

# ALGORITMOS DE COMPRESIÓN PARA OPTIMIZAR EL CONSUMO DE BATERÍA EN GANADERÍA DE PRECISIÓN.

John Esteban Castro Ramírez  
Universidad Eafit  
Colombia  
jecastror@eafit.edu.co

Carlos Gustavo Vélez Manco  
Universidad Eafit  
Colombia  
cgvelezm@eafit.edu.co

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

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

## RESUMEN

La existencia del ganado es primordial para los humanos, ya que proporciona el 33% de nuestra dieta. Por tanto, es muy habitual encontrarse con el uso de herramientas tecnológicas que optimicen los procesos que se desarrollan con estos animales, como por ejemplo la ganadería de precisión (GdP), la cual monitoriza y automatiza continuamente la salud animal; sin embargo, hacen falta algunos ajustes, pensando en el público que suele usar este recurso, por lo que se hacen necesarios programas que minimicen diferentes recursos que en muchas ocasiones son escasos. Existen gran variedad de dificultades enlazadas con la anteriormente descrita, las cuales se analizarán, para encontrar una solución más completa y/o robusta.

En este trabajo, clasificaremos la salud de los animales a través de un algoritmo que se encargue de comprimir y descomprimir imágenes del ganado, con el fin de generar un mayor ahorro de energía y tiempo.

## Palabras clave

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

## 1. INTRODUCCIÓN

La historia de la ganadería en América, se remonta a la época de la conquista española, el ganado llegó con los españoles que lo usaban para su alimentación ya que los distintos pueblos indígenas no tenían animales domésticos para el consumo; en el siglo XX esta industria tomó gran importancia para la sociedad, debido al

aumento demográfico (1).

Debido al incremento constante del consumo de los productos derivados del ganado, han surgido necesidades para optimizar las labores de los trabajadores de este sector y garantizar elementos de calidad; como respuesta a este fenómeno, aparecen distintas herramientas tecnológicas, como la ganadería de precisión, que busca monitorear y automatizar los diferentes procesos de manejo de estos animales, este componente ha sido de gran ayuda, pero aún existen ciertos problemas que impiden el acceso a estos instrumentos para todo el público involucrado, por lo que se hacen esenciales ciertas modificaciones que permitan llegar a una cantidad masiva de clientes.

Para darle solución a este problema, se deben de tener en cuenta diferentes factores en cuanto a recursos, sobre todo tiempo y memoria. Cumpliendo así el objetivo principal de optimizar recursos.

### 1.1. Problema

El horizonte de este trabajo es desplegar un algoritmo, haciendo uso de estructuras de datos, que permita comprimir y descomprimir las imágenes del ganado, generando así una mayor optimización del tiempo y memoria requeridos tanto para procesar la información como para clasificar las reses según su salud.

La resolución de esta problemática, sería de gran ayuda para empresas, ganaderos y zootecnistas que usan la ganadería de precisión, ya que les permitiría usar el

algoritmo incluso con una conexión a internet inestable, que es lo que comúnmente sucede en los sitios en los cuales se encuentran alojados estos animales; y así favorecer la labor de estos entes, asegurando a la población un consumo de leche y carne de calidad.

## 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.

En primera instancia, desarrollamos un algoritmo para la compresión y descompresión con pérdidas: ‘escalamiento de la imagen por medio del vecino más cercano’; ya que teniendo en cuenta el tipo de imágenes que vamos a manejar, lo más factible es comprimir la imagen por medio de la reducción de su tamaño y la pérdida de un poco de su calidad comparado con la remoción de elementos que utiliza el tallado o el gran costo que puede significar la compresión fractal, para efectos posteriores del uso del algoritmo para la clasificación de salud animal, es más viable tener la imagen estilo pixel art que una imagen baja de calidad.

Luego, es evidente que esta compresión y descompresión con pérdidas puede generar una reducción considerable de la precisión del algoritmo de clasificación de salud animal, por lo que se hizo necesario implementar un segundo algoritmo para la compresión sin pérdidas.

## 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 Visual localization and individual identification of Holstein Friesian Cattle via Deep Learning

El ganado Holstein Friesian es el mayor productor de leche tipo bovina y de allí su importancia económica, por lo que su identificación es importante para la exportación y demanda de los clientes, este proceso se suele realizar con distintas marcas como etiquetas en sus orejas, tatuajes, entre otros, que claramente son cuestionables. Por esto, este trabajo busca implementar la automatización de reconocimiento a través de diferentes patrones en su pelaje, lo cual, contribuirá a una granja más eficaz y al bienestar del animal.

En primer lugar, se encargaron de la identificación del ganado, para esto, se dieron a conocer los 3 tipos de identificación de ganado que existen, patrones de bozal o *Cattlemuzzle patterns*, y escaneos faciales, de retinao de cuerpo y escaneos de piel, centrándose en este último. En el cual, se pudo realizar a través de algoritmos como *Scale-Invariant feature Transform* (SIFT) y *Speeded-Up Robust features* (SURF) a través de la captura de fotos de 3840x2160 pixeles tomadas a 5 metros del suelo (2).

Gracias al *Regional Convolutional Neural Networks* (RCNN), se logra seleccionar un objeto, en este caso un animal, se determina su localización y se genera un ID buscando los patrones que hay en el dorso del animal. Toda esta información queda guardada en forma de

secuencia, usando el *Long short-TermMemory* (LSTM), que organiza toda la información que recopila dependiendo de la cantidad de ganado que se tenga en la granja. A continuación, se presentan los resultados obtenidos, luego de varias validaciones.

Task	mAP (%)		
	Fold 1	Fold 2	Average
Detection & Localisation	99.02	99.59	<b>99.3</b>

### **Ilustración 1.** Precisión en la detección de especies

Como se puede observar, en promedio la precisión es de 99,3 %; demasiado buena.

## **2.2 Ganadería de precisión en vacuno de carne**

La universidad de Zaragoza realiza este trabajo con el fin de describir una gran cantidad de datos y características acerca del ganado, usando parámetros biométricos.

En principio, se seleccionaron cerca de 15.000 referentes de la utilización de GdP, y gracias a esto, el trabajo tuvo por objetivo principal analizar el vacunado de carne con la ayuda de herramientas tecnológicas que optimicen los procesos necesarios para esta tarea.

La identificación electrónica de estos animales se clasifica en dos tipos; identificación por radio de frecuencia (RFID), la cual genera una captura automática de datos e identifica animales, objetos, y hasta personas por medio de etiquetas que poseen un único número de ID que luego son puestos en objetos como placas para orejas. Una debilidad de este método es que no funciona tan bien para granjas con muy pocos animales, o para cuando existen grandes cantidades de animales muy cercanos entre sí. El otro tipo de identificación es la que se basa en parámetros biométricos o morfológicos gracias al análisis de imágenes por medio del uso de

algoritmos, por ejemplo para los humanos, se tienen datos como huellas dactilares, reconocimiento de cara e iris, entre otros, y para los bovinos se tiene también varias características como impresión de hocico, imágenes faciales y reconocimiento de iris, los cuales son procedimientos que no generarán ningún tipo de dolor o molestia en los animales, porque por ejemplo los chips pueden ser molestos para ellos.

Se realizó un estudio muy detallado acerca de varios factores como reproducción del ganado, peso de los animales, vallado virtual, bienestar del animal, alimentación, tiempo de rumia, medio ambiente, etc. Pero para este tipo de análisis es de vital importancia el uso de las dimensiones corporales.

Junto con la condición corporal, estos dos factores hablan mucho acerca del animal que se está analizando, por tanto, es necesario tenerlos en cuenta a la hora de mejorar tanto la productividad del animal como la calidad del producto que estos nos brindan, ya sea carne o leche. Por otro lado, gracias a la recopilación de estos factores, se pueden optimizar procesos tales como tiempo y esfuerzo por parte humana, y niveles de dolor y estrés por la parte bovina. Todo esto, se logra gracias al sensor LIDAR o *Light Detection and Rangin* que obtuvo las siguientes mediciones: Altura de la cruz, profundidad del pecho, altura dorsal, longitud de cuerpo y altura de la grupa(3).

## **2.3 Cloud services integration for farm animals' behavior studies based on smartphones as activity sensors**

En este trabajo, se utilizaron celulares, en particular Iphone 4s y 5s, como sensores para obtener ciertas medidas acerca del ganado.

Muchas características tanto de animales como de humanos, se pueden medir gracias al uso de celulares, por ejemplo, micrófono, sensores de presión, electromiografía, locación, acelerómetros, entre otros. Por ejemplo, el uso de un chip GPS y un

acelerómetro en el cuello de un bovino genera un cambio aproximadamente del 90% en el comportamiento del animal (4). Los celulares poseen unidades de medida inerciales, las cuales reconocen cierto tipo de señales que pueden medir, por lo que una gran ventaja de esto es que no será necesario el uso de hardware adicional ni de herramientas que midan sólo un tipo de señal.

Se concluyó que ambos celulares poseen una precisión muy parecida, con errores entre sí de aproximadamente el 0,1%.

#### **2.4 Programa de bienestar animal para grandes sistemas ganaderos.**

El monitoreo de la salud animal en las granjases esencial, ya que a través de este se puede obtener información importante y prevenir algunas enfermedades (5).

Esto se realiza a través de sensores en forma de collar, colocados en los animales; los cuales funcionan para recolectar datos en tiempo real como la geolocalización y su actividad o bienestar, que luego se cargan a la nube y se puede acceder a ellos a través de una aplicación móvil. Para optimizar, sobre todo la batería, y asegurar la transmisión de gran cantidad de datos desde los collares hasta los dispositivos Edge para luego ser montados en la nube diariamente, se utilizan 2 canales de comunicación, el primero con un pequeño rango de comunicación, altas velocidades de transmisión de datos y gran consumo de energía, en tanto que el segundo comprende un rango más amplio, menor consumo de energía y menor velocidad de transmisión; evidentemente la comunicación se dará más a través del segundo canal para optimizar la energía.

Con este servicio los ganaderos tienen la capacidad de conocer los movimientos de sus animales y estar informados; a través de los datos cargados en la nube se pueden obtener múltiples informaciones y estadísticas que les ayudan a tomar decisiones más acertadas en cuanto al futuro de sus animales y así obtener una

mayor utilidad y/o beneficio.

### **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 Liquid rescaling**

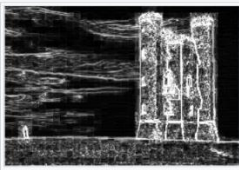
Es un algoritmo para el "re-tallado" de imágenes desarrollado por Shai Avidan y Ariel Shamir, consiste en establecer una serie de costuras o caminos de menor

importancia, una vez aplicado se pueden agregar o eliminar lo que amplía o reduce el tamaño de la imagen respectivamente.

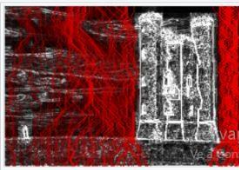
Permite definir de manera manual las áreas en las que no se desea que se modifiquen los píxeles y presenta la habilidad de remover objetos completos de las fotografías



1) Imagen inicial



2) Se calcula el peso, densidad y energía de cada píxel



3) Se hace una lista de costuras, las cuales se clasifican según la energía del píxel.



4) Se remueven todas las costuras de energía baja.



5) Imagen final.

**Ilustración 2.** Proceso del algoritmo LiquidRescaling.

Fuente:

[https://en.wikipedia.org/wiki/Seam\\_carving](https://en.wikipedia.org/wiki/Seam_carving)

### 3.2.2 IMAGE SCALING

Se refiere al ajuste de tamaño de una imagen digital, que en el medio de tecnología de video se conoce como “upscaling”; al momento de escalar un vector gráfico, los elementos primitivos que componen la imagen se pueden modificar usando transformaciones

geométricas sin pérdida de la calidad de imagen, cuando se escalan gráficos ráster se debe generar otra imagen con un mayor o menor número de píxeles, en caso de que decrezcan (scaling down) resulta con una reducción de la calidad de imagen.



**Ilustración 3.** Ejemplo de un televisor a escala Fuente:

[https://en.wikipedia.org/wiki/Image\\_scaling](https://en.wikipedia.org/wiki/Image_scaling)

Desde el punto de vista del procesamiento de señal digital, la escala de los gráficos ráster es un ejemplo bidimensional de conversión de frecuencia de muestreo, la conversión de una señal discreta de una frecuencia de muestreo (en este caso la frecuencia de muestreo local) a otra.

### 3.2.3 Compresión fractal

La palabra fractal es término reciente propuesto en el año de 1975, el cual se define como un patrón geométrico que se repite en varias escalas.

Por tanto, como su nombre lo indica, la compresión fractal es un método de compresión basado en fractales, este es el más apropiado para imágenes o texturas naturales, porque ciertas partes de la imagen se parecen a otras.

Como se deben buscar similitudes propias de la imagen, este método puede llegar a ser bastante costoso, aunque, por otro lado, la decodificación de la imagen se hace de una manera bastante rápida. Por tanto, la eficiencia del algoritmo va a depender mucho en la complejidad de la imagen y la



profundidad del color.

Una de las principales características de la compresión fractal, es que las imágenes se vuelven independientes de la resolución (escalado fractal), y esto hace posible, que la imagen se pueda pasar una mejor resolución (interpolación fractal).

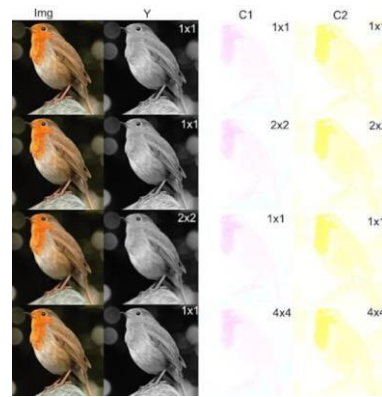


**Ilustración 4.** Ejemplo de compresión fractal. Fuente: <http://webfractales.free.fr/es/rapport.php?see=16>

### 3.2.4 Compresión JPEG

Este algoritmo, pierde información y esto se refleja en una disminución de la calidad de la imagen; pero tal pérdida no es visible para el ojo humano. El funcionamiento del algoritmo se explica a continuación: se inicia con una imagen en donde cada píxel tiene 3 canales de color (rojo, azul y verde) y lo primero que hace el algoritmo es transformar estos 3 canales en 2 de color y uno de brillo; luego, a los canales de color les baja la resolución y allí es donde se generan las pérdidas ya que elimina o modifica alguna información, ya que el ojo humano no nota estas modificaciones del color, pero si lo notase en el brillo.

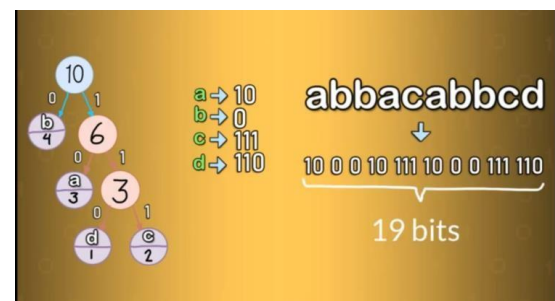
Luego divide cada canal en cuadros de 8x8 píxeles y a cada cuadro por separado le aplica la transformada discreta bidimensional del coseno y cuantificación perceptual que lo que realiza es suavizar las variaciones bruscas de brillo y color, y ya por último se aplica un algoritmo de compresión sin pérdida el cual se explica líneas adelante, el algoritmo de Huffman.



**Ilustración 5.** Funcionamiento compresión JPEG.

### 3.2.5 Codificación de Huffman

Es un procedimiento que permite asignar a los diferentes símbolos a comprimir, un código binario. Este algoritmo crea un árbol de nodos; primero se debe establecer un orden prioritario, el más importante es el símbolo que aparece con menor frecuencia en la cadena, luego, se eliminan los dos símbolos más prioritarios, construyendo así un nuevo “padre” que es el resultado de la suma de las frecuencias eliminadas y se ubica nuevamente en la cola de prioridad, iterativamente se realiza este proceso hasta que sólo quede un elemento, así queda construido el árbol. Luego, para asignar el código binario se contarán los pasos efectuados para llegar a cada símbolo del árbol (movimiento a la izquierda=0, a la derecha=1), obteniendo el valor de cada uno y por último reemplazándolos en la cadena.



**Ilustración 6.** Ejemplo codificación de Huffman.

### 3.2.6 Transformación de Burrows

La transformación de Burrows-Wheeler

reorganiza una cadena de caracteres en series similares. Es un algoritmo que prepara los datos para su posterior uso con técnicas de compresión, al ingresar una cadena de caracteres, la transformación conmuta su orden.

El algoritmo toma la cadena de entrada y realiza todas sus rotaciones hasta que el carácter inicial vuelva a quedar en su misma posición, luego con los resultados de todas estas rotaciones los ordena por orden alfabético (los símbolos especiales como \*,|,)se toman como últimos y ya una vez organizados, se toma el último carácter de cada cadena y así se obtiene la salida esperada.

1	mississippi	1	i	mississip	p	1	p	1	p	i
2	ississippi	2	i	ppinissis	s	2	s	2	s	i
3	ssissippi	3	i	ssippinis	s	3	s	3	s	i
4	ssissippi	4	i	ssissippi	n	4	n*	4	n*	i
5	ssippiniss	5	n	ississipp	i	5	i	5	i	m
6	ssippinissi	6	p	inississip	p	6	p	6	p	p
7	sippinissis	7	p	pinississ	i	7	i	7	i	p
8	ippinississ	8	s	ppinissis	s	8	s	8	s	s
9	ppinississ	9	s	issippini	s	9	s	9	s	s
10	pinississip	10	s	sippiniss	i	10	i	10	i	s
11	inississipp	11	s	ssissippi	i	11	i	11	i	s

**Ilustración 7.** Ejemplo transformación de Burrows-Wheeler.

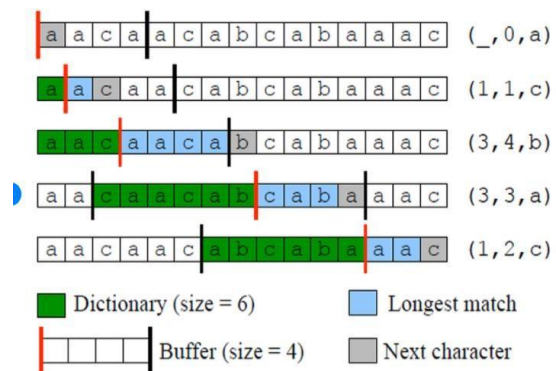
Fuente:

<https://bioinfoperl.blogspot.com/2011/07/transformada-de-burrows-wheeler-para.html>

### 3.2.7 LZ77

Es un compresor basado en algoritmo sin pérdida, este algoritmo es un tipo de codificador de diccionario en el cual existen los literales, banderas y palabras claves; se empieza a recorrer la cadena y si se encuentra con un literal lo deja totalmente igual, si encuentra una bandera específica si lo que sigue es un literal o un comprimido y si lo anterior es un comprimido (que es una especie de palabra clave) se lleva a una posición en un diccionario que arroja que bytes continúan. Por ejemplo, en una imagen las esquinas pueden ser iguales y al comprimir tales esquinas lo que hacemos

es que se tenga que guardar una sola vez y por así decirlo evitar las repeticiones.



**Ilustración 8.** Ejemplo compresión con LZ77

### 3.2.8 LZ78

El algoritmo LZ78 usa una estructura de datos, específicamente trie, basada en diccionario para comprimir los datos. Este nació para mejorar el rendimiento respecto al LZ77, es por así decirlo, una segunda versión mejorada.

Respecto al funcionamiento para la compresión, se lee una cadena carácter por carácter y verificamos si el nodo actual (Cada nodo está marcado con el índice del diccionario) tiene algún borde de salida que contenga al carácter leído; en caso positivo configuramos el nodo actual como el nodo encontrado y repetimos el proceso, en caso negativo, se crea un borde de salida con el carácter leído se crea un nodo nuevo, el cual pasa a ser el nodo raíz e igualmente se repite el proceso.

No	Pattern	No	Pattern	No	Pattern	No	Pattern	No	Pattern	No	Pattern
0	D	0	D	0	D	0	D	0	D	0	D
1		1	A	1	A	1	A	1	A	1	A
2		2		2	D	2	D	2	D	2	D
3		3		3		3	DA	3	DA	3	DA
4		4		4		4		4	DAB	4	DAB
5		5		5		5		5		5	DAD

**Ilustración 9.** Ejemplo compresión usando LZ78 de la cadena “DAD DADABDAD”

## 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

Para almacenar los datos usamos una lista enlazada, y dentro de ella en cada posición tenemos un diccionario que contiene el nombre del archivo y una matriz que contiene los valores de los pixeles, utilizamos el diccionario ya que nos permitía conservar los nombres originales del archivo luego de la compresión y descompresión.

Para el efecto del algoritmo usamos sólo la matriz de pixeles.

127	94	76	57
0	128	56	87
27	115	110	95
123	112	82	53

**Ilustración 10.** Matriz de 4x4, que contiene los pixeles de cada imagen.

Luego de la compresión claramente la matriz se reducirá y luego de la descompresión volverá a su tamaño original.

### 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 y descompresión de las imágenes con pérdidas usamos el escalado por medio del vecino más cercano.

Este método es muy sencillo, lo único que se hace es dividir la matriz de pixeles de la imagen en submatrices de igual tamaño y escogemos cualquier pixel dentro de cada submatriz, pero siempre el de la misma posición, es decir si en la primera submatriz escogemos el pixel de la mitad en todas las submatrices también deberemos de seleccionar el de la mitad.

127	94	76	57
0	128	56	87
27	115	110	95
123	112	82	53




127	76
27	110

**Ilustración 11.** Compresión mediante el algoritmo del vecino más cercano.

Para la descompresión lo que hacemos es replicar el valor que obtuvimos de la compresión a lo largo del tamaño de la submatriz definido y así obtendremos nuevamente la cantidad de pixeles originales.

127	76
27	110



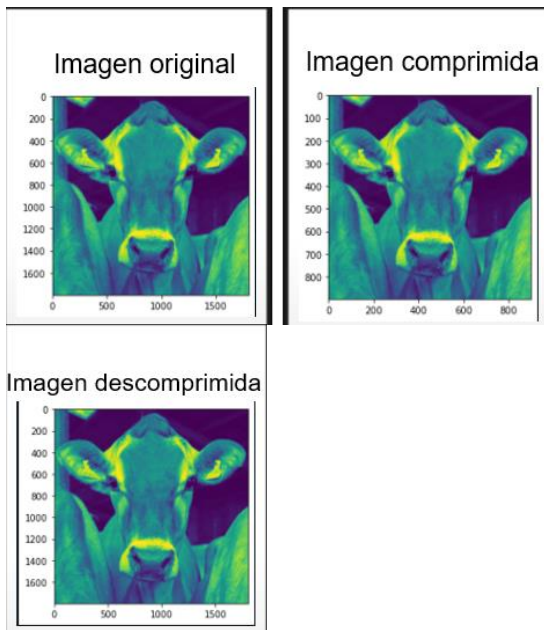
127	127	76	76
127	127	76	76
27	27	110	110
27	27	110	110

**Ilustración 12.** Descompresión mediante el algoritmo del vecino más cercano.

En la siguiente imagen podemos observar los resultados de la compresión y descompresión con una de las imágenes del conjunto de datos.

<sup>1</sup><https://github.com/johnesteban/ST0245-002/tree/master/proyecto>





**Ilustración 13.** Ejemplo de una imagen comprimida y descomprimida

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

##### REFERENCIAS

1. Fedegan. Nuestra Hlstory. 2013.  
<https://www.fedegan.org.co/quienes-somos/nuestra-historia>
2. Andrew, W and Greatwood, C and Burghardt, T. *Visual Localization and*
3. *Individual Identification of Holstein Friesian Cattle via Deep Learning*. Bristol. 2007.
4. Alzate, C. *Ganadería de precision en vacuno de carne*. Zaragoza. 2020.
5. Debauche, O and Mahmoudi, S and Andriamandroso, A and Manneback, P and Bindelle, J and Lebeau, F. *Cloud services integration for farm animals' behavior studies based on smartphones as activity sensors*. Alemania. 2018.
6. Doulgerakis, V and Kalyvas, D and Bocaj, E and Giannousis, C and Feidakis, M and Laliotis, G and Patrikakis, C and Bizelis, I. *An animal welfare platform for*

*Extensive Livestock Production Systems.*

7. *Explicación detallada del algoritmo de intepolacion bilineal de imágenes Python realiza el algoritmo de interpolación bilineal – programador clicl.(s.f.)*  
Programador clicl. Recuperado 11 de abril de 2021, de <https://programmerclick.com/article/9212974781/>