**Compresión y análisis de imágenes por medio de algoritmos.**

|  |  |  |  |
| --- | --- | --- | --- |
| David Agudelo Tapias  Universidad EAFIT  Colombia  dagudelot@eafit.edu.co |  | Simón Marín Universidad Eafit Colombia smaring1@eafit.edu.co | Mauricio Toro  Universidad Eafit  Colombia  mtorobe@eafit.edu.co |

# **RESUMEN**

El problema que se quiere llegar a solucionar en este proyecto de la materia de Estructuras de datos y algoritmos, es llegar a descifrar unas imágenes, las cuales tienen en ellas animales, siendo más específicos, animales vacunos; en las cuales hay que identificar si el animal es sano, o sea, si está en buenas condiciones para luego ser tenido en cuenta en el proceso de selección de la ganadería, o si está enfermo, para saber si se llega a descartar. Todo esto por medio de un algoritmo de compresión, el cual permite tomar las imágenes y llevarlas a una examinación de estas en el código, donde no siempre los resultados van a ser cien por ciento exactos, pero lo que permite a este código ser eficiente, es que se trabaja con aprendizaje de máquina, lo que quiere decir que mientras más información toma, más precisos van a ser los resultados sin traer consigo afectaciones en general.

Esta problemática es de gran impacto en la sociedad, ya que si en una gran empresa del extranjero, o del mismo país, no se tiene un buen control de la información, pueden llegar a haber pérdidas a futuro no solo en lo económico, sino también en las relaciones con las demás compañías.

# **1. INTRODUCCIÓN**

La motivación para resolver esta problemática planteada para el proyecto, es llegar a poder comprimir las imágenes de los animales bovinos, para hacer que el código funcione de la manera más eficiente posible y así permitirle saber a las personas que lo usen, qué salud animal tiene cada uno de estos y poder llevar un mejor control, preferiblemente más preciso, de sus ganados en la ganadería.

# **1.1. Problema**

El problema que se trabajará es poder identificar por medio de un código y unas imágenes, qué vacunos tiene una salud con el que se puede considerar sano o enfermo cada animal analizado. El impacto que esto genera es que permite que el tratado de estos animales en específico sea el más adecuado para los negocios, ya que al no usarlo o darle un mal uso, se están perdiendo oportunidades en las grandes compañías que trabajan en este sector agrícola*.*

**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 Localisation and Individual Identification of Holstein Friesian Cattle via Deep Learning.**

Existe una raza de animales vacunos la cual es Holstein Friesian, y esta es representativa para la ganadería por su gran producción de leche. El problema que querían solucionar era identificar por medio de una red de cámaras fijas, y también de vehículos aéreos no tripulados (UAV), el pelaje del animal para saber si era de esta raza tan demandada; estas cámaras tomaban fotos de 3840x2160 pixeles, de una forma rectangular y estando ubicadas a una altura de cinco metros. Usando Long Short-Term Memory se organizan los patrones en las pieles según los animales que haya en el lugar. Los resultados fueron: “Demostramos que la detección y localización del ganado frisón se puede realizar de forma robusta con una precisión del 99,3% en estos datos”, “También evaluamos la identificación a través de una tubería de procesamiento de video en 46,430 cuadros provenientes de 34 clips (aproximadamente 20 s de longitud cada uno) de imágenes de UAV tomadas durante el pastoreo (23 individuos, precisión = 98.1%)”.

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

Para este trabajo, se quería transferir, almacenar, tratar y compartir datos y para esto se usan dos modelos de iPhone, que son los modelos 4s y 5s, ya que los celulares contienen gran cantidad de sensores de todo tipo.

“Se ejemplifica una aplicación al estudio del comportamiento del ganado en pastos basada en los datos registrados con la IMU de iPhone 4s. También se logra una comparación de rendimiento entre el iPhone 4s y el iPhone 5s. El paquete también viene con una interfaz web para codificar el comportamiento real observado en los videos y sincronizar las observaciones con las señales del sensor. Finalmente, el uso de Edge computing en el iPhone redujo en un 43,5% en promedio el tamaño de los datos sin procesar al eliminar las redundancias. La limitación del número de dígitos en una variable individual puede reducir la redundancia de datos hasta en un 98,5%”

## **2.3 An Animal Welfare Platform for Extensive Livestock Production Systems**

El problema a solucionar es acoplarse a las nuevas reglas en la UE para demostrar que los bovinos están en las mejores condiciones, para al final censar que “problemas de salud, estado y bienestar de la granja”. La solución que plantearon fue usar un sensor inalámbrico “basado en algoritmos de reconocimiento de patrones de redes neuronales”, dando información de sus movimientos, velocidad y donde se encuentran. Los resultados fueron resultados de los patrones, que los agricultores usan para tomar las mejores medidas para sus animales.

## **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 .

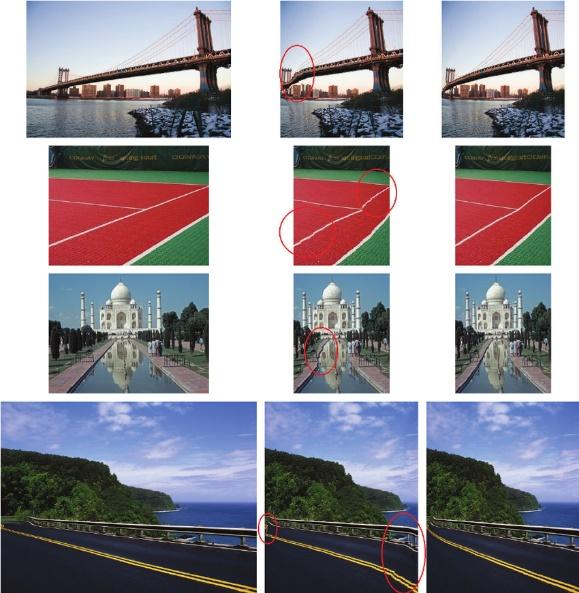
Por último, utilizando el conjunto de datos de entrenamiento, entrenamos una red neuronal convolucional para la clasificación binaria de imágenes utilizando *Teachable Machine* de Google disponible en https://teachablemachine.withgoogle.com/train/image.

## **3.2 Alternativas de compresión de imágenes con pérdida**

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

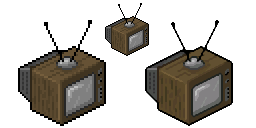
**3.2.1 Liquid Rescaling**

Es un algoritmo que permite que se agreguen o eliminen píxeles, si es que importan o no respectivamente, ampliando o reduciendo la imagen a analizar.



**3.2.2 Image Scaling**

Se trata de un algoritmo que “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”.



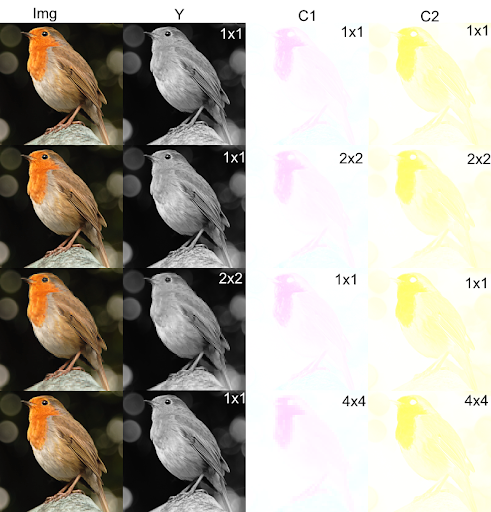
**3.2.3 Compresión Fractal**

Es un algoritmo el cual, por medio de patrones geométricos, la imagen que es analizada pasa a una mejor resolución, y además es óptimo al momento de analizar imágenes de la naturaleza, ya que hay fractales que se repiten.



**3.2.4 Compresión JPEG**

Este algoritmo usa los píxeles de una imagen y los agrupa, es poco usado ya que representa una gran pérdida de calidad en la imagen, después de este proceso, y al final de esto se le aplica el algoritmo de Huffman.

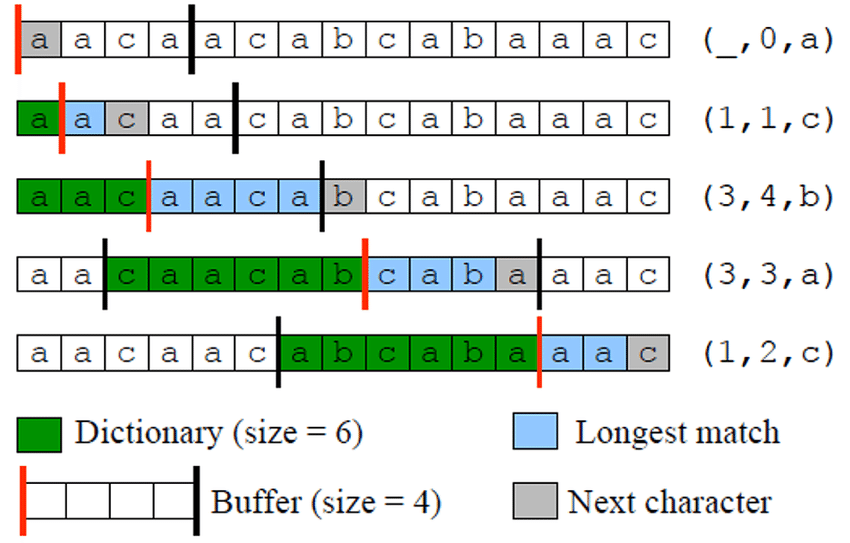


## **3.3 Alternativas de compresión de imágenes sin pérdida**

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

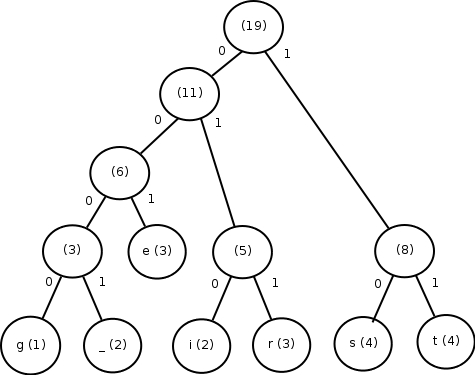
**3.3.1 LZ77**

Este algoritmo trata de un codificador de diccionario, donde LZ77 “codifica y decodifica desde una ventana deslizante sobre los caracteres vistos anteriormente” y así poder encontrar coincidencias. Su complejidad en memoria es de O(n), pero en el tiempo es más demorado comparado con algunos y su complejidad es O(n2).



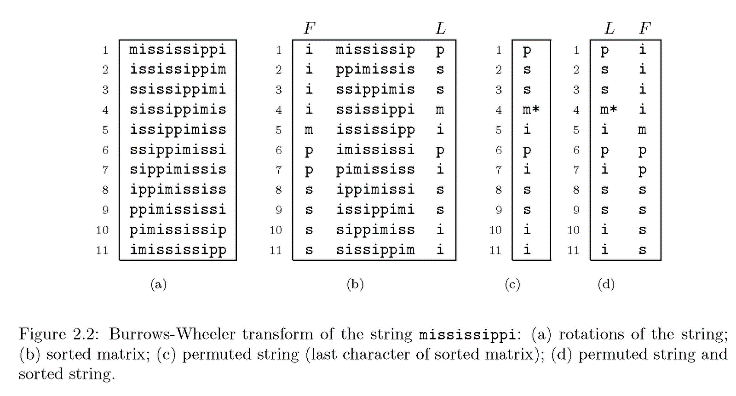
**3.3.2 Codificación Huffman**

Este tipo de algoritmo se trata de un árbol de nodos donde este “permite asignar a los diferentes símbolos a comprimir, un código binario”. Su complejidad en memoria es O(n) ya que es lineal, y en el tiempo tiene una complejidad de O(n\*Log(n)).



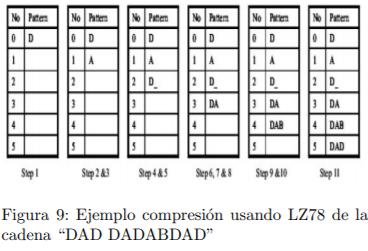
**3.3.3 Borrow-Wheeler transform**

“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” [2]. Su complejidad en memoria y en tiempo son de O(n).



**3.3.4 LZ78**

Este algoritmo es una versión mejorada del algoritmo LZ77, donde necesita un diccionario para así, comprimir los datos recibidos. Su complejidad es muy parecida a la de LZ77, “este tiene unas mejoras respecto al consumo de memoria, pero tarda más tiempo”.

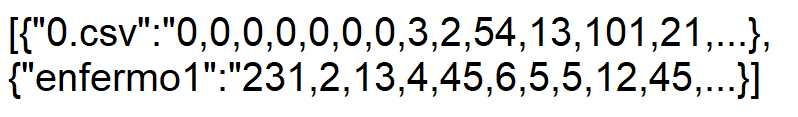


## **4. DISEÑO E IMPLEMENTACIÓN DE LOS ALGORITMOS**

## 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**

La estructura de datos que se usó es el diccionario, donde se guarda tanto el nombre como los valores de cada pixel de las imagenes archive .csv.

**Figura 1:**

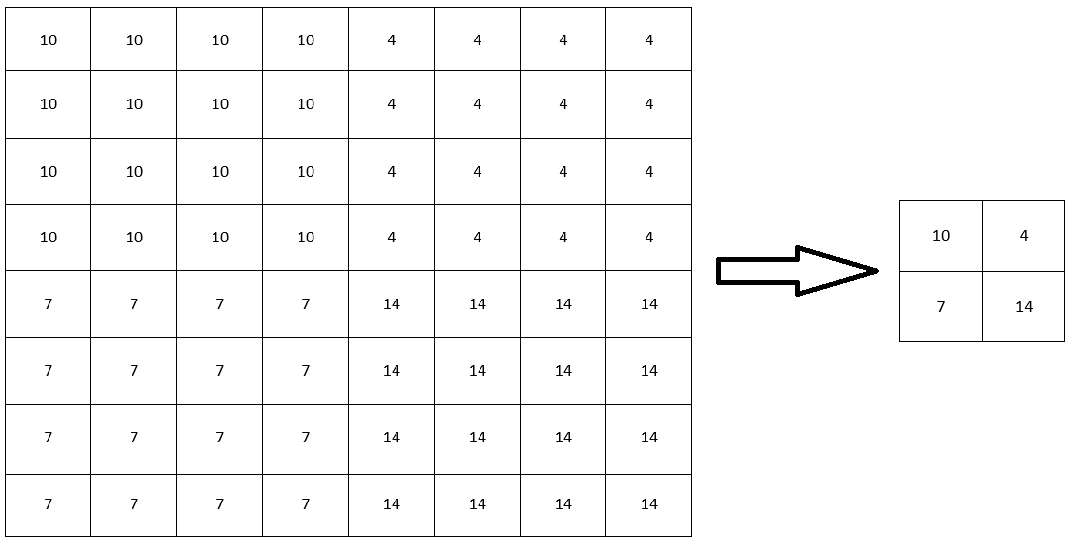
**4.2 Algoritmos**

En este trabajo, se propone un algoritmo de compresión que es de un algoritmo de compresión de imágenes con pérdida. También explicamos cómo funciona la descompresión para el algoritmo propuesto.

**4.2.1 Algoritmo de compresión de imágenes con pérdida**.

**“K-Nearest-Neighbor** es un algoritmo basado en instancia de tipo supervisado de Machine Learning. Puede usarse para clasificar nuevas muestras (valores discretos) o para predecir (regresión, valores continuos). Al ser un método sencillo, es ideal para introducirse en el mundo del aprendizaje Automático. Sirve esencialmente para clasificar valores buscando los puntos de datos “más similares” (por cercanía) aprendidos en la etapa de entrenamiento y haciendo conjeturas de nuevos puntos basado en esa clasificación.” [7]

“Es un método que simplemente busca en las observaciones más cercanas a la que se está tratando de predecir y clasifica el punto de interés basado en la mayoría de datos que le rodean” [7]



**Referencias.**

1). Wikipedia. *LZ77 and LZ78*. Wikipedia: la enciclopedia libre. Retrieved Agosto 14, 2021, from https://en.wikipedia.org/wiki/LZ77\_and\_LZ78

2). Castro, J. E., Vélez, C. G., & Mesa, J. E. *Algoritmos de compresión para la optimización del consumo de baterías en ganadería de precisión*. Retrieved August 15, 2021, from https://osf.io  
  
3). Fernández, S. *Código Huffman*. Retrieved August 14, 2021, from https://es.slideshare.net/mejiaff/cdigo-huffman

4). Debauche, O. *Cloud services integration for farm animals’ behavior studies based on smartphones as activity sensors*. Retrieved August 16, 2021, from https://link.springer.com/article/10.1007/s12652-018-0845-9

5).Andrew, W., Greatwood, C., & Burghardt, T. *Visual Localisation and Individual Identification of Holstein Friesian Cattle via Deep Learning*. Retrieved August 16, 2021, from https://openaccess.thecvf.com/content\_ICCV\_2017\_workshops/w41/html/Andrew\_Visual\_Localisation\_and\_ICCV\_2017\_paper.html

6). Doulgerakis, V., Kalyvas, D., Bocaj, E., Giannousis, C., Feidakis, M., Laliotis, G. P., Patrikakis, C., & Bizelis, I. *An Animal Welfare Platform for Extensive Livestock Production Systems*. ResearchGate. Retrieved August 16, 2021, from https://www.researchgate.net/publication/338595895\_An\_Animal\_Welfare\_Platform\_for\_Extensive\_Livestock\_Production\_Systems

# 7). Bagnato, J. I. Clasificar con K-Nearest-Neighbor ejemplo en Python. Retrieved October 10, 2021, from https://www.aprendemachinelearning.com/clasificar-con-k-nearest-neighbor-ejemplo-en-python/#:~:text=K%2DNearest%2DNeighbor%20es%20un,el%20mundo%20del%20Aprendizaje%20Autom%C3%A1tico.

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