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

ABSTRACT

Classify livestock images in precision farming to solve for animal health detention. Farmers incur huge cost because it is hard to manually detent animal health deterioration early enough to allow for successful treatment. Using image compression running on O(n)(being the number of pixels in an image) and machine learning models we present a solution that empowers farmers to keep track of their animals ' health.

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 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.

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.

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.

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%.

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.

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.

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.

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).

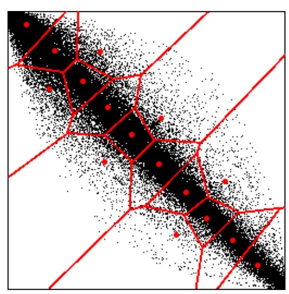
Complexity: O(n^2

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.

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.

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.

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.

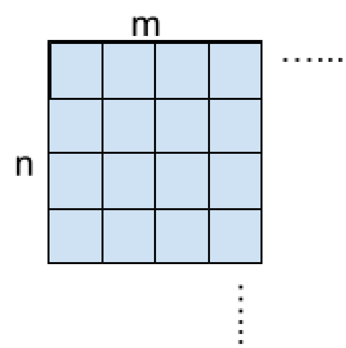
complexity : O(n)

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

We made use of a (n x m) matrix where n is the number of datapoint lines and m is the number of datapoints in each line. The matrix is made of static arrays whose lengths are calculated only once prior to filling up the matrix with data.



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.

4.2.1 Lossy image-compression algorithm

We choose to use the interpolation algorithm. It works by dividing pixel data into blocks of 4 values and finding the average value between all four points. We thus sacrificed file resolution for file size.

---

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

The decompression algorithm works by duplicating pixels nxn. Meaning files increase in size but keep the same underlying data as the compressed file.

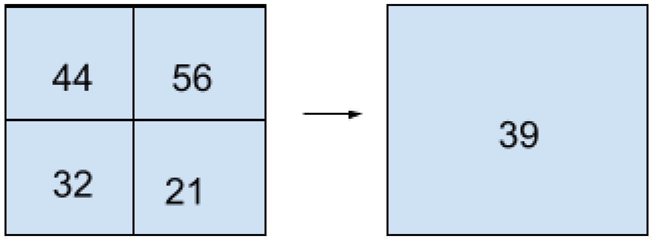


Figure 2

4.2.2 Lossless image-compression algorithm

Run-Length encoding with the subtraction base algorithm.

The algorithm first works by converting every pixel into the difference between itself and the last. This enables us to increase the frequency of some numbers and thus increase the efficacy of the runth length algorithm that saves repeating values as a value followed by the multiple of repetitions.

4.3 Complexity analysis of the algorithms

The worst case scenario is seen in every use case. Because the algorithm behaves the same no matter the input.

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

n = total pixels.

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

Algorithm	Memory Complexity
Compression	O(n)
Decompression	O(n)

n = total pixels.

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

4.4 Design criteria of the algorithm

From the very beginning we understood that we would need to optimize and prioritize speed over anything else. Our model thrives on massive amounts of information and works best when inputs are diverse and plentiful. Because of this we decided that we would trade some compression for a reduction in run time.

We first tried using the lz77 algorithm but quickly realized the algorithm was too slow *O(n^2)(with n being the number of pixels in the photograph)* .We then tried the huffman algorithm but decided it was not a viable option because the image had to many unique data points and thus it was impossible to keep binary codes from becoming too large.

We then decided to use a version of the run length algorithm that incorporated a subtraction algorithm to accomplish a 20 : 13 compression ratio while keeping a time complexity of *O(n)(with n being the number of pixels in the photograph).*

5. RESULTS

5.1 Model evaluation

In this section, we present some metrics to evaluate the model. Accuracy is the ratio of the 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

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.

	Average execution time (s)	***Average file size (B)***
Compression	.29	133.504 B
Decompression	.40	123.770 B

Table 6: Run-Length algorithm with subtraction base algorithm

5.3 Memory consumption

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

	Average memory consumption (MB)	***Average file size (B)***
Compression	4 MB	133.504 B
Decompression	9 MB	123.770 B

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

5.3 Compression ratio

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

	Healthy Cattle	***Sick Cattle***
Average compression ratio	20:13	20:13

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

6. DISCUSSION OF THE RESULTS

We are extremely happy with the compression algorithm. Even Though we understand there are algorithms that accomplish a better compression ratio than our solution did. Our algorithm works best for this specific use case because it accomplished manageable compression ratio 20:13, while keeping time complexity to the minimum.

6.1 Future work

Create a LZSS algorithm and connect that algorithm with huffman encoding to create the Deflate Algorithm. This would enable us to use the powerful huffman encoding algorithm in a manner that would benefit the nature of the Huffman encoding.

ACKNOWLEDGEMENTS

We thank Simon and Kevin for being instrumental in guiding our understanding of compression algorithms. They taught us there is no perfect algorithm for everything, there is only a perfect algorithm for a specific task or goal.

We also thank Maurizio for all the support and the willingness to jump on video calls.

REFERENCES

G. Boopathy, and S. Arockiasamy. “Implementation of Vector Quantization for Image Compression - A Survey.” Core, Apr. 2010, core.ac.uk/download/pdf/231160989.pdf.

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

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

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

Ivanov, Ivan Assen. “Image Compression with Vector Quantization.” Gamasutra, 16 Apr. 2001,

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

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

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

the Discrete Cosine Transform.” Dct, www.math.cuhk.edu.hk/~lmlui/dct.pdf.

Boyu Ji. 2017. model development for solving heat stress problemsondairyfarms.(September2017). https://www.researchgate.net/publication/320577951_PLF_technologies_model_development_for_solving_heat_stress_problems_on_dairy_farms

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

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

C. Patiño-Forero, M. Agudelo-Toro, and M. Toro. Planning system for deliveries in Medellín. ArXiv e-prints, Nov. 2016. Available at: https://arxiv.org/abs/1611.04156