión, para minimizar el uso de equipo de granja y tener mejor compresión de datos.

1.1. Problema

El objetivo a desarrollar es crear un algoritmo con el cual se haga un eficiente guardado y búsqueda de archivos para aquellas empresas o personas que manejan una cantidad masiva de datos para su rendimiento y optimización de energía, logrando así una mejor tasa de compresión posible para el mejor consumo de tiempo y de memoria.

1.2. Solución

Para el problema de la ganadería bovina, se encontró una solución a este problema, lo que se hará es que se le tomara una foto a cada vaca, y luego esta foto será procesada por un sistema, el cual usara una compresión

de imágenes principalmente con pérdida, y se usara la compresión con pérdida de tallas de costuras ya que este es un algoritmo para el cambio del tamaño de las imágenes conscientes del contenido, este funciona estableciendo una serie de uniones en una imagen y elimina automáticamente las uniones para reducir el tamaño de la imagen o inserta uniones para extenderlo, también permite definir manualmente áreas en las que no se pueden modificar los píxeles.

El propósito del algoritmo es la reorientación de imágenes, que es el problema de mostrar imágenes sin distorsión en medios de varios tamaños. Calcular una costura consiste en encontrar un camino de mínimo costo energético de un extremo de la imagen a otro. Esto se puede hacer a través del algoritmo de Dijkstra, la programación dinámica, el algoritmo codicioso o cortes de gráficos, entre otros

2. TRABAJOS RELACIONADOS

2.1 Revisión sistemática de la literatura sobre el uso del aprendizaje automático en precisión ganadería

Este artículo presenta una revisión bibliográfica sistemática de trabajos recientes sobre el uso del aprendizaje automático en la ganadería de precisión, centrándose en dos áreas de interés: pastoreo y salud animal. Esta revisión:

1. destaca las oportunidades para el LA en el sector ganadero
2. Muestra los sensores, software y técnicas actuales para el análisis de datos
3. Detalla la creciente apertura de las fuentes de datos.

Se encontró que el uso de ML en la ganadería de precisión se encuentra en una etapa de desarrollo y tiene varios desafíos de investigación. Ejemplos de tales desafíos son:

1. desarrollar modelos híbridos de diagnóstico y prescripción como herramienta para la prevención y control de enfermedades animales
2. Aumentar los temas de pastoreo y salud animal.
3. Dar autonomía a la ganadería de precisión mediante ciclos autónomos de tareas de análisis de datos y meta-aprendizaje

4. Reunir las variables de suelo y pasto porque, tanto para la salud animal como para el pastoreo, las variables utilizadas son solo conductuales y ambientales.

Diosdado y col. presentaron una combinación de vecinos más cercanos k fuera de línea y un algoritmo de aprendizaje en línea, que los autores consideran como un mecanismo útil para el seguimiento a largo plazo en el sistema del campo. El algoritmo propuesto clasifica el comportamiento relevante de las ovejas, utilizando información de un dispositivo de borde integrado que incluye un acelerómetro triaxial y sensores de giroscopio triaxial.

Otra alternativa al algoritmo para la clasificación del comportamiento es un ML jerárquico, se propuso el uso de un método ML jerárquico para clasificar el comportamiento del ganado. En primer lugar, clasificaron el comportamiento del ganado en dos categorías de comportamiento, el activo o el inactivo.

Rodrigo García, José Aguilar, Mauricio Toro, Ángel Pinto, Paul Rodríguez. Computers and Electronics in Agriculture. Computers and Electronics in Agriculture volumen 179, diciembre 2020, 105826.

2.2 Una plataforma de bienestar animal para la ganadería extensiva Sistemas de producción

El presente estudio presenta una solución para el seguimiento y la monitorización de la actividad y el comportamiento de los animales en las explotaciones ganaderas, obteniendo indicadores que sustenten el bienestar de los animales. La solución aprovecha:

1. Un único tipo de sensor inalámbrico (dispositivo de collar) para registrar la actividad de los animales con una baja implementación.
2. Dispositivos de computación de borde con capacidades computacionales, capaces de realizar el procesamiento de datos fuera de línea y en tiempo real para el reconocimiento de patrones a través de Neuronas Profundas.
3. Computación en la nube tanto para el almacenamiento de datos como para el almacenamiento de modelos de Deep Learning.
4. Visualizaciones utilizables y eficaces en dispositivos móviles que proporcionen a los

usuarios finales (agricultores) información valiosa.

La arquitectura de la nube está compuesta por Java Spring Boot con Hibernate Object-Relational Mapping y una base de datos PostgreSQL para almacenar los datos. Java Spring Boot la cual es una extensión de Spring Framework que proporciona una forma sencilla de crear aplicaciones independientes basadas en Spring de nivel de producción, que se utiliza para crear el servicio REST que conecta el dispositivo Edge y la aplicación móvil con la nube.

2.3 Visual Localización and Individual Identificación of Holstein Friesian Cattle via Deep Learning Demostraron que la detección automática y la identificación individual es posible via Deep Learning.

Demostraron que la detección automática y la identificación individual es posible via Deep Learning. Esta identificación se da por medio del reconocimiento de su estructura dorsal únicos que se da por la especie misma, como sus colores o sus diferentes patrones de manchas, parches y marcas. Para lograr esto usaron un dron que filma a 5 metro sobre el suelo, este reconoce los patrones anteriormente mencionados y se usa como entidad biométrica característica de este trabajo por lo que hicieron 3 diferentes tareas durante el proceso:

- A. Detección de ganado, especie y localización
- B. Identificación de un solo cuadro
- C. Identificación basada en video

Implementaron para su desarrollo el uso de una API de Python, un software de Foster R-CNN que se basa en la biblioteca de aprendizaje profundo de Café.

3. MATERIALES Y METODOS

3.2 Alternativas de compresión de imágenes con perdida

3.2.1 El tallado de costuras

Es un algoritmo para el cambio del tamaño de las imágenes conscientes del contenido, este funciona estableciendo una serie de uniones en una imagen y elimina automáticamente las uniones para reducir el

tamaño de la imagen o inserta uniones para extenderlo, también permite definir manualmente áreas en las que no se pueden modificar los píxeles.

El propósito del algoritmo es la reorientación de imágenes, que es el problema de mostrar imágenes sin distorsión en medios de varios tamaños.

Las costuras pueden ser verticales u horizontales. Una costura vertical es una ruta de píxeles conectados de arriba a abajo en una imagen con un píxel en cada fila. Una costura horizontal es similar con la excepción de que la conexión es de izquierda a derecha.

El proceso es:

1. Comenzar con una imagen.
2. Calcular el peso, densidad, energía de cada pixel, se usa magnitud de gradientes.
3. A partir de la energía, hacer una lista de costuras, las costuras se clasifican por energía.
4. Quitar las costuras de baja energía según sea necesario.
5. Imagen final.



3.2.2 Compresión fractal

Método de compresión con pérdida para imágenes digitales que se basa como su nombre lo dice en fractales. Este se usa en su mayoría para imágenes naturales, se basa en que ciertas partes de una imagen, a veces, se parecen a otras de la misma.

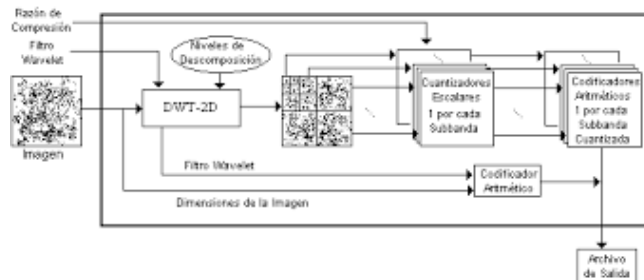
Estos algoritmos convierten estas partes mencionadas anteriormente en datos matemáticos para que se conviertan en código para poder así recrear la imagen codificada.

Este dentro de su código presenta ciertas características como lo es que a la hora de su codificación este sea cara a nivel computacional pero su decodificación es bastante rápida.



3.2.3 Compresión de Wavelet

Desarrollado por la empresa Leica Geosystems y funciona mediante técnicas de ondículas. Este reduce considerablemente el tamaño de los archivos, pero sin reducir su calidad gráfica y permitiendo que esta compresión sea lo suficientemente rápida y que su descompresión no genere tanto gasto de la memoria RAM.



3.2.4 Escala de imagen

El escalado de imágenes se refiere al cambio de tamaño de una imagen digital. En la tecnología de video, la ampliación de material digital se conoce como mejora de la resolución o escalamiento.

Al escalar una imagen de gráfico que sea vectorial, las primitivas gráficas que componen la imagen se pueden escalar utilizando transformaciones geométricas, sin

tener una pérdida de calidad de imagen. Al escalar una imagen de gráficos de mapa de bits, se debe generar una nueva imagen con un número mayor o menor de píxeles. En el caso de disminuir el número de píxeles, esto generalmente da como resultado una pérdida de calidad visible. Desde el punto de vista del procesamiento de señales digitales, el escalado de gráficos de trama es un ejemplo bidimensional de conversión de frecuencia de muestreo, la conversión de una señal discreta de una frecuencia de muestreo a otra



3.3 Alternativas de compresión de imágenes sin pérdida.

3.3.1 La transformación de Burros-Wheeler

también conocida como compresión por clasificación de bloques, esta reordena una cadena de caracteres en varias series de caracteres similares a este, esto es algo que nos ayuda al momento de compresión, ya que es más fácil comprimir una cadena que tiene ejecuciones de caracteres repetidos con la técnica como la transformación del movimiento al frente y la codificación del tamaño de la ejecución. Y la transformación es reversible sin almacenar ningún dato adicional solo con la posición del primer carácter.

El BWT es un método "gratuito" para mejorar la eficiencia de los algoritmos de compresión de texto, que cuesta sólo algunos cálculos adicionales. Un dato importante es que cuando el BWT transforma una cadena de caracteres, la transformación permuta el orden de los caracteres.

Se utiliza para la preparación de datos para su uso con técnicas de compresión de datos como bzip2, el cual se basa en una transformación inédita. El algoritmo se puede implementar de manera eficiente haciendo uso de una matriz de sufijos la cual es una matriz ordenada de todos los sufijos de

una cadena, alcanzando así una complejidad de tiempo lineal.

I		i	S(i)	L(i)
0	T A G A C A G A \$	0	1 \$ T A G A C A G	A
1	\$ T A G A C A G A	1	2 A \$ T A G A C A	G
2	A \$ T A G A C A G	2	6 A C A G A \$ T A	G
3	G A \$ T A G A C A	3	4 A G A \$ T A G A	C
4	A G A \$ T A G A C	4	8 A G A C A G A \$	T
5	C A G A \$ T A G A	5	5 C A G A \$ T A G	A
6	A C A G A \$ T A G	6	3 G A \$ T A G A C	A
7	G A C A G A \$ T A	7	7 G A C A G A \$ T	A
8	A G A C A G A \$ T	8	0 T A G A C A G A	\$

a)

b)

3.3.2 Codificación Huffman

Algoritmo desarrollado por David A. Huffman, es un código de prefijo que se usa para la compresión de datos sin pérdidas. Este se puede ver como una tabla de códigos de longitud variable para codificar un símbolo fuente (como un carácter en un archivo). El algoritmo deriva esta tabla de la probabilidad o frecuencia estimada de ocurrencia para cada valor posible del símbolo fuente. El método de Huffman se puede implementar de manera eficiente, encontrando un código en el tiempo lineal al número de ponderaciones de entrada si estas ponderaciones se ordenan.

El codificador Huffman crea una estructura arbórea ordenada con todos los símbolos y la frecuencia con que aparecen. Las ramas se construyen en forma recursiva comenzando con los símbolos menos frecuentes.

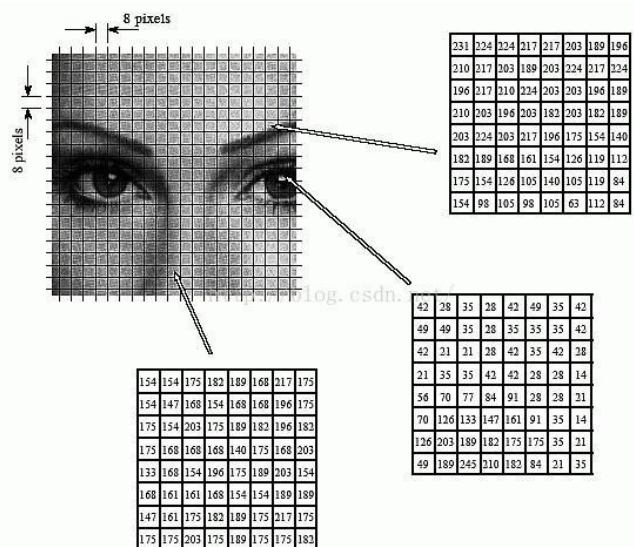
Este árbol se construye ordenando los símbolos según su frecuencia de aparición. Los dos símbolos que tengan menos frecuencia de aparición se eliminan sucesivamente y se conectan a un nodo cuyo peso es igual a la suma de la frecuencia de los dos símbolos. El símbolo con menor peso es asignado a la rama 1, el otro a la rama 0 y así sucesivamente, considerando cada nodo formado como un símbolo nuevo, hasta que se obtiene un nodo principal llamado raíz. El código de cada símbolo corresponde a la sucesión de códigos en el camino, comenzando desde este carácter hasta la raíz. De esta manera, cuanto más dentro del árbol esté el símbolo, más largo será el código.



3.3.3 Compresión de imagen basada en diccionario bidimensional real en escala de grises

Método que se usa inicialmente para reducir la correlación unidimensional de los datos, ya que estos están diseñados para comprimir texto por lo que no aprovechan el hecho de que los píxeles adyacentes están correlacionados en dos dimensiones. Existen dos métodos de compresión uno con pérdida que conserva un aproximado de la imagen y uno sin que conserva todos los datos de la imagen esto conlleva a que ciertos archivos sean de mayor tamaño que otros dependiendo de su formato.

Por eso se propone un esquema compresión de imágenes sin pérdida, pero basado en un diccionario bidimensional, este esquema al hacerse pruebas demuestra que supera el rendimiento de compresión de un JPEG

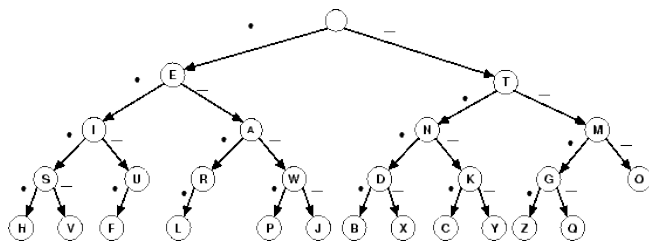


3.3.4 Compresión LZS

Es una compresión de datos sin pérdidas algoritmo que utiliza una combinación de la LZ77 algoritmo de compresión de ventana deslizante y se fija la codificación de Huffman. Esta luego de varios cambios se adaptó se adaptó para la compresión de disco duro.

Este funciona de la siguiente forma:

- 1) Utiliza los últimos 2KB de datos sin comprimir como un diccionario de venta deslizante
- 2) Luego un compresor busca coincidencias entre los
- 3) Si se encuentra una coincidencia codifica una longitud de diccionario



4. DISEÑO E IMPLEMENTACIÓN DE LOS ALGORITMOS

4.1 Estructuras de datos

La estructura de datos a utilizar en este proyecto será los árboles ya que nos ayudan a sintetizar más fácil el uso de nuestro sistema, los arboles en la estructura de datos son un método de búsqueda simple, dinámico y eficiente considerado como uno de los fundamentales en la ciencia de la computación. Es una estructura de datos en la cual se tiene nodos que a su vez pueden tener un hijo izquierdo y uno derecho, esto quiere decir que no pueden tener más de dos hijos.

Este es similar a una lista enlazada, pero puede tener varios siguientes que son sus diferentes ramas, al tener diferentes ramas quiere decir que representa una estructura jerárquica.

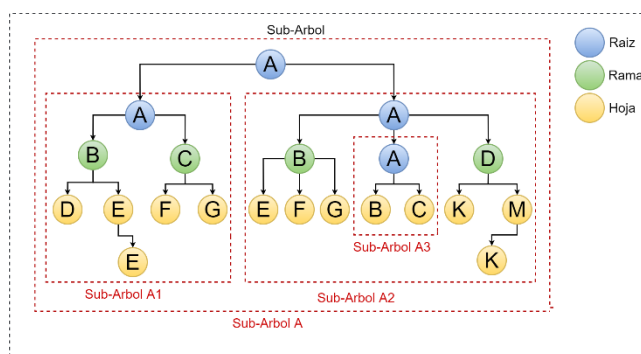
Este tiene una estructura propia entre las cuales son:

Nodos: Se le llama Nodo a cada elemento que contiene un Árbol.

Nodo Raíz: Se refiere al primer nodo de un Árbol, Solo un nodo del Árbol puede ser la Raíz.

Hoja: Son todos aquellos nodos que no tienen hijos, los cuales siempre se encuentran en los extremos de la estructura.

y este a su vez puede tener subárboles como lo son el subárbol derecho y el izquierdo.



4.2 Algoritmos

En este trabajo, se hará uso de un algoritmo de compresión con pérdida el cual es escalamiento de imagen mejor conocido como image scaling, a continuación, se explicará cómo funciona este algoritmo.

4.2.1 tallado de costuras (comprensión con pérdida)

Para el uso del algoritmo tallado de costuras se debe seguir un proceso:

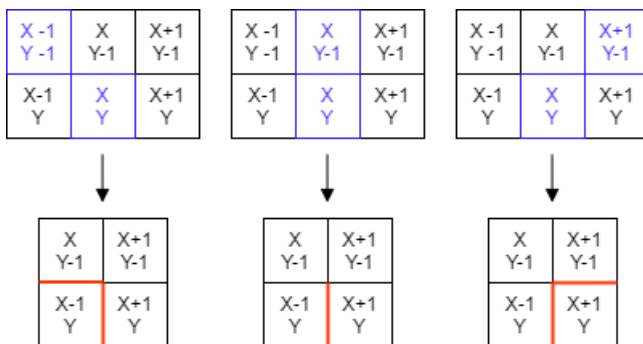
1. Se comienza con la imagen
2. Se calcula el peso, la densidad, energía de cada pixel (colores). Esto se hace mediante varios algoritmos: magnitud de gradiente, entropía, prominencia visual, movimiento de la mirada.
3. A partir de la energía, se hace una lista de costuras. Las costuras se clasifican por energía, siendo las costuras de baja energía las de menor importancia para el contenido de la imagen. Las costuras se pueden calcular

mediante el método de programación dinámica.

4. Se quitan las costuras de baja energía.

5. Se llega a la imagen final.

En la computación de costuras consiste en encontrar un camino de mínimo costo energético de un extremo de la imagen a otro. Esto se puede hacer a través del algoritmo de Dijkstra, la programación dinámica, el algoritmo codicioso o cortes de gráficos, entre otros. La programación dinámica es un método de programación que almacena los resultados de los subcálculos para simplificar el cálculo de un resultado más complejo, esta se puede utilizar para calcular uniones. Si se intenta calcular una costura vertical (trayectoria) de menor energía, para cada píxel de una fila calculamos la energía del píxel actual más la energía de uno de los tres posibles píxeles por encima de él.



El algoritmo funciona de la siguiente manera:

1. Asignar un valor de energía a cada píxel
2. Encuentre una ruta de 8 píxeles conectados con la menor energía
3. Eliminar todos los píxeles en la ruta
4. Repita 1-3 hasta que se elimine el número deseado de filas / columnas



4.3. Análisis de la complejidad de los algoritmos

En los algoritmos usados en este proyecto, las complejidades son: para el algoritmo de tallado de costuras se usó una complejidad de $O(w)$, pero en el peor de los casos sería una complejidad de $O(W \times H)$ si la imagen es W píxeles de ancho y H píxeles de alto, la complejidad del tiempo, para el algoritmo de Huffman Su complejidad es del orden de $O(nL)$, siendo L el tamaño de la fila de la matriz.

Algoritmo	La complejidad del tiempo
Compresión con pérdida	$O(W \times H)$
Compression sin pérdida	$O(nL)$

Algoritmo	La complejidad del memoria
Compresión con pérdida	$O(W)$
Compression sin pérdida	$O(nL)$

4.4 Criterios de diseño del algoritmo

Para la solución del algoritmo con pérdida se intento usar el algoritmo del vecino mas cercano, luego después de pensarlo bien nos arriesgamos por algo mejor que seria el tallado de costuras, al ver este algoritmo nos intereso mucho su funcionamiento y como lo íbamos a realizar, nos adentramos al código, y pudimos mostrar las imágenes en jpg luego de mucho tiempo, buscamos ejemplos en internet pero no logramos encontrar buenas referencias, a lo cual acudimos a uno de nustrs amigos que van en semestres superiores y nos ayudaron a entender el funcionamiento, luego plasmamos nuestra idea en el código, algo que no tiene bueno nuestro código es que puede demorarse un poco alrededor de 2 a 4 minutos, lo cual es notable. Por otro lado con el algoritmo de compresión sin pérdida usamos el de huffman, al principio pensamos en hacer el algoritmo lzs el cual combinaba el algoritmo de lz77 y el algoritmo de huffman pero al ver esto, mejor escogimos el algoritmo de huffman, empezamos buscando como funcionaba

este algoritmo y luego como lo podríamos juntar con lo que llevavamos anteriormente, luego de pensar mucho logramos realizar el algoritmo el cual también nos devolvería la imagen en jpg.

5. DISCUSIÓN DE LOS RESULTADOS

Los resultados obtenidos al momento de probar el código que se creo es muy positivo ya que para ser algo tan pesado de hacer se soluciona rápidamente, lo único que si tiene problemas es que dependiendo de el tamaño de la imagen, la velocidad del internet y las costuras de alta energía, el proceso podría tomar un poco mas de tiempo alrededor de 1 minuti mas de tiempo como máximo, cuanto al consumo de memoria, es positivo ya que al comprimir la imagen se logra mermar el ppeso de esta siendo asi un ejemplo, se pasa de tenr una imagen de 60 kb a una de 40kb hasta 20 kb, para los algoritmos que usamos tenemos una buena velocidad y buena memoria, ya que no toma tanto tiempo y la imagen se comprime muy bien, casi siendo perfecto, logrando asi mermar el espacio y logrando una mejor visualización de la imagen, logrando que sea mas fácil para las empresas de ganadería bovina.

6.1 Trabajos futuros

Para el futuro queremos lograr que nuestro algoritmo tenga un mejor tiempo de solución, queremos empezar estas vacaciones mermando el tiempo de 1 minuto en un 30% y ya para el siguiente año poder mermar este tiempo un 50%, queremos aprender mas acerca de la comprensión con ondelas y la transformación de coseno discreto para ver que tanto nos podría ayudar con nuestros algoritmos, mermando nuestro tiempo y

REFERENCIA

La referencia se obtiene usando el formato de referencia ACM. Lea las directrices de ACM en <http://bit.ly/2pZnE5g>

1. Anonimo. Burrows–Wheeler transform, from Wikipedia, the free encyclopedia last edited on 2 july 2021.
https://en.wikipedia.org/wiki/Burrows%E2%80%93Wheeler_transform

2. Anonimo. LZ77 and LZ78 From Wikipedia, the free encyclopedia. Last edited on 1 august 2021.
https://en.wikipedia.org/wiki/LZ77_and_LZ78

3. Anonimo. Huffman coding from Wikipedia the free encyclopedia. Last edite on 10 july 2021.
https://en.wikipedia.org/wiki/Huffman_coding

4. Anonimo. Lempel-Ziv-Stac. From Wikipedia the free encyclopedia. Last edite on 14 august 2021.
<https://en.wikipedia.org/wiki/Lempel%E2%80%93Ziv%E2%80%93Stac>

5. Nathanael J. Brittain, Mahmoud R. El-Sakka. Grayscale true two-dimensional dictionary-based image compression. February 2007.
<https://www.sciencedirect.com/science/article/abs/pii/S104732030600054X>

6. Anonimo, Seam carving. From Wikipedia the free encyclopedia. Last edited on 5 july 2021.
https://en.wikipedia.org/wiki/Seam_carving

7. Anonimo. Image scaling, from Wikipedia the free encyclopedia. Last edite on 9 july.
https://en.wikipedia.org/wiki/Image_scaling.

8. Rodrigo García, Jose Aguilar , Mauricio Toro , Angel Pinto , Paul Rodríguez. A systematic literature review on the use of machine learning in precision livestock farming. Contents lists available at ScienceDirect

9. Vasileios Doulgerakis¹, Dimitrios Kalyvas¹, Enkeleda Bocaj¹, Christos Giannousis¹, Michalis Feidakis¹, George P. Laliotis², Charalampos Patrikakis¹, Iosif Bizelis². An Animal Welfare Platform for Extensive Livestock Production Systems

10. Olivier Debauche, Saïd Mahmoudi¹ Andriamasinoro Lalaina Herinaina Andriamandroso, Pierre Manneback, Jérôme Bindelle, Frédéric Lebeau. Cloud services integration for farm animals' behavior studies based on smartphones as activity sensors. Received: 15 February 2018 / Accepted: 15 April 2018 / Published online: 14 May 2018.

11. Andrew, W., Greatwood, C., & Burghardt, T. (2018). Visual Localisation and Individual Identification of Holstein Friesian Cattle via Deep Learning. In 2017. Link to published version (if available): 10.1109/ICCVW.2017.33

12. Anonimo, Nearest-neighbor interpolation. From Wikipedia the free encyclopedia. Last edited on 1 July 2021. https://en.wikipedia.org/wiki/Nearest-neighbor_interpolation.

13. Oscar Blancarte, Estructura de datos – Árboles. from <https://www.oscarblancarteblog.com/>. Last edited 22 of August 2014. <https://www.oscarblancarteblog.com/2014/08/22/estructura-de-datos-arboles/>

14. Karthik Karanth, Implementación de tallado de costuras con Python, published on May 28, 2018. <https://karthikkaranth.me/blog/implementing-seamcarving-with-python/>