

## GANADERÍA INTELIGENTE, SOSTENIBLE Y EFICIENTE

Daniel Posada Murillo  
Universidad Eafit Colombia  
dposad21@eafit.edu.co

Jaime Uribe  
Universidad Eafit Colombia  
jruribem@eafit.edu.co

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

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

### RESUMEN

En la actualidad, el 33% de las proteínas consumidas por los seres humanos provienen de la ganadería convirtiéndose así en un tema sumamente importante a tratar y mejorar de la mano con la tecnología para poder recolectar datos en tiempo real, mejorar así la productividad, detectar patologías con anticipación y hacer de esto algo más amigable con el medio ambiente. Estas tecnologías (GdP) consiguen una optimización de los recursos aumentando el rendimiento, controlando el impacto ambiental y mejorando el bienestar de los animales. Este mejor aprovechamiento de los recursos se lleva a cabo monitoreando la salud y producción, para ello se miden indicadores biométricos de carácter fisiológico y morfológico.

Para esta actividad, el equipo llegó a la conclusión que el mejor sistema que se podía utilizar el vecino más **cercano (k-nearest neighbors (K-nn))**, debido a es un programa sencillo de manejar, de implementar y comprensible, además ayuda para el manejo alto de información, esto lo demuestran las métricas obtenidas en la compresión y descompresión de las imágenes que se implementaron en las pruebas (*ver tablas punto 5*) donde, con una gran cantidad de archivos, es capaz de lograr su fin en tiempos mínimos con alta tasa de eficiencia.

### Palabras clave

Algoritmos de compresión, aprendizaje de máquina,

aprendizaje profundo, ganadería de precisión, salud animal.

### 1. INTRODUCCIÓN

En la actualidad, la ganadería de precisión es un tema que cada día va convirtiéndose en algo más necesario y útil donde se combinan las ciencias agropecuarias con las tecnologías de la información. Esto es con el fin de aumentar la producción y a su vez, el bienestar del animal recolectando datos en tiempo real que servirán para la temprana detección de patologías, conocer que sucede con el animal, crear

parámetros para su habitat, reducir los costos de inversión entre otros

#### 1.1. Problema

El problema al cual nos enfrentamos con los algoritmos es que se puede disminuir los tamaños a tal punto que con esto puede reducir la calidad de la imagen y se perdería información, sin embargo, a medida que surgen nuevos algoritmos, la pérdida de calidad de un archivo es menor, para permitir su transmisión y almacenamiento de forma rápida y eficiente.

Cuanto más grande sea el contenido de la información, más tiempo tomará enviarlo a su destino, y en algunos casos, disminuye la cantidad de bytes disponibles para enviar información, por ejemplo

#### 1.2 Solución

De acuerdo con la complejidad de la información que vamos a manejar, se decidió usar el algoritmo del vecino más cercano (**k-nearest neighbors (K-nn)**), es un algoritmo de aprendizaje supervisado, es decir, que a partir de un juego de datos inicial su objetivo será el de clasificar correctamente todas las instancias nuevas. El juego de datos típico de este tipo de algoritmos está formado por varios atributos descriptivos y un solo atributo objetivo (también llamado clase).

#### 1.3 Estructura del artículo

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

### 3.1 “Precision livestock farming technologies for welfare management in intensive livestock systems”

Las patologías respiratorias son uno de los mayores problemas en los cerdos siendo esta una de las mayores causas de mortalidad de estos. Es por eso por lo que se desarrolló un modelo para clasificar sonidos de los cerdos y detectar a tiempo una tos fuera de lo normal evitando así su propagación.

[https://tice.agrocampus-ouest.fr/pluginfile.php/55014/mod\\_resource/content/17/res/u324\\_Berckmans\\_RevSciTechOffIntEpiz.PDF](https://tice.agrocampus-ouest.fr/pluginfile.php/55014/mod_resource/content/17/res/u324_Berckmans_RevSciTechOffIntEpiz.PDF)

### 3.2 ¿Ganadería de precisión como herramienta para el desarrollo sustentable?

<http://sigloxxi.esпам.edu.ec/Ponencias/VIII/I%20FISPAM C/SALA2/EPP-002-2019.pdf>

## 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>.



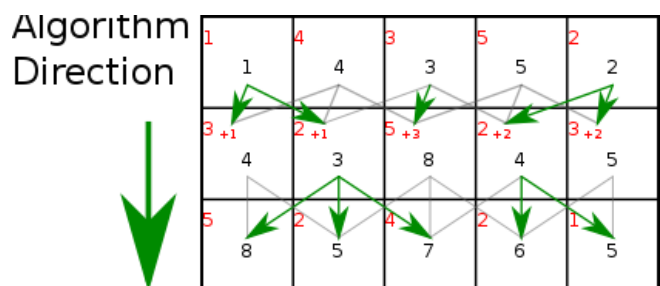
### 3.2 Alternativas de compresión de imágenes con pérdida

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

#### 3.2.1 Seam Carving

Es un algoritmo que nos permite modificar las imágenes sin pérdida de información del contenido.

“En pocas palabras, a cada valor de píxel se le asigna un valor de energía, y luego, en función de este valor de píxel, se utilizan 8 dominios conectados para la programación dinámica para obtener el valor mínimo, y luego, aplicando este algoritmo fila por fila o columna por columna obtendrá una línea de energía, que en realidad es la fase La línea del píxel con el valor de píxel más pequeño de dos filas (columnas) adyacentes se elimina de la imagen original. Cuántas de esas líneas se pueden eliminar depende de la escala que se va a recortar.”

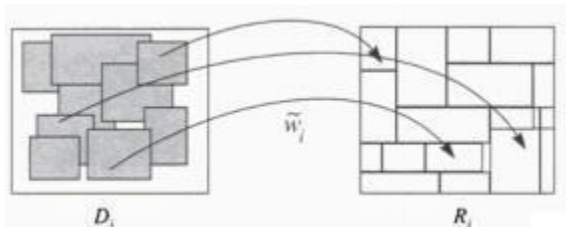


#### 3.2.2 Escalada de imágenes

Es el cambio de tamaño de una imagen digital donde al escalar una imagen, se genera una nueva imagen con un número mayor o menor de píxeles

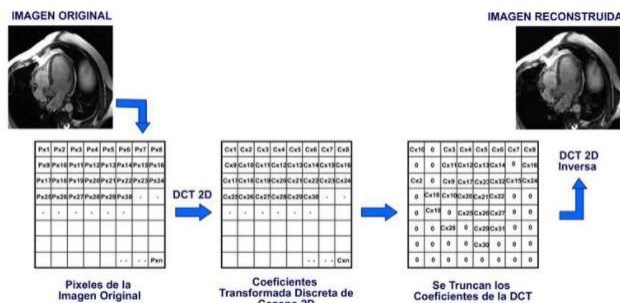
#### 3.2.3 Compresión fractal

Los algoritmos Fractales se utilizan para describir formas complejas con una reducida cantidad de información mediante sencillos procesos iterativos. Estas técnicas tienen especial interés en el tratamiento de imágenes, tanto en la generación y simulación como en la compresión.



### 3.2.4 Transformada discreta de cosenos

La imagen se debe dividir en bloques de 8 x 8 píxeles, para aplicar mejor la redundancia de información. El proceso de codificación se realiza con la fórmula principal de la transformada. Como la imagen es una señal bidimensional, la ecuación de la transformada debe adaptarse como una ecuación bidimensional.



### 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 La transformada de Burrows-Wheeler

Este algoritmo tiene como función el reordenar una cadena de caracteres en series de caracteres similares arrojando como resultado los mismos datos de entrada, distorsionando su ordenamiento; dicha distorsión tiene la capacidad de revertirse.

		$F$	$L$			$L$	$F \leftarrow$
1	mississippi	1	i mississipp	p	1	p	i
2	ississippim	2	i ppimississ	s	2	s	s
3	ssissippimi	3	i ssippimis	s	3	s	s
4	sissippimis	4	i ssissippi	m	4	m*	i
5	issippimiss	5	m ississipp	i	5	i	m
6	ssippimissi	6	p inississi	p	6	p	p
7	sippimissis	7	p pimississ	i	7	i	p
8	ppimississ	8	s ipmississ	s	8	s	s
9	ppimississi	9	s issippimi	s	9	s	s
10	pimississip	10	s sippimiss	i	10	i	s
11	imississipp	11	s ssissippim	i	11	i	s

### 3.3.2 LZ77

Este algoritmo pertenece al grupo de comprensores sin Pérdida. Funciona mediante el concepto de Ventana móvil el cual tiene la función de restringir la búsqueda de coincidencias a una parte de los datos de longitud previamente determinada. La ventaja es que el número de bits para indicar referencias se conoce de antemano y no puede crecer.

						s	a	l	s	$\rightarrow (0, 0, s)$
					s	a	l	s	a	$\rightarrow (0, 0, a)$
				s	a	l	s	a	$\square$	$\rightarrow (0, 0, 1)$

Una vez que la segunda **s** pasa a ser el carácter futuro inicial, se produce una coincidencia de longitud 2, la cadena **sa**, y hay que ir 3 cuadros hacia el pasado para encontrarla:

			s	a	l	s	a	u	s
--	--	--	---	---	---	---	---	---	---

 $\rightarrow (3, 2, u)$ 

El resto de la codificación, hasta llegar al fin de los datos sería:

	s	a	l	s	a	␣	s	a	l	→ (0, 0, s)
s	a	l	s	a	␣	s	a	l	a	→ (6, 3, a)
a	␣	s	a	l	a	d	a			→ (0, 0, a)
␣	s	a	l	a	d	a				→ (4, 1, #)

### 3.3.3 Codificación Huffman

Es un algoritmo usado para la comprensión de datos usando como método la construcción de un árbol binario donde a cada símbolo le sigue un código. Dicho proceso termina cuando solo hay un nodo representando así el fin del árbol.

Para este proceso se calcula la probabilidad que tiene cada carácter para aparecer, luego a los dos menos probables se les asigna un 0 y un 1 respectivamente. Luego se suman las probabilidades de ambos caracteres la cual se lleva a un tercer carácter



#### 4.2.2 Algoritmo de compresión de imágenes sin pérdida

La compresión sin pérdida se utiliza cuando se quiere conservar la calidad de la imagen. De este modo se organizan los datos de la imagen con mayor eficiencia sin borrar información lo que nos ayuda a tener una buena conversión de la imagen.

La compresión también depende del contenido de la imagen. A mayor detalle menos puede comprimirse. Los dos archivos de imagen que tienen la opción lossless (sin pérdida) más usados son el formato TIFF y el PNG.

La codificación Huffman crea una tabla de códigos con las combinaciones de bits que más se repiten estadísticamente. A estas sentencias se asignan códigos cortos y a otros menos probables se asignan códigos de una longitud mayor, pero provoca problemas al cambiar la imagen original.

El problema que presenta es que un diccionario de claves único tiene resultados muy diferentes en distintos originales.

El funcionamiento de Huffman es:

- Contar cuantas veces aparece cada carácter
- Crear una lista enlazada con la información de caracteres y frecuencias.
- Ordenar la lista de menor a mayor en función de la frecuencia.
- Convertir cada elemento de la lista en un árbol.
- Fusionar todos estos árboles en uno único, para hacerlo se sigue el siguiente proceso, mientras la lista de árboles contenga más de un elemento:
  - o Con los dos primeros árboles formar un nuevo árbol, cada uno de los árboles originales en una rama.
  - o Sumar las frecuencias de cada rama en el nuevo elemento árbol.
  - o Insertar el nuevo árbol en el lugar adecuado de la lista según la suma de frecuencias obtenida.
- Para asignar el nuevo código binario de cada carácter sólo hay que seguir el camino adecuado a través del árbol. Si se toma una rama cero, se añade un cero al código, si se toma una rama uno, se añade un uno.
- (Cero a la izquierda y Uno a la derecha)

#### 4.3 Análisis de la complejidad de los algoritmos

Nuestro algoritmo tiene un comportamiento estándar sin verse afectado tras la cantidad de archivos que se desean procesar.

Algoritmo	La complejidad del tiempo
Compresión	$O(n*m^3)$
Descompresión	$O(n*m)$

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

Algoritmo	Complejidad de la memoria
Compresión	$O(n*m)$
Descompresión	$O(n*m)$

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

#### 4.4 Criterios de diseño del algoritmo

Como objetivo principal, siempre tuvimos nuestra mirada en un algoritmo rápido, pues como equipo, no somos practicantes del viejo paradigma “Lento pero seguro”. Por eso fue descartado el algoritmo Lz77 debido a su complejidad. Pasamos al uso de tallado de costuras (Seam Carving) pero nos encontramos con muchos inconvenientes y bastante pérdida de la imagen. Por eso, teniendo en cuenta que nuestra primordial desventaja era la memoria, encontramos la manera de disminuir la complejidad de memoria aplicando el algoritmo en los archivos planos para las carpetas de archivos de manera unitaria disminuyendo a  $O(n*m^3)$

### 5. RESULTADOS

#### 5.1 Evaluación del modelo

#### 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.

	<i>Tiempo promedio de ejecución (s)</i>	<i>Tamaño promedio del archivo (MB)</i>
<i>Compresión</i>	1.64 s	0.235 MB
<i>Descompresión</i>	1,04 s	0.161 MB

**Tabla 6:** Tiempo de ejecución de los algoritmos (*Knn*) 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.

	<i>Consumo promedio de memoria (MB)</i>	<i>Tamaño promedio del archivo (MB)</i>
Compresión	5.69 MB	0.235 MB
Descompresión	1.59 MB	0.161 MB

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

### 5.4 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	620:255	469:225

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

## 6. DISCUSIÓN DE LOS RESULTADOS

El algoritmo usado nos arroja buenos resultados debido a que es capaz de comprimir en sus respectivas carpetas las imágenes, se observa además que con muy poca pérdida de imagen siendo un aspecto positivo y con un muy buen tiempo. Se logra el objetivo principal de la clase a la hora de implementar un algoritmo en casos de la vida real como lo es la ganadería de precisión.

## RECONOCIMIENTOS

Agradecemos a la universidad por el apoyo que nos brindó en todo lo que va del semestre y de la carrera, nos reta cada día más a ser mejores

## REFERENCIAS

- D. Berckmans, Precision livestock farming technologies for welfare management in intensive livestock systems, VisualVM, desarrollado por Oracle, <http://docs.oracle.com/javase/7/docs/technotes/guides/visualvm/profiler.html>. Para Python, usa el *C profiler*. [https://tice.agrocampus-ouest.fr/pluginfile.php/55014/mod\\_resource/content/17/res/u324\\_Berckmans\\_RevSciTechOffIntEpiz.PDF](https://tice.agrocampus-ouest.fr/pluginfile.php/55014/mod_resource/content/17/res/u324_Berckmans_RevSciTechOffIntEpiz.PDF)
- Ernesto, Hurtado - Fátima, Arteaga-Chávez, ¿GANADERÍA DE PRECISIÓN COMO HERRAMIENTA PARA EL DESARROLLO SUSTENTABLE?, 2019 <http://sigloxxi.espm.edu.ec/Ponencias/VIII/I%20FISPAMC/SALA2/EPP-002-2019.pdf>
- Gloria Puetamán Guerrero - Hernán Salazar Escobar, Compresión de imágenes usando wavelets [https://repository.eafit.edu.co/bitstream/handle/10784/155/Gloria\\_PuetamanGuerrero\\_2007.pdf?sequence=1](https://repository.eafit.edu.co/bitstream/handle/10784/155/Gloria_PuetamanGuerrero_2007.pdf?sequence=1)
- Fernando Chamizo, Resumen Codificación y compresión. <http://matematicas.uam.es/~fernando.chamizo/asignaturas/model1415/entrop.pdf> <https://www.fotonostra.com/fotografia/compresion.htm>