Evaluating Binary Encoding Techniques for WiSARD

Andressa Kappaun, Karine Camargo, Fabio Rangel, Fabrício Firmino,
Priscila Machado Vieira Lima, Jonice Oliveira
Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brazil
e-mail: {andressakappaun, karinecamargo, fabiorangel, firminodefaria}@ufrj.br,
priscila.lima@nce.ufrj.br, jonice@dcc.ufrj.br

***Abstract*—Many weightless neural networks, such as WiSARD, are RAM-based classifiers that receive binary data as input. In order to convert raw data into binary input, several techniques are applicable. This work evaluates the impact of some of these binarization techniques on the accuracy of two types of classifiers: WiSARD model and WiSARD with bleaching mechanism. The binary encoding techniques explored were: (i) *thermometer*, (ii) threshold, (iii) local threshold, (iv) Marr-Hildreth filter, and (v) Laplacian filter. The MNIST digit dataset was used to compare the accuracy obtained by each encoding technique. Results showed a difference of more than 20% in the accuracy due to the choice of encoding approach.**

I. Introduction

Weightless neural networks, such as WiSARD (Wilkie, Stonham & Aleksander's Recognition Device) [1], are based on networks of Random Access Memory (RAM) nodes [2]. This set of classifiers require's binary data input to classification tasks, despite few works such as the one proposed by Souza in 2014 [3], which uses a ternary codification system. In the WiSARD model, binary data input is used to address memory positions. Binary encoding process usually impacts on the classification results, mainly because it usually implies in loss of information. Many techniques can be applied in order to obtain a binary data representation, and choosing the ideal technique for the application is crucial to obtain good classification results.

Literature presents many approaches for binary encoding over the input data when dealing with weightless neural networks. For instance, in [4] WiSARD is used to perform opinion mining (text categorization) over social network data sets. To encode the text into a feature vector, the work applied bag-of-words model. Bag-of-words model had to be slightly changed, considering only the presence of each term, ignoring the number of appearances or frequencies. The presence and absence of each term is already a binary representation, enabling the use of a weightless neural network.

When developing a binary encoding for the WiSARD input, one must take in account the Hamming distance [5] between different patterns. A naive approach for the data binarization could compromise the classification accuracy. In the work of [6], a part-of-speech tagger was created using the WiSARD architecture. Part of the input was composed of probabilities associated with terms. As probabilities lies in the continuous range $[0, 1]$, it was necessary to discretize it. Furthermore, the discrete space had to be binary. The solution was the creation of a bit string using d bits, simulating a scale that was named as *thermometer*. The change in data representation implied in a loss of information, since close probabilities were represented as the same. *Thermometer* encoding considers the Hamming distance between patterns.

Many works deal with image classification using weightless neural networks. When dealing with grayscale images [7], it is possible to implement a binary encoding using a threshold. This is made by analyzing the intensity value of the pixel. If this value is greater than the threshold, it is encoded by one, zero otherwise. However, it is possible to implement the *thermometer* technique in the same scenario, reducing the loss of information that a simple threshold could entail. In [8], a similar approach was used to an image classification task. In this case, colored images were used and each color channel was represented by a *thermometer* of 192 bits.

MNIST database of handwritten digits [9] is a well known dataset used to compare the performance of many classification models by evaluating the error rate (%) in its test data [10]. In [7], MNIST was used to evaluate the performance of WiSARD, but did not compare different binary encoding approaches, neither presented the binary encoding used. However, it explored a comparison between different numbers of bits for WiSARD, which is an important parameter for good classification results. The work described in [7] also introduced an improvement in WiSARD: the bleaching mechanism.

The present work compares many possible binary encoding approaches, evaluating WiSARD accuracy and standard deviation, using and not using bleaching. Aiming the direct comparison with [7], MNIST was used as dataset. Considering that corners and edges are the most important regions for an image, i.e. regions with high level of information, this work also applies image processing techniques, such as Marr-Hildreth and Laplacian filters [11], in order to obtain these regions, changing the image representation to a binary encoding. As main result, it was possible to find that the best binary encoding method for this dataset was the *thermometer*, which led WiSARD to an accuracy of 94%, however it increased the number of features, if compared with other approaches. The worst binary encoding tested was the threshold higher than

200. Another interesting finding is the fact that *thermometer* size has low impact in the 32 bits WiSARD.

A review of WiSARD is presented in the Section II; Section III describes the dataset, the binary encoding techniques used, and the experiment design; Section IV discuss the results obtained; and the conclusions and future works are presented in the Section V.

II. WiSARD Model

WiSARD (Wilkie, Stonham & Aleksander's Recognition Device) is RAM-based classifier, also considered a weightless neural network model. WiSARD was introduced in [1], but it is based in the N-Tuple Classifer, presented in [12]. N-Tuple Classifier is an one-class classifier, and WiSARD architecture is composed by many of them, each one is called a Discriminator associated to one category of the classification problem.

Initially, all memories are set to zero. A pseudo-random mapping is created in order to split the input pattern into n tuples. In the training phase, when a pattern from a specified category is presented, the Discriminator responsible for this category uses the tuples to address its RAMs. Each tuple is responsable for the address of a RAM and the positions accessed during the training step are set to one. The Figure 1 shows one Discriminator training an input pattern, where the connections represent the pseudo-random mapping.

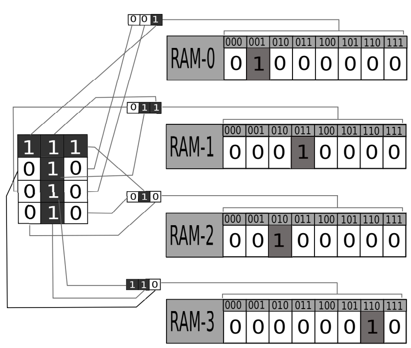

Fig. 1. WiSARD Discriminator training an input pattern [6].

When classifying the category of an input pattern, all Discriminators receive the pattern and use the same pseudo-random mapping as the training to address the RAMs. Each Discriminator calculates the number of non-zero accessed positions. This number is the example's degree of membership for the Discriminator category. The Figure 2 shows a pattern being presented to a Discriminator. It is possible to notice that the RAM number 3 accessed a zeroed position. At the end, WiSARD classify the example as the category of the Discriminator which responds the higher degree, comparing all discriminators. The Figure 3 presents the degree of membership of a given example for each Discriminator.

In [7], WiSARD was improved with a technique called bleaching mechanism. Bleaching was created aiming to work around the saturation problem: when a large number of examples are trained, any noise in the training data can spoil the classification. Using WiSARD with bleaching, memories

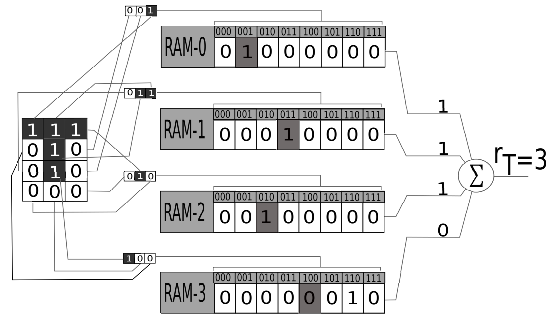

Fig. 2. WiSARD Discriminator verifying the degree of membership of an input pattern [6].

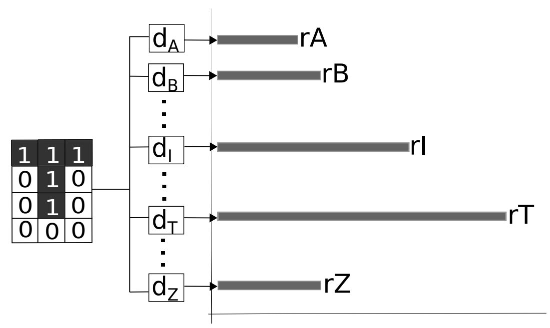

Fig. 3. Discriminators response after the presentation of a pattern to be classified [6].

must store an integer, in order to accumulate the training. When a pattern is presented during the training, the accessed positions are increased by 1. During the classification, each Discriminator calculates the number of accessed positions with stored value higher than a bleaching value b. The b value is incremented while the confidence c in the classification is not higher than a confidence threshold. The confidence c is calculated using the higher and the second higher degree of membership from the Discriminators. The Equation 1 shows how the confidence is calculated, and its value lies in the range $[0, 1]$, and r_1 is the higher degree as long the r_2 is the second higher. The Figure 4 presents the bleaching mechanism inside a Discriminator. In Figure 4, the actual bleaching value is 3, and only memories accessing positions in which values stored are equal or higher than 3 are going to be counted in the summing.

$$c = 1 - \frac{r_2}{r_1} \tag{1}$$

III. Methodology

In this section, the used binary encoding techniques are introduced. The present work compares these techniques, evaluating WiSARD's accuracy and standard deviation, using three different number of bits (8, 16 and 32), and the same dataset as in [7].

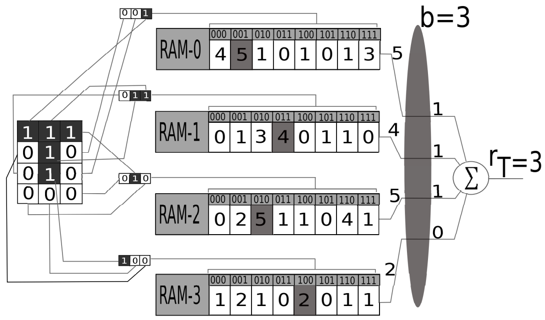

Fig. 4. Classification with bleaching mechanism inside a Discriminator [6].

A. Binary Encoding Approaches

MNIST dataset has the grayscale intensity for each pixel. The following techniques basically changes the pixel values for a new representation. This representation may produce feature vectors with different number of features from the input space. In addition to the binary encoding presented in the field of weightless neural networks, this work presents image processing techniques, such as Difference of Gaussians (Marr-Hildreth filter) and Laplacian filter, which can be found in [11].

1) Threshold: Threshold is the most used technique for binary encoding. Using this technique, if the pixel value is equal or higher than the threshold, the pixel is changed to 1. Otherwise, it is changed to 0. In this work, thresholds are tested between the range [1,254] in order to find the best threshold.

2) Thermometer: Thermometer encoding defines a zeroed vector of d length such that each position is a threshold encoding. For each pixel, the vector positions will be changed to one, from the begin until the first threshold higher than the pixel value. After, the vector will be the representation of this pixel. In the end, the length of the feature vector will be $d \times n$, where n is the number of pixels. The input pixel in the example on Figure 5 has intensity value of 130, the binary encoding for this pixel is: 1111100000, $d = 10$ and the thresholds are: 25, 51, 76, 102, 127, 153, 178, 204, 229, 255. The thresholds were obtained dividing the pixel space in d parts. This work used 3 distinct *thermometers* with $d = 5, 10, 20$.

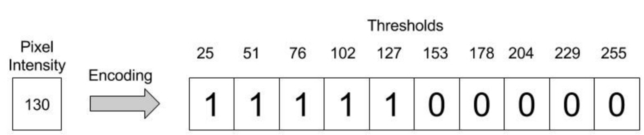

Fig. 5. *Thermometer* binary encoding example.

3) Marr–Hildreth Filter: The most widely used smoothing filter is the Gaussian filter [13]. Difference of Gaussian filter is a fast edge enhancement algorithm that involves subtracting a blurred image from a less blurred one, from the same image. Also known as Marr-Hildreth algorithm for detecting edges, based on Laplacian of the Gauss function [14]. After the filtering, it is possible to obtain the binary encoding by applying a zero threshold.

4) Laplacian Filter: Discrete Laplacian operator was applied in order to enhance the image edges. This approach is close to the previous one, but the used convolution kernel was not considering the diagonal positions. The kernel can be seen in the Equation 2. After the convolution, it is possible to obtain the binary encoding by the same way as Marr–Hildreth filter.

$$kernel = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{2}$$

5) Local Threshold: Local threshold is an adaptive threshold approach for binary encoding. It consists in calculating the average local intensity, using a window. This value is used as threshold for the pixels in the current window. The present work used a 2×2 window to establish the local area. This binary encoding technique was applied for a motion track challenge in the paper of Nascimento *et al.* [15].

B. Dataset

MNIST Database of Handwritten Digits [16] is a famous dataset used to evaluate the performance of many classification algorithms [10]. The dataset has a training set of 60,000 examples and a testing set of 10,000 examples. Each example is a grayscale image of 28×28 pixels, representing a handwritten digit from 0 to 9.

C. Experiment Design

To provide a clean comparison, firstly the best threshold was found. Then, the three *thermometers* were compared to understand their impact. The last analysis compares the best thresholds and *thermometers* with other binary encoding approaches.

WiSARD uses a random mapping to generate the tuples. This random mapping may influence the accuracy and, after the training phase, same models can present different results. Moreover, each model was trained with 6000 random examples. Due to these characteristics, the validation had to be repeated, which was done 30 times. The provided accuracy is the average accuracy, with the respective standard deviation. The validation consisted in training a model with 6000 random examples and testing in the testing set, which has 10000 examples. It is important to consider that WiSARD using bleaching needed an additional parameter: confidence threshold. This work fixed the confidence threshold $c = 0.1$. The experiment used an open source library for WiSARD implementation, and it is available on Github [1].

[1]https://github.com/firmino/libwann

IV. Results

A. Threshold

WiSARD can use bleaching, and the number of bits is an essential parameter for the model. WiSARD with and without bleaching can be addressed to different domains or purposes. The threshold influence for models not using bleaching is presented in the Figure 6, and the models using bleaching are presented in the Figure 7.

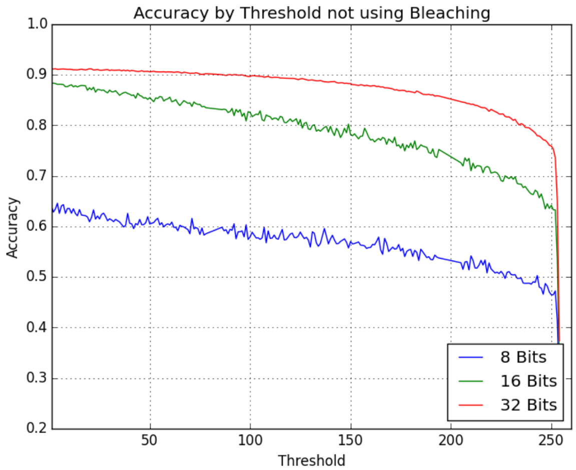

Fig. 6. Accuracy by threshold of a standard WiSARD with 8, 16 and 32 bits.

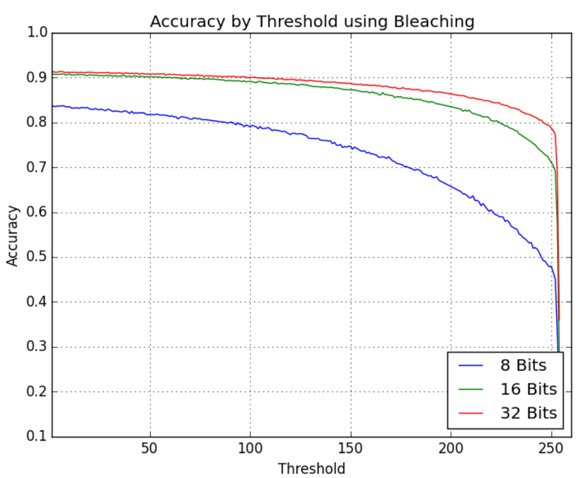

Fig. 7. Accuracy by threshold of a bleaching WiSARD with 8, 16 and 32 bits.

WiSARD using 32 bits was superior than 16 and 8 bits in both threshold analysis presented in Figure 6 and Figure 7. Furthermore, it is possible to notice that bleaching made the classification more stable when varying the threshold for 8 and 16 bits. This is an important finding for WiSARD research field.

This analysis also presented 1-threshold as the best threshold for this dataset, which resulted in an accuracy superior than 90% for both cases. This visualization also gives an important information: higher thresholds result in lower accuracies.

B. Thermometer

Three different size of *thermometers* were compared together: 5, 10 and 20. The first comparison is presented in Figure 8, in which WiSARDs were not using bleaching mechanism. For these models, the accuracy increased as the *thermometer* size increased. One must take in account that increasing the *thermometer* size also increases the computational cost, as it enlarges the feature vector size. The best number of features for all tested *thermometers* was 32 bits.

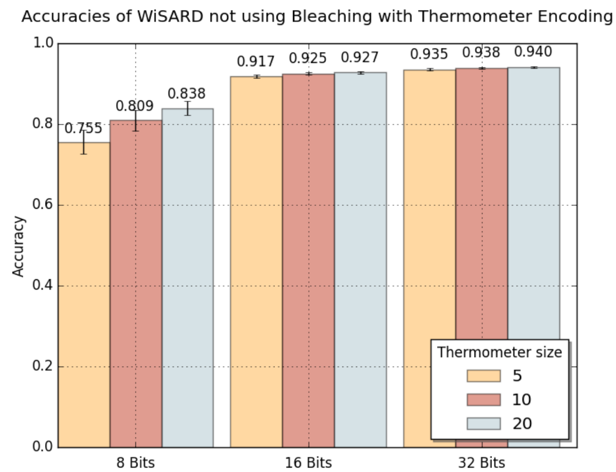

Fig. 8. *Thermometer* encoding analysis using three different *thermometers*. Standard WiSARD using 8, 16 and 32 bits accuracies evaluation.

The Figure 9 presents the result of the three *thermometers* encoding for WiSARD using bleaching mechanism. For these models, the accuracy decreased as the *thermometer* size increased, different from models using bleaching. This behavior may be caused by the fixed confidence threshold parameter c, which was 0.1. Different values for c could lead to better results, but it would demand a hyper-parameter optimization (model selection) which is the problem of choosing a good set of hyper-parameters for a learning algorithm [17].

C. Image Processing Approaches

The Figures 10 and 11 present the analysis using image processing approaches, respectively for standard WiSARD (not using bleaching) and WiSARD using bleaching. For the WiSARD not using bleaching, the three approaches presented close results for 8 bits. For 16 bits the Difference of Gaussians provided a better accuracy, and the local threshold, though competitive, had a standard deviation of 2.5%. For the model using 32 bits, local threshold accuracy was quite superior than other approaches accuracies.

The WiSARD with bleaching had superior results than the standard version for the analysis using image processing. Local threshold had better results, presenting close standard deviations to others. The Laplacian filter presented the worst

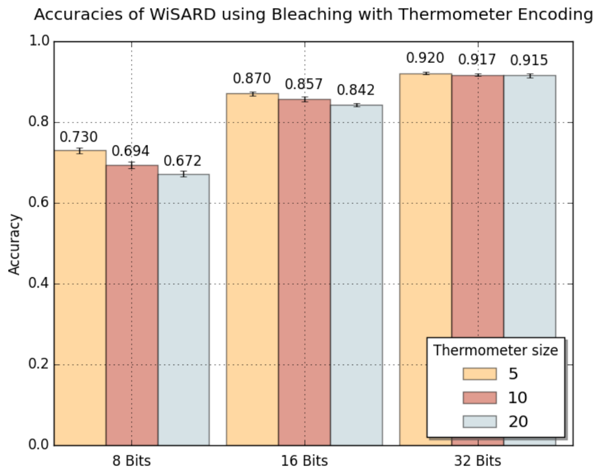

Fig. 9. *Thermometer* encoding analysis using three different *thermometers*. WiSARD with bleaching using 8, 16 and 32 bits accuracies evaluation.

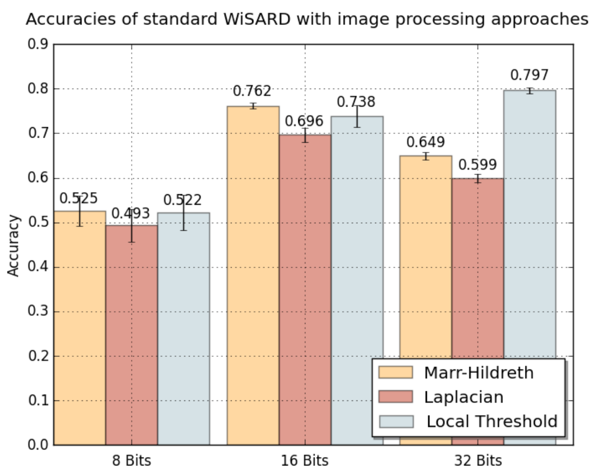

Fig. 10. Image processing approaches evaluation. WiSARD not using bleaching with 8, 16 and 32 bits.

accuracies. Both Laplacian and Marr-Hildreth filters were discarded for the the joint analysis due the non competitive accuracies.

D. Joint Analysis

The joint analysis compares the best results from the previous analysis. In the Figure 12, the accuracy classification of WiSARD without bleaching is compared using the 1-threshold, the local threshold, and the 20-*thermometer*. The 20-*thermometer* provided an accuracy of 94%, which is superior than the accuracies obtained by the other techniques. Although, it is important to consider that 1-threshold had a good performance and is much faster than the 20-*thermometer* encoding, given the size of the feature vector.

Analyzing Figure 13, WiSARD with bleaching had superior accuracy using the 1-threshold for 8 and 16 bits with standard deviation inferior to 0.5%. For 32 bits, this model presented

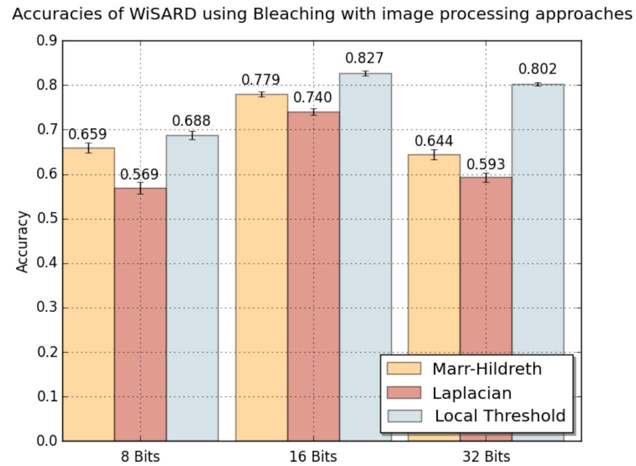

Fig. 11. Image processing approaches evaluation. WiSARD using bleaching mechanism with 8, 16 and 32 bits.

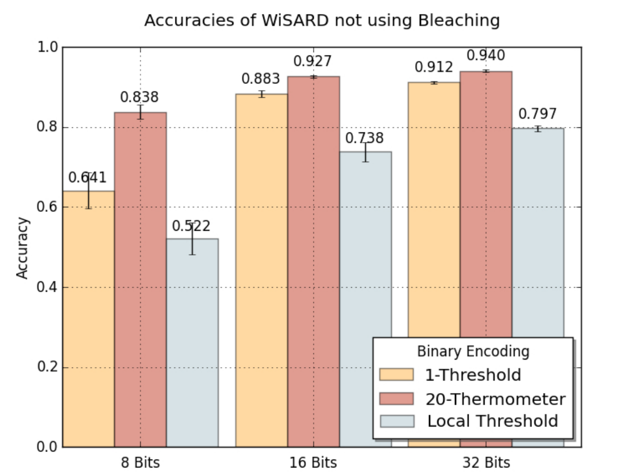

Fig. 12. Best *thermometer*, threshold and image processing approach compared. Standard WiSARD using 8, 16, and 32 bits.

the higher accuracy with 5-*thermometer* encoding, although the difference to the 1-threshold was less than 1%, with a feature vector 5 times smaller. Both standard deviations were inferior than 0.3%.

One possible reason for the image processing techniques results is that these techniques generate feature vectors with higher sparsity. This condition may be ”forcing” WiSARD to classify examples for what they ”do not have”. One possible way to deal with sparse input patterns to WiSARD is discussed in [4]. The present work did not analyze the sparsity in the feature vectors after the application of binary encoding techniques.

V. Conclusion and Future Works

This work presented an evaluation of binary encoding techniques using WiSARD. The encoding techniques used

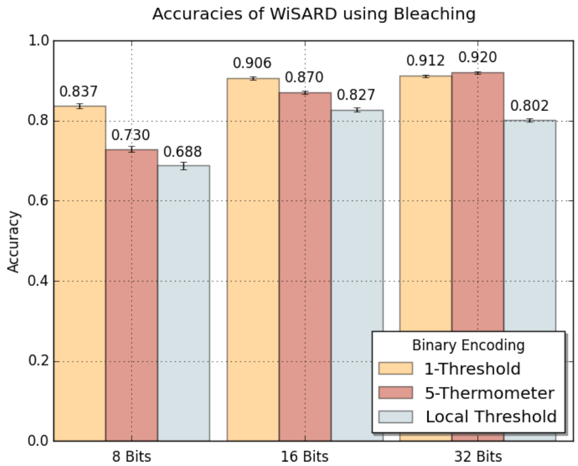

Fig. 13. Best *thermometer*, threshold and image processing approach compared. WiSARD with bleaching mechanism using 8, 16, and 32 bits.

were: threshold, *thermometer* encoding, local threshold, Mar-Hildreth filter and Laplacian filter. MNIST database was used as data set, which is an important data set for many classifiers evaluation [10].

The threshold analysis provided a visualization such that it is possible to notice that WiSARD using bleaching mechanism has a different behavior than the standard one, when varying the threshold. Moreover, the same analysis showed that the best threshold for this data set is 1-threshold, and the best models use 32 bits. The *thermometer* encoding analysis presented the best *thermometer* as the 20-*thermometer* for 32 bits WiSARD model, resulting in a accuracy of 94%. Another contribution is the presented behavior for models using bleaching mechanism, when binary encoding with *thermometers*, where the larger *thermometer* results in worse accuracies. The use of image processing techniques to binary encoding the input was tested, but was not successful.

For future works, the computational time could be added to evaluate the use of *thermometers*, also contemplating *thermometers* with larger sizes. Also, varying the confidence threshold could help to understand the reason WiSARD using bleaching had bad results for larger *thermometers*. The adaptation of WiSARD model provided in [4], which enhances the classifier to sparse data input, could be used to evaluate the image processing binary encoding techniques. Another future work is the application of statistical methods to verify the improvement when using *thermometers*.

ACKNOWLEDGMENT

We would like to thank CNPq (National Counsel of Technological and Scientific Development), CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), and FAPERJ (Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro).

REFERENCES

[1] I. Aleksander, W. Thomas, and P. Bowden, "Wisard a radical step forward in image recognition," *Sensor review*, vol. 4, no. 3, pp. 120–124, 1984.

[2] I. Aleksander, M. De Gregorio, F. M. G. França, P. M. V. Lima, and H. Morton, "A brief introduction to weightless neural systems." in *ESANN*. Citeseer, 2009.

[3] D. F. De Souza, F. M. França, and P. Lima, "Spatio-temporal pattern classification with kernelcanvas and wisard," in *Intelligent Systems (BRACIS), 2014 Brazilian Conference on*. IEEE, 2014, pp. 228–233.

[4] F. Rangel, F. Faria, P. M. V. Lima, and J. Oliveira, "Semi-supervised classification of social textual data using wisard," in *ESANN*. Citeseer, 2016.

[5] R. W. Hamming, "Error detecting and error correcting codes," *Bell System technical journal*, vol. 29, no. 2, pp. 147–160, 1950.

[6] H. C. Carneiro, F. M. França, and P. M. Lima, "Multilingual part-of-speech tagging with weightless neural networks," *Neural Networks*, vol. 66, pp. 11–21, 2015.

[7] B. P. Grieco, P. M. Lima, M. D. Gregorio, and F. M. França, "Producing pattern examples from "mental" images," *Neurocomputing*, vol. 73, no. 7–9, pp. 1057 – 1064, 2010.

[8] M. Gregorio and M. Giordano, "Change detection with weightless neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 403–407.

[9] Y. LeCun, C. Cortes, and C. J. Burges, "The mnist database of handwritten digits, 1998," *Available electronically at http://yann. lecun. com/exdb/mnist*, 2012.

[10] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[11] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[12] W. W. Bledsoe and I. Browning, *Pattern recognition and reading by machine*. PGEC, 1959.

[13] M. Basu, "Gaussian-based edge-detection methods-a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 32, no. 3, pp. 252–260, 2002.

[14] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.

[15] D. N. do Nascimento, R. L. de Carvalho, F. Mora-Camino, P. V. Lima, and F. M. França, "A wisard-based multi-term memory framework for online tracking of objects," in *ESANN 2015, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2015, pp. 978–287 587 014.

[16] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[17] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 281–305, 2012.