# **How to improve your farm with Precision Livestock Farming.**

|  |  |  |  |
| --- | --- | --- | --- |
| Camilo Palacio Restrepo  Universidad Eafit  Colombia  cpalacior@eafit.edu.co | Sebastian Salazar Osorio  Universidad Eafit  Colombia  ssalazaro1@eafit.edu.co | Simón Marín Universidad Eafit Colombia smaring1@eafit.edu.co | Mauricio Toro  Universidad Eafit  Colombia  mtorobe@eafit.edu.co |

# **ABSTRACT**

The objective of this proyect is to design an algorithm to compress and decompress images in order to optimize the energy consumption in the context of the precision livestock farming (GdP on Spanish). For this reason, the compressed images will be tested with a clasification algorithm of images about animal health. The compression algorithms can be designed based on Hash Tables, Red-Black Tress or both of them.

The main goal of this algorithms is to reach the best possible rate of compression without the accuracy of the classification being affected on more than 5%. Regarding the time consumption and memory, time consumption has a higher priority.

## **Keywords**

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

# **1. INTRODUCTION**

# Thanks to the fact that we are in the Information Age, it’s essential for the mankind to create and learn how to use loads of tools in order to have access to a bigger amount of information at once. Those tools should be able to help us to organize, stock and be efficient so it can be easier to handle. We will implement a file system which will help to manage the files, the images and free up space.

The usage that we are going to give to this system is to improve the performance of the energy used by the farmers, the way they store the information and the time they spend doing it. We will use this system to compress images to classify animal health so it becomes easier to manage the information.

# **1.1. Problem**

*O*ne of the problems is that the majority of the data of the farms are in notebooks or in zoologist´s databases. Another is the automatic digitalization of the useful data to the construction of the precision livestock farming. Furthermore we have to use data compression algorithim to capture data from diferent sourses

**1.2 Solution**

For this proyect, 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 infraestructure is very limited, thus data compression is required.

For this problem, we decided to use the Box Sampling compression algorithm for the images compression. We chose this algorithm specifically due to it’s loseless characteristics.

**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 Studies of the farm animals’ behavior** They used the sensors of the Iphone 4s and Iphone 5s to study the behavior of the cattle. They notticed that the study of animal behavior requires the storage of many high frecuency variables from a large number of individuals and their processing through various relevant variables combinations for modeling and decition making so they used a lambda cloud architecture coupled to a scientific sharing platform used to process high-frecuency data. They achieved reduce data redundancy ut to 98.5 % with the use of Edge computing.

Debauche, O., Mahmoudi, S., Andriamandroso, A.L.H., Manneback, P., Bindelle, J., Lebeau, F., 2019.

## **2.2 Platform to assure animal welfare**

The University of West Attica and the Agricultural University of Athens researched about the agricultural reforms of EU and how different techniques were created in order to comply with these regulations. The main goal of the reforms is to assure the welfare of the animals used in farming in the European Union. One of the solutions used consists of implanting sensors on the animals, which are in charge of capturing the data. This data is processed locally on an Edge Device and then, the results are pushed to the cloud, from where they can be accessed through a companion mobile application. For the purpose of data collection, a prototype device is implemented, which would be carried by the animal at all times. The device is designed as a collar, in order to be easy to handle and would not introduce any constraints unfamiliar to the animal.

All the gathered data is uploaded in the cloud and the mobile application. The mobile application informs accordingly the user about the animals’ location and their well-being.

[1] Vasileios Doulgerakis, Dimitrios Kalyvas, Enkeleda Bocaj, Christos Giannousis, Michalis Feidakis, George P. Laliotis, Charalampos Patrikakis, Iosif Bizelis. 2020. University of West Attica, Department of Electrical & Electronics Engineering 250 Thivon & P. Ralli, 12241, Athens, Greece. Agricultural University of Athens, Department of Animal Breeding & Husbandry 75 Iera Odos, 11855, Athens, Greece.

## **2.3 Indivual identification of cattle via deep learning**

They demostrate that computer vision pipelines using deep neutral architectures are well-suited to perform Holstein Friesian cattle detection as well as the individual identification in agriculturally relevant setups. This work is the first to apply deep learning to the task of automated visual bovine identification. With this (the Friesian cattle detection and localization) can be performed with an accuracy of 99 .3 % on this data.

[1] Andrew, W., Greatwood, C., & Burghardt, T. (2018). Visual Localisation and Individual Identification of Holstein Friesian Cattle via Deep Learning. In 2017 IEEE International Conference of Computer Vision Workshop (ICCVW 2017) (pp. 2850-2859). Institute of Electrical and Electronics Engineers (IEEE).

## **2.4 Alternatives for improvement in Precision Livestock Farming**

This article presents a systematic literature review of recent works on the use of machine learning (ML) in precision livestock farming (PLF), focusing on two areas of interest: grazing and animal health. This review: highlights opportunities for ML in the livestock sector; shows the current sensors, software and techniques for data analysis; details the increasing openness of data sources. It was found that the use of ML in PLF is in a stage of development and has several research challenges. Examples of such challenges are: to develop hybrid models for diagnosis and prescription as a tool for the prevention and control of animal diseases; to bring together the grazing and animal health issues; to give autonomy to PLF using autonomous cycles of data analysis tasks and meta-learning; and to bring together soil and pasture variables because, for both, animal health and animal grazing, the variables used are only behavioral and environmental.

[1] Rodrigo García , Jose Aguilar , Mauricio Toro , Angel Pinto , Paul Rodríguez. 2020. A systematic literature review on the use of machine learning in precision livestock farming. Department on Informatics and Systems, Universidad del Sinú, Montería, Colombia b CEMISID. Universidad de Los Andes, Merida, Venezuela c GIDITIC. Universidad EAFIT, Medellin, Colombia d Faculty of Economics, Administration and Accounting. Universidad del Sinú, Montería, Colombia

## **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](https://github.com/mauriciotoro/ST0245-Eafit/tree/master/proyecto/datasets)s .

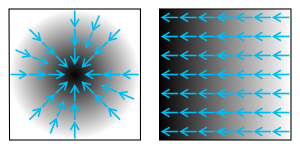
Finally, using the training data set, we trained a convolutional neural network for binary image-classification using Google Teachable Machine available at <https://teachablemachine.withgoogle.com/train/image>.

## **3.2 Seam carving**

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

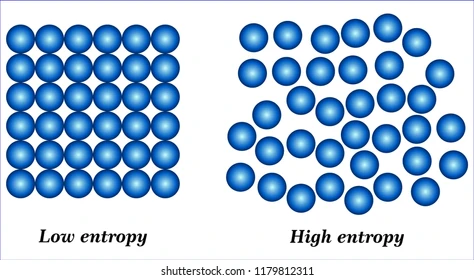
**3.2.1 Image gradient**

Is a directional change in the intensity or color in an image. The most common way to approximate the image gradient is to convolve an image with a kernel, such as the Sobel operator or Prewitt operator. Each pixel of a gradient image measures the change in intensity of that same point in the original image, in a given direction. To get the full range of direction, gradient images in the x and y directions are computed

.

**3.2.2 Entropy**

The entropy or average information of an image can be determined approximately from the histogram of the image. The histogram shows the different grey level probabilities in the images.



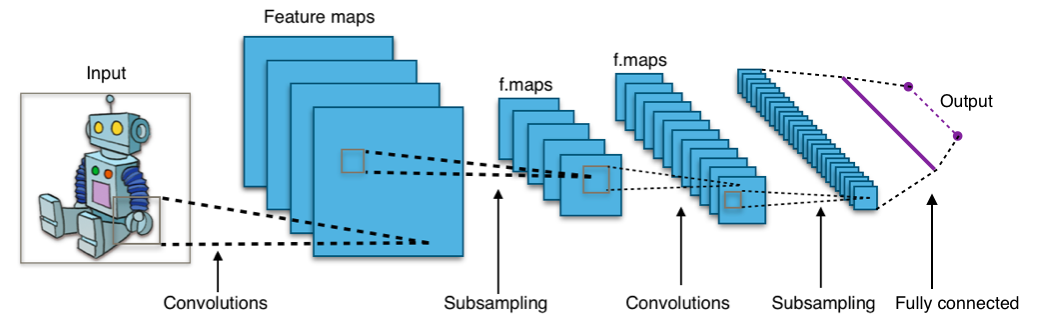
**3.2.3 Visual saliency**

A saliency map is an image that shows each pixels´unique quality. The goal of a saliency map is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Saliency estimation may be viewed as an instance of image segmentation. In computer vision, image segmentation is the process of partitioning a digital image into multiple segments.



**3.2.4 eye-gaze movement**

Interpretation of the data that is recorded by the various types of eye-trackers employs a variety of software that animates or visually represents it, so that the visual behavior of one or more users can be graphically resumed

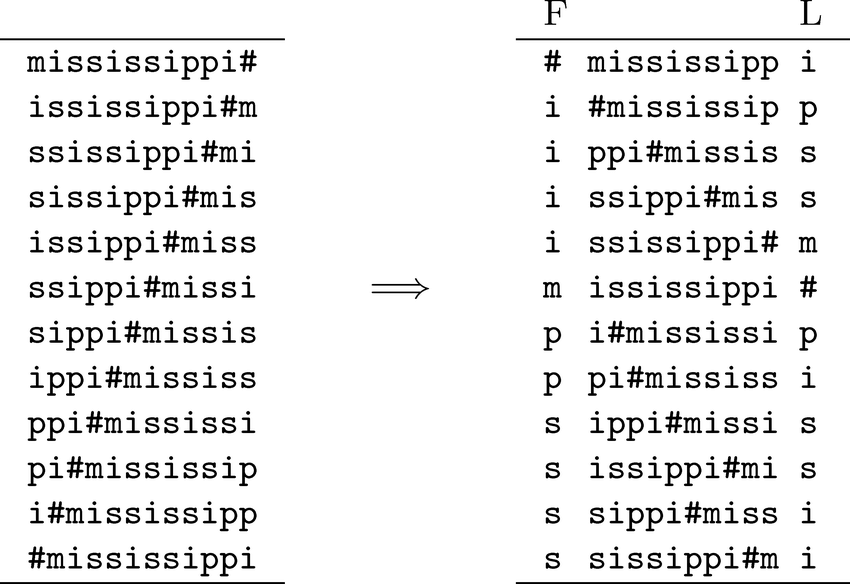


## **3.3 Lossless Image-compression alternatives**

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

**3.3.1 Burrows–Wheeler transform**

It rearranges a character string into runs of similar characters. It uses techniques like move-to-front transform and run-length encoding. And the most important thing is that the transformation is reversible without needing to store any additional data except the position of the first original character.

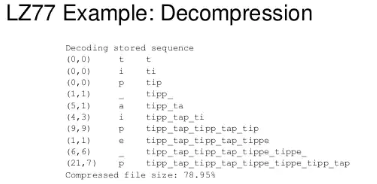


**3.3.2 LZ77 and LZ78**

They are both theoretically dictionary coders. LZ77 mantains a sliding window during compression. This was later shown to be equivalent to the explicit dictionary constructed by lZ78.

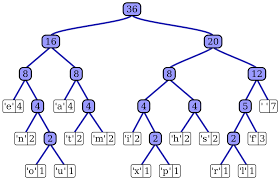
LZ77 algorithms achieve compression by replacing repeated occurrences of data with references to a single copy of data exsisting earlier in the uncompressed data stream

LZ78 algorithms achieve compression by replacing repeated occurrences of data with references to a dictionary that is built based on the input data stream.



**3.3.3 Huffman coding**

The output from Huffman's algorithm can be viewed as a variable-length code table for encoding a source symbol (such as a character in a file). The algorithm derives this table from the estimated probability or frequency of occurrence (weight) for each possible value of the source symbol.



**3.3.4 Gray-scale compression**

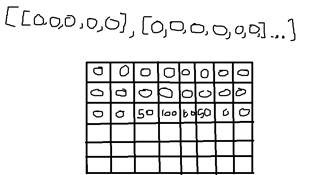
The results also show that it slightly outperforms JPEG-2000s compression performance, when it operates in its lossless mode, and it is comparable to JPEG-LS’s and CALIC’s compression performance, where JPEG-2000 and JPEG-LS are the current image compression standards, and CALIC is a Context-based Adaptive Lossless Image Coding scheme.

## **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[[1]](#footnote-2).

## **4.1 Data Structures**

## The data structure we are using is the matrices from the NumPy library, it consists of a table with a width of the number of data (numbers) in each line and a large of the number of lines

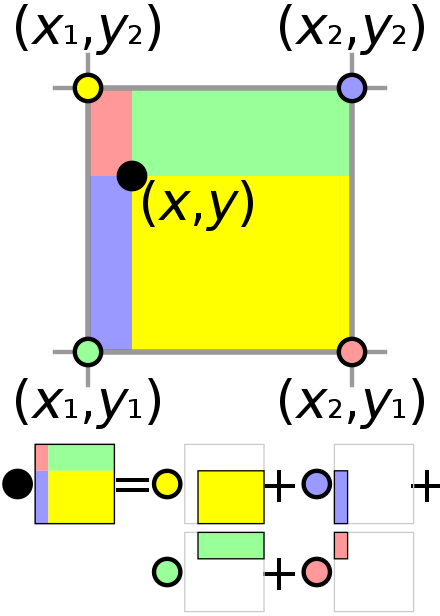


**Figure 1:** A numpy matrix made by the library NumPy

**4.2 Algorithms**

The algorithm that we are implementing is the image scaling algorithm, in this case, the bilineal algorithm, which works by interpolating pixel color values, introducing a continuous transition into the output even where the original material has discrete transitions. Although this is desirable for continuous-tone images, this algorithm reduces contrast (sharp edges).

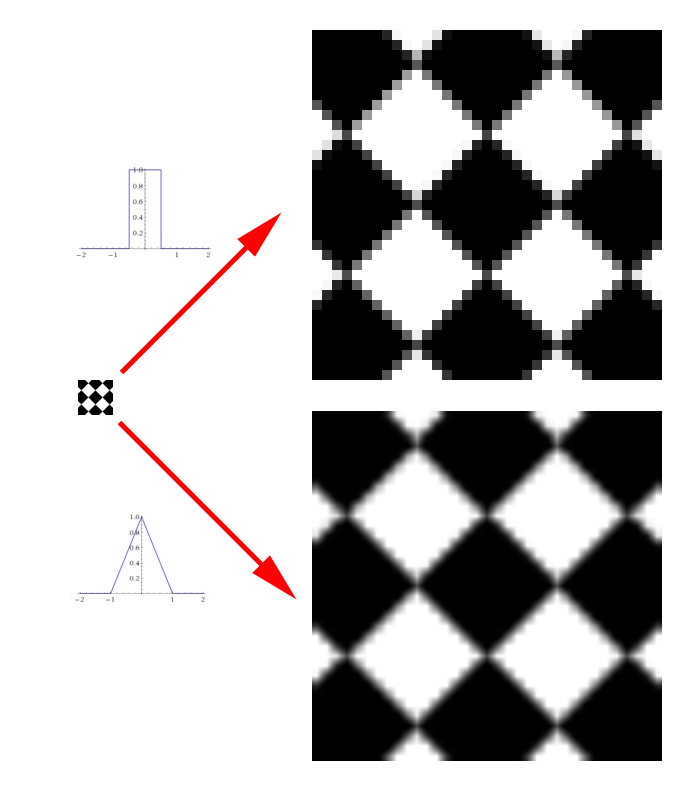
Basically we are grabing four pixels from the image and making an average of those, and then asign that value to only 1 pixel. The image generated is lighter than the previous but you can’t recover the exact image that you had.



**4.2.1 Lossy image-compression algorithm**

Image scaling

It refers to the resizing of a digital image. When scaling an image, a new image generates with a higher or lower number of pixels, and in our case (scaling down) it results in a visible quality loss.



**4.2.2 Lossless image-compression algorithm**

**LZ77**

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.

**4.3** **Complexity analysis of the algorithms**

Explain, in your own words, the analysis for the worst case using O notation. How did you calculate such complexities. Please explain briefly.

|  |  |
| --- | --- |
| **Algorithm** | **Time Complexity** |
| Compression | O(N2\*M2) |
| Decompression | O(N3\*M\*2N) |

**Table 2:** Time Complexity of the image-compression and image-decompression algorithms. *(Please explain what do N and M mean in this problem).*

|  |  |
| --- | --- |
| **Algorithm** | **Memory Complexity** |
| Compression | O(N\*M\*2N ) |
| Decompression | O(2M\*2N) |

**Table 3:** Memory Complexity of the image-compression and image-decompression algorithms. *(Please explain what do N and M mean in this problem).*

**4.4 Design criteria of the algorithm**

Explain why the algorithm was designed that way. Use objective criteria. Objective criteria are based on efficiency, which is measured in terms of time and memory consumption. Examples of non-objective criteria are: “I was sick”, “it was the first data structure that I found on the Internet”, “I did it on the last day before deadline”, etc. Remember: This is 40% of the project grading.

**5. RESULTS**

**5.1 Model evaluation**

In this section, we present some metrics to evaluate the model. Accuracy is the ratio of number of correct predictions to the total number of input samples. Precision. is the ratio of successful students identified correctly by the model to successful students identified by the model. Finally, Recall is the ratio of successful students identified correctly by the model to successful students in the data set.

**5.1.1 Evaluation on training data set**

In what follows, we present the evaluation metrics for the training data set in Table 3.

|  |  |
| --- | --- |
|  | ***Training data set*** |
| *Accuracy* | 0.02 |
| *Precision* | 0.03 |
| *Recall* | 0.01 |

## **Table 3.** Binary image-classification model evaluation on the training data set.

**5.1.2 Evaluation on test data set**

In what follows, we present the evaluation metrics for the testing dataset in Table 4 without compression and, in Table 5, with compression.

|  |  |
| --- | --- |
|  | ***Testing data set*** |
| *Accuracy* | 0.01 |
| *Precision* | 0.012 |
| *Recall* | 0.013 |

## **Table 4.** Binary image-classification model evaluation on the testing data set without image compression.

|  |  |
| --- | --- |
|  | ***Testing data set*** |
| *Accuracy* | 0.001 |
| *Precision* | 0.0012 |
| *Recall* | 0.0013 |

## **Table 5.** Model evaluation on the testing data set with image compression.

**5.2 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.

Compute execution time for each image in Github. Report average execution time Vs average file size.

## 

|  |  |  |
| --- | --- | --- |
|  | ***Average execution time (s)*** | ***Average  file size (MB)*** |
| *Compression* | 100.2 s | 12.4 MB |
| *Decompression* | 800.1 s | 12.4 MB |

## **Table 6:** Execution time of the *(Please write the name of the algorithms, for instance, seam carving & LZ77)* algorithms for different images in the data set.

## **5.3 Memory consumption**

We present memory consumption of the compression and decompression algorithms in Table 7.

|  |  |  |
| --- | --- | --- |
|  | ***Average memory consumption (MB)*** | ***Average file size (MB)*** |
| Compression | 634 MB | 3.12 MB |
| Decompression | 9 MB | 878.12 MB |

## **Table 7:** Average Memory consumption of all the images in the data set for both compression and decompression.

## To measure memory consumption, you should use a profiler. A very good one for Java is VisualVM, developed by Oracle, <http://docs.oracle.com/javase/7/docs/technotes/guides/visualvm/profiler.html>. For Python, use C Profiler.

## **5.3 Compression ratio**

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

|  |  |  |
| --- | --- | --- |
|  | ***Healthy Cattle*** | ***Sick Cattle*** |
| Average compression ratio | 1:23 | 1:34 |

## **Table 8:** Rounded Average Compression Ratio of all the images of Healthy Cattle and Sick Cattle.

## **6. DISCUSSION OF THE RESULTS**

Explain the results obtained. Are precision, accuracy and sensibility appropriate for this problem? Is the model over-fitting? Is memory consumption and time consumption appropriate? Is compression ratio appropriate? Does compression changes significantly precision on the test data set? *(In this semester, according to the results, can this improve animal-health classification in the context of PLF?)*

**6.1 Future work**

Answer, what would you like to improve in the future? How would you like to improve your algorithm and its implementation? What about using discrete cosine transform or wavelet compression?

# **ACKNOWLEDGEMENTS**

Identify the kind of acknowledgment you want to write: for a person or for an institution. Consider the following guidelines: 1. Name of teacher is not mentioned because he is an author. 2. You should not mention websites of authors of articles that you have not contacted. 3. You should mention students, teachers from other courses that helped you.

As an example: This research was supported/partially supported by [Name of Foundation, Grant maker, Donor].

We thank for assistance with [particular technique, methodology] to [Name Surname, position, institution name] for comments that greatly improved the manuscript.

# **REFERENCES**

1. 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

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

3. <https://en.wikipedia.org/wiki/Eye_tracking>

4. <https://en.wikipedia.org/wiki/Saliency_map>

5. <https://stats.stackexchange.com/questions/235270/entropy-of-an-image>

6. [1] Rodrigo García , Jose Aguilar , Mauricio Toro , Angel Pinto , Paul Rodríguez. 2020. A systematic literature review on the use of machine learning in precision livestock farming. Department on Informatics and Systems, Universidad del Sinú, Montería, Colombia b CEMISID. Universidad de Los Andes, Merida, Venezuela c GIDITIC. Universidad EAFIT, Medellin, Colombia d Faculty of Economics, Administration and Accounting. Universidad del Sinú, Montería, Colombia

7. [1] Vasileios Doulgerakis, Dimitrios Kalyvas, Enkeleda Bocaj, Christos Giannousis, Michalis Feidakis, George P. Laliotis, Charalampos Patrikakis, Iosif Bizelis. 2020. University of West Attica, Department of Electrical & Electronics Engineering 250 Thivon & P. Ralli, 12241, Athens, Greece. Agricultural University of Athens, Department of Animal Breeding & Husbandry 75 Iera Odos, 11855, Athens, Greece.

1. <http://www.github.com/> ????????? /proyecto/ [↑](#footnote-ref-2)