# **COMPRESSION AND OPTIMIZATION IN THE PRECISION LIVESTOCK FARMING CONTEXT**

|  |  |  |
| --- | --- | --- |
| Julian Peña Ochoa  Universidad Eafit  Colombia  jpenao@eafit.edu.co | Simón Marín Universidad Eafit Colombia smaring1@eafit.edu.co | Mauricio Toro  Universidad Eafit  Colombia  mtorobe@eafit.edu.co |

# **ABSTRACT**

In the precision livestock farming context, we want to compress images without loss in the most effective way. To do this, we will use a compression algorithm.

This is important because helps efficiency and the animals health in the process. We want exact and precise data in the most efficient way possible, not something perfect but something enough.

## **Keywords**

|  |
| --- |
| Compression algorithms, machine learning, deep learning,   precision livestock farming, animal health. |

# **1. INTRODUCTION**

In traditional livestock raising animals are obtained products for human consumption. As a result of this, precision farming arose in which information and communication technologies are added to improve processes. But there is the problem of how to do the automatic digitization of all the useful data of a farm and how to make the system more efficient in terms of energy. This problem is important because when solving it there would be energy savings not only in farm equipment but also the energy consumption of the GDP battery is optimized for this, the total data of the farm must be systematized in a single file that belongs to it.

# **1.1. Problem**

The problem is that by not compressing an image well, the data recorded by the system is not known exactly, which can affect our health because various foods from animals can be of poor quality or in the worst cases could contract a disease and it would affect the health of the consumer but if we solve this problem, in addition to avoiding food infected with a disease or of poor quality, it would also lower costs in technological or basic instruments for the time of observation of each animal, in addition, they could lower a certain degree of contamination as it would not require much energy to shed any data from any animal

In this work, we used a convolutional neural network to classify animal health, in cattle, in the context of precision livestock farming (PLF). A common problem in PLF is that networking infrastructure is very limited, thus data compression is required.

# **1.2. Solution**

Basically, in the lossy-data compression we implemented image scaling using one algorithm. We choose nearest neighbor interpolation, this method simply determines the “nearest” neighbouring pixel and assumes the intensity value of it. The main reason we decided to implement it, is because this algorithm has an appropriate complexity to work with a huge amount of data (the time complexity for the worst case O(nxm)) and also because, you don’t lose much information when you compress the image.

**1.3 Article structure**

In what follows, in Section 2, we present related work to the problem. Later, in Section 3, we present the data sets and methods used in this research. In Section 4, we present the algorithm design. After, in Section 5, we present the results. Finally, in Section 6, we discuss the results, and we propose some future work directions.

**2. RELATED WORK**

## In what follows, we explain four related works on the domain of animal-health classification and image compression in the context of PLF.

## **3.1 Cloud services integration for farm animals’ behavior studies based on smartphones as an activity sensor**

The problem that they analyzed was the optimization that

the sensors need.  First because of the power consumption,

second the storage and the processing of large amounts of

data and third the matching of data with complementary

data.  They describe a new infrastructure which brings ben-

eats in storage, real-time processing, and abilities for large

scale data storage and analytics that allows to collect, store,

treat and share information between scientists.  In the case

of the compression, they performed it in two ways:  First

by eliminating redundancies and replacement of redundant

data by a time interval during which the value remains con-

stint was applied to preserve data integrity.  And second by

truncating data to 3, 4 and 5 decimal digits [11].

**3.2 Precision livestock farming for pigs**

Pig Cough Monitoring Respiratory problems are very common in pig herds, causing significant economic losses. Early treatment of problems is crucial for reducing the economic losses and the amount of antibiotics used in the process. Early warning for the problem allows earlier treatment, causing fewer animals to be infected and taking less time to cure the animals. The Pig Cough Monitor is a tool for the automated and continuous monitoring of coughs in a pig herd (compartment level). The Pig Cough Monitor can be used as an early warning tool, and it also demonstrates the effects of treatment and preventive measures (e.g., difference between different vaccines) (Finger et al., 2014; Genzow et al., 2014) [12].

**3.3 Visual Localization and Individual Identification of Holstein Friesian Cattle via Deep Learning**

They demonstrate that computer vision pipelines utilizing deep neural architectures are well-suited to perform automated cattle detection as well as individual identification in agriculturally relevant setups. They show that off-the-shelf networks can perform end-to-end identification of individuals in top-down still imagery acquired from fixed cameras.

After some test, they conclude that particularly when videoing small herds in uncluttered environments, an application of marker-less Friesian cattle identification is not only feasible using standard deep learning components, but it also appears robust enough to assist existing tagging methods [10].

**3.4 An Animal Welfare Platform for Extensive Livestock Production Systems**

This study presents a way of study the animal behavior, an automated system with a single type of wireless sensor able to record indicators of animal’s well-being (i.e., movement, speed, and geolocation information of the animal) with low implementation cost, based on Deep Neural Network pattern recognition algorithms. The solution also provides end-users (farmers) with usable and effective information visualizations, so that they take proper actions [11].

## **3. MATERIALS AND METHODS**

In this section, we explain how the data was collected and processed and, after different image-compression algorithm alternatives to solve improve animal-health classification.

## **3.1 Data Collection and Processing**

We collected data from Google Images and Bing Images divided into two groups: healthy cattle and sick cattle. For healthy cattle, the search string was “cow”. For sick cattle, the search string was “cow + sick”.

In the next step, both groups of images were transformed into grayscale using Python OpenCV and they were transformed into Comma Separated Values (CSV) files. It was found out that the datasets were balanced.

The dataset was divided into 70% for training and 30% for testing. Datasets are available at <https://github.com/mauriciotoro/ST0245-Eafit/tree/master/proyecto/dataset>s.

Finally, using the training data set, we trained a convolutional neural network for binary image-classification using Google Teachable Machine available at <https://teachablemachine.withgoogle.com/train/image>.

## **3.2 Lossy Image-compression alternatives**

**3.2.1 Fractal compression**

Fractal compression is a lossy compression method for digital images, based on fractals.  This method uses an algorithm to convert these parts into fractal codes which are used to recreate the encoded image [1].

Graphical user interface

Description automatically generated with low confidence

**3.2.2 Image scaling**

Image scaling refers to resizing or resampling a digital image age. Depending on the images, they can be scaled using geometric transformations or create a new one with a higher or lower number of pixels [2].

There exist three common scaling algorithms:

1. Nearest Neighbor Scaling.

2. Bilinear

3. Bicubic Interpolation

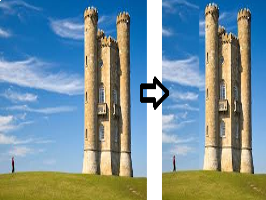
A picture containing application

Description automatically generated

**3.2.3 Seam carving**

**Seam carving** (or **liquid rescaling**) is an algorithm for content-aware [image resizing](https://en.wikipedia.org/wiki/Image_scaling). It functions by establishing a number of *seams* (paths of least importance) in an image and automatically removes seams to reduce image size or inserts seams to extend it. Seam carving also allows manually defining areas in which pixels may not be modified and features the ability to remove whole objects from photographs.

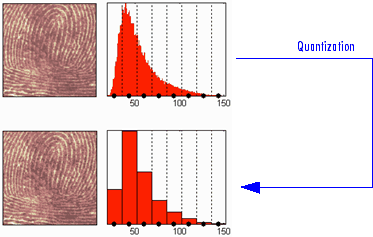
The purpose of the algorithm is image retargeting, which is the problem of displaying images without distortion on media of various sizes (cell phones, projection screens) using document standards, like HTML, that already support dynamic changes in page layout and text but not images. [8]



**3.2.4 Wavelet compression**

Wavelet compression is a form of [data compression](https://en.wikipedia.org/wiki/Data_compression" \o "Data compression) well suited for [image compression](https://en.wikipedia.org/wiki/Image_compression" \o "Image compression) (sometimes also [video compression](https://en.wikipedia.org/wiki/Video_compression" \o "Video compression) and [audio compression](https://en.wikipedia.org/wiki/Audio_compression_(data)" \o "Audio compression (data))). The goal is to store image data in as little space as possible in a [file](https://en.wikipedia.org/wiki/Computer_file" \o "Computer file). Wavelet compression can be either [lossless](https://en.wikipedia.org/wiki/Lossless_data_compression" \o "Lossless data compression) or [lossy](https://en.wikipedia.org/wiki/Lossy_data_compression). Wavelet coding is a variant of [discrete cosine transform](https://en.wikipedia.org/wiki/Discrete_cosine_transform" \o "Discrete cosine transform) (DCT) coding that uses wavelets instead of DCT's block-based algorithm.

Wavelet compression is not good for all kinds of data: transient signal characteristics mean good wavelet compression, while smooth, periodic signals are better compressed by other methods, particularly traditional harmonic compression (frequency domain, as by Fourier transforms and related) [9].



## **3.3 Lossless Image-compression alternatives**

## In what follows, we present different algorithms used to compress images.

**3.3.1 Huffman coding**

The Huffman coding is an algorithm to compress data, basically for compressing files. The idea of the algorithm is to assign variable-length codes to input characters considering which are used and unused to turn out the most optimal code lengths [3][4].

There are mainly two major parts in Huffman Coding:

1. Build

a Huffman Tree from input characters.

2. Traverse a Huffman Tree and assign codes

To characters.

Diagram

Description automatically generated

**3.3.2 Burrows-wheeler transformation**

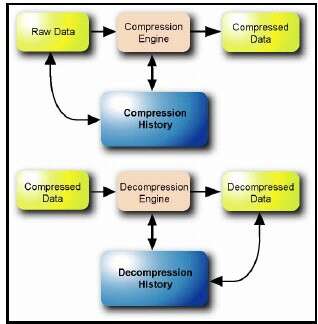
The burrows-Wheeler transformation rearranges a character string into similar character strings. This is useful for compression and decoding using algorithms, as the-move- to-front transform technique and run-length encoding make it easy to compress a string that has repeated character runs. The potential utility of the BWT of large amounts of shortread data (“reads”) has not been fully studied. The BWT basically serves as a dictionary of lossless reads. For example, in reading genomes, unlike the results of heuristic mapping and lossy reads that are conventionally obtained, the use of BWT is usually much more efficient. In the future, this is expected to lead to the development of sensitive methods for analyzing short-read data.

Graphical user interface, text, application, table

Description automatically generated

**3.3.3 LZS Compression**

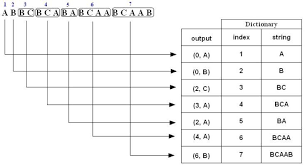
LZS compression and decompression uses an [LZ77](https://en.wikipedia.org/wiki/LZ77" \o "LZ77) type algorithm. It uses the last 2 KB of uncompressed data as a sliding-window dictionary. An LZS compressor looks for matches between the data to be compressed and the last 2 KB of data. If it finds a match, it encodes an offset/length reference to the dictionary. If no match is found, the next data byte is encoded as a "literal" byte. The compressed data stream ends with an end-marker [6].



**3.3.4 LZ77 and LZ78**

They are both theoretically [dictionary coders](https://en.wikipedia.org/wiki/Dictionary_coder" \o "Dictionary coder). LZ77 maintains a sliding window during compression. This was later shown to be equivalent to the *explicit dictionary* constructed by LZ78—however, they are only equivalent when the entire data is intended to be decompressed.

Since LZ77 encodes and decodes from a sliding window over previously seen characters, decompression must always start at the beginning of the input. Conceptually, LZ78 decompression could allow random access to the input if the entire dictionary were known in advance. However, in practice the dictionary is created during encoding and decoding by creating a new phrase whenever a token is output [7].



**4. ALGORITHM DESIGN AND IMPLEMENTATION**

In what follows, we explain the data structures and the algorithms used in this work. The implementations of the data structures and algorithms are available at GitHub

**4.1. Data structures**

The application of nearest neighbor interpolation with data bases is with Content-based image retrieval, also known as query by image content (QBIC) and content-based visual information retrieval (CBVIR), is the application of computer vision techniques to the image retrieval problem, that is, the problem of searching for digital images in large databases (see this survey for a scientific overview of the CBIR field). Content-based image retrieval is opposed to traditional concept-based approaches (see Concept-based image indexing).

"Content-based" means that the search analyzes the contents of the image rather than the metadata such as keywords, tags, or descriptions associated with the image. The term "content" in this context might refer to colors, shapes, textures, or any other information that can be derived from the image itself. CBIR is desirable because searches that rely purely on metadata are dependent on annotation quality and completeness.

Having humans manually annotate images by entering keywords or metadata in a large database can be time-consuming and may not capture the keywords desired to describe the image. The evaluation of the effectiveness of keyword image search is subjective and has not been well-defined. In the same regard, CBIR systems have similar challenges in defining success. "Keywords also limit the scope of queries to the set of predetermined criteria." and, "having been set up" are less reliable than using the content itself

<http://www.github.com/>AlterBiscuit/ST0245-001/

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza baja

**Figure 1:** Nearest general scheme of content-based image retrieval Nearest

**4.2 Algorithms**

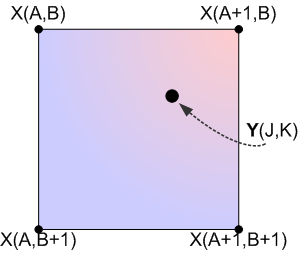
In this work, we propose a compression algorithm which is a combination of a lossy image-compression algorithm and a lossless image-compression algorithm. We also explain how decompression for the proposed algorithm works.

We have implemented nearest neighbor interpolation transform merged with Huffman coding. Firstly, we applied the nearest neighbor interpolation transform and it returns a csv file, which is the input to apply the compression with Huffman coding. Due to when we applied the Huffman compression it gives us a binary file, we have to apply the Huffman decompression. Finally to get the size of the original image, we applied the nearest neighbor interpolation. In the following section the two algorithms selected are explained deeper.

**4.2.1 Lossy image-compression algorithm**

**Nearest neighbor interpolation**

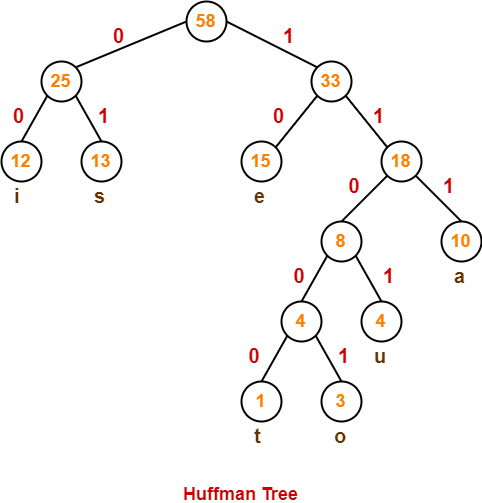
nearest neighbor interpolation is the simplest approach to interpolation. Rather than calculate an average value by some weighting criteria or generate an intermediate value based on complicated rules, this method simply determines the “nearest” neighbouring pixel, and assumes the intensity value of it.



**4.2.2 Lossless image-compression algorithm**

**Huffman coding**

Huffman process is a lossless compression algorithm that works bye analyzing the frequencies that certain symbols appear in a message. Those symbols that appear more often will be encoded as a shorter-bit string while symbols that aren't used as much will be encoded as longer strings. Basically, we created a dictionary where we store the frequencies of each data and we created a priority queue with a minheap. Later, we implement a binary tree where the nodes store the dictionary, and sequentially it merges the pairs of nodes which sum the less frequencies, creating an intermediate node which stores that sum. Finally, it is as-signed the respective code for each symbol, each code is the sequence from the root until each character following the labels of the Edge



# **REFERENCES**

1. Wikipedia contributors. (2021, 18 January). Fractal

compression. Wikipedia.

<https://en.wikipedia.org/wiki/Fractalcompression>.

2. Tabora, V. JPEG Image Scaling Algorithms, 2019.

From:

<https://medium.com/hd-pro/jpeg-image-scaling-algorithms-913987c9d588>

3. Huffman coding compression algorithm. From:

<https://www.techiedelight.com/huffman-coding>.

4. Huffman coding. 2021. From:

<https://www.geeksforgeeks.org/huffman-codinggreedy-algo-3>

5. Steven L. Brunton,J. Nathan Kutz (2019) Data-DrivenScience and Engineering:  Machine Learning, DynamicalSystems, and Control.

6. Wikipedia contributors. (2021, 14 August). Lempel-Ziv-Stac. Wikipedia.

https://en.wikipedia.org/wiki/Lempel%E2%80%93Ziv%E2%80%93Stac

7. Wikipedia contributors. (2021, 1 August). LZ77 and LZ78. Wikipedia.

<https://en.wikipedia.org/wiki/LZ77_and_LZ78>

8. Wikipedia contributors. (2021, 5 July). Seam carving.

https://en.wikipedia.org/wiki/Seam\_carving

9. Wikipedia contributors. (2021, 9 August). Wavelet transform.

<https://en.wikipedia.org/wiki/Wavelet_transform>

10. William Andrew, Colin Greatwood, Tilo Burghardt; Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2850-2859

11. Debauche, O., Mahmoudi, S., Andriamandroso,

A.L.H., Manneback, P., Bindelle, J., Lebeau, F., 2019.

Cloud services integration for farm animals' behavior

studies based on smartphones as activity sensors. J.

Ambient Intell. Humanized Comput. 10 (12), 4651-4662

11. Andrew, W., Greatwood, C., Burghardt, T., 2017. Visual localization and individual identification of holstein friesian cattle via deep learning. In: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 2850–2859.

12. Genzow, M., C.O. Duran, K. Strutzberg-Minder, G. Finger, and M. Hemeryck. 2014. Monitoring of a commercial fattening herd by means of the Pig Cough Monitor and oral fluid diagnostics. In: Proceedings of the 23rd IPVS Congress, Cancun, Mexico. p. 205– 206.

13. Nearest neighbor interpolation coding compression algorithm. From:

<https://www.giassa.net/?page_id=207>

14. Wikipedia contributors. (2021, 11 October) Content-based image retrieval

https://en.wikipedia.org/wiki/Content-based\_image\_retrieval

15. Huffman coding image. From:

https://www.gatevidyalay.com/huffman-coding-huffman-encoding/