

# OPTIMIZACIÓN DE ENERGÍA EN GANADERÍA DE PRECISIÓN CON ALGORITMOS DE COMPRESIÓN

Samuel Salazar  
Universidad Eafit  
Colombia  
Ssalazar1@eafit.edu.co

Carla Sofia Rendon  
Universidad Eafit  
Colombia  
csredonb@eafit.edu.co

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

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

**Texto rojo = Comentarios**

**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

La ganadería de precisión ha llegado para poder mejorar de forma significativa el manejo del ganado pero con esto ha llegado ciertos retos a superar, uno de estos retos es la forma en la cual se puede optimizar la obtención de datos, específicamente cómo podemos automatizar o digitalizar estos datos, también otro desafío es cómo se puede ahorrar energía al momento de tomar la información y más. Los algoritmos que vamos a proponer para este proyecto son dos, interpolación lineal y el famoso LZ77. Hablando específicamente de cada uno el LZ77 es un algoritmo con pérdida, que reducirá la calidad de la imagen, pero nos dará una reducción en general del archivo y el LZ77 que también nos dará un archivo reducido pero la diferencia es que este no tiene pérdida, por lo tanto la imagen se mantendrá igual, por ello al final de nuestro proyecto tendremos dos algoritmos eficientes para la reducción del tamaño de archivos .csv y tendremos varios datos diciéndonos que tan eficiente son y varias comparaciones a peso con tiempo.

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

## 1. INTRODUCCIÓN

La ganadería de precisión es un concepto con unos resultados que se esperan ser muy buenos por la organización que este trae a la ganadería y más que eso, a la sostenibilidad que ofrece. Específicamente la motivación que hay ahora mismo es la toma de información de la salud del ganado a partir de la toma y compresión de las imágenes de estos animales para el contexto de la ganadería.

### 1.1. Problema

El problema que tenemos en la ganadería de precisión es que necesitamos encontrar formas de ahorrar energía y de poder minimizar los recursos usados (al fin y al cabo ese es el mismo propósito de la ganadería de precisión), en este caso

uno de los problemas más importantes que tenemos ahora mismo es en las imágenes, específicamente las compresiones de esta. Es importante que podamos comprimir estas imágenes para que no usen tanto espacio y podamos seguir usando ese espacio ahorrado para más cosas como datos o hasta otras imágenes ya que con estas vamos a clasificar 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 compresión de los datos.

La solución que estamos implementando para este problema es el uso de un algoritmo de compresión de imágenes para reducir el tamaño de estas y así liberar más espacio, el algoritmo que estamos usando para esta solución se llama interpolación lineal, el cual reduce el tamaño de la imagen haciendo un promedio ciertos puntos de la imagen, reduciendo su tamaño.

### 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 La falta de control en los cultivos debido a factores externos.

La necesidad de dar un uso adecuado a los suelos y producir alimentos de calidad de forma sostenible, usualmente se caracteriza por tener muchas dificultades, debido a que los cultivos en los campos, su producción no es homogénea y cada zona de los campos varía, por ello gracias a la agricultura de precisión (PA) nos permite el desarrollo un sistema de captura y procesamiento de imágenes de cultivos

agrícolas, para poder entender las variaciones de los campos y así mejorar la producción. El diseño de la solución incluye el desarrollo de software y hardware para la adquisición de imágenes utilizando un modelo de helicóptero equipado con cámaras de video con el propósito del procesamiento de imágenes.

El Indexes Platform realiza el procesamiento de datos y entrega de resultados para diagnosticar el estado del cultivo. En este software, las imágenes capturadas por la cámara de espectro visible e infrarrojo se cargan junto con el Archivo Log.txt que es entregado por la estación de telemetría, y el archivo histórico gpsbase.txt. La interfaz realiza tres funciones básicas: procesa imágenes, realiza la corrección diferencial DGPS y muestra datos y parámetros finales. Este software desarrolla un algoritmo de procesamiento de imágenes que permite la caracterización y diagnóstico de un producto unidad agrícola.

## **2.2 Limitaciones en la transmisión y almacenamiento de imágenes médicas con el formato DICOM**

Un archivo DICOM contiene por un lado la información de una o varias imágenes y por otro la información del contexto en el que se ha tomado la imagen, estos archivos suelen caracterizarse por el gran espacio de almacenamiento que ocupan ya que contienen una gran cantidad de datos y de allí surge la necesidad de crear un sistema que permita la fácil transmisión de este tipo de archivos sin afectar la calidad de la imagen y así no alterar el diagnóstico médico, por ello como solución se desarrolló un programa con el que se pudiera comprimir y descomprimir las imágenes médicas. Se desarrolló un algoritmo tanto de compresión como descompresión de imágenes mediante familias wavelets basadas en Funciones Atómicas. El algoritmo de compresión de imágenes está compuesto por tres etapas que son en la transformación, cuantización y codificación, en el caso del algoritmo de descompresión se realiza el proceso contrario a la compresión de imágenes.

## **2.3 El gas metano en la producción ganadera y alternativas para medir sus emisiones y aminorar su impacto a nivel ambiental y productivo**

Los rumiantes son uno de los ganados más importantes en la ganadería y a su vez es uno de los actores más influyentes en la contaminación de la capa de ozono ya que actualmente estos animales contribuyen al 14% a la contaminación por dióxido de carbono.

Se han desarrollado con el tiempo varias formas de controlar estos gases lanzados por estos animales, una forma ha sido controlar la dieta de estos animales. Ellos comen primordialmente del pasto pero si se hace una intervención en su dieta, se puede disminuir los niveles de metano producidos por ellos, controlando el medio ambiente y mejorando en general la producción. Otra forma de controlar a estos animales es con un sistema silvopastoril la cual trata en controlar el ambiente del animal para controlar lo que esta haga. Por último, una alternativa que hay es que si se

encuentra una forma de usar este gas metano como recurso, se podría usar como una fuente de energía dando como resultado ahorro en ciertos sistemas.

## **2.4 Compresión de imágenes digitales utilizando redes neuronales de base radial**

Cada vez crece más la necesidad de transmisión y de almacenamiento de imágenes digitales y esto también conlleva a un gasto muy alto por el uso de memoria y tiempo y es por eso que la compresión de imágenes es una nueva solución a este problema ya que es una forma económica de ahorrar más espacio y de volver mas rapido la transmisión de estas fotos. La técnica que se presenta para poder hacer esta compresión de las imágenes es a través de una inteligencia artificial diseñada como redes neuronales de base radial para mantener la calidad de esta. Para lograr ello se hizo un estudio sobre las fotos, los colores y también una investigación sobre cómo crear redes neuronales artificiales. Con el experimento hecho junto con la comparación de una imagen tipo JPEG, se concluyó que la técnica planteada reduce la imagen en un 50% manteniendo la calidad de esta.

## **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>.

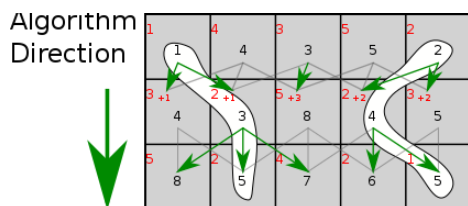
### 3.2 Alternativas de compresión de imágenes con pérdida

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

#### 3.2.1 Tallado de costuras

El tallado de costuras o también llamado reescalado líquido es un algoritmo hecho para el cambio de tamaño de imágenes. El algoritmo lo que hace es establecer una serie de trayectorias de menor importancia en la imagen y elimina estas trayectorias reduciendo el tamaño de la imagen. Estas trayectoria también permite poder definir pixeles para no ser modificados o eliminar objetos completos de una imagen.

El problema de este algoritmo es que al cambiar la imagen, hace que ciertas partes se recorten o se modifique, distorsionando la imagen.

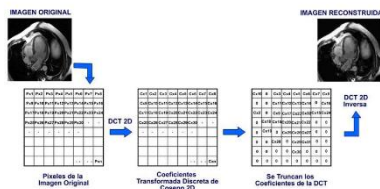


#### 3.2.2 Transformada discreta del coseno

La transformada discreta del coseno es un algoritmo que se usa muchas veces para compresión de imágenes JPEG. Esta está basada en la transformada discreta de Fourier.

Lo que hace este algoritmo es que descompone la imagen en sumas de cosenos ya que no se necesita tantas funciones de coseno para expresar las condiciones de la imagen.

Este algoritmo está expresado en varias fórmulas matemáticas lo cual hace que no sea tan complicado de usar.



#### 3.2.3 Transformada de ondeleta

La transformada de ondeleta es una función matemática que permite seleccionar o separar datos en pequeños

componentes para así poder ser vistos uno por uno. La ondeleta se realiza a través de una fórmula matemática, como dicho antes esta permite la división específica de un conjunto de datos y a diferencia con la transformada de Fourier, está puede encontrar discontinuidades, siendo más precisa que la transformada mencionada antes.

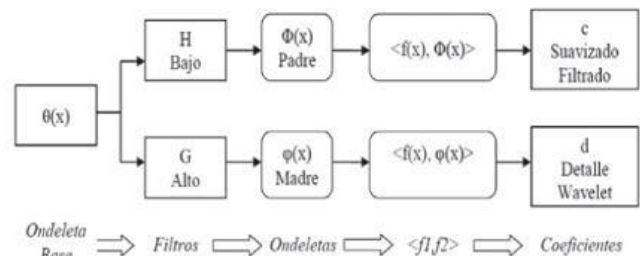


Figura 1. Esquema transformación directa con ondeletas

Fuente: Beltrán y Colmenares (2011).

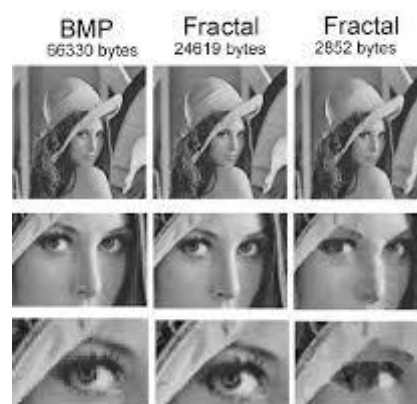
#### 3.2.4 Compresión fractal

La compresión fractal es una forma de compresión fractal basada en fractales, lo que estos hacen es coger las partes de la imagen que se parecen a otras partes de la misma imagen y convertirlas en datos matemáticos llamados códigos fractales para recrear la imagen codificada.

Este código es muy caro y toma tiempo cuando es una imagen muy asimétrica pero cuando son por ejemplo paisajes o parecidos es en realidad un método fácil y rápido para la compresión.

#### Fractal Image Compression

Lenna 256 colors (gray scale)



### 3.3 Alternativas de compresión de imágenes sin pérdida

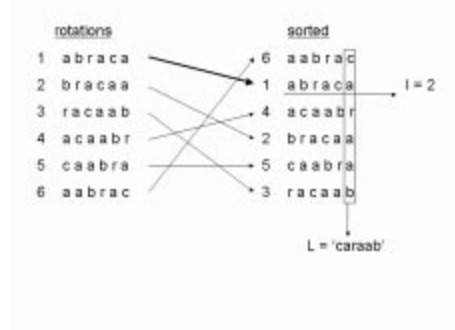
En lo que sigue, presentamos diferentes algoritmos usados para comprimir imágenes sin pérdida.).

#### 3.3.1 Compresión de Burrows-Wheeler

Compresión de Burrows-Wheeler o transformada de Burrows-Wheeler es un algoritmo popular usado en técnicas de compresión de datos.

Este algoritmo lo que hace es reordenar una cadena de caracteres a una cadena mucho más simple. Este algoritmo es muy usado en la compresión ya que comprime de forma muy sencilla los caracteres repetidos y también no requiere de ningún tipo de datos adicional para revertirlo, es por esto que el algoritmo es muy sencillo y de poca dificultad.

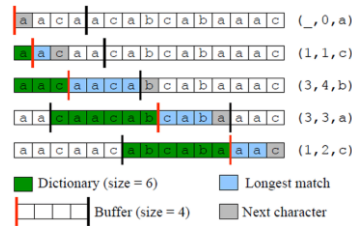
Lo que hace este algoritmo es transformar una cadena de caracteres permutando su orden, primero lo que hace es clasificar todos los desplazamientos del texto en orden lexicográfico sacando la última columna y el índice de la cadena original en el conjunto de permutaciones hechas.



### 3.3.2 LZ77 y LZ78

LZ77 y LZ78 son dos algoritmos de compresión de datos publicados por Abraham Lempel y Jacob Ziv en 1977 y 1978. Estos algoritmos son considerados muy importantes ya que son la base de muchas variaciones como LZW, LZSS y más. Además de esto formaron la base de varios archivos importantes como GIF y el algoritmo DEFLATE para PNG y ZIP.

LZ77 lo que hace es iterar secuencialmente a partir de el input de un string y guarda algún match en el buffer del buscador, después a partir de otros parámetros este empieza a codificar moviéndose entre la secuencia dada. La gran diferencia entre LZ77 y LZ78 es que el último dicho no usa una "Sliding Window" como LZ77, sino que LZ78 ya tiene esa equivalencia construida en sí mismo y por eso estos dos algoritmos son altamente complejos.



### 3.3.3 LZS

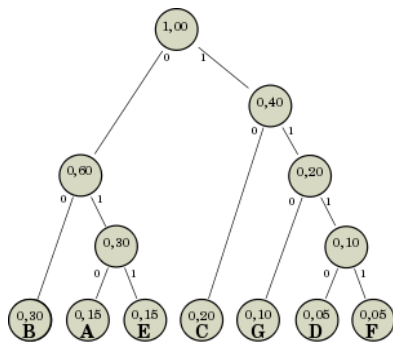
El LZS es un algoritmo de compresión de datos que usa el "sliding window" de LZ77 con codificación de Huffman.

Este compressor busca coincidencias entre la data que se imprimirá y los últimos 2kb de la "Sliding window", si se encuentra una coincidencia, crea una referencia offset para el diccionario pero si no coincide, marca el siguiente byte como un "byte" literal. Es un algoritmo que no es tan complicado como lo es el LZ77 y se podría decir que de una forma lo simplifica.

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

### 3.3.4 Codificación Huffman

La codificación Huffman es un algoritmo de compresión de datos creado en 1951 por David Huffman, la forma en al que funciona este algoritmo es asignando el número de repeticiones a cada valor de la frase a codificar, después se crea un árbol binario con estos datos y se le pone un número específico a cada valor y con esto queda terminada la codificación. Este algoritmo no es en realidad tan complejo y no requiere de mucho trabajo.



## 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<sup>1</sup>.

### 4.1 Estructuras de datos

La estructura de datos que usaremos para el algoritmo es de vectores dinámicos. Los vectores dinámicos son una estructura de datos que se puede encontrar en la gran mayoría de los lenguajes modernos de programación. Esta estructura es caracterizada por ser sencilla y bastante básica pero a la vez capaz de almacenar mucha información, esta tiene muchas similitudes con un arreglo ya que son prácticamente la misma estructura pero el vector dinámico tiene una diferencia clave que la hace mucho más útil que los arreglos en ciertos casos y es que los vectores dinámicos pueden expandirse de forma dinámica en el momento de ejecución mientras que un arreglo no puede. Los vectores dinámicos tienen una complejidad de  $O(1)$  para el acceso del vector, mientras que para búsqueda, inserción y borrado tiene una complejidad de  $O(n)$ . y esta complejidad es tanto para el caso promedio como el peor caso.

**Vector dinámico antes de extenderse:**



**Vector dinámico después de extenderse:**



**Gráfica 1: Demostración como un ArrayList crece.**

### 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.

El algoritmo que usaremos aplicará interpolación lineal, básicamente lo que pasara es que formaremos grupos de cuatro con los datos del archivo csv, con estos datos lo que haremos es sumarlos y después dividirlos entre 4, es decir, sacarles el promedio a cada pareja. Este proceso se repetirá por todo el archivo, así que básicamente este algoritmo lo que hará es recortar por la mitad la imagen.

**4 grupos de 4 valores  
antes del algoritmo**

90	95	106	109
96	89	85	98
74	72	56	84
101	82	87	111

**Valores simplificados  
Después del algoritmo**

93	100
82	85

**Gráfica 2: Demostración algoritmo en los archivos csv**

### 4.3 Análisis de la complejidad de los algoritmos

Obtener la complejidad en el peor caso de los dos algoritmos no fue lo mas complicado de sacar, al observar los dos algoritmos y sacar la complejidad a cada línea que esta tiene, vemos que en algún punto del código hay ciclos anidados, por lo tanto esto nos dice que los dos algoritmos en su peor caso tendrá que recorrer  $n$  veces el primer ciclo y  $n$  veces el segundo ciclo, es por ello que la complejidad de tiempo es de  $O(n^2)$

Interpolación lineal	La complejidad del tiempo
Compresión	$O(N*M)$
descompresión	$O(N*M)$

**Tabla 1:** Complejidad de tiempo del algoritmo de compresión de interpolación lineal siendo  $N$  la cantidad de columnas y  $M$  la cantidad de filas.

<sup>1</sup><http://www.github.com/ ???????? /proyecto/>

<b>LZ77</b>	<b>La complejidad del tiempo</b>
Compresión	$O(N*S)$
Descompresión	$O(N)$

**Tabla 2:** Complejidad de tiempo del algoritmo de compresión LZ77 siendo N la fila de caracteres y S el tamaño de la ventana del buffer.

<b>Interpolación lineal</b>	<b>Complejidad de la memoria</b>
Compresión	$O(N*M)$
Descompresión	$O(N*M)$

**Tabla 3:** Complejidad de memoria del algoritmo de compresión de interpolación lineal siendo N la cantidad de columnas y M la cantidad de filas.

<b>LZ77</b>	<b>Complejidad de la memoria</b>
Compresión	$O(N*M)$
Descompresión	$O(N*M)$

**Tabla 4:** Complejidad de memoria del algoritmo de compresión LZ77 siendo N la cantidad de columnas y M la cantidad de filas.

#### 4.4 Criterios de diseño del algoritmo

Los algoritmos los diseñamos de la forma más fiable a como estos fueron creados anteriormente, es decir, nos enfocamos mucho en el proceso de este y que cada uno hiciera su proceso tal como dice en papel. Para el algoritmo con pérdidas, específicamente interpolación lineal, la información de este se encuentra de una forma muy matemática entonces fue un poco complicado de conseguirla, al contrario de LZ7, este algoritmo es mas relacionado en la computación entonces no hay tanto problema en poder saber como implementarlo en algún lenguaje.

## 5. RESULTADOS

### 5.1 Evaluación del modelo

En esta sección, presentamos algunas métricas para evaluar el modelo. La exactitud es la relación entre el número de predicciones correctas y el número total de muestras de entrada. La precisión es la proporción de estudiantes exitosos

identificados correctamente por el modelo a estudiantes exitosos identificados por el modelo. Por último, sensibilidad es la proporción de estudiantes exitosos identificados correctamente por el modelo a estudiantes exitosos en el conjunto de datos.

**Tabla 5.** Evaluación del modelo de clasificación de imágenes, con el conjunto de datos de prueba, con compresión.

### 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.

	<b>Tiempo promedio de ejecución (s)</b>	<b>Tamaño promedio del archivo (MB)</b>
Compresión	326.5s	0.4885 MB

**Tabla 6:** Tiempo de ejecución de los algoritmos (LZ77 e Interpolación lineal) para diferentes imágenes en el conjunto de datos.

### 5.3 Consumo de memoria

Presentamos el consumo de memoria de los algoritmos de compresión y descompresión en la Tabla 7.

	<b>Consumo promedio de memoria (MB)</b>	<b>Tamaño promedio del archivo (MB)</b>
Compresión	634 MB	3.12 MB
Descompresión	9 MB	878.12 MB

**Tabla 7:** Consumo promedio de memoria de todas las imágenes del conjunto de datos, tanto para la compresión como para la descompresión.

### 5.3 Tasa de compresión

Presentamos los resultados de la tasa de compresión del algoritmo en la Tabla 8.

	<i>Ganado sano</i>	<i>Ganado enfermo</i>
Tasa de compresión promedio	1:8	1:9

**Tabla 8:** Promedio redondeado de la tasa de compresión de todas las imágenes de ganado sano y ganado enfermo.

## 6. DISCUSIÓN DE LOS RESULTADOS

Según los resultados que pudimos obtener, los archivos exitosamente se pudieron comprimir aplicando los dos algoritmos de compresión con pérdida y sin pérdida. Los algoritmos al correr leen el archivo .csv y dejan otro archivo en la misma carpeta que se puede ejecutar con Notepad y será el archivo .csv comprimido.

### 6.1 Trabajos futuros

En el futuro nos gustaría tener una mejor compresión de los algoritmos y tener un mejor entendimiento de ellos, nos gustaría entender mejor como se crea los archivos en la carpeta y optimizar estos para que sean mas rápido, a la vez nos gustaría tal vez rediseñar los algoritmos para que estos lean cada archivo que haya en la carpeta, verifique que pueda ser comprimido y hacerlo, sino simplemente pasar hacia otro.

En general este proyecto tiene mucho espacio de mejora y potencial muy chévere para realizarlo y entenderlo, pero puede ser complicado y tedioso a veces, en especial porque no está en Google casi información de estos algoritmos.

### REFERENCIAS

1.Carmona, J., Bolívar, D. and Giraldo, L., 2016. El gas metano en la producción ganadera y alternativas para medir sus emisiones y aminorar su impacto a nivel ambiental y productivo. *Revistas.udea.edu.co*.

Carmona, J., Bolívar, D. and Giraldo, L., 2016. El gas metano en la producción ganadera y alternativas para medir sus emisiones y aminorar su impacto a nivel ambiental y productivo. *Revistas.udea.edu.co*.

Jimenez, A., Salamanca, J., Jisell, M. and Acevedo, O., 2015. Vista de Diagnóstico de Cultivos Utilizando Procesamiento Digital de Imágenes y Tecnologías de Agricultura de Precisión. *Revistascientificas.cuc.edu.co*.

2. Miranda Asturizaga, J., 2009. Compresión de imágenes digitales utilizando redes neuronales de base radial. *Repositorio.umsa.bo*.