Compression algorithms and their functionality for animal health.

Valentina Ochoa Universidad Eafit Colombia vochoaa@eafit.edu.co

Katherine Benjumea Universidad Eafit Colombia kbenjumeao@eafit.edu.co

Simón Marín Universidad Eafit Colombia smaring1@eafit.edu.co

Mauricio Toro Universidad Eafit Colombia mtorobe@eafit.edu.co

ABSTRACT.

It is necessary to optimize energy consumption and facilitate data storage in precision agriculture to generate a more profitable process for everyone, especially for farmers. As is known, livestock is part of the diet of many people, so it is important to classify animals suitable and healthy for human consumption. It seeks to reduce the levels of diseases transmitted by sick animals, in addition to generating income at low cost. It also seeks to promote better performance in production. The algorithms used were the LZ77 and RLE of these two algorithms, results were obtained in which it is shown that the LZ77 algorithm is more efficient in execution time than the RLE algorithm. It can be concluded that for future implementations it is recommended to use a lossy algorithm and with the results obtained from this algorithm, perform a lossless compression to be more efficient and thus have the combination of the two algorithms.

Keywords.

Compression algorithms, machine learning, deep learning, precision agriculture, animal health.

1. INTRODUCTION.

Cattle throughout history have been part of our diet, contributing an important part of protein to our diet. For this reason, it seeks to generate an acceleration in the process of classification of animals, thus achieving that farmers can optimize resources and produce good quality products, suitable for all. It is also expected to reduce execution time and memory storage.

1.1. problem.

It seeks to develop an algorithm with which images can be compressed and decompressed to achieve a more compact data storage and faster to process to reduce energy consumption and guarantee the health of the animals to be consumed and thus generate sustainable livestock.

1.2 Solution.

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.

The solution that has been developed for this problem is to implement an algorithm in which a lossy compression method and a lossless compression method interact, to determine which is the best result.

The lossy compression algorithm to be implemented for this problem is image scaling. This algorithm seeks to reduce the space occupied in memory, thus making this solution much more manageable, optimal and adequate for the process of passing it through the neural network for the identification of cattle.

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.

2.1 Integration of cloud services for smartphone-based behavioral studies of farm animals as activity sensors.

The problem that was solved in this article was the study of animal behavior using smartphones as it requires the storage of different variables and individuals, so transferring, storing, and treating this amount of data is the great challenge for precision agriculture. We want to solve this difficulty by using a coupled lambda cloud architecture for a scientific exchange platform for archiving and processing data.

An open-source observer-based classification algorithm is used in iPhone IMU that can be used and shared by researchers.

With the data stored in the cloud, the development and validation of new models will be allowed and thus provide new research opportunities for animal feeding, behavior, and health, with this it is expected to detect diseases or changes in animals in time before they become a bigger problem. The metric that was obtained was that smartphones used as sensors gave similar results in terms of measurements. When doing the process of compressing the data there are different rates; for acceleration and position data it is obtained that they are weakly compressible and for speed and altitude they are highly compressible. For other data, the same compression rate of 24.13% is obtained since they are linked data.

2.2 design of an electronic identification system and its potential use in the traceability of beef.

The problem that was solved in the article is the incorrect identification of animals for livestock activity since it hinders the organization of reliable records for the monitoring of health events and animal production. In traditional livestock, fire marks are used that cause accidents and significant damage to the skins, and these practices are losing strength in the face of new animal welfare policies. The need for reliable identification systems is created because of world politics, created by the problems caused by a neurodegenerative disease (BSE), found in cattle and transmissible to humans. To solve the problem, a device is created that has three components: an RDFID reader, the tags (label) and the WEB-oriented information system. Ceramic encapsulated Tags that were induced by a probe were used. The reader emits a magnetic field necessary for the hosted tag to transmit the identification code and send it to the information system that was developed with an incremental model supported by the graphical tool for structured diagrams DIA. All the information was stored in tables with which the system will be in charge of working through the users, favoring control through defined roles and passwords. As a result, it is given that the reader was sufficient to achieve the identification of the animals, but it can be improved for future prototypes, although the device was only tested for livestock, it is considered that it is not limited in its storage capacity and could be included in other areas of agriculture.

2.3 A systematic review of the literature on the use of machine learning in precision farming.

This article seeks to select and analyze recent works on the use of machine learning (ML) in precision farming (PLF), focusing on two areas of interest: grazing and animal health. A systematic review of the literature is made of AA techniques used in the context of LP, as well as of the different forms of data acquisition to train ML models. Research questions were posed, search strategies, selection criteria, inclusion and exclusion of documents are explained, the quality evaluation is detailed, and a summary of the selection process is detailed. To solve this, they used Boolean search equations that combine the keywords that could answer each research question. The sources were databases (Google Scholar, IEEE Xplore, Scopus, and Springer). Then they apply some questions that allow evaluating the quality of the documents. At the end of the selection, 35 documents were collected to analyze. The results are divided into two parts. First, the results regarding AA for grazing, in the context of PLF, and, later, the results on ML for animal health in the context of PLF. Regarding how ML has been used to improve grazing in PLF, 15 documents were selected, of which 12 are in magazines and 3 in conferences. Regarding how the SL has been used to improve health in the PLF domain, 20 documents were selected, of which 11 are in journals, 8 in conferences and 1 in literature review.

2.4 An animal welfare platform for extensive livestock production systems

The problem that was resolved was the growing need to raise indicators that support animal welfare during the processes that involve the production of food for humans derived from animal protein, for this purpose, all the behaviors evidenced in the study were stimulated and evaluated in a more specific way. the life span of the animals used during a certain production time. Sensors and receivers (IR temperature, humidity and pressure sensors, accelerometer, gyroscope, magnetometer) were used which transmit all the processes and results to a mega computer that automatically files them in the cloud. With all these studies, the researchers found that the level of heat stress (HS) is inversely proportional to dairy production while maintaining the nutritional intake at the same levels, something that increases the cost of production for the farmer. By being able to monitor the characteristics and other vital signs of farm animals, they found valuable information about their health and activity level, and thus managed to protect them from diseases and provide useful details and metrics in terms of agricultural management. The data from the collected sensors was processed in the cloud to evaluate the efficiency of the cattle in different actions: The performance was reduced because analyzing these scenarios requires the exploration of an adequate machine learning with algorithms capable of handling this complexity. A mobile application was used that provides visualizations of the animals' habits. These provide farmers with clear information to make adjustments for their growth.

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

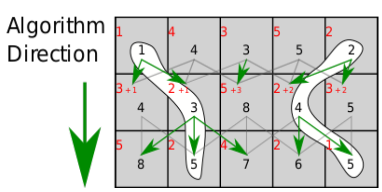
https://github.com/mauriciotoro/ST0245-Eafit/tree/master/project/datasets.
Finally, using the training data set, we trained a convolutional neural network for binary image-classification using Google Teachable Machine available at https://teachablemachine.withgoogle.com/train/image

3.2 Lossy Image-compression Alternatives.

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

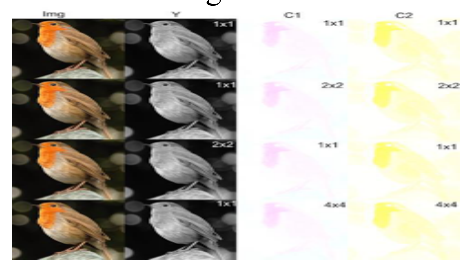
3.2.1 Carving seams.

It is an algorithm based on image resizing and its operation is by establishing a series of unions in an image that can automatically eliminate unions to reduce the size of the image and in turn can insert unions to extend it. This algorithm also allows you to manually define areas that cannot be modified by pixels. The seams can be vertical or horizontal: Vertical seams allow you to establish a path of connected pixels from top to bottom; horizontal stitching works in a similar way with the difference that the connection is from left to right.



3.2.2 JPEG compression algorithm.

This algorithm is a lossy compression algorithm, this indicates that it loses information and in turn its image quality the algorithm works as follows: an image is split where each pixel or point of the image is made up of an intensity of red, another of green and another of blue. So, the first thing the algorithm does is transform the way the image is rendered to the original three red, green, and blue color channels, then pixelate the image. It is done by assigning the same color to each block, ensuring that it is as close as possible to the original colors of the image.

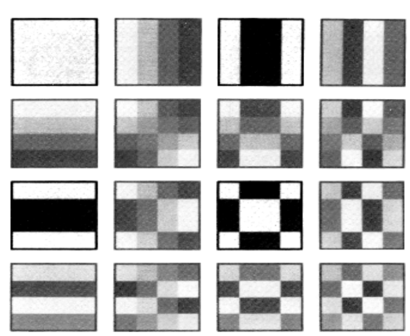


3.2.3 Fractal compression.

It is a lossy compression method for digital images, based on fractals. The method is best suited for natural images and textures, since parts of an image often resemble other parts of the same image. Fractal algorithms convert these parts into mathematical data called "fractal codes" which are used to recreate the encoded image.

3.2.4 Discrete cosine transform.

The image is divided into 8 x 8-pixel blocks to better apply information redundancy. The process and coding are carried out with the main formula of the transform. The image being a two-dimensional signal, the transformed equation must be adapted as a two-dimensional equation. Bearing in mind that the limit of the sum is the length of each block.

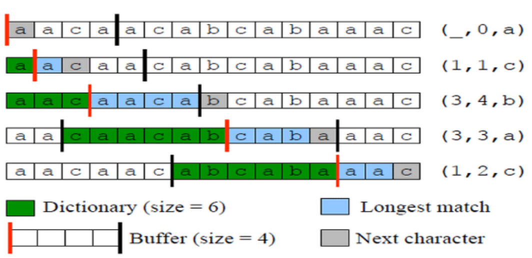


3.3 Lossless Image-compression alternatives.

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

3.3.1 Lz77 algorithm.

Lz77 is a dictionary-based compression model for text compression. It is widely used because it is easy to implement and quite efficient. The output always consisted of offsets or shifts and sizes of the previously viewed text. The next byte after a match was also outputting because the context (last seen bytes) of this byte is the phrase, and if it was not part of the phrase (the match), then it might not have been compressed. This algorithm uses a prefix, a bit that acts as a flag, like a two-state switch that lets us know what kind of data is coming next.



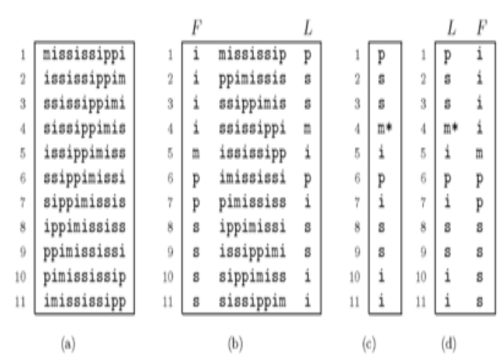
3.3.2 LZW.

The LZW compressor is a very fast compression / decompression system that is based on the multiplicity of characters in the string to be encoded. From the chain he created patterns that he integrated into a dictionary. The LZW works with bits and not bytes, which achieves great compatibility when processing data. This format is widely used in understanding TIFF or GIF images.

	Compressed Output	Dictionary	Buffer	Uncompressed Input
a)			0	100110101
b)	0	2(0,1)	0	100110101
c)	01	3(1,0)	1	00110101
d)	010	4(0,0)	0	0110101
e)	010		0	110101
f)	0102	5(0,1,1)	2	10101
g)	0102		1	0101
h)	01023	6(1,0,1)	3	101
i)	01023		1	01
j)	01023		3	1
k)	010236		6	

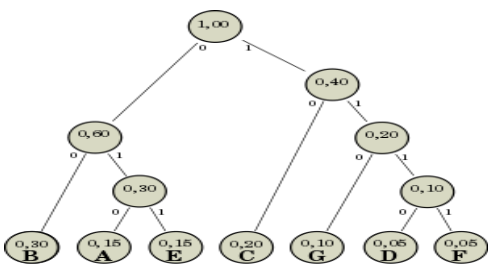
3.3.3 Burrows-Wheeler compression.

BWT also known as block order compression), is an algorithm used in data compression techniques such as bzip2. It is based on a transformation previously discovered by Wheeler that is not published. When a character string is transformed using the BWT, none of its characters change value. The transformation permutes the order of the characters. If the original string contains many substrings that occur often, then the transformed string will contain multiple positions where the same character is repeated multiple times in a row. This is useful for compression, as it tends to be easy to compress a string containing repeating sequences of characters with techniques such as move-to-front transform and run-length encoding.



3.3.4 Huffman algorithm.

It is an algorithm for the construction of Huffman codes. It takes an alphabet of n symbols, along with their associated occurrence frequencies, and produces a Huffman code for that alphabet and those frequencies. The algorithm consists of creating a binary tree that has each of the symbols per leaf and constructed in such a way that following it from the root to each of its leaves, the Huffman code associated with it is obtained.



4. DESIGN AND IMPLEMENTATION OF ALGORITHMS

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

Hash tables

Hash tables or also known as associative matrix are a data structure that, as its name indicates, associates' keys to specific values, it can also be called a type of dictionary.

Hash tables are usually implemented in dynamic arrays. These tables have a complexity in the worst case of O (n) and in these you can do the operations of insert, delete and search for elements.

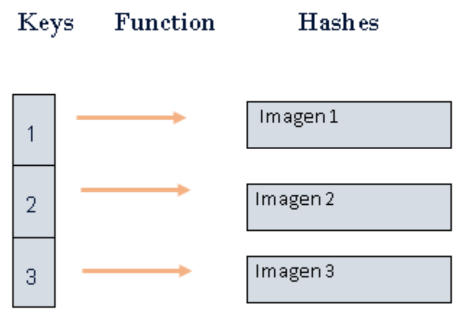


Figure 1: illustration hash tables

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.

4.2.1 Lossy image-compression algorithm

For this understanding, what was done was to pass the csv files to a Pandas data frame, then this became a numpy array, when doing this it is read as an image and it is compressed with the image scaling method, using the del function nearest neighbor. This allows the image to be transformed and can be saved in a new, smaller image. To decompress an image obtained, it is passed into an array to finally convert it into csv data.

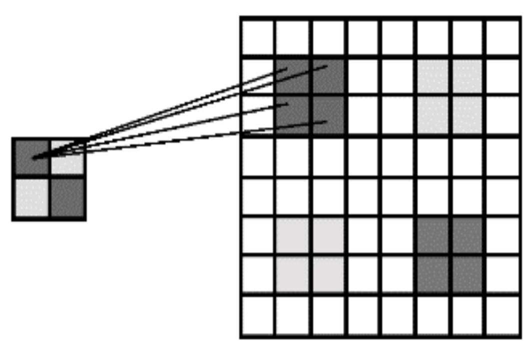


Figure 2: image scaling by nearest neighbor

4.2.2 Lossless image compression algorithm

For compression there is a repetition of consecutive elements, the algorithm first receives a csv file which is converted into text, since these algorithms work with characters. Then this text goes through the compression method that is used and is compressed.

At the end of this, the decompression is performed in which the original file is returned, which is verified by converting it into an image without loss. After the comma, the RLE returns the number of repetitions of a value and then the repeated value.

After the comma, the LZ77 is based on the multiplicity of the characters in the string, replaces the characters with an index code and progressively builds its dictionary.

4.3 Analysis of the complexity of the algorithms

To calculate the complexity of the algorithms, the properties of the O notation were used. According to the structure that is evidenced in each of the algorithms, the following results can be obtained.

For the lossless compression algorithms, we have:

LZ77 algorithm	Time complexity
Compression	O(N)
Decompression	O(N)

Table 1: *Temporal complexity of the image compression and decompression algorithms.*

RLE algorithm	Time complexity
Compression	O(N)
Decompression	O(N*M)

Table 2: *Temporal complexity of the image compression and decompression algorithms.*

LZ77 algorithm	Memory complexity
Compression	O(N)
Decompression	O(N)

Table 3: *Memory complexity of the image compression and decompression algorithms.*

RLE algorithm	Memory complexity
Compression	O(N)
Decompression	O(N)

Table 4: *Memory complexity of the image compression and decompression algorithms.*

For the lossy compression algorithm, we have:

Nearest neighbor algorithm	Time complexity
Compression	O(N)
Decompression	O(N)

Table 5: *Temporal complexity of the image compression and decompression algorithms.*

Nearest neighbor algorithm	Memory complexity
Compression	O(N)
Decompression	O(N)

Table 6: *Memory complexity of the image compression and decompression algorithms.*

4.4 Algorithm design criteria

The algorithm was designed in this way because it was desired to obtain a percentage compression rate that could be executed in a reasonable time, from this a strategy was sought in which the implementation of a code with these qualities was reflected.

5. RESULTS

5.1 compression ratio

LZ77 algorithm	compression ratio
Healthy cattle	2:1
Sick Cattle	2:1

Table 7: *Rounded average compression ratio of all healthy and diseased cattle images.*

RLE algorithm	compression ratio
Healthy cattle	2:1
Sick Cattle	2:1

Table 8: *Rounded average compression ratio of all healthy and diseased cattle images.*

5.2 execution time

Average execution time (seconds)	Average file size (GB)
146	280.337

Table 9: *LZ77 algorithm execution time.*

execution time(seconds)	Average file size (GB)
317	280.337

Table 10: *RLE algorithm execution time*

5.3 conclusion

It is concluded with the data set that the most efficient algorithm to solve this problem is the LZ77 due to the results it produces.

6 DISCUSSION OF THE RESULTS

With the results obtained, the lz77 algorithm is more efficient in terms of execution time for solving this problem, the compression applied in precision farming is effective since it helps a more optimal classification with less execution consumption.

6.1 Future work

For future work we want to implement a lossy compression algorithm, to the result that this yield applying a lossless compression to obtain a combination of algorithms that can significantly improve the compression rate and the execution time so that the solution reaches be more efficient and feasible to apply in this PLF context.

ACKNOWLEDGMENTS

The first author is supported by a Sapiencia scholarship, the second author is supported by My talent, my future scholarship. All the authors would like to thank Isabel Piedrahita Velez and Kevin Sossa Chavarria.

REFERENCES

1. García R, Agilar J, Toro M, Pinto A, y Rodríguez P. A systematic literature review int the use of machine learning in precision livestock fa rming .2020.

2. Doulgeraskis V, Kalyvas D, Bocaj E, Gia nnousis C, Feidakis M, Laliotis G, Patrikakis C y Bizelis I. An Animal Welfare Platform for Extensive Livestock Production Systems.2019.

3. Debauche O, Mahmoudi S, Hereriana A, Manneback P, Bindelle J y Lebeau F. Cloud services integration for farm animals 'behavior studies based on smartphones as activity sensors.2018

4. Reyes H, Vacca A y Gongora A. Desing an electronic identification system and its potential use inbeeftraceability.2010

5. Wikipedia. Seam carving. https://en.wikipedia.org/wiki/Seam_carving

6. Wikipedia. LZ77 and LZ78. https://en.wikipedia.org/wiki/LZ77_and_LZ78

7. Wikipedia.Fractal compression. https://es.wikipedia.org/wiki/Compresi%C3%B3n_fractal

8. Wikipedia. LZW. https://es.wikipedia.org/wiki/LZW

9. Wikipedia. Burrows–Wheeler transform. https://en.wikipedia.org/wiki/Burrows%E2%80%93Wheeler_transform

10. Wikipedia. Algoritmo de Huffman. https://es.wikipedia.org/wiki/Algoritmo_de_Huffman

11 Castro J. Analysis of image compression algorithms for implementation on mobile deviC