Algorithms to Further the Development of PLA technologies

Tomas Calle
Universidad Eafit
Colombia
tcallee@eafit.edu.co

Juan Camilo Salazar
Universidad Eafit
Colombia
csalazaru@eafit.edu.co

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

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

For each version of this report: 1. Detele all text in red. 2. Adjust spaces among words and paragraphs. 3. Change the color of all the texts to black.

Red text = Comments

Black text = Simón and Mauricio's contribution

Green text = To complete for the 1st deliverable

Blue text = To complete for the 2nd deliverable

Violet text = To complete for the 3rd deliverable

ABSTRACT

To write an abstract, you should answer the following questions in a single paragraph: What is the problem? Why is the problem important? Which are the related problems? Which is the algorithm you proposed? What results did you achieve? What are the conclusions of this work? The abstract should have **at most 200 words**. (*In this semester, you should summarize here execution times, memory consumption, compression ratio and accurracy*).

Keywords

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

1. INTRODUCTION

Precision Livestock farming is a type of farming where technology is integrated to optimize resources. Animals are analyzed with technology to identify and solve problems when they are presented. So in general, livestock is one of the main sources of nutrition to humans and is produced in large quantities. This job has been really manual throughout the years because it consists in doing specific tasks and repeating them each time. For this reason, people have started implementing some automated solutions to this task, such as counting the number of cows, making robots for milking the cow, etc.

A great challenge in this industry is the data management, now that most of the data is stored in paper which is prone for errors. According to Moore's Law, the number of transistors on a microchip doubles every two years which means that computers increment the speed each two years. The problem is that data is doubling at faster rates where the computer becomes slower. So the challenge is to optimize the storage by compressing data to ensure the speed and storage of the devices. Furthermore, people are searching to use the least resources possible, by compressing information to save energy consumption which may seem small, but on a large scale it is immense. Fortunately, there are some compression algorithms that can provide us with an accurate solution to the problem.

Explain the motivation, in the real world, that leads to the problem. Include some history of this problem. *(In this semester, motivation is why we need to compress images to classify animal health in the context of precision livestock farming).*

1.1. Problem

Compressing information minimizes energy consumption and in large scale projects, it can save lots of resources from the managers. In this case, we will compress cow images to classify their health based on the context of precision livestock farming. So this activity will help process hundreds of photos and classify them based on their help and this will save a lot of workforce from the farmer. The idea is to use the best compression possible in order to have less than 5% of error in the health classification

In a few words, explain the problem, the impact that has on society and why it is important to solve the problem. *(In this semester, the problem is to compress images to classify animal health in the context of precision livestock farming).*

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.

Explain, briefly, your solution to the problem *(In this semester, the solution is an implementation of compression algorithms. Which algorithms did you choose? Why?)*

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.

Explain four (4) articles related to the problem described in Section 1.1. You may find the related problems in scientific journals. Consider Google Scholar for your search. *(In this semester, related work is research on animal-health classification and data compression, in the context of PLF).*

3.1 Automatic cough detection for bovine respiratory disease in a calf house

Bovine respiratory disease (BRD) is a major problem for livestock farmers. It forces the farmers to incur huge costs on treatment and professional treatment. Researchers were successful in developing an algorithm that listed for cough sounds and served as an early warning system for farmers. Using 664 hand picked sound references the algorithm reached an efficacy between 53% and above 80%.

Anon. 2018. Automatic cough detection for bovine respiratory disease in a calf house. (July 2018). Retrieved February 14, 2021 from https://www.researchgate.net/publication/326240118_Automatic_cough_detection_for_bovine_respiratory_disease_in_a_calf_house

3.2 Model development for solving heat stress problems on dairy farms

Stress by heat is the major deterrent of optimal dairy production in the world. Even Though it is the biggest cause of decreasing darty production of an animal, it is impossible for farmers to have a live view of their animals' stress levels. Using image processing and tecention algorithms researchers were able to detect animals' stress levels and change diets accordingly.

Boyu Ji. 2017. model development for solving heat stress problems on dairy farms. (September 2017). https://www.researchgate.net/publication/320577951_PLF_technologies_model_development_for_solving_heat_stress_problems_on_dairy_farms

3.3 An ethogram of biter and bitten pigs during an ear biting event

Pigs raised on industrial farming facilities tend to develop violet tendencies. Understanding the violent tendencies will help farmers better understand how to help animals cope with industrial style farming. The researchers were able to detect, and identify violent events inside farming societies, thus helping farmers better understand what really is going on.

Alessia Diana. 2019. An ethogram of biter and bitten pigs during an ear biting event: first step in the development of a Precision Livestock Farming tool. (March 2019). Retrieved February 14, 2021 from https://www.researchgate.net/publication/332073052_An_ethogram_of_biter_and_bitten_pigs_during_an_ear_biting_event_first_step_in_the_development_of_a_Precision_Livestock_Farming_tool

3.4 Automatic lameness detection in intensive livestock systems

Lameness is one of the most common production diseases affecting the livestock business. Even Though it is preventable, it is hard to diagnose without a close look into each individual animal. Rechers developed algorithms that helped in the early detection of the disease.

Samaneh Azarpajouh. 2020. automatic lameness detection in intensive livestock systems. (June 2020). https://www.researchgate.net/publication/342110673_Precision_livestock_farming_automatic_lameness_detection_in_intensive_livestock_systems

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 and 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/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. *(In this semester, examples of such algorithms are Seam carving, image scaling, discrete cosine transform, wavelet compression and fractal compression).*

3.2.1 Discrete cosine transform

The image breaks into 8x8 boxes of pixels(orthogonal matrix). Then, DCT(discrete cosine transform) is applied to each block which will be compressed with quantization(making a number each time smaller). This will make an array of quantized blocks and is stored in a small amount of space. To decompress the information, you need to use the Inverse Discrete Cosine Transform(IDCT).

Complexity: O(n^2

Cabeen, Ken, and Peter Gent. “Image Compression and

the Discrete Cosine Transform.” Dct,

www.math.cuhk.edu.hk/~lmlui/dct.pdf.

3.2.2 Fractal compression

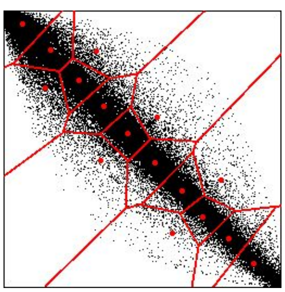
It consists in creating fractals of the image inputted. So this enters in a loop of transforming the image inputted and “copying” the image in smaller parts. So basically, the program will store the numbers that define the affine transformations(geometric transformation that preserves lines and parallelism) and then generate the image with those numbers whenever needed.

Texas Instruments Europe. “An Introduction to Fractal Image Compression.” Ti, Oct.

1997, ti.com/lit/an/bpra065/bpra065.pdf.

Please explain the algorithm, its complexity and include a vector Figure.

3.2.3 Vector Quantization



Vector Quantization processes the input in groups and separates them into a set of well-defined vectors with a measure of distortion. So the idea of setting vectors, is to have approximately the same vectors as the rest of the sets and each one has its centroid point which is its representation. By dividing the vectors into sets, it will require less storage space in order to compress the image easily. The density-matching property gives a percentage of error to the data compression which is inversely proportional to its density.

G. Boopathy, and S. Arockiasamy. “Implementation of

Vector Quantization for Image Compression -

A Survey.” Core, Apr. 2010, core.ac.uk/download/pdf/231160989.pdf.

Ivanov, Ivan Assen. “Image Compression with Vector Quantization.” Gamasutra, 16

Apr. 2001,

www.gamasutra.com/view/feature/131499/image_compression_with_vector_.php.

Complexity : O(log2n)

Please explain the algorithm, its complexity and include a vector Figure.

3.2.4 Generalized Lloyd Algorithm

This algorithm has 2 fases.the codebook(set of vectors) assignment phase and the codebook adjustment phase. The program works by getting each of the vectors from the input set and assigning it to the nearest vector from the codebook. Each of these codebook vectors are replaced with the centroid of all input vectors assigned to it. People call this process convergent.

Ivanov, Ivan Assen. “Image Compression with Vector Quantization.” Gamasutra, 16

Apr. 2001,

www.gamasutra.com/view/feature/131499/image_compression_with_vector_.php.

Complexity: O(n^2)

3.3 Lossless Image-compression alternatives

In what follows, we present different algorithms used to compress images. *(In this semester, examples of such algorithms are Borrows & Wheeler Transform, LZ77, LZ78, Huffman coding and LZS).*

3.3.1 LZ77

The algorithm works by having a sliding window roll over a stream of data. The window rolls over depending on the matches in finds between the star of the window and the end data values after the end of the window. It records the offset, the length of the match and the next unmaxed data value.

complexity: O(n)

3.3.2 LZ78

The algorithm works by creating a hashmap or dictionary. the entries in the dictionary will consist of indexes of the relevant peace of data and the next piece of data. If we encounter a repeated sequence that is already in the dictionary, there is no need to duplicate the entry but rather store only the index and the pointing figure to the original sequence.

complexity: O(n)

3.3.3 Huffman

The algorithm works by first creating a table of all pixel data broken down by frequency of equal information. It then creates an ordered list of nodes containing the pixel data with each relevant frequency. It then starts merging the nodes with every single adjacent node until a node tree is built with the parent node equal to the number of base nodes.

complexity: O(nlogn)

3.3.4 Shannon-fano

The algorithm works by first describing the probability of each piece of data. It then groups data with equal probability sums. It repeatedly divides the groups into 2 groups, the small probability sum and the biggest probability sum.

complexity = 0(n)

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].

4.1 Data Structures

Explain the data structure used to make the image compression and make a figure explaining it. Do not use figures from the Internet. *(In this semester, example of the data structures are trees and hash tables)*

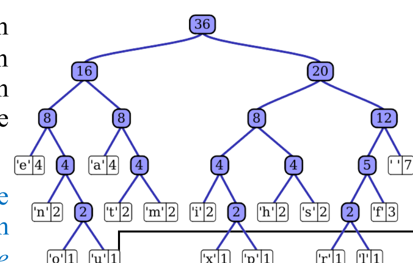


Figure 1: Huffman tree generated from the exact frequencies of the text "this" (Please, feel free to change this Figure if you use a different data structure).

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.

Explain the design of the algorithms to solve the problem and make a figure. Do not use figures from the Internet, make your own. *(In this semester, one algorithm must be a lossy image-compression algorithm such as image scaling, seam carving or wavelet compression and the second algorithm must be a lossless image-compression algorithm such as Huffman coding, LZS or LZ77).*

4.2.1 Lossy image-compression algorithm

Explain, briefly, how did apply a lossy image-compression algorithm such as seam carving or image scaling. Explain also decompression.

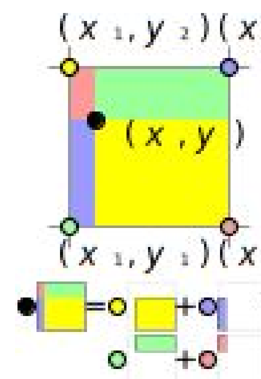


Figure 2: Image scaling using bi-lineal interpolation. (Please, feel free to change this Figure if you use a different data structure).

4.2.2 Lossless image-compression algorithm

Explain, briefly, how did you apply a lossless image-compression algorithm such as Huffman coding, LZS or LZ77. Explain also decompression.

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	$O(N^2*M^2)$
Decompression	$O(N^3*M*2^N)$

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	$O(N*M*2^N)$
Decompression	$O(2^{M*}2^N)$

[1]https://www.github.com/tomasCalletce/Algorithms-to-Further-the-Development-of-PLA-technologies

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

Reference sourced using ACM reference format. Read ACM guidelines in http://bit.ly/2pZnE5g

As an example, consider this two references:

1.Adobe Acrobat Reader 7, Be sure that the references sections text is Ragged Right, Not Justified. http://www.adobe.com/products/acrobat/.

2. Fischer, G. and Nakakoji, K. Amplifying designers' creativity with domainoriented design environments. in Dartnall, T. ed. Artificial Intelligence and Creativity: An Interdisciplinary Approach, Kluwer Academic Publishers, Dordrecht, 1994, 343-364.

Please remove the references above, they are only an example.