# Algorithm and Implementation of an Associative Memory for Oriented Edge Detection Using Improved Clustered Neural Networks

Robin Danilo*, Hooman Jarollahi†, Vincent Gripon‡, Philippe Coussy*, Laura Conde-Canencia*, and Warren J. Gross†

* Lab-STICC, Université de Bretagne-Sud, Morbihan, France

† Department of Electrical and Computer Engineering, McGill University, Montreal, Québec, Canada, H3A 0E9

‡ Electronics Department, Télécom Bretagne, Brest, France

***Abstract*—Associative memories are capable of retrieving previously stored patterns given parts of them. This feature makes them good candidates for pattern detection in images. Clustered Neural Networks is a recently-introduced family of associative memories that allows a fast pattern retrieval when implemented in hardware. In this paper, we propose a new pattern retrieval algorithm that results in a dramatically lower error rate compared to that of the conventional approach when used in oriented edge detection process. This function plays an important role in image processing. Furthermore, we present the corresponding hardware architecture and implementation of the new approach in comparison with a conventional architecture in literature, and show that the proposed architecture does not significantly affect hardware complexity.**

## I. Introduction

Human brain is a powerful embedded machine able to quickly apprehend its environment in order to provide coherent responses. It is one of the reasons why our numerical systems are more and more inspired by the human brain. For example, the human visual system is an important inspiration for image processing systems.

The smart encoding realized in the human visual cortex is reproduced in bio-inspired image processing through particular convolution filters [1], [2]. For example, the orientation selectivity of simple cells of primary visual cortex can be modeled with Gabor filters [3].

In this paper, we present an Oriented Edge Detection (OED) algorithm and its hardware implementation using a variant of a biologically-inspired associative memory known as Clustered Neural Networks (CNNs) [4], [5].

An efficient fully-parallel hardware implementations of it has been presented in [6]. We show that this implementation cannot be used in its current form for an OED application. We present an algorithm and architecture for the target application without significantly affecting hardware complexity of the conventional implementation.

The paper is organized as follows: In Section II, conventional CNNs are briefly discussed. In Section III, the intended OED application based on a retina-like processing is presented. In Section IV, a new retrieval algorithm and its hardware implementation are introduced for use in the application. Furthermore, the new retrieval algorithm is simulated and compared to the previous algorithms [7] in terms of efficiency. Section V concludes the paper.

## II. Clustered Neural Networks

A CNN-based associative memory stores a set of messages $\mathcal{M}$ of length $c$ over a finite alphabet $\mathcal{A}$. The network is then able to retrieve message $m \in \mathcal{M}$, when $\tilde{m}$, that is a version of $m$ wherein some elements are erased, is presented to the network. For better readability and without loss of generality, we consider $\mathcal{A}$ to be the set of integers between 1 and $\ell = |\mathcal{A}|$. The network is composed of $c$ clusters (one per element of $m$) each containing $\ell$ nodes (one per value of $\mathcal{A}$). During storage and retrieval (decoding) processes, an input message leads to the activation of a few nodes (i.e. setting their states to one) in each cluster.

During storage process, only one node per cluster is activated and the connections between them are stored in the adjacency matrix $W(\mathcal{M})$.

During retrieval process, an erased element leads to the activation of all nodes of the corresponding cluster. Then, an activation rule is iterated in order to retrieve the erased elements, that is until a single node per cluster remains activated. This activation rule is the following, a node must be active at the previous iteration and share at least one connection with an active node in all other clusters of the network. This convenient activation rule leads to efficient implementations [6], [8].

An extension of this "classical CNN" has been proposed [7] for which the stored messages can have missing elements. We call such networks "sparse CNN". During the retrieval process, there is no distinction between erased and missing elements, resulting in a more difficult retrieval problem.

With this modification to the model, the number of storable messages can grow quadratically with respect to the linear growth of both $c$ and $\ell$ [7]. This algorithm requires a maximum-detection function during retrieval process, which is resource hungry in hardware, and does not permit large networks as show in [9] for a similar architecture.

## III. Oriented Edge Detection (OED)

The role of OED in image processing is to detect oriented edges at several intensities in an image. An image in grey lev-

978-1-4799-8391-9/15/$31.00 ©2015 IEEE

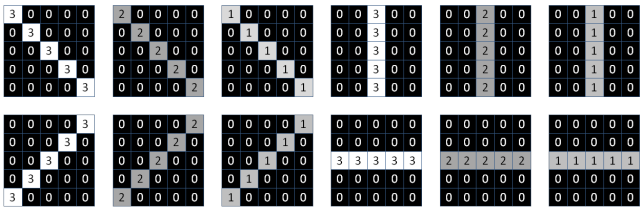

Fig. 1. An example set of patterns of size $5 \times 5$ pixels, with 4 quantization levels and 4 orientations

els is first pre-processed before it is presented to an associative memory module to recover the pattern.

The pre-processed image is obtained by using a Laplacian of Gaussian (LoG) filter and some approximations. The LoG filter is traditionally used to model the behaviors of the retina cells [10] and allows to provide an image in contrast levels.

The first approximation is a large sub-quantization of the filtered image in $I$ quantization levels. This sub-quantization is motivated by two reasons. Firstly, a low number of quantization levels leads to less sensitiveness to noise. Secondly, the associative memory module used for the final edge detection can be implemented in hardware only if the number of quantization levels is not too large.

The second approximation is a sub-sampling of the image by using a $max$ operation. This approximation aims to decrease the sensitivity to position. The $max$ operation has been proposed in the HMAX model [11] as an invariant mechanism performed by the human visual cortex.

Patches of size $N \times N$ pixels are extracted from the pre-processed image and are then provided to the associative memory module to detect the oriented edge. Therefore, the associative memory module needs to have stored the patterns previously.

The patterns are oriented edges of $I-1$ quantization levels higher than 0 (no black pixels) surrounded by 0 intensity pixels (black pixels). Fig. 1 shows an example set of patterns.

These oriented edge patterns must be retrieved inside the patch, that we consider as noisy input to the associative memory module. The alteration considered here is the substitution of some non black pixels in black pixels.

## IV. Proposed CNN-based associative memory for use in OED

To store and retrieve patterns with a classical CNN-based associative memory for use in OED, the network is split into $c$ clusters, with $c = N \times N$, each one containing $\ell = I$ nodes. Thus, each pixel in the patch is associated to a cluster while each intensity level is associated to a node. In this solution, the nodes associated to black pixels are overused, leading to nonuniform distribution of the stored patterns, and thus poor performance of the CNN.

Our proposed solution is to disregard black pixel elements during retrieval process. Consequently, the network becomes sparse and only the connections between the nodes associated to the non black pixels of the pattern are stored in the weight matrix. The characteristics of the network are thus $c = N \times N$ and $\ell = I - 1$.

We propose a new activation rule using only boolean operations along with an associated hardware module designed to meet the application specifications.

### A. Proposed Activation Rule

When the stored messages have missing elements, some clusters are not concerned by the message. As a consequence, there is no node in these clusters which sharing a connection with the active nodes located in the other clusters. Consequently, the activation rule used for classical CNNs [7] cannot be applied directly. The reason is that, to be activated, a node must share a connection with an active node in all other clusters.

To take in account the missing elements of the message during the retrieval process, the principle of **neutral cluster** is introduced. A neutral cluster is a cluster with no associated value, it is the case for the missing elements of the stored message and the erased elements. To remove the participation of the neutral clusters, all their nodes are deactivated and a bypass mechanism is inserted in the original activation rule. The state of each cluster is stored in a vector $d$, if $d_i = 1$, then, the cluster $i$ is neutral. The original activation rule is thus updated as follows:

$$v_{ij}^{t+1} = (v_{ij}^t \vee d_i^t) \wedge \bigwedge_{i' \neq i} \left( \bigvee_{j'} \left( W(\mathcal{M})_{ij,i'j'} \wedge (v_{i'j'}^t \vee d_{i'}^t) \right) \vee d_i^t \right), \tag{1}$$

where $d_i^t = \overline{\bigwedge_j v_{ij}^t}$. In other words, to be activated, a node must be active at the previous iteration or belong to a neutral cluster, as well as share at least one connection with an active node in each non-neutral cluster.

### B. Proposed Hardware Architecture and Implementation

Fig. 2 depicts the system-level architecture of a fully-parallel CNN employing the new decoding rule (node rule) shown in Equation 1. First, a set of messages are trained to the network by storing the corresponding links in link storage module. Then, retrieval process can begin by presenting partial inputs. Retrieval (decoding) process is performed by local decoding followed by an iterative global decoding.

The architecture of link storage module is similar to the presented architecture in [12]. It includes independently-accessed registers to store the connections between the nodes in parallel during storage and to retrieve them in parallel during decoding. Therefore, an input message of length $c \times \kappa$ with $\kappa = \log_2 \ell$ requires a single clock cycle to be stored. Local mapping circuit (LPC) maps a $\kappa$-bit input to an $\ell$-bit output by converting the integer value of an input to the set the corresponding index of the output to '1', and setting the rest to '0's. LPC is used during both storage and local decoding.

Node registers are used during decoding to store the value of the nodes after local mapping, and after each iteration. Therefore, $c \times l$ registers are dedicated in them. The new activation rule presented in Equation 1 has been implemented in hardware using a Altera Stratix V 5SGXMABN3F45C2 FPGA as shown in Fig. 3. Both after local decoding and

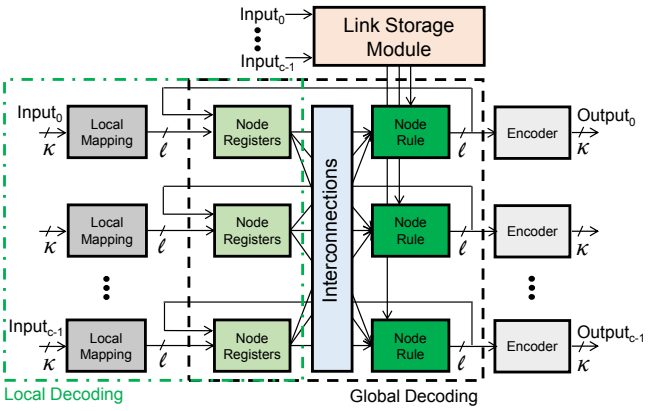


Fig. 2. Simplified system level architecture of a fully-parallel CNN-based associative memory employing the new activation rule

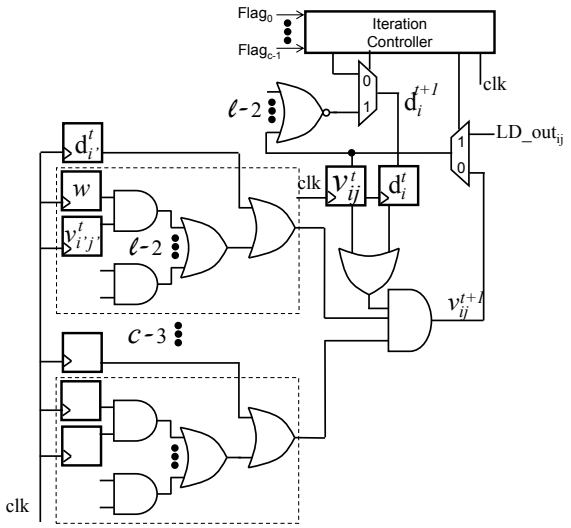


Fig. 3. A slice of the hardware architecture of the proposed algorithm computing the next value of node $v_{ij}$ using the proposed node activation rule

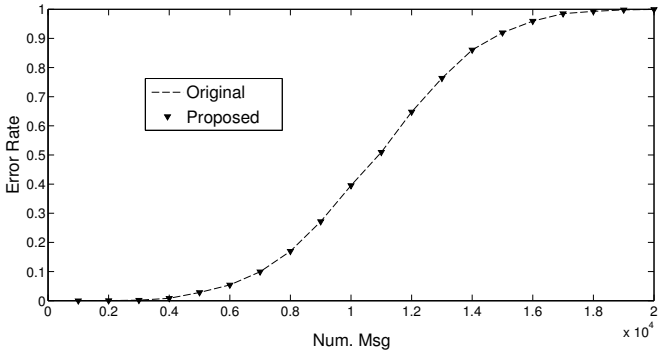


Fig. 4. Comparison between the retrieval algorithm in [7] and the proposed algorithm ($c = 32$, $\ell = 64$, message of 10 elements and 5 erased elements).

every iteration of global decoding, the value of node $j$ of cluster $i$, $v_{ij}^t$, is updated and is stored in its dedicated node register in parallel with that of other nodes. Depending on the iteration, the iteration controller decides whether the inputs of the node registers should be connected to the outputs of the local mapping module (LD_out$_{ij}$), or to the output of the node rule module after each iteration.

A cluster status register is dedicated for each cluster to store a flag that indicates whether or not a particular cluster is neutral. The value of the status register of cluster $i$, $d_i^t$ is updated in parallel with that of all node registers of the same cluster. If none of the nodes are activated after an iteration in a particular cluster, or the cluster is a neutral cluster from the beginning (indicated by a Flag bit in input messages), the dedicated status register stores a '1' to bypass the cluster during the decoding process in the next iteration. A cluster's status function is implemented using an $\ell$-input NOR gate during global decoding. Table I summarizes hardware resource allocation for implementing an CNN-based associative memory including the new decoding rule with various patch sizes and two different intensity levels corresponding to the number of nodes per cluster ($\ell = 8$ and $\ell = 16$).

A variant of the conventional architecture in [12] has also been implemented using the same FPGA and parameters for comparison purposes. The variant architecture uses a similar mapping circuit in local decoding as in the proposed architecture.

Based on experimental results, for $\ell = 8$, the maximum size of a patch that fits on the selected FPGA is a $9 \times 9$ patch, whereas for $\ell = 16$, the maximum patch size is a $6 \times 6$. Aside from the application-specific algorithmic advantages, for identical network parameters, the proposed architecture utilizes the hardware resources slightly more efficiently than the conventional architecture although the maximum frequency in the worst case scenario ($f_{Max}$) is also slightly slower mainly due to the existence of additional $\ell$-input NOR gates.

### C. Simulation Results

In this section, the new activation rule has been simulated and compared with the conventional algorithm used for sparse CNNs [7]. First, the comparison is done for uniform distribution of the set of stored messages $\mathcal{M}$. Fig. 4 draws the evolution of the message retrieval error rate in function of the size of $\mathcal{M}$ for $c = 32$ and $\ell = 64$ and messages of 10 elements. The number of erased elements is equal to $\frac{c}{2}$ and the number of iterations is limited to 4. Both the new activation rule and the conventional algorithm give the same error rate, whereas the new activation rule uses only boolean operations.

Secondly, we simulate the edge detection application to control if our associative module is able to manage the particular distribution of messages. The patterns are generated for $N = (5; 7)$, $I = (9; 17)$ and a fixed number of orientation $or = (8; 16)$. The resulting networks are $c = (25, 49)$, $\ell = I - 1 = (8, 16)$ and the number of stored messages is equal to $or \times (I - 1)$. During the retrieval process the altered patches are provided to the CNN with a number of erased clusters equal to $\frac{c}{2}$ rounded to the next integer. We consider that a pattern is present if at least the half of non black pixels are present in the patch. The proposed work is compared with:

TABLE I
HARDWARE RESOURCE ALLOCATION

| Parameter | [12] | This Work | | | | |
|---|---|---|---|---|---|---|
| Logic Utilization (ALMs) | 29,976/ 359,200 (8%) | 29,104/ 359,200 (8%) | 59,356/ 359,200 (17%) | 111,704/ 359,200 (31%) | 112,316/ 359,200 (31%) | 279,595/ 359,200 (78%) |
| Registers | 38,604 | 39,227 | 82,226 | 153,323 | 154,627 | 324,434 |
| $f_{\text{Max}}$ (MHz) | 254.32 | 216.31 | 165.43 | 124.38 | 140.61 | 101.67 |
| No. of Clusters ($c$) | 25 | 25 | 36 | 49 | 25 | 36 |
| No. of Nodes/Cluster ($\ell$) | 8 | 8 | 8 | 8 | 16 | 16 |
| Retrieval cycles | 1+iter | | | | | |
| Storage cycles | $\mathcal{M}$ | | | | | |

TABLE II
ERROR RATE FOR EDGE DETECTION APPLICATION

| | $N=5$ | | $N=7$ | | | |
|---|---|---|---|---|---|---|
| | $or=8$ | | $or=8$ | | $or=16$ | |
| | $I=9$ | $I=17$ | $I=9$ | $I=17$ | $I=9$ | $I=17$ |
| Proposed | 0.0728 | 0.0732 | 0 | 0 | 0.1278 | 0.1230 |
| SCNN | 0.0728 | 0.0732 | 0 | 0 | 0.1278 | 0.1230 |
| CCNN | 0.7608 | 0.7502 | 0.6972 | 0.7062 | 0.9694 | 0.9690 |
| ML | 0.0728 | 0.0732 | 0 | 0 | 0.1278 | 0.1230 |

1) The classical CNN (CCNN) [4], [5].
2) The sparse CNN (SCNN) [7].
3) A brute-force retrieval algorithm based on maximum likelihood (ML).

The results are shown in the Table II. For all configurations of $N$, $I$, the proposed activation rule gives the same results compared to the original algorithm for SCNN and also compared to the brute force retrieval algorithm. These results also show that CCNN is not suitable for this application due to the particular distribution of messages.

## V. CONCLUSIONS

In this paper, algorithm and architecture of an associative memory are presented for use in oriented edge detection, which is an important function in image processing. A variant of a recently introduced family of associative memories known as clustered neural networks has been exploited. The proposed associative memory is able to retrieve oriented edges in a patch extracted from a pre-processed image. The original activation rule used for classical CNNs has been modified in order to take into account the required function. The new activation rule proposed here allows to reach similar performance of the existing algorithm for sparse CNNs without significantly affecting the hardware complexity.

We believe that this architecture is generic and could apply to other problems where messages to store are sparse. Future work include looking at more complicated scenarios where inputs are not only erased but approximate or erroneous as well as testing the proposed solution in an image processing chain.

## ACKNOWLEDGMENT

This work has received a French government support granted to the CominLabs excellence laboratory and managed by the National Research Agency in the "Investing for the Future" program under reference ANR-10-LABX-07-01.

## REFERENCES

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[2] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust object recognition with cortex-like mechanisms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 411–426, 2007.
[3] J. P. Jones and L. A. Palmer, "An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex," *Journal of neurophysiology*, vol. 58, no. 6, pp. 1233–1258, 1987.
[4] V. Gripon and C. Berrou, "A simple and efficient way to store many messages using neural cliques," in *Proceedings of IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain*, Paris, France, April 2011, pp. 54–58.
[5] ——, "Sparse neural networks with large learning diversity," *IEEE Transactions on Neural Networks*, vol. 22, no. 7, pp. 1087–1096, July 2011.
[6] H. Jarollahi, N. Onizawa, V. Gripon, and W. J. Gross, "Algorithm and architecture of fully-parallel associative memories based on sparse clustered networks," *Journal of Signal Processing Systems, Springer*, vol. 76, no. 3, pp. 235–247, Sep. 2014.
[7] B. K. Aliabadi, C. Berrou, V. Gripon, and X. Jiang, "Storing sparse messages in networks of neural cliques," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, pp. 980–989, 2014.
[8] H. Jarollahi, V. Gripon, N. Onizawa, and W. J. Gross, "Algorithm and architecture for a low-power content-addressable memory based on sparse-clustered networks," *IEEE Transactions on Very Large Scale Integration Systems*, pp. 1–12, Apr. 2014.
[9] H. Jarollahi, N. Onizawa, V. Gripon, and W. J. Gross, "Architecture and implementation of an associative memory using sparse clustered networks," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, Seoul, Korea, May 2012, pp. 2901–2904.
[10] D. Marr and E. Hildreth, "Theory of edge detection." *Proceedings of the Royal Society of London. Series B, Containing papers of a Biological character. Royal Society (Great Britain)*, vol. 207, no. 1167, p. 187, 1980.
[11] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature neuroscience*, vol. 2, no. 11, pp. 1019–1025, 1999.
[12] H. Jarollahi, N. Onizawa, V. Gripon, and W. J. Gross, "Reduced-complexity binary-weight-coded associative memories," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, May 2013, pp. 2523–2527.