DATA STRUCTURES FOR IMAGE ANALYSIS AND COMPRESSION

Daniel Vélez
Eafit University
Colombia
dvelezd2@eafit.edu.co

Juan José Villada
Eafit University
Colombia
jjvilladac@eafit.edu.co

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

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

SUMMARY:

The idea with this report is to analyze and mainly solve the problem were currently having, which is compressing images in an efficient way in order to determine the health condition of a defined cattle.

What we're looking for with this project is to develop an algorithm capable of identifying, trough a data structure, whether the cattle are in optimal health conditions, or are sick. The fact of providing a solution to this problem would become into a huge step, due to the huge technology lack in the cattle raising industry, which is making the study of this area to be way more difficult than it should.

Key words:

Compression algorithms, machine learning, deep learning, precision cattle raising, animals' health condition.

1. INTRODUCTION:

Nowadays, compression algorithms are taking a very important role inside the precision cattle raising, because of this there are many systems/apps/programs that help people know the possible health conditions their cattle can be in.

This is why were going to use compression algorithms, because they're designed to represent certain amount of information while also using fewer space, and then, after being decompressed, it won't lose information in the process, making it very efficient in the precision cattle raising world, because it helps their study in order to collect data in a more general way.

1.1. Problem:

The problem we're now facing is basically to create, through a data structure, a CSV images network where, using these images, and in an efficient way, we can find between the animals, which one is not in the best health conditions.

It is really important to give a proper solution to this problem, because when developing these types of programs, there will be lots of people, inside de precision cattle raising, such as farmers or even whole business, being benefited from it.

1.2. Solution:

In this project, we will be using a convolutional neural network to classify animal health, in cattle, in the context of precision farming (GdP). A common problem in GoP is that the network infrastructure is very limited, so data compression is required.

1.3 Article structure:

In what's next, in Section 2, we present papers related to the problem. Later in Section 3, we present the data sets and methods used in this investigation. In Section 4, we present the design of the algorithm. Then, in Section 5, we present the results. Finally, in Section 6, we discuss the results and propose some future directions of work.

2. TRABAJOS RELACIONADOS:

Now, we will explain four related projects, in the domain of animal health classification and data compression, in the context of PLF.

2.1 Cálculo automatizado del vector eléctrico integral de la actividad ventricular cardíaca en equinos:

The problem that was solved here was the development of an algorithm that found the automated determination of the position of the middle VEI of the QRS complex in equines.
In the algorithm, a pseudo-code in Java was used initially, which had as main data related to the variations in the ventricles of each animal studied and all the previous data were taken to the final result in order to calculate the efficiency and veracity of the program.
To find the result, they first made manual calculations about the position of the electrical axis and then inserted the same problem in the program to compare and finally determine that the two data coincided

2.2 Caracterización de variables utilizando inteligencia computacional para identificar alteraciones en la salud:
The problem that arises in this case is to develop a program to find alterations in the health of cattle, determining FP-GROWTH as the best algorithm for solving the problem.
The conclusions that were obtained in the study is that first the number of variables present is very extensive, making the investigation with each data very meticulous and also the FP-GROWTH algorithm identifies patterns of irregularity through homagram tests.

2.3 DESARROLLO DE UNA BASE DE DATOS CON IMÁGENES TERMOGRÁFICAS PARA USO EN ALGORITMOS DE VISIÓN E INTELIGENCIA ARTIFICIAL:

What is presented as a problem is that a study is being carried out to develop a program that, through artificial intelligence, can identify thermal indicators in order to visualize reality from different ranges of the electromagnetic spectrum.

What was concluded is that the number of pregnant images is not so far removed from the number of non-pregnant sheep images, as much as digital images in the visible spectrum of thermograms.

2.4 Desarrollo e implementación de un dispositivo de adquisición y almacenamiento de sonidos para ganadería de precisión:
The problem here is that we want to find the implementation of a signal acquisition and storage device for the monitoring of feeding activities in cattle that can capture variables such as sound during feeding without disturbing the animal.

As a solution, the program was successful, but even so they decided to implement an improvement in terms of data management, the measure they took was that they divided the tasks according to the respective computational load.

3. MATERIALS AND METHODS:
In this section, we explain how the data was collected and processed, and then different alternative image compression algorithms to improve the classification of animal health.

3.1 Data collecting 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 sets of images were grayscale using Python OpenCV and transformed into comma separated value (CSV) files, data sets were balanced.

The data set was divided into 70% for training and 30% for testing. The data sets are available at https://github.com/mauriciotoro/ST0245-Eafit/tree/master/proyecto/datasets.

Lastly, using the training dataset, we trained a convolutional neural network for binary image classification using Google's 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 lossy images.

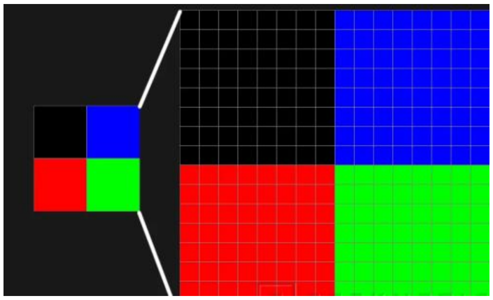
3.2.1 Algoritmo de Talla de Costura:

Simply put, each pixel value is assigned an energy value, and based on that pixel value, eight connected domains are used in dynamic programming to get the smallest value. Then apply this row-by-row or column-by-column algorithm to: A power line is actually a phase. Rows of pixels with the smallest pixel value in two adjacent rows (columns) will be removed from the original image. The number of these rows that can be removed depends on the scale you are cutting. The above energy is actually the same as the pixel gradient.

$$e_1(\mathbf{I}) = |\frac{\partial}{\partial x}\mathbf{I}| + |\frac{\partial}{\partial y}\mathbf{I}|$$

3.2.2 Integer Scalling:

Integer scaling, defined by Intel, is a scaling technique that simply scales existing pixels using a multiplier such as an integer, a number that can be described without having a decimal component. For example, 1 or 3 are whole numbers that can be used with this scale, but negative or decimal numbers are not completely valid. This resizing uses a "near neighbor" interpolation algorithm, also known as near interpolation, to resize the image from the original information and resize the image based on taste-related colors. The original pixel is closest to the target pixel.

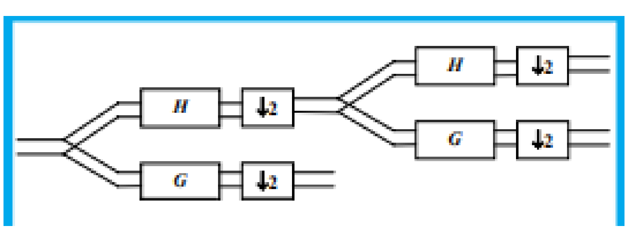


3.2.3 Compresión fractal:

This algorithm is based on that it analyzes the image and looks for patterns in which there is similarity and depending on this it classifies them into fractals, which in the end will be converted into mathematical data as functions and thus recreate the total image.

3.2.4 Compresión por wavelets:

This compression method begins by using a wavelet transformed to a test image, this is correlated with the pixel values in the original image, thus bringing this information to relatively small coefficients.



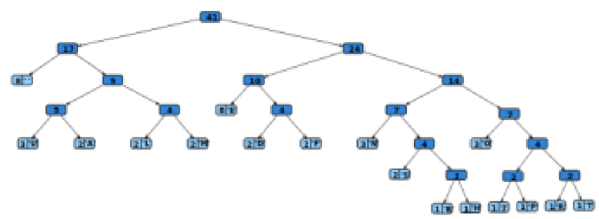
3.3 Lossless Image Compression Alternatives:

In what follows, we present different algorithms used to compress images without loss.

3.3.1 Codificación Huffman:

This type of encoding refers to encoding a specific symbol using variable length encoding, compiled in a specific way based on the estimated probability of occurrence of all possible values for the above notation.

Huffman encoding uses specific methods to select a representation for each symbol. This makes it a prefix code, using the shortest bit string. Represents the most common character of shorter bits and vice versa.



3.3.2 Transformación Burrows-Wheeler:

The Burrows-Wheeler conversion is a text transformer that accepts an input string and produces an output string that contains a large number of repeating characters. For example, the word "banana" becomes "annb $ aa." This output can be efficiently compressed.

The transformation is carried out by ordering all the rotations of the text in lexicographic order, and selecting the last column.

Transformación			
Entrada	Todas las Rotaciones	Ordenar Filas	Salida
^BANANA@	^BANANA@ @^BANANA A@^BANAN NA@^BANA ANA@^BAN NANA@^BA ANANA@^B BANANA@^	ANANA@^B ANA@^BAN A@^BANAN BANANA@^ NANA@^BA NA@^BANA ^BANANA@ @^BANANA	BNN^AA@A

3.3.3 LZ77:

The LZ77 acronyms comes from Lempel and Ziv, which are the last name of the creators of the algorithm, whose names are Abraham and Jacob, respectively. The 77 comes from 1977, being this the year when everything was done.

What differentiates this algorithm from the rest is that it does not present data omission when compressing it. This algorithm is made up of the following data types, literals, flags, and keywords.

First, before using this data, use is made of a bit that will act as a flag that will help the program to know what type of data comes next. If the prefix is 0, then what comes is an uncompressed byte. If instead the prefix is 1, then what follows next is an offset / size pair. In the next step are the keywords, which are groups of bits or bytes that contain information required by the compressor and the decompressor, and finally there will be the literal that will be an uncompressed byte

- **Literals:** They're simply non compressed bytes.
- **Key Words:** In our case they represent the displacement.
- **Flags:** They show whether if the date we Will see next will be a literal or a key word.

```
Obtener ‘a’.  Sin coincidencia.  Bandera 0. Literal ’a’.
Obtener ‘b’.  Sin coincidencia.  Bandera 0. Literal ’b’.
Obtener ‘ ’.  Sin coincidencia.  Bandera 0. Literal ’ ’.
Obtener ‘a’.  Coincidencia.      Bandera 1. Palabra clave: desplazamiento = 0, tamaño = 2.
```

3.3.4 LZW:

The LZW acronym means Lempel–Ziv–Welch, which are the 3 last names of the creators of the algorithm, which names are Abraham Lempel, Jacob Ziv, and Terry Welch respectively

This algorithm is, so to speak, an improved version of the LZ78, it serves to compress sequences of bits without taking into account the type of information, its principle consists of replacing patterns with an index code and progressively building a dictionary.

For the construction of the dictionary, it happens that it begins with the 256 values of the ASCII table. The file to compress is divided into strings of bytes each of these strings is compared to the dictionary and added if it is not there, which would be the total compression process. During decompression, the algorithm rebuilds the dictionary in the opposite direction; therefore, it does not need to be stored.

```
Carácter:  Código emitido  Entrada en el diccionario:
           (salida):

T          20 =  10100
O          15 =  01111      28: TO
B           2 =  00010      29: OB
E           5 =  00101      30: BE
O          15 =  01111      31: EO <--- se agotaron los códigos de 5 bits
R          18 = 010010      32: OR <--- se comienza a usar códigos de 6 bits
N          14 = 001110      33: RN
O          15 = 001111      34: NO
T          20 = 010100      35: OT
TO         28 = 011100      36: TT
BE         30 = 011110      37: TOB
OR         32 = 100000      38: BEO
TOB        37 = 100101      39: ORT
EO         31 = 011111      40: TOBE
RN         33 = 100001      41: EOR
OT         35 = 100011      42: RNO
#           0 = 000000      43: OT#
```

REFERENCIAS:

1. Revista de Salud Animal. Cálculo automatizado del vector eléctrico integral de la actividad ventricular cardíaca en equinos. Retrieved August 13, 2021. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0253-570X2016000100007.

2. Ciencia y Agricultura de la Universidad Pedagógica y Tecnológica de Colombia. Caracterización de variables utilizando inteligencia computacional para identificar alteraciones en la salud. Retrieved August 13, 2021. https://www.redalyc.org/pdf/5600/560058660005.pdf

3. Pérez, S. DESARROLLO DE UNA BASE DE DATOS CON IMÁGENES TERMOGRÁFICAS PARA USO EN ALGORITMOS DE VISIÓN E INTELIGENCIA ARTIFICIAL. Retrieved August 13, 2021. https://www.redalyc.org/pdf/5600/560058660005.pdf

4. Chelotti, J. Desarrollo e implementación de un dispositivo de adquisición y almacenamiento de sonidos para ganadería de precisión. Retrieved August 13, 2021. http://sedici.unlp.edu.ar/handle/10915/42007

5. Programador Click. algoritmo de talla de costura. Retrieved August 14, 2021. https://programmerclick.com/article/4510360417/

6. Bercial, J. Integer Scalling: Qué es y como funciona. Retrieved August 14, 2021. https://www.geeknetic.es/Guia/1662/Integer-Scaling-Que-es-y-como-funciona.html

7. Osuna, J. Introducción a lo Fractales y a la compresión Fractal. Retrieved August 14, 2021. http://www.iespravia.com/carpetas/recursos/mates/recursos_2005/fotografia/intro/intro.htm

8. Ingeniería de la Universidad Distrital Francisco José de Caldas. Compresión de imágenes con Wavelets y Multiwavelets. Retrieved August 14, 2021. https://www.redalyc.org/pdf/4988/498850160008.pdf

9. Wikipedia. Codificación Huffman. Retrieved August 15, 2021. https://es.wikipedia.org/wiki/Codificaci%C3%B3n_Huffman

10. Wikipedia. Compresión de Burrows-Wheeler. Retrieved August 15, 2021. https://es.wikipedia.org/wiki/Compresi%C3%B3n_de_Burrows-Wheeler

11. Wikipedia. LZSS. Retrieved August 15, 2021. https://es.wikipedia.org/wiki/LZSS#Caracter%C3%ADsticas

12. Wikipedia. LZW. Retrieved August 15, 2021. https://es.wikipedia.org/wiki/LZW