Compression algorithms to optimize battery consumption in precision livestock farming.

Isabella Montoya Henao
Universidad Eafit
Colombia
imontoyah@eafit.edu.co

María Isabel Arango Palacio
Universidad Eafit
Colombia
miarangop@eafit.edu.co

Andrés Felipe Agudelo
Universidad Eafit
Colombia
afagudeloo@eafit.edu.co

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

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

ABSTRACT

The 34% world supply of food proteins comes from livestock [1] and the need to supplement it, makes that the number of animals rearing increases day per day. Nowadays, this process is not effective due to the farmers not having the correct tools and devices to minimize their energy consumption. In line, the objective of this project is to design an algorithm that helps to compress and decompress images to optimize the energy that is required for classify and obtain the information of the animals.

Keywords

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

1. INTRODUCTION

Livestock farming has played a very important economic and socio-cultural role throughout history. A few years ago, a multidisciplinary science called the Precision Livestock Farming (PLF) came up with the purpose of getting better livestock activity using information and communication technology. Currently PLF has some important challenges like data digitization, reducing data dimensions and some other that are important to apply the system correctly in the farms. During the years, some experts like Debauche had concluded that the best way to assume those challenges is with an algorithm that helps to compress the data [2] to determine some characteristics of the animal like its health, behavior, and other.

1.1. Problem

We are facing a problem that is designing an algorithm to compress and decompress images to reduce the energy consumption in the context of **precision** livestock farming. Moreover, the compress images will be evaluated through a classification algorithm that determines the animal health. The objective is to get the more accurate compression and try to have an error rate less than 5%.

According to D.Berckmans´s article a major problem of the livestock sector is and will be the continuous monitoring of animal health within the big groups of animals[3]. And it is because as decades have passed the number of animals has increased but the number of farmers has decreased. For the above reasons and more is important to develop an efficient solution that makes farmers able to have more information about the health and other conditions of the animals.

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.

Basically, we implemented image scaling using two different algorithms. Firstly, we choose nearest neighbor interpolation, this method simply determines the “nearest” neighbouring pixel and assumes the intensity value of it. The main reason we decided to implement it, is because this algorithm has an appropriate complexity to work with a huge amount of data (the time complexity for the worst case is O(nxm)) and also because compared with other algorithms like bilinear interpolation, you don't lose much information when you compress the image.

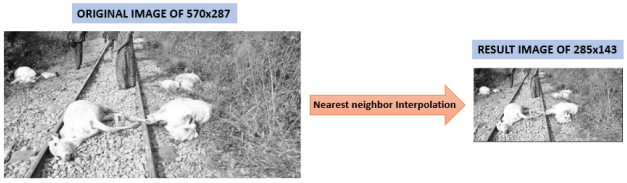

Illustration Image Scaling using Nearest neighbor interpolation

Secondly, we used Fast Fourier Transformation (FFT), this is commonly use to modify an image between the spatial and frequency domain. This algorithm transforms our image matrix into a matrix of coefficients which are frequencies, and if we only keep the higher of them it will reduce a lot the size of the matrix. For example, we can only keep 5% of those coefficients and the loss of information would be really small. Also, this algorithm has an appropriate time complexity, it is O(m*n*log(n*m)) for the worst case.

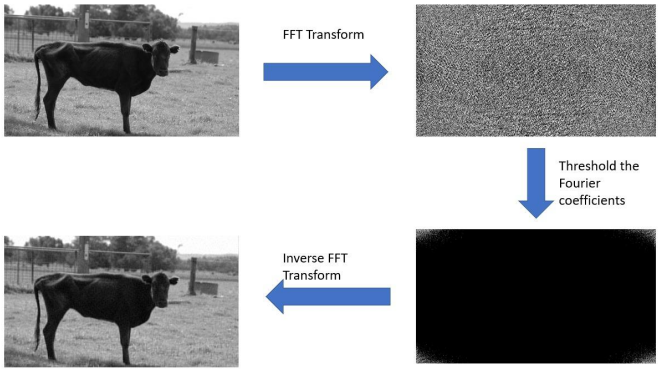

Illustration Image Scaling using FFT.

Lastly, we tested other method that is Singular Value Decomposition (SVD), this algorithm creates 3 variables from the image matrix and if you multiply these variables, it will be equal to the original matrix: Knowing this we can choose fewer elements from those 3 variables and create an approximation that can keep a lot of information with a reduction of almost 80% of its size. All this with a time complexity of O (max (m, n) ^2).

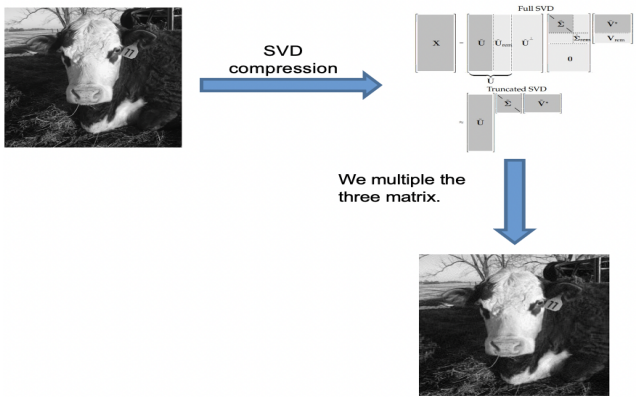

Illustration Compression image using SVD .

The algorithm with lossy image-compression that we chose to solve the problem is the Fast Fourier Transform (FFT), because this method has an appropriate time complexity to compute large amounts of data. Also, it has a great compression/lost information correlation of compressing.

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 An Animal Welfare Platform for Extensive Livestock Production Systems

Currently, agriculture policy reforms in the USA have favored livestock production showing respect to animal welfare. In order, this project was based on giving a solution for tracking and monitoring animal activity and behavior in livestock farms, to get the respective indicators about animal well-being. They designed an automated system with a type of wireless sensor, in this case, a collar device, to monitor the animal`s well-being according to their movement, speed, and geolocation information, with low implementation cost.

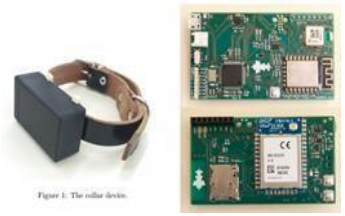

Illustration 1 Collar device to monitor the animals

The system was able to perform offline and real-time data processing for pattern recognition through Deep Neural Network pattern recognition algorithms. Finally, cloud computing processed both data and Deep Learning model storage, and the significant data were presented to the farmers in mobile devices.[4]

2.2 Visual Localisation and Individual Identification of Holstein Friesian Cattle via Deep Learning

In this work they prove using computer vision pipelines full of architectures of neural networks that the identification of individual Holstein Friesian can occur automatically and non-intrusively. The deep network that they use to address the problem of detecting and locating Holstein through images is the R-CNN adaptation of the VGG CNN M 1024 network published as part of several other network architecture proposals. And to detect them through videos, they use the standard Kernelized Correlation Filter (KCF) tracking algorithm.

The results they obtained were really good to prove what they want, however there were some mistakes and

identifications that were not well-done, some false positives, some identifications were less than what they expected and so on [5].

2.3 Cloud services integration for farm animals´ behavior studies based on smartphones as an activity sensor The problem that they analyzed was the optimization that the sensors need. First because of the power consumption, second the storage and the processing of large amounts of data and third the matching of data with complementary data. They describe a new infrastructure which brings benefits in storage, real-time processing and abilities for large scale data storage and analytics that allows to collect, store, treat and share information between scientists.

In the case of the compression, they performed it in two ways: First by eliminating redundancies and replacement of redundant data by a time interval during which the value remains constant was applied to preserve data integrity. And second by truncating data to 3, 4 and 5 decimal digits [6].

2.4 A systematic literature review on the use of machine learning in precision livestock farming
The project aims to reduce livestock environmental impact and identify the most appropriate process in livestock. With the recent concept of Precision Livestock Farming (PLF), which focuses on making decisions based on quantitative data obtained in real-time, they want to improve the farming process and propose technological solutions in agriculture and livestock production systems.

To get and study the data, this system uses data analysis, machine learning, control systems, and ICT.[7]

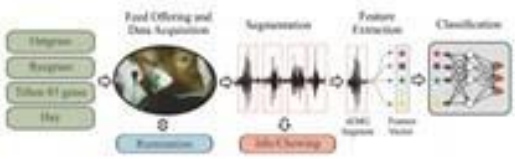

Illustration 2 Process to get data and study animals.

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/ST0245Eafit/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.

3.2.1 Fractal compression

Fractal compression is a lossy compression method for digital images, based on fractals. This method uses an algorithm to convert these parts into fractal codes which are used to recreate the encoded image. [8]

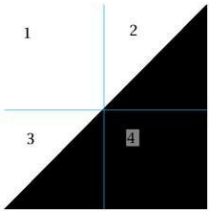

Illustration 3 Fractal compression method.

3.2.2 Seam Carving

Seam carving is an algorithm for content-aware image resizing. It establishes several seams in an image and automatically removes seams to reduce image size or inserts seams to extend it. How does it work? First, it receives an image. After that, it calculates the weight/density/energy of each pixel, this is done by an algorithm called gradient magnitude.

Illustration 4 Seam carving process.

From the energy, make a list of seams. Seams are ranked by energy, with low energy seams being of least importance to the content of the image. It removes low-energy seams as needed and finally, it returns the image.[9]

3.2.3 Discrete cosine transform

The Discrete Cosine Transform (DTC) is very important to the process of image/video compression because it has a very strong energy compaction [10]. It uses many arithmetic operations for that cause is very important to make the implementation of the DTC efficient in the real-time image transforming.

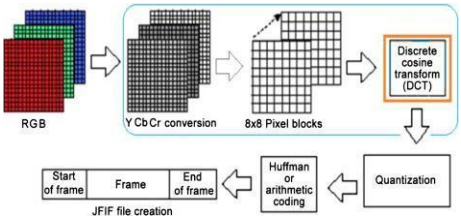

Illustration 5 Discrete cosine transform

Generally, to use the DTC´s algorithm, you need to get the image file information, then you divide it in a block of 8x8 matrix and the you apply the discrete cosine (a sum of cosine functions oscillating at different frequencies) [11][12]

3.2.4 Image Scaling

The image scaling refers to resizing or resampling a digital image. Depending on the images, they can be scaled using geometric transformations or create a new one with higher or lower number of pixels [13]. There exist three common scaling algorithms: 1. Nearest Neighbor Scaling. 2.Bilinear and 3. Bicubic Interpolation [14].

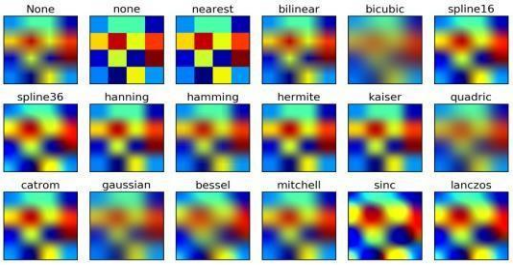

Illustration 6 Common scaling algorithm.

3.3 Lossless Image-compression alternatives

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

3.3.1 Huffman coding

The Huffman coding is an algorithm to compress data, basically for compressing files. The idea of the algorithm is to assign variable-length codes to input characters considering which are used and unused to turn out the most optimal code lengths [15]. There are mainly two major parts in Huffman Coding: 1. Build a Huffman Tree from input characters. 2. Traverse a Huffman Tree and assign codes to characters [16].

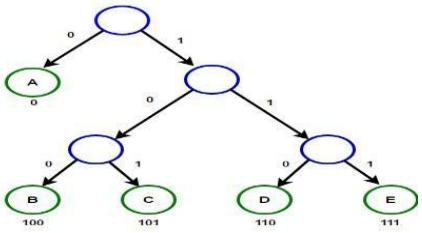

Illustration 7 Huffman coding process.

3.3.2 LZ77

The LZ77 algorithm is used to analyze the input data and find out how to reduce the size by replacing redundant information with metadata(XML-formatted data that defines the characteristics of an update) . Steps for the LZ77 algorithm: 1. Set the coding position to the beginning of the input stream. 2. Find the longest match in the window for the lookahead buffer. 3. If a match is found, output the pointer P. Move the coding position (and the window) L bytes forward. 4. If a match is not found, output a null pointer and the first byte in the lookahead buffer. Move the coding position (and the window) one byte forward. 5. If the lookahead buffer is not empty, return to step 2 [17].

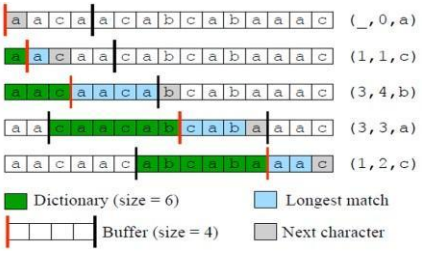

Illustration 8 LZ77 algorithm.

3.3.3 Burrows-Wheeler transformation

The Burrows-Wheeler transformation rearranges a character string into similar character strings. This is useful for compression and decoding using algorithms, as the move-tofront transform technique and run-length encoding make it easy to compress a string that has repeated character runs. The potential utility of the BWT of large

amounts of shortread data ("reads") has not been fully studied. The BWT basically serves as a dictionary of lossless reads. For example, in reading genomes, unlike the results of heuristic mapping and lossy reads that are conventionally obtained, the use of bwt is usually much more efficient. In the future, this is expected to lead to the development of sensitive methods for analyzing short-read data.[18]

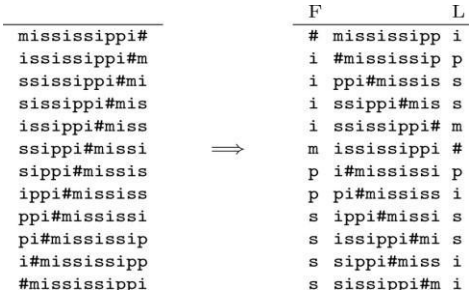

Illustration 9 Burrows-Wheeler transformation.

3.3.4 LZMA

In LZMA compression, the compressed sequence is a sequence of bits, encoded by an adaptive binary range encoder. The stream is divided into packets, each packet describing a single byte or LZ77 sequence with its length and distance encoded either implicitly or explicitly within each one. In addition, each part of each packet is modeled with independent contexts, so its probability predictions are correlated to each bit and this in turn is correlated with the values of the related bits of the same field) in previous packets of the same type.[19]

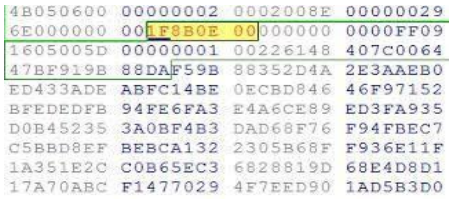

Illustration 10 LZMA algorithm.

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[ ]

4.1 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.

4.1.1 Lossy image-compression algorithm

Fast Fourier Transform (FFT)

The FFT is an optimized way to implement the Discrete Fourier Transform, in which the image´s data (the domain of pixels) is transformed into a set of frequencies in terms of sines and cosines. The above process generates a coefficient matrix where the greater coefficients represent the highest frequencies, generally we keep around the 5% to 1% of the coefficients of the matrix with greater frequency because they are the ones that contribute the most in the generation of the new data matrix that will be the pixels of the compressed image. [20]

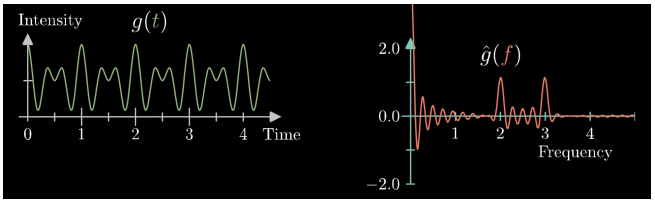

Illustration 11 Function in terms of frequency.

What we do with the minor coefficients is that we change them to 0 because they are insignificant in the processing data, so we don´t take it into account. Finally, we use a sparse matrix, implemented with a linked list of linked lists, in other words a graph, which skips the weight of the zeros in the file and just stores the most important data, and that makes that image´s weight less and the compression effectively.

Next to that, we load the image and make the decompression.

The decompression for this algorithm requires inversing the DFT matrix. To decompress, we are given a value representation of our polynomial and we want to find the coefficient representation, which means we have to multiply the value representation by the inverse of the DFT matrix.

REFERENCES

1. From Las Naciones Unidas Para La Alimentación y la Agricultura: http://www.fao.org/animalproduction/es/
2. Toro M. Marín,S. Proyecto Final ED1-Algoritmos de compresión para optimizar el consumo de batería en la ganadería de precisión.
3. D.Berckmans. General Introduction to precision livestock farming. January 2017. From academic oup:https://watermark.silverchair.com/6.pdf?token=AQECAHi208BE49Ooan9kkhW_Ercy7Dm3ZL_9Cf3qfKAc485ysgAAApMwggKPBgkqhkiG9w0BBwagggKAMIICfAIBADCCAnUGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQMraBI362XzIiqKIqAgEQgIICRqmH6juVF69UkXon95rvDfZ0Ty-CtAyDMInt1y_Ez980in2CePHe6fluNQucrTt_8g5s5aNmeTs53IFcamOcoayMEDKt1i5dH-mKlyc2Lb4wqYlKMmRdHBsMlPAt0UJkdKhKBiwKzP_tutvpgfTSPDNEuTiWxSzo8W9TJexyRVJTjpQkQ1bXBTA6HYY_Yau3sXyQXPcCMQPOXCjGBPMHjSUHkC95zRymZJY7jsgJjbGPxoUIrL6STbBznK6mUBGa6f9FegLe_I6rRMoy381qWQdQSMk3qkmt5PdjEhSf4dL7dR4bghyWrOp9W58n1BwLZ1sspEVpwI8bLfmGdymCGUl4ObPjhwkCtyFJvWNkkcRIRm_woLH9hj0CrCgksttwDZBvZPdeVrpymXpiPQEKlp1H4frGwurKLpTSAizToJdfgWb7tazx2Jep3N2bZt60fFMA_8soOfShkMjUVVWmPf-htP_TxRkFufu4IsC9DS5w9qipcOiOjmbTcSKazhH3rXTtn2GHJZk7brJEUGCSDNxGW_QCbUxYHtYJ5WwprotSpKlRbj580dTl57P_nQCAhKy_0h4gwtrNddSJKDHygzFXPVnM7U8BbJuCrai9HZxoTeLsQVT5eDbjNnkaW5Dprb4tuJe2MyBES3zLfuHlJk6rTgNkc0mSlA9KuZusvBm_fHup5x36uY4gFOYRTjLBQfO6h3FChE8ldcypcfNZ-iPEHOloJu-WHdY3yyjGQGoORGydEtu8NmOfMRrttwAanKA
4. Doulgerakis, V., Kalyvas, D. Bocaj, E. Giannousis, C., Feidakis, M. Laliotis, G. P. Patrikakis, C. Bizelis, I. , 2019. An animal welfare platform for extensive livestock production systems. In: CEUR Workshop Proceedings, vol. 2492.
5. Andrew, W., Greatwood, C., Burghardt, T., 2017. Visual localisation and individual identification of holstein friesian cattle via deep learning. In: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 2850–2859.
6. 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
7. Rodrigo Garcia, Jose Aguilar, Mauricio Toro, Angel Pinto, Paul Rodriguez. A systematic literature review on the use of machine learning in precision livestock farming. Computers and Electronics in Agriculture Volume 179, December 2020.
8. Wikipedia contributors. (2021, 18 enero). Fractal compression. Wikipedia. https://en.wikipedia.org/wiki/Fractal_compression
9. Wikipedia contributors. (2020, 7 noviembre).Seam carving.Wikipedia.https://en.wikipedia.org/wiki/Seam_carving
10. Xiuhua, J., Caiming, Z., Jiaye, W., Kai, W. 2009. A Fast 2D*8x8 Discrete Cosine Transform Algorithm for Image Coding. Science in China Series F Information Sciences 52(2):215-225 11. Kumar, A. 2021. Discrete Cosine Transform (Algorithm and Program). From:https://www.geeksforgeeks.org/discretecosine-transform-algorithm-program/
12. Wikipedia Contributors. Discrete Cosine Transform, 2020. From: https://en.wikipedia.org/wiki/Discrete_cosine_transform
13. Wikipedia Contributors. Image Scaling, 2021. From: https://en.wikipedia.org/wiki/Image_scaling
14. Tabora, V. JPEG Image Scaling Algorithms, 2019. From:https://medium.com/hd-pro/jpeg-imagescaling-algorithms-913987c9d588#:~:text=JPEG%20Scaling%20Techniques,nearest%20pixel%20in%20the%20output.&text=Bilinear%20%E2%80%94%20This%20interpolates%20pixels%20much%20better%20than%20Nearest%20Neighbor.
15. Huffman coding compression algorithm. From: https://www.techiedelight.com/huffman-coding.
16. Huffman coding. 2021. From : https://www.geeksforgeeks.org/huffman-codinggreedy-algo-3
17. LZ77 Compression algorithm, March, 2020. From: https://docs.microsoft.com/enus/openspecs/windows_protocols/mswusp/fb98aa28-5cd7-407f-8869-a6cef1ff1ccb.
18. Kimura, K., & Koike, A. (2015, 9 diciembre).

Analysis of genomic rearrangements by using the Burrows-Wheeler transform of short-read data.BMC Bioinformatics. https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-16-S18-S5

19. lzma — Compression using the LZMA algorithm — Python 3.9.1 documentation. (2021, enero). Compression using the LZMA algorithm. https://docs.python.org/3/library/lzma.html
20. Steven L. Brunton,J. Nathan Kutz(2019)Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control
21. Steve Brunton. (2020, 31 marzo). The Fast Fourier Transform (FFT). YouTube. https://www.youtube.com/watch?v=E8HeD-MUrjY&t=116s
22. Reducible. (2020, 14 noviembre). The Fast Fourier Transform (FFT): Most Ingenious Algorithm Ever? YouTube. https://www.youtube.com/watch?v=h7apO7q16V0&t=789s
23. Steve Brunton. (2020b, abril 4). The Fast Fourier Transform Algorithm. YouTube. https://www.youtube.com/watch?v=toj_IoCQE-4&t=175s
24. Steve Brunton. (2020c, junio 8). Image Compression and the FFT (Examples in Python). YouTube. https://www.youtube.com/watch?v=uB3v6n8t2dQ
25. Creative. (2019, 10 agosto). Fast Fourier Transform (FFT) Algorithm. ALLSIGNALPROCESSING.COM. https://allsignalprocessing.com/lessons/fast-fourier-transform-fft-algorithm/

IMAGES` REFERENCE

1. Doulgerakis, V., Kalyvas, D. Bocaj, E. Giannousis, C., Feidakis, M. Laliotis, G. P. Patrikakis, C. , Bizelis, I. , 2019. An animal welfare platform for extensive livestock production systems [Fotography]. In: CEUR Workshop Proceedings, vol. 2492.
2. Rodrigo Garcia, Jose Aguilar, Mauricio Toro, Angel Pinto, Paul Rodriguez. A systematic literature review on the use of machine learning in precision livestock farming [Fotography]. Computers and Electronics in Agriculture Volume 179, December 2020.
3. Wikipedia contributors. (2021, 18 enero). Fractal compression [Photography]. Wikipedia. https://en.wikipedia.org/wiki/Fractal_compression
4. Wikipedia contributors. (2020, 7 noviembre).Seam carving [Photography] .Wikipedia.https://en.wikipedia.org/wiki/Seam_carving.
5. Spie Digital Library (2014, september). Discrete Cosine Transformation [Fotography]. https://www.spiedigitallibrary.org/journals/journal-of-electronic-imaging/volume-23/issue6/061110/Adaptive-discrete-cosine-transformbased-image-compression-method-ona/10.1117/1.JEI.23.6.061110.short?SSO=1&tab=A rticleLink
6. Super User (2016, August). Image Scaling Algorithm [Photography]. https://superuser.com/questions/375718/whichresize-algorithm-to-choose-for-videos
7. Techie Delight. Huffman Coding [Fotography]. https://www.techiedelight.com/huffman-coding/.
8. Shi, P., Li, B., Thinke, B., Ding, L. (2018, January). LZ77 encoding example [Photography]. https://www.researchgate.net/figure/An-exampleof-LZ77-encoding_fig4_322296027
9. Ferragina, P. (2005, july). Burrows wheeler transform [Photography]. https://www.researchgate.net/figure/Example-ofBurrows-Wheeler-transform-for-the-string-Tmississippi-The-matrix-on-the_fig5_220430619.
10. Waste, G. (2014, 8 october). LZMA compression method [Fotography]. LZMA compression method in GZIP files. https://forum.xentax.com/viewtopic.php?f=21&t=12065.