USE OF ALGORITHMS TO IDENTIFY ANIMAL HEALTH STATUS

Juan José Quintana Guzmán
Universidad Eafit
Colombia
jjquintang@esfit.edu.co

Dayana Arrieta Regino
UniversiGive Eafit
Colombia
dparrietar@eafit.edu.co

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

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

ABSTRACT
The problem is about the lack of information technologies in the livestock sector, since the counts of the cows are made manually. Therefore, there are things that already have to be done by the technology without affecting the work of the peasants. Besides, all the problem to change from manual work to a computerized work goes back in a problem associated, for example, the entrance to a gym that was with a security guard that could let you go in or not was a bit tedious, but now with the implementation of technology already can go in without the need of the surveillance of somebody and this is possible thanks to the electronic cards.

The algorithms that we used for image compression were Lz77 and RLE, which are without image loss, starting with Lz77, remember that this algorithm works in that we start passing the csv file to a text type, then those characters of the type text and goes backwards from that character position checking if there are equal items since the memory must be spending Bytes in these elements and thus save memory, on the other hand the RLE algorithm was very useful to save memory as well since it joins the similar characters in a row so as not to waste memory.

The results that we obtained from this project beyond the operation of the code were the different types of algorithms both with loss and without loss since, although we did not use all of them, we had the opportunity to know them in order to select the one we would use.

The conclusion we have for this work is the importance of algorithms without image loss, which can be reflected in other projects such as no data loss, which is essential to give a better data delivery when decompressing them later.

Keywords

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

1.INTRODUCTION

The manual work in some fields is becoming obsolete since the technology is taking big force and precision inside a lot of things included the mobility of data and information, therefore, this is giving more usability to the technology.

1.1 Problem

Lor that treats in this problem is delicate largely by the tall consumption so much of dairy as of meat, therefore, the good classification is paramount since if there is not good classification the product will be in bad state

1.2 Solution

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

For this solution we choose the algorithm of the nearest neighbor, this is an image scaling algorithm that effectively helps us to our image classification, which for our cattle problem is an important factor, even though it is an image loss algorithm, we believe It is the one indicated to do the compression and also the label of the 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 dates 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 Binary code that solve the compression lost dates

The loss of information was one of the problems related since compressing an information (image) lost a part of the image therefore they fixed this problem with the help of the binary code and space in the memory for like this can do that the information compress and decompress complete.

3.2 Helping hardware that helps in livestock farming

The welfare of the calves is something fundamental because it will be the livestock afterwards and has to be healthy for a good product, therefore, to the hand with the hardware invented a necklace to measure some parameters for like this have monitored the children essentially

3. MATERIALS AND METHODS

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

3.1 it Dates Collection and Processing

We collected Dates 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 Plough available at https://github.com/mauriciotoro/st0245-eafit/tree/master/proyecto/datasets.

Finally, using the training dates set, we trained to 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 Seam carving

It´s an algorithm for the change of image resizing it works by establishing a series of unions and then delete or insert more unions. Another function of this algorithm is manually defined areas where pixels can´t be modified and features the ability to remove entire objects.

3.2.2 Image scaling

Nearest neighbor interpolation is one of the most important, because it consist in select pixels nearest to the exit to scale up. So, it’s so usual to see this algorithm in the compression of images.

3.2.3 Discrete cosine transform.

This one is a lossy image compression algorithm because is so heavy for the memory and it’s too much information, so the user loses a small number of bits but at the end of the compression there is some information that is lost.

3.2.4 fractal compression

This algorithm is more functional in cases where the image needs to be compressed are related to nature because there are parts of the image that resemble each other; therefore, it is the most used in this type of images because it converts these parts of the image in fractal codes

3.3 Lossless Image-compression alternatives

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

3.3.1 Borrows & Wheeler Transform

Also known as block sort compression, when a character string is transformed using the BWT (Burrows-Wheeler Transformation), 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.

3.3.2 LZ77

In this algorithm, the encoder analyzes the text as a sequence of characters, through a sliding window made up of two parts; a look-ahead buffer, which is the option that is about to be encoded, and a search buffer (the window itself), which is the part where sequences equal to those in the look-ahead buffer are searched.

3.3.3 Huffman coding

Huffman encoding is an algorithm used for data compression. The term refers to the use of a variable length code table to encode a certain symbol, where the table has been filled in a specific way based on the estimated probability of occurrence of each possible value of said symbol.

3.3.4 LZ78 and LZS

LZ78 has high space requirements because the dictionary can occupy all free memory. There are several ways to avoid this problem. If we run out of memory, we can freeze the dictionary, or delete the entire dictionary and start creating a new one. There are other methods as well.

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[1].

4.1 Data Structures.

The data structure that we used was one implemented in python called numpy arrays, which basically consists of a system of vectors or arrays which have positions that can be used effectively. therefore, it helps us a lot for the problem to be solved.

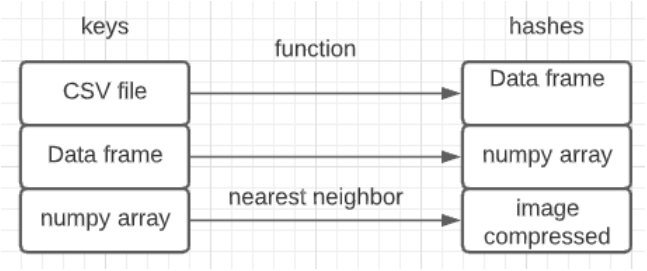

Figure 1: Data structure made in hash tables showing the compression processes of the problem.

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.

The first thing is to pass the csv files to a pandas dataframe, then that dataframe is used to convert it into a numpy array, when converting it into a numpy array it is read as an image and this image is compressed with the image scaling method, with the function from the nearest neighbor, when doing this, the images are resized by a smaller one, to complete the compression process and it is saved in a new image.

Figure 2: The algorithm is composed of an interpolation of a scaled image in which the nearest neighbor is used.

[1]http://www.github.com/dparrietar/ST0245-002/tree/master/proyecto

4.2.2 Lossless image-compression algorithm

In the process of decompression of images, it will be the opposite of the compression process, that is, when you have the compressed image, it passes it to a numpy array, then to the dataframe and finally to the csv file.

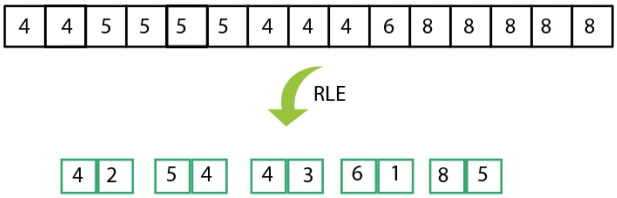

Figure 3: This is the design of the RLE algorithm.

4.3 Complexity analysis of the algorithms

Algorithm	Time Complexity
Compression	O(N)
Decompression	O(N)

Table 2: Time Complexity of the image-compression and image-decompression algorithms of LZ77.

Algorithm	Memory Complexity
Compression	O(N)
Decompression	O(N*M)

Table 3: Time Complexity of the image-compression and image-decompression algorithms of RLE.

4.4 Design criteria of the algorithm

In this part we will explain why we develop the algorithms in this way, so let's start with the compression, we did it this way since both algorithms offered us savings in memory and in lines of code, that is, for example, for the Lz77 when passed to a text file and taking the position of a character and checking from this position if there are characters that are repeated since these characters are already occupying space in memory, therefore, here we note their savings. In addition, RLE also works by saving memory as it groups repeated and followed characters.

Now, we will look at why we did not choose other algorithms that were available for our compression, starting

with Huffman coding, this algorithm is a bit similar to RLE, only that when the character is repeated, one is added to its frequency, it is also useful per the at the time of application, we found that RLE was simpler and saved us more memory. Now for delivery two that asked us for one with image loss, we chose the image scaling because when comparing it with the others which we saw, it seemed easier to apply and for example the seam carving what it did was create unions or erase them in order to do so. do the compression process, but the image scaling gave us more effectiveness when it came to compression.

5. RESULTS.

5.1 Execution times

In what follows we explain the relation of the average execution time and average file size of the images in the data set, in Table 6.

Average execution time(s)	**Average File size (kb)**
146	280.337

Table 4. LZ77 execution time

Average execution time(s)	**Average File size (kb)**
317	280.337

Table 5. RLE execution time

5.2 Compression ratio

We present the average compression ratio of the compression algorithm in Table 8.

LZ77	**Healthy cattle**	**Sick Cattle**
Average Compression ratio	2:1	2:1

Table 7. Compression ratio of LZ77.

RLE	**Healthy cattle**	**Sick Cattle**
Average Compression ratio	2:1	2:1

Table 8. Compression ratio of RLE.

6. DISCUSSION OF THE RESULTS

6.1 Future Works

It would be very interesting to use the discrete cosine transform since we would apply it to a matrix and that each position of the matrix is a pixel and when compressed, the pixels are brought together using cosines. In the future we would like to implement in database methods that when the data is compressed it does not have any type of loss, this would come together with the decompression of the data, although it is already very similar to how it is established since it is with images, but it would be interesting with a database. As for how we would like to improve our project, it would be the speed of the execution time and improvement of its complexity.

ACKNOWLEDGEMENTS

We would like to Isabel Piedrahita and Kevin Sossa who helped us with some doubts we had. This work was made with the support of “Generación-E(excelencia)” scholarship and “Hijo del empleado” scholarship.

REFERENCES

1. Fox Pamela. Compression of images without loss. Khan Academy: https://es.khanacademy.org/computing/ap-computer-science-principles/x2d2f703b37b450a3:Digital-information/x2d2f703b37b450a3:lossless-date-compression/to/simple-image-compression

2. Doulgerakis Vasileios. An Animal welfare platform for extensive livestock production System. University of west Attica

3.Ecured.LZ77:https://www.ecured.cu/Lz77#Funcionamiento