**AUTOMATIZACIÓN DE DATOS CON EL MENOR GASTO DE ENERGIA Y MEMORIA APLICADO EN EL SECTOR GANADERO**

|  |  |  |  |
| --- | --- | --- | --- |
| Silvia Katheryn Caceres Vasquez  Universidad Eafit  Colombia  skcaceresv@eafit.edu.co | Alejandro Suarez Sanchez  Universidad Eafit  Colombia  asuarezs@eafit.edu.co | Simón Marín Universidad Eafit Colombia smaring1@eafit.edu.co | Mauricio Toro  Universidad Eafit  Colombia  mtorobe@eafit.edu.co |

**Para cada versión de este informe: 1. Elimine todo el texto en rojo. 2. Ajustar los espacios entre las palabras y los párrafos. 3. Cambiar el color de todos los textos a negro.**

**Texto rojo =** Comentarios

**Texto negro =** Ccontribución 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**

Con este proyecto se desea analizar y resolver la deficiencia que se encuentra a la hora de guardar y registrar datos en la zona ganadera.

Lo que se busca es encontrar y utilizar una estructura de datos que nos permita, guardar componentes con el menor consumo de energía posible, buen rendimiento, eficacia y accesibilidad para los ganaderos; evitando así que estos guarden sus registros de variables como la salud, la alimentación, entre otros, en hojas de cuadernos, las cuales se pueden perder o dañar fácilmente y la búsqueda de datos de esta manera se hace mucho más tediosa.

.¿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 sistema de producción sostenible que reduce el impacto ambiental mediante el uso eficiente de los recursos, contribuyendo a su vez con el bienestar del ganado y un control de calidad; está asociada con las tecnologías de la información y comunicación, por lo que enfrenta un desafío relacionado con la digitalización automática de los datos. Una de las alternativas de solución a dichos problemas es la estructura de datos y los algoritmos de compresión, que permiten organizar grandes cantidades de datos para que sean usados de manera óptima, y con lo cual puede ser posible el ahorro de energía de la batería de los equipos de las granjas.

# **Problema**

El problema que se busca resolver, principalmente se basa en crear a través de una estructura de datos un sistema de archivos eficiente en cuanto a las especificaciones de un cliente con el menor gasto de energía y memoria para registrar fácilmente las diferentes variables que se requieran dentro del sector ganadero y así llevar un monitoreo mucho más eficiente sobre los animales. Resolver este problema sería un avance en cuanto a la reducción de recursos en los sistemas de archivos y almacenamiento que maneja un sistema operativo, debido a que con el uso menor de memoria y energía, puede servir para aquellas empresas que trabajan con cantidades enormes de información.

**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 relacional con el problema. Más adelante, en la Ssección 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 Integración de servicios en la nube para estudios de comportamiento de animales de granja basados ​​en teléfonos inteligentes como sensores de actividad.**

Los teléfonos inteligentes, particularmente el iPhone, pueden ser instrumentos relevantes para los investigadores en comportamiento animal porque están fácilmente disponibles en el planeta, contienen muchos sensores y no requieren desarrollo de hardware.

El estudio del comportamiento animal utilizando teléfonos inteligentes requiere el almacenamiento de muchas variables de alta frecuencia de un gran número de individuos y su procesamiento a través de diversas combinaciones de variables relevantes para el modelado y la toma de decisiones. Transferir, almacenar, tratar y compartir tal cantidad de datos es un gran desafío.

## **3.2 Localización visual e identificación individual del ganado Holstein Friesian a través del aprendizaje profundo**

Se demuestra que las tuberías de visión por computadora que utilizan arquitecturas neuronales profundas son adecuadas para realizar la detección automatizada de ganado Holstein Friesian, así como la identificación individual en configuraciones relevantes para la agricultura. A nuestro leal saber y entender, este trabajo es el primero en aplicar el aprendizaje profundo a la tarea de identificación visual automatizada de bovinos. Demostramos que las redes disponibles en el mercado pueden realizar la identificación de un extremo a otro de los individuos en imágenes fijas de arriba hacia abajo adquiridas con cámaras fijas. Luego, presentamos una tubería de procesamiento de video compuesta de componentes estándar para procesar de manera eficiente la filmación dinámica de rebaños. por vehículos aéreos no tripulados (UAV). Informamos sobre estas configuraciones, así como el contexto, entrenamiento y evaluación de sus componentes. Publicamos junto con nuevos conjuntos de datos: FriesianCattle2017 de imágenes de arriba hacia abajo en el establo, y AerialCattle2017 de imágenes de ganado al aire libre filmadas por un UAV DJI Inspire MkI. Demostramos que la detección y localización del ganado frisón se puede realizar de manera robusta con una precisión del 99,3% en estos datos. Evaluamos la identificación individual explotando la singularidad del pelaje en 940 RGB todavía extraídas después del ordeño en el establo (89 individuos, precisión = 86,1%). También evaluamos la identificación a través de una tubería de procesamiento de video en 46.430 fotogramas que se originan en 34 clips (aproximadamente 20 s de longitud cada uno) de imágenes de UAV tomadas durante el pastoreo (23 individuos, precisión = 98.1%). Estas pruebas sugieren que, particularmente al grabar en video rebaños pequeños en ambientes despejados, una aplicación de identificación de ganado frisón sin marcadores no solo es factible usando componentes estándar de aprendizaje profundo, sino que parece lo suficientemente robusta como para ayudar a los métodos de etiquetado existentes.

## **3.3 La identificación biométrica automatizada de bovinos**

## Los patrones del hocico forman la base de mucha investigación sobre la identificación bovina semiautomatizada métodos. Las muestras aleatorias para la extracción y el filtrado de características permiten lograr una precisión de identificación del 93,3% en las imágenes del hocico.

## Los enfoques alternativos para la identificación de bovinos incluyen construcción de representaciones faciales de ganado basadas en características de textura de patrón binario local.

## **3.4 Una plataforma de bienestar animal para sistemas extensivos de producción ganadera**

## Actualmente, la gente desea que la producción agrícola favorezca tanto a los humanos como a los animales, el estudio sobre el comportamiento animal, su estado y bienestar, se ha convertido en algo de suma importancia.

## Los avances tecnológicos recientes ofrecen un seguimiento de indicadores vitales que se pueden registrar y analizar a fondo, esto puede permitir la identificación temprana de problemas de salud, estado y bienestar de los animales a nivel de la granja, así como la intervención oportuna.

## Aunque los avances tecnológicos permiten el desarrollo de dispositivos de última generación para registrar el comportamiento animal, su uso es bastante costoso para el productor.

## **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 .

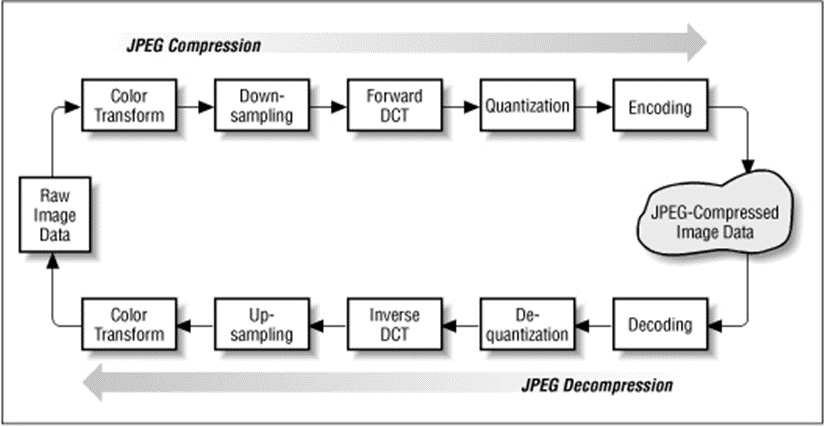
Por último, utilizando el conjunto de datos de entrenamiento, entrenamos una red neuronal convolucional para la clasificación binaria de imágenes utilizando *Teachable Machine* de Google disponible en https://teachablemachine.withgoogle.com/train/image.

## **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 JPEG (Joint Photographic Experts Group)**

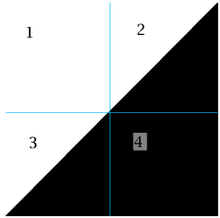
Es un algoritmo diseñado para comprimir imágenes con 24 bits de profundidad o en escala de grises. JPEG es también el formato de archivo que utiliza este algoritmo para almacenar las imágenes comprimidas. JPEG sólo trata imágenes fijas, pero existe un estándar relacionado llamado MPEG para videos. El formato de archivos JPEG se abrevia frecuentemente .jpg debido a que algunos sistemas operativos sólo aceptan tres letras de extensión. JPEG es un algoritmo de compresión con pérdida. Esto significa que al descomprimir la imagen no obtenemos exactamente la misma imagen que teníamos antes de la compresión. Si especificamos una compresión muy alta se perderá una cantidad significativa de calidad, pero obtendremos archivos de pequeño tamaño. Con una tasa de compresión baja obtenemos una calidad muy parecida a la del original, y un archivo mayor. La compresión con pérdida del formato no conviene a diagramas que incluyen textos y líneas.



**3.2.2**

**La compresión fractal**

Es un método de compresión con pérdida de imágenes digitales, basado en fractales. El método es más adecuado para texturas e imágenes naturales, ya que se basa en el hecho de que partes de una imagen a menudo se parecen a otras partes de la misma imagen. Los algoritmos fractales convierten estas partes en datos matemáticos llamados "códigos fractales" que se utilizan para recrear la imagen codificada.



**3.2.3 Enhanced Compression Wavelet (ECW)**

Es formato de archivo privativo para imágenes raster que presenta unos ratios muy altos de compresión, desde 10:1 hasta de 50:1, mediante el uso de técnicas de ondículas. Debido a esto se reduce considerablemente el tamaño de los archivos, manteniendo una alta calidad gráfica y permitiendo una rápida compresión y descompresión mediante un uso escaso de la memoria RAM.

La imagen comprimida ECW se procesa línea a línea directamente a partir de la imagen original. La técnica de compresión ECW puede comprimir imágenes de cualquier tamaño usando un algoritmo recursivo de segmentación que no precisa del uso de almacenamiento en disco mientras se realiza la DWT.

**3.2.4 Better Portable Graphics (BPG)**

Es un formato de archivo para codificar imágenes digitales. Se basa en la codificación intracuadro del estándar de compresión de video High Efficiency Video Coding (HEVC). El formato ha sido diseñado para ser portátil y funcionar en entornos de poca memoria, y se utiliza en dispositivos portátiles de mano y de IoT.

## **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 Borrows y Wheeler, LZ77, LZ78, la codificación Huffman y LZS).*

**3.3.1 Compression de Burrows Wheeler**

 Es un [algoritmo](https://es.wikipedia.org/wiki/Algoritmo) usado en técnicas de [compresión de datos](https://es.wikipedia.org/wiki/Compresi%C3%B3n_de_datos) como en [bzip2](https://es.wikipedia.org/wiki/Bzip2). Se basa en una transformación previamente descubierta por Wheeler que no se encuentra publicada.

Cuando se transforma una [cadena de caracteres](https://es.wikipedia.org/wiki/Cadena_de_caracteres) mediante la BWT, ninguno de sus caracteres cambia de valor. La transformación [permuta](https://es.wikipedia.org/wiki/Permutaci%C3%B3n) el orden de los caracteres. Si la cadena original contiene muchas subcadenas que aparecen a menudo, entonces la cadena transformada contendrá múltiples posiciones en las que un mismo carácter esté repetido varias veces en una fila. Esto es útil para la compresión, ya que tiende a ser fácil comprimir una cadena que contiene secuencias de caracteres repetidos con técnicas como [move-to-front transform](https://es.wikipedia.org/w/index.php?title=Move-to-front_transform&action=edit&redlink=1" \o "Move-to-front transform (aún no redactado)) y [run-length encoding](https://es.wikipedia.org/wiki/Run-length_encoding).

Imagen que contiene Texto

Descripción generada automáticamente

**3.3.2 LZ77 y LZ78**

Ambos son, teóricamente, [codificadores de diccionario](https://en.wikipedia.org/wiki/Dictionary_coder) . LZ77 mantiene una ventana deslizante durante la compresión. Más tarde se demostró que esto era equivalente al *diccionario explícito* construido por LZ78; sin embargo, solo son equivalentes cuando se pretende descomprimir todos los datos.

Dado que LZ77 codifica y decodifica desde una ventana deslizante sobre los caracteres vistos anteriormente, la descompresión siempre debe comenzar al comienzo de la entrada. Conceptualmente, la descompresión del LZ78 podría permitir el acceso aleatorio a la entrada si se conociera todo el diccionario de antemano. Sin embargo, en la práctica, el diccionario se crea durante la codificación y decodificación creando una nueva frase cada vez que se emite un token.

**3.3.3 Codificación Huffman**

la **codificación Huffman** es un [algoritmo](https://es.wikipedia.org/wiki/Algoritmo) usado para [compresión de datos](https://es.wikipedia.org/wiki/Compresi%C3%B3n_de_datos). El término se refiere al uso de una tabla de códigos de longitud variable para codificar un determinado símbolo (como puede ser un carácter en un archivo), donde la tabla ha sido rellenada de una manera específica basándose en la probabilidad estimada de aparición de cada posible valor de dicho símbolo.

 La codificación Huffman usa un método específico para elegir la representación de cada símbolo, que da lugar a un [código prefijo](https://es.wikipedia.org/wiki/C%C3%B3digo_prefijo) (es decir, la cadena de bits que representa a un símbolo en particular nunca es prefijo de la cadena de bits de un símbolo distinto) que representa los caracteres más comunes usando las cadenas de bits más cortas, y viceversa. Huffman fue capaz de diseñar el método de compresión más eficiente de este tipo: ninguna representación alternativa de un conjunto de símbolos de entrada produce una salida media más pequeña cuando las frecuencias de los símbolos coinciden con las usadas para crear el código. Posteriormente se encontró un método para llevar esto a cabo en un tiempo lineal si las probabilidades de los símbolos de entrada (también conocidas como "pesos") están ordenadas.

Pantalla negra con letras blancas

Descripción generada automáticamente

**3.3.4 LZSS**

El algoritmo de compresión **lz77** (acrónimo de *Abraham****L****empel*, *Jacob****Z****iv* y 19**77**) pertenece a la familia de compresores sin pérdida, también llamados *compresores de texto*, a los cuales se les llama así porque no omiten información del archivo al comprimirlo, al contrario que los compresores que utilizan [algoritmos](https://es.wikipedia.org/wiki/Algoritmos) del tipo [lossy](https://es.wikipedia.org/w/index.php?title=Lossy&action=edit&redlink=1" \o "Lossy (aún no redactado)), que omiten algo de información pero que disminuyen considerablemente el [tamaño del archivo](https://es.wikipedia.org/wiki/Tama%C3%B1o_de_archivo) original, el cual es el caso de los archivos [MP3](https://es.wikipedia.org/wiki/MP3), [MPG](https://es.wikipedia.org/wiki/MPG), [jpeg](https://es.wikipedia.org/wiki/Jpeg" \o "Jpeg), etc.



## **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[[1]](#footnote-1).

## **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:** Árbolde 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:** Escaladode 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(N2\*M2) |
| Descompresión | O(N3\*M\*2N) |

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

|  |  |
| --- | --- |
| **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 deejecución 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:** Consumopromedio 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.

|  |  |  |
| --- | --- | --- |
|  | ***Ganado sano*** | ***Ganado enfermo*** |
| 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**

**Bibliografía**

(Junio 19 de 2021). Lossy Compression. Wikipedia Recuperado de https://en.wikipedia.org/wiki/Lossy\_compression

(Julio 15 de 2021). *Joint Photographic Experts Group*. Wikipedia Recuperado de https://es.wikipedia.org/wiki/Joint\_Photographic\_Experts\_Group

Enkeleda, B. Dimitrios, K. Christos, G. Michalis, F. (Noviembre 13 de 2019). An Animal Welfare Platform for Extensive Livestock Production Systems. ResearchGate Recuperado de <https://www.researchgate.net/publication/338595895_An_Animal_Welfare_Platform_for_Extensive_Livestock_Production_Systems>

Octubre 31 2020) LZSS. Wikipedia Recuperado de <https://es.wikipedia.org/wiki/LZSS>

(Agosto 17 2021) LZ77 y LZ78. Wikipedia Recuperado de <https://en.wikipedia.org/wiki/LZ77_and_LZ78>

(Febrero 14 2021) Codificacion Huffman. Wikipedia Recuperado de <https://es.wikipedia.org/wiki/Codificaci%C3%B3n_Huffman>

(Febrero 12 2021) Burrows Wheeler. Wikipedia Recuperado de <https://es.wikipedia.org/wiki/Compresi%C3%B3n_de_Burrows-Wheeler>

1. http://www.github.com/ ????????? /proyecto/ [↑](#footnote-ref-1)