

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. *¿Cuál es el algoritmo propuesto? ¿Qué resultados obtuviste? ¿Cuáles son las conclusiones de este trabajo? El resumen debe tener como máximo 200 palabras. (En este semestre, deberías resumir aquí los tiempos de ejecución, el consumo de memoria, la tasa de compresión y la exactitud).*

Palabras clave

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.

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

1.3 Estructura del artículo

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

Explique cuatro (4) artículos relacionados con el problema descrito en la sección 1.1. Puede encontrar los problemas relacionados en las revistas científicas, en lo posible, en inglés. Considere Google Scholar para su búsqueda. (En este semestre, el trabajo relacionado es la investigación sobre la clasificación de la salud animal y la compresión de datos, en el contexto de la GdP).

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>.

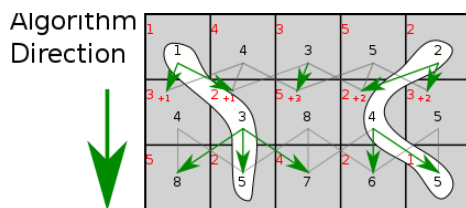
3.2 Alternativas de compresión de imágenes con pérdida

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

3.2.1 Tallado de costuras

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.

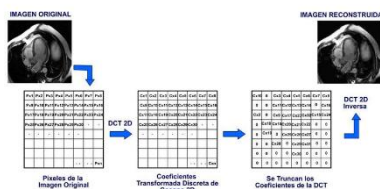


3.2.2 Transformada discreta del coseno

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.

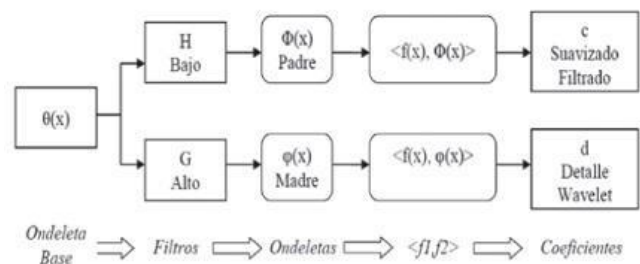


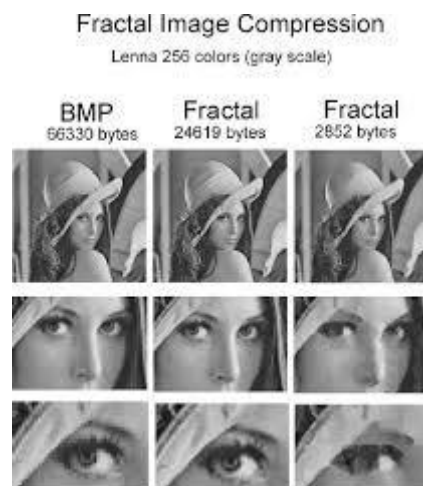
Figura 1. Esquema transformación directa con ondeletas

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.



3.3 Alternativas de compresión de imágenes sin pérdida

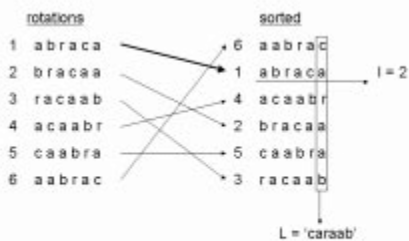
En lo que sigue, presentamos diferentes algoritmos usados para comprimir imágenes sin pérdida. *(En este semestre, ejemplos de tales algoritmos son la transformada de Burrows y Wheeler, LZ77, LZ78, la codificación Huffman y LZS).*

3.3.1 Compresión de Burrows-Wheeler

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.

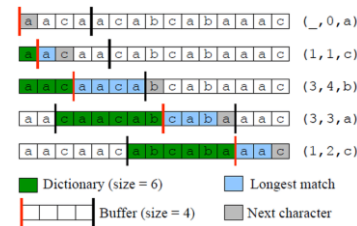


3.3.2 LZ77 y LZ78

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

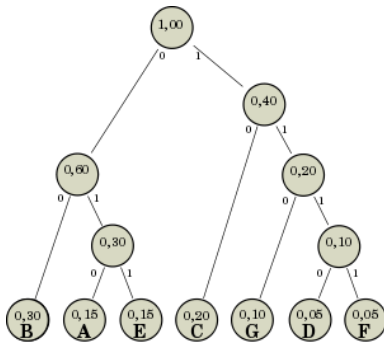
El LZS es un algoritmo de compresión de datos que usa el “sliding window” de LZ77 con codificación de Huffman.

Este compresor 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	S T A G A C A G	A	
1	S T A G A C A G A	1	A S T A G A C A	G	
2	A S T A G A C A G	2	A C A G A S T A	G	
3	G A S T A G A C A	3	A G A S T A G A	C	
4	A G A S T A G A C	4	A G A C A G A S	T	
5	C A G A S T A G A	5	C A G A S T A G	A	
6	A C A G A S T A G	6	G A S T A G A C	A	
7	G A C A G A S T A	7	G A C A G A S T	A	
8	A G A C A G A S T	8	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 Github¹.

4.1 Estructuras de datos

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

Figura 1: Árbol de Huffman generado a partir de las frecuencias exactas del texto "this". (Por favor, no dude en cambiar esta figura si utilizan una estructura de datos diferente).

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.

Expliquen el diseño de los algoritmos para resolver el problema y hagan una figura. No uses figuras de Internet, haz las tuyas propias. (En este semestre, un algoritmo debe ser un algoritmo de compresión de imágenes con pérdidas, como el escalado de imágenes, el tallado de costuras o la compresión con ondeletas, y el segundo algoritmo debe ser un algoritmo de compresión de imágenes sin pérdidas, como la codificación Huffman, LZS o LZ77).

4.2.1 Algoritmo de compresión de imágenes con pérdida

Explique, brevemente, cómo se aplicó un algoritmo de compresión de imágenes con pérdidas, como, por ejemplo, el tallado de costuras o el escalado de imágenes. Explique también la descompresión.

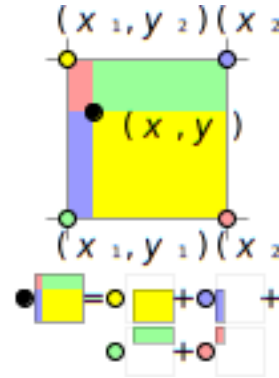


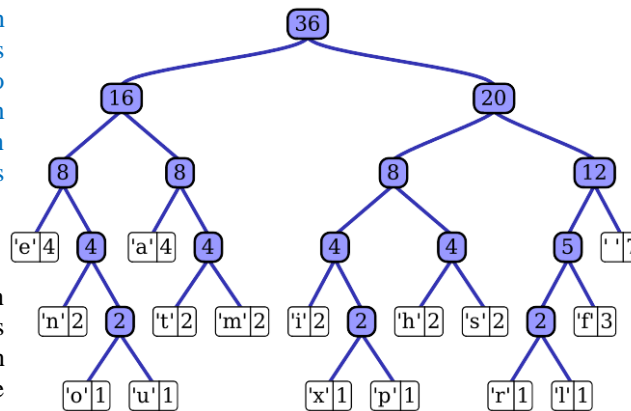
Figura 2: Escalado de la imagen mediante interpolación bilineal. (Por favor, siéntase libre de cambiar esta figura si utiliza una estructura de datos diferente).

4.2.2 Algoritmo de compresión de imágenes sin pérdida

Explique brevemente cómo aplicó un algoritmo de compresión de imágenes sin pérdidas como la codificación Huffman, LZS o LZ77. Explique también la descompresión.

4.3 Análisis de la complejidad de los algoritmos

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



Algoritmo	La complejidad del tiempo
Compresión	$O(N^2 * M^2)$
Descompresión	$O(N^3 * M * 2^N)$

Tabla 2: Complejidad temporal de los algoritmos de compresión y descompresión de imágenes. (Por favor, explique qué significan N y M en este problema).

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

Algoritmo	Complejidad de la memoria
Compresión	$O(N*M*2N)$
Descompresión	$O(2M*2N)$

Tabla 3: Complejidad de memoria de los algoritmos de compresión y descompresión de imágenes. (Por favor, explique qué significan N y M en este problema).

4.4 Criterios de diseño del algoritmo

Explica por qué el algoritmo fue diseñado de esa manera. Use un criterio objetivo. Los criterios objetivos se basan en la eficiencia, que se mide en términos de tiempo y consumo de memoria. Ejemplos de criterios no objetivos son: "Estaba enfermo", "fue la primera estructura de datos que encontré en Internet", "lo hice el último día antes del plazo", etc. Recuerde: Este es el 40% de la calificación del proyecto.

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.

5.1.1 Evaluación del conjunto de datos de entrenamiento

A continuación presentamos las métricas de evaluación del conjunto de datos de entrenamiento en la Tabla 3.

	Conjunto de datos de entrenamiento
Precisión	0.02
Precisión	0.03
Recordar	0.01

Tabla 3. Evaluación del modelo de clasificación de imágenes con el conjunto de datos de entrenamiento.

5.1.2 Evaluación del conjuntos de datos de prueba

A continuación presentamos las métricas de evaluación del conjunto de datos de prueba, en la Tabla 4, sin compresión y, en la Tabla 5, con compresión.

	Conjunto de datos de prueba
Exactitud	0.01
Precisión	0.012
Sensibilidad	0.013

Tabla 4. Evaluación del modelo de clasificación de imágenes, con el conjunto de datos de prueba, sin compresión.

	Conjunto de datos de prueba
Exactitud	0.001
Precisión	0.0012
Sensibilidad	0.0013

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.

	Tiempo promedio de ejecución (s)	Tamaño promedio del archivo (MB)
Compresión	100.2 s	12.4 MB
Descompresión	800.1 s	12.4 MB

Tabla 6: Tiempo de ejecución de los algoritmos (Por favor, escriba el nombre de los algoritmos, por ejemplo, tallado de costuras y LZ77) 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.

	Consumo promedio de memoria (MB)	Tamaño promedio del archivo (MB)

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.

Para medir el consumo de memoria, deberían usar un generador de perfiles. Uno muy bueno para Java es VisualVM, desarrollado por Oracle, <http://docs.oracle.com/javase/7/docs/technotes/guides/visualvm/profiler.html>. Para Python, usa el *C profiler*.

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	1:23	1:34

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

6. DISCUSIÓN DE LOS RESULTADOS

Explique los resultados obtenidos. ¿Son la exactitud, la precisión y la sensibilidad apropiadas para este problema? ¿El modelo está sobreajustado? ¿Es apropiado el consumo de memoria y el consumo de tiempo? ¿Es la relación de compresión apropiada? ¿Cambia la compresión significativamente la exactitud con el conjunto de datos de la prueba? (En este semestre, según los resultados, ¿puede la compresión mejorar la clasificación de la salud animal en el contexto del PLF?)

6.1 Trabajos futuros

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

RECONOCIMIENTOS

Identifique el tipo de reconocimiento que quiere escribir: para una persona o para una institución. Considere las siguientes pautas: 1. El nombre del profesor no se menciona porque es un autor. 2. No debe mencionar sitios web de autores de artículos que no ha contactado. 3. Debe mencionar estudiantes, profesores de otros cursos que le hayan ayudado.

Como ejemplo: Esta investigación fue apoyada/parcialmente apoyada por [Nombre de la Fundación, Donante, Beca].

Agradecemos la asistencia con [técnica particular, metodología] a [Nombre Apellido, cargo, nombre de la institución] por los comentarios que mejoraron enormemente el manuscrito o la codificación del algoritmo.

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*.