Making the Precision Livestock Farming consume less energy

David Cuadros
Universidad Eafit
Colombia
Dacuadrosm@eafit.edu.co

Juan Jácome
Universidad Eafit
Colombia
jsjacomeb@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. Delete all text in red. 2. Adjust spaces among words and paragraphs. 3. Change the color of all the texts to black.

Red text = Comments

Violet text = To complete for the 3rd deliverable

The Precision Livestock Farming (PLF) which objective is to improve the process of the livestock farming at the fact of include information and communication technology, which is many times ignored, because Stefanova found that the data of the farm is in notebooks or into the zootechnical information systems.

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

The goal of this project improves the energy conception of the PLF which is going to give to the farmers more useful information of the animals and consume less time.

1.1. Problem

The problem is to create an algorithm which can compress and decompress images which can optimize the information about the animal's health, which is consumes more of the necessary time.

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.

The first compressor that we used is the seam carving which removes or insert seam in case of wanting to remove the image size or extend it, and choose which one may not be modified, and features the ability to remove whole object from photographs. To later use the Huffman coding which is a lossless one which find an estimated probability or frequency of occurrences for each possible value of the source symbol.

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 IoT based Animal Health Monitoring with Naïve Bayes Classification.

This study is to analyze parts of the Wireless Sensor Network like the body temperature sensor, heart rate sensor, etc. To monitor the health of the animals that ended increasing the productivity parts like the nodeMCU microcontroller, animal body temperature, humidity, etc. and implementing the Naïve Bayes algorithm.

Taken from:

https://web.archive.org/web/20180423042443id_/http://www.ijett.in/index.php/IJETT/article/viewFile/323/211

2.2 Standardising Syndromic Classification in Animal Health Data.

This study is classification of the animal health data into syndromes with syndromic surveillance, standardizing the classification of records into syndromes that helps to record the data using an institution´s own vocabulary, compare Veterinary Syndromic Surveillance (VSS) outputs within and between countries, and more time development of VSS systems.

Taken from:

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4512370/#__sec4title

2.3. Classification of cattle behavior using convolutional neural networks.

This study tested acceleration-based activity collar collars can through Convolution Neural Networks (CNN) which were able to classify states of rumination, eating, and other with an overall F1 score of 82% which were compared to reported collar classification with an overall F1 score of 72%.

Taken from:

https://strathprints.strath.ac.uk/73143/1/Pavlovic_etal_EAAP2020_Classification_of_cattle_behaviour_using_convolutional_neural_networks.pdf

2.4 Web-based cattle behavior service for researchers based on the smartphone inertial central.

This study proposes and describe infrastructures allowing to collect, store, threat, and share information between scientists, showing that Docker can be faster to Flock port (LXC) if it uses multi-layer unification files system without using of network translation modules, and that the use of a GPS and an accelerometer implanted on the neck of a cow to detect behavior with more than 90% already achieved in Gonzales et al[1].

Proposed new data storage architecture. Lambda is bale to collect data at high frequency and is adaptable, and the IPhones are inexpensive mean of measuring cow behavior, but their gateway by using the UDP protocol can cause a data packet loss problem when it needs to collect data from several IPhones simultaneously.

Taken from: https://pdf.sciencedirectassets.com/280203

3.1 A Plant Image Compression Algorithm Based on Wireless Sensor Network

The problem is the need to transmit plant information effectively, the solution uses a high-availability image compression algorithm which effectively reduces the volume of transmitted data and uses less energy. The aspects of improvement are in the pixels which can be reduced and in the use of the JPEG image compression algorithm and changing the value of the quantization table in the algorithm.

Sun, G. , Chu, Y. , Liu, X. and Wang, Z. (2019) A Plant Image Compression Algorithm Based on Wireless Sensor Network. Journal of Computer and Communications, 7, 53-64. doi: 10.4236/jcc.2019.74005.

3.2 Development and validation of a visual image analysis for monitoring the body size of sheep, Journal of Applied Animal Research

The problem solved is the measurement of the body size of sheep in order to reflect its growth development, production performance and genetic characteristics. The algorithms used are an automatic foreground extraction and a measuring point detection algorithm. The experimental evidence suggests the method is effective and the errors in measurement are within 3%

A. Li Na Zhang, B. Pei Wu, C. Xin Hua Jiang, D. Chuan Zhong Xuan, E. Yan Hua Ma & F. Yong An Zhang (2018) Development and validation of a visual image analysis for monitoring the body size of sheep, Journal of Applied Animal Research, 46:1, 1004-1015, DOI: 10.1080/09712119.2018.1450257

3.3 A lossless compression method for multi-component medical images based on big data mining

The problem solved is the lossless compression of disease diagnosis, in order to reduce local storage space and communication bandwidth used. The algorithm proposed is a soft compression algorithm for multi component medical images. As a conclusion they define soft compression as a much better method than traditional Huffman coding in lossless image compression, because it aims to remove coding and spacial redudancy.

Xin, G., Fan, P. A lossless compression method for multi-component medical images based on big data mining. Sci Rep 11, 12372 (2021). https://doi.org/10.1038/s41598-021-91920-x

3.4 Modified HuffBit Compress Algorithm – An Application of R

The problem solved is that the databases of genomic sequences are growing at an explicative rate because of the increasing growth of living organisms. They propose “HuffBit Compress” as a solution, an algorithm based on the concept of an Extended Binary Tree using the R language as it givers the best case of the compression ratio but using an extra 6 bits. They conclude that the “HuffBit Compress” is 16.18% faster in R language and 11.12% over the “2-bits encoding method”

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/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.

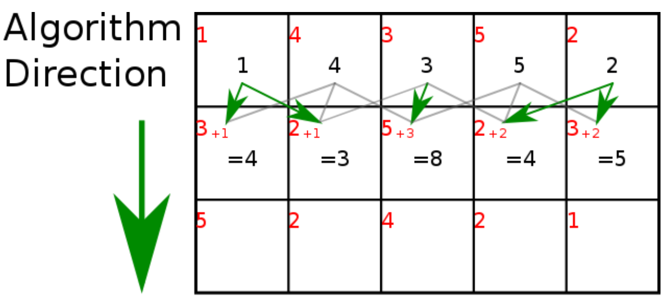
3.2 Lossy Image-compression alternatives

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 *Seam carving*

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

Algorithm for content-aware image resizing. It was developed by Shai Avidan and Ariel Shamir. How it works is that it establishes a number of paths of least importance in an image and eliminates them in order to reduce the image size. For a W×H image, the time complexity is O(W×H+W+H).

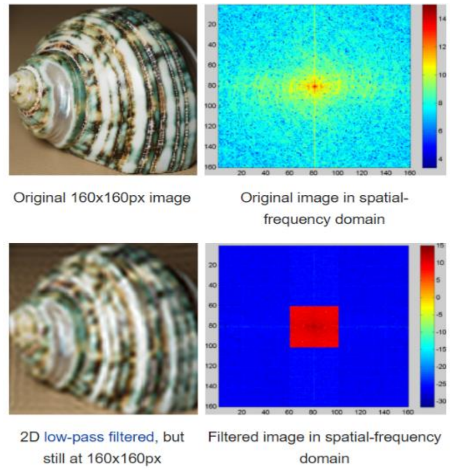


From: https://en.wikipedia.org/wiki/Seam_carving

3.2.2 *image scaling*

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

image scaling refers to the resizing of a digital image. In video technology, when scaling a vector graphic image, the graphic primitives that make up the image can be scaled using geometric transformations, with no loss of image quality.



From: https://en.wikipedia.org/wiki/Image_scaling#Vectorization

3.2.3 *discrete cosine transform*

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

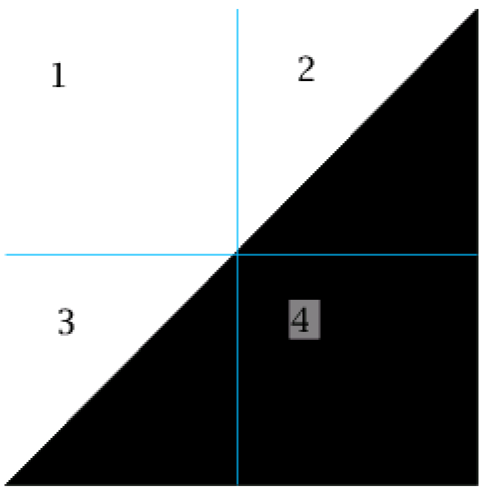
A discrete cosine transform (DCT) expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. The DCT, first proposed by Nasir Ahmed in 1972, is a widely used transformation technique in signal processing and data compression.

From: https://en.wikipedia.org/wiki/Discrete_cosine_transform

3.2.4 *fractal compression*

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

Fractal compression is a lossy compression method for digital images, based on fractals. The method is best suited for textures and natural images, relying on the fact that parts of an image often resemble other parts of the same image. Convert these parts into mathematical data called "fractal codes" which are used to recreate the encoded image.



From: https://en.wikipedia.org/wiki/Fractal_compression

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 *Borrows & Wheeler Transform*

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

The Burrows–Wheeler transform is an algorithm used to prepare data for use with data compression techniques such as bzip2. It was invented by Michael Burrows and David Wheeler in 1994. This is useful for compression, since it tends to be easy to compress a string that has runs of repeated characters by techniques such as move-to-front transform and run-length encoding.

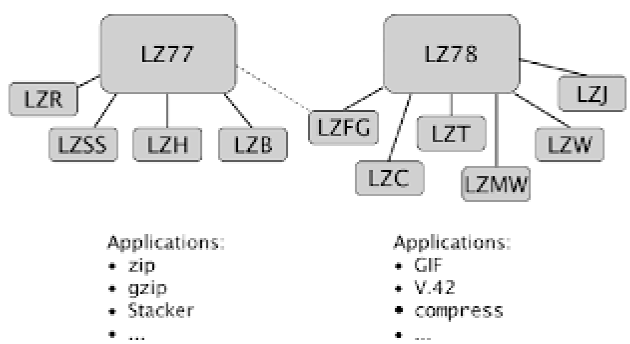
Transformation				
1. Input	2. All rotations	3. Sort into lexical order	4. Take the last column	5. Output
^BANANA|	^BANANA| |^BANANA A|^BANAN NA|^BANA ANA|^BAN NANA|^BA ANANA|^B BANANA|^	ANANA|^B ANA|^BAN A|^BANAN BANANA|^ NANA|^BA NA|^BANA ^BANANA| |^BANANA	ANANA|^B ANA|^BAN A|^BANAN BANANA|^ NANA|^BA NA|^BANA ^BANANA| |^BANANA	BNN^AA|A

From: https://en.wikipedia.org/wiki/Burrows%E2%80%93Wheeler_transform

3.3.2 *LZ77 and LZ78*

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

LZ77 and LZ78 are the two lossless data compression algorithms published in papers by Abraham Lempel and Jacob Ziv in 1977 and 1978. They are also known as LZ1 and LZ2 respectively. They are both theoretically dictionary coders. LZ77 maintains a sliding window during compression. LZ77 algorithms achieve compression by replacing repeated occurrences of data with references to a single copy of that data existing 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.



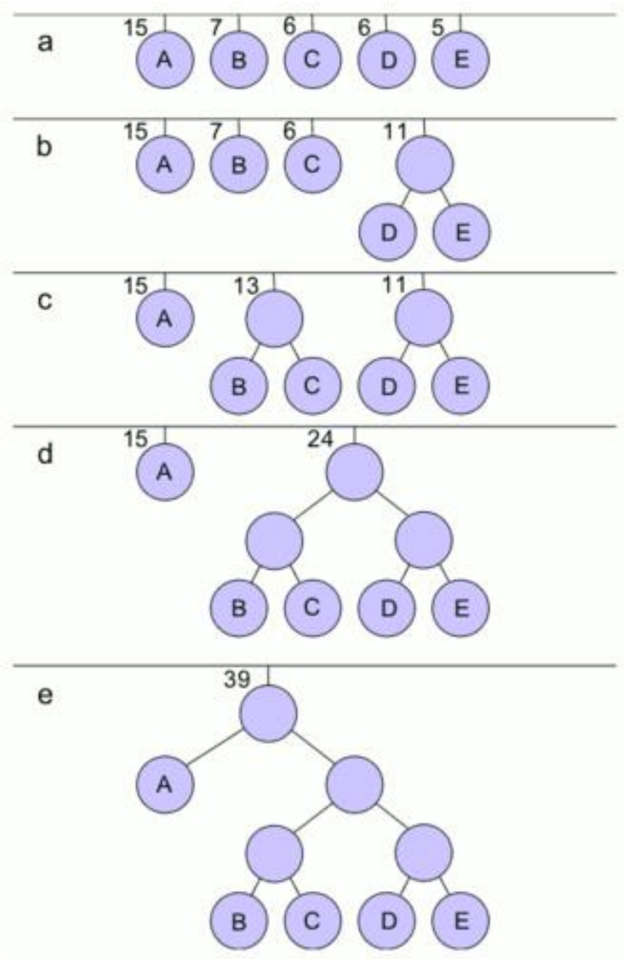
Image_From_:https://www.semanticscholar.org/paper/The-Lempel-Ziv-Algorithm-Zeeh/6227063dd63b22d18e3dad2a6f247c4f80561081

Information ftom: https://en.wikipedia.org/wiki/Burrows%E2%80%93Wheeler_transform

3.3.3 *Huffman coding*

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

Huffman code is a particular type of optimal prefix code that is commonly used for lossless data compression, developed by David A. Huffman. The idea is to assign variable-length codes to input characters, lengths of the assigned codes are based on the frequencies of corresponding characters. The most frequent character gets the smallest code and the least frequent character gets the largest code.



From : https://en.wikipedia.org/wiki/Huffman_coding

3.3.4 *LZS*

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

. Is a lossless data compresion algorithm, it basically is as follows: a 0 bit indicates that the next eight bits are just a literal byte. A 1 bit indicates either a string to copy from the previous data or the end of the compressed data depending on the subsequent bits. The 1 bit is followed by an offset field of either 8 or 12 bits, and that is followed by a length field of variable length of 2, 4, 8, 12, etc. bits depending on the length. Or the 1 is followed by a specific 9-bit end marker.

From :https://en.wikipedia.org/wiki/Lempel%E2%80%93Ziv%E2%80%93Stac

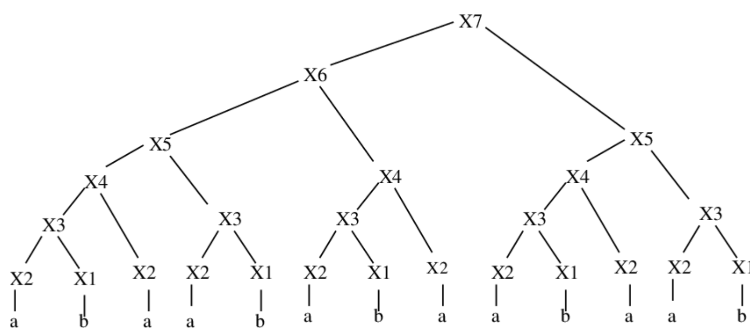


Image: https://www.semanticscholar.org/topic/Lempel%E2%80%93Ziv%E2%80%93Stac/2659343

4. ALGORITHM DESIGN AND IMPLEMENTATION (Array)

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

An array is a container object that holds a fixed number of values of a single type.

[1]http://www.github.com/ ????????? /proyecto/

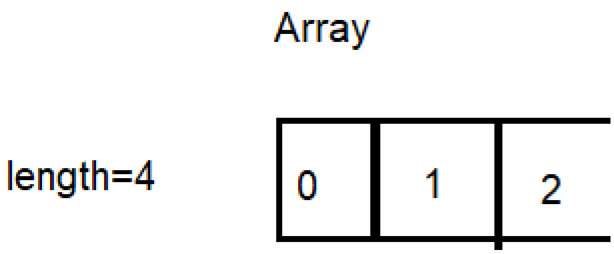


Figure 1: Representation of an Array with its respective index.

4.2 Algorithms

4.2.1 Seam Carving

Seam carving (or liquid rescaling) is an algorithm for content-aware image resizing, It functions by establishing a number of seams (paths of least importance) in an image and automatically removes seams to reduce image size or inserts seams to extend it.

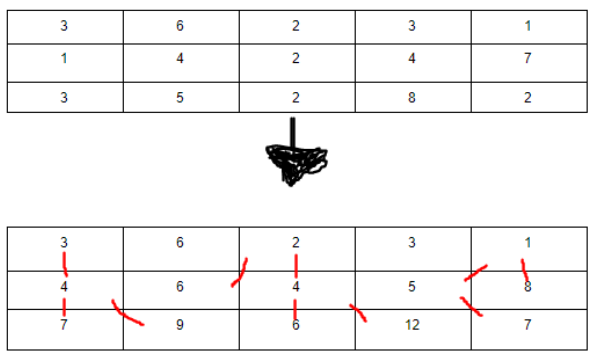


Figure 2: graphical description Seam Carving

4.2.2 Lossless image-compression algorithm

Huffman coding: A minimal variable-length character coding based on the frequency of each character. First, each character becomes a one-node binary tree, with the character as the only node. The character's frequency is the tree's frequency.

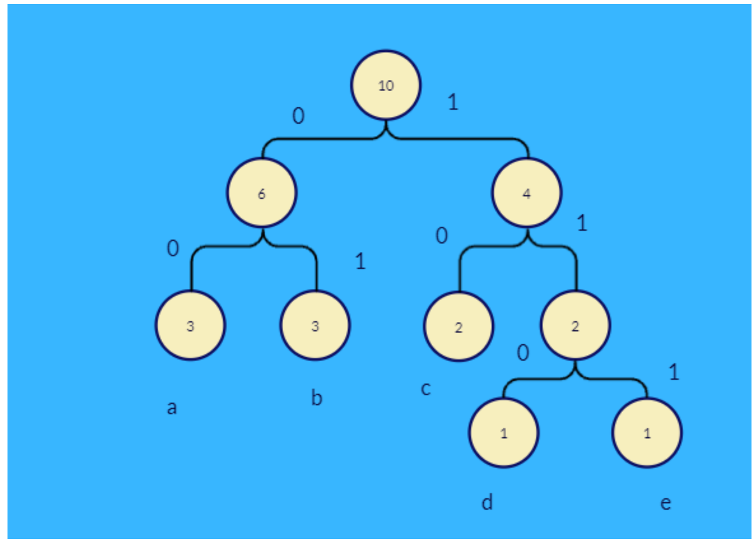


Figure 3 Huffman coding python implements

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

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. González LA, Bishop-Hurley GJ, Handcock RN, Crossman C. Behavioral classification of data from collars containing motion sensor in grazing cattle. Computers and Electronics in Agriculture. 2015; 110: 91-102. DOI: 10.1016/j.compag.2014.10.018