**Compresión de imágenes para la ganadería de precisión**

|  |  |  |  |
| --- | --- | --- | --- |
| Samuel Palacio Jimenez  Universidad Eafit  Colombia  spalacioj@eafit.edu.co | Juan Esteban Jaramillo Ramos  Universidad Eafit  Colombia  jejaramilr@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**

El problema que buscamos encontrarle una solución es con relación a la ganadería de precisión el cual trata la ganadería con el uso de tecnologías, en nuestro caso tratemos los animales sanos ya que muchas veces no se sabe si esta enfermo por lo cual nosotros consumimos esta carnes con enfermedades por lo cual esto propaga enfermedades uno de los tantos problemas relacionados que tenemos es por ejemplo la proteína que contiene la comida con la que se alimentan a el ganado.

## **Palabras clave**

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

# **INTRODUCCIÓN**

Tenemos que en el mundo hay una población aproximada de 7.8 billones de personas, en Colombia hay 50,34 millones de personas y tenemos en cuenta que esto va en aumento por lo cual se necesita cumplir las necesidades básicas una de ellas el hambre y se están sobreexplotando los recursos naturales como agua y animales para satisfacer estas por lo cual necesitamos que estos animales aparte de dar comida suficiente den los nutrientes y proteína buena entonces nosotros buscamos saber si estos animales están en un estado de salud sano para que se pueda comer y darse a una población y de estar enfermo, encontrar cual es la enfermedad para que no se propague por esta misma. Aparte si lo hacemos con imágenes comprimida. estos procesos de clasificar el ganado se harán mucho más eficiente y rápido que con una imagen sin comprimir al consumir mucho tiempo para esta entonces no habría ningún beneficio.

# **1.1. Problema**

el principal problema que se plantea solucionar es la calidad del ganado ya que sabemos que un ganado enfermo lo que hará es no dar los suficientes nutrientes y proteína que se necesita que de, por ejemplo, un ganado enfermo no comerá lo mismo que uno sano y eso no es todo ya que puede contagiar de enfermedades. La importancia de resolver esto es que sabemos que cada vez hay mas escases de alimentos debido a la población que va en aumento.

**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.

## 2.1 En este artículo, se muestra la visión por computadora.

Para la realización de la detección automática de ganado individual en la agricultura relacionada, primero hicieron la prueba con imágenes fijas de arriba hacia abajo con cámaras fijas y después usaron nos drones para probar el algoritmo con videos a 43 fotogramas por segundo

El algoritmo que usaron se basa en obtener imágenes desde un punto de vista aéreo que detecta el conjunto limitante de la vaca que es la tierra a su alrededor y se conecta con varias capaz para llegar a una red neuronal que es donde hace el proceso de validación de la vaca individual

2.2 En este Proyecto se demuestra como, mediante los avances tecnológicos se es capaz de genera un reconocimiento de los animales ofreciendo información de problemas de salud, estado, y bienestar a nivel de la granja

[1] Andrew, Greatwood & Burghardt T.(2018) Visual Localisation and Individual Identification of Holstein Friesian Cattle via Deep LearninG. University of Bristol

El algoritmo que usan se aplica mediante la utilización de collares puestos en los animales. Para el procesamiento de los datos primero se pasa la información por un filtro de corrección de errores o de ruidos durante la captación de esta información, después esta información llega a una red neuronal donde puede dar un resultado dependiendo de esta informacion

[1] Vasileios, Dimitrios, Enkeleda, Christos, Michalis, George. An Animal Welfare Platform for Extensive Livestock Production Systems. University of West Attica

Agricultural University of Athens

2.3 En este proyecto se descubrieron que mediante los celulares se pudo almacenar muchas variables sobre los individuos en cuestión y mediante una plataforma para archivar y procesar los datos de alta frecuencia

El algoritmo usado en este proyecto fue utilizar celulares de la marca iPhone, lo colocaron en el animal en cuestión y mediante sus sensores pudieron ir almacenando datos para recolectarlo. También tuvieron que hacer pruebas con sus baterías de los celulares y como comprimían los datos para poder usarlos

[1]Debabuche, Mahmoudi, Manneback, Bindelle, Lebeau (2018) Cloud services integration for farm animals’ behavior studies based on smartphones as activity sensors

## 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).*

## **3.1 Escriba un título para el primer problema**

Deberían mencionar el problema que resolvieron, el algoritmo que usaron, la métrica que obtuvieron y la cita de ACM.

## **3.2 Escriba un título para el segundo problema**

Deberían mencionar el problema que resolvieron, el algoritmo que usaron, la métrica que obtuvieron y la cita de ACM.

## **3.3 Escriba un título para el tercer problema**

Deberían mencionar el problema que resolvieron, el algoritmo que usaron, la métrica que obtuvieron y la cita de ACM.

## **3.4 Escriba un título para el cuarto problema**

Deberían mencionar el problema que resolvieron, el algoritmo que usaron, la métrica que obtuvieron y la cita de ACM.

## **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.

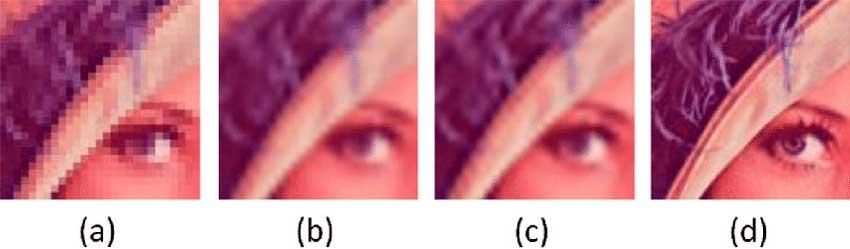
**3.2.1 talla de costuras**

En este algoritmo a cada valor de pixel de una amigan se le asigna un valor de energía y con programación dinámica y con una función se sacar el valor mínimo del pixel y así con cada pixel y los pixeles con un valor mas pequeño se eliminan de la original así cambia la imagen, pero la parte importante de la imagen se mantiene

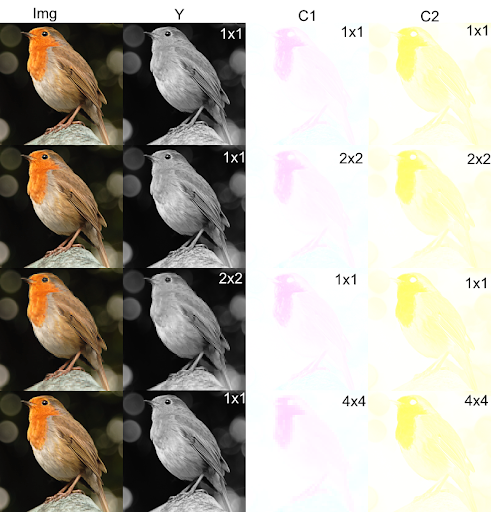


**3.2.2 Escalado de imágenes**

En este algoritmo cuando se trata de aumentar la resolución de una imagen se crean mas pixeles, pero estos no tienen color entonces copian el pixel de alrededor y algunas ocasiones se pueden



**3.2.3 algoritmo de compresión de JEPG**

Este algoritmo divide la imagen en dos canales de color y un de brillo para después quitar resolución a estos canales de resolución es decir pixelea la imagen ya que el ojo tiene más resolución de brillo. y quitan unos brillos que al final no se nota en la imagen final 

**3.2.4 Compresión fractal**

El método es el más apropiado para texturas e imágenes naturales, basándose 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» los cuales se usan para recrear la imagen codificada.



## **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 Compresión de Burrows-Wheeler**

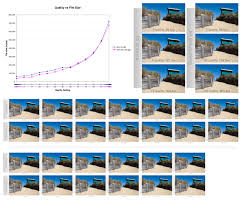
La transformación de Burrows-Wheeler ha demostrado ser fundamental para las aplicaciones de compresión de imágenes. Por ejemplo, mostró una tubería de compresión basada en la aplicación de la transformación Burrows-Wheeler seguida de codificadores de inversión, longitud de ejecución y aritmética.

**3.3.2 LZ77**

Este algoritmo el código analiza el texto en nuestro caso los archivos csv en busca de secuencias iguales en un buffer de anticipación que guarda la información y así poder comprimir algunas partes de las imágenes sin perder pixeles

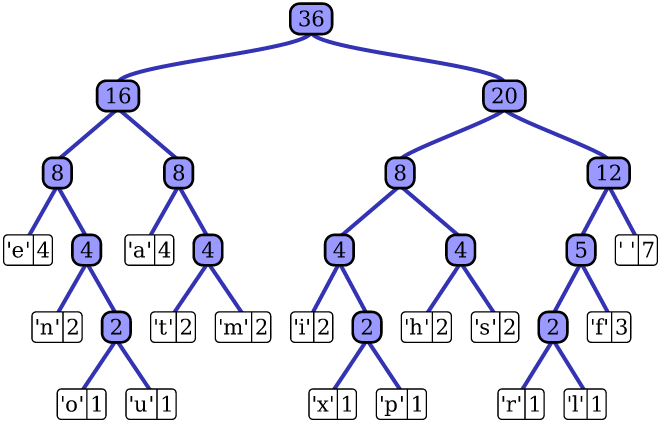
**3.3.3 LZW**

La diferencia de este algoritmo y los otros es que se puede crear mediante el uso entonces como una versión muy mejorada del LZ77 y LZ78 y si tenemos un diccionario que se le van añadiendo elementos aun así puede seguir trabajando comprimiendo estos



**3.3.4 Codificación de Huffman**

Par este algoritmo se usa una tabla de códigos de longitud de variables con la cual se pueden cambiar por una letra por ejemplo y en ese caso estamos comprimiendo una imagen sin que haya ningún tipo de perdida



## **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**

Programmerclick: algoritmo de talla de costura Obtenido de <https://programmerclick.com/article/4510360417/>

Carlos López , 2021 ¿Qué es la compresión LZW y cómo funciona? Obtenido de https://es.ccm.net/contents/731-compresion-lzw

Ecured Lz77 obetenido de <https://www.ecured.cu/Lz77>

Kripkit Burrows-Wheeler transformación Obtenido de

<https://kripkit.com/burrows-wheeler-transformacin/>

Wikipedia 2021 Compresión fractal Obtenido de <https://es.wikipedia.org/wiki/Compresi%C3%B3n_fractal>

Netinbag.com ¿Qué es el escalado de imagen? Obtenido de https://www.netinbag.com/es/internet/what-is-image-scaling.html

PensamientosComputables Como el algoritmo de compresión JPEG engaña al ojo humano

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