Efficient neural models for visual attention

Sylvain Chevallier, Nicolas Cuperlier, Philippe Gaussier

Efficient neural models for visual attention

Sylvain Chevallier, Nicolas Cuperlier, and Philippe Gaussier

ETIS - Neurocybernetic team,
ENSEA - University Cergy-Pontoise - CNRS UMR 8051
firstname.name@ensea.fr,
F-95000 Cergy, France

Abstract. Human vision rely on attention to select only a few regions to process and thus reduce the complexity and the processing time of visual task. Artificial vision systems can benefit from a bio-inspired attentional process relying on neural models. In such applications, what is the most efficient neural model: spiked-based or frequency-based? We propose an evaluation of both neural model, in term of complexity and quality of results (on artificial and natural images).

1 Introduction

Biological inspiration aims at adapting biological mechanisms to design efficient artificial systems benefiting from the natural solutions. An essential mechanism of the human visual system is visual attention, which allows higher cognitive processes (such as learning or recognition) to concentrate on few regions of the visual scene, selected by the attention. Adapting attention in artificial systems may be a way to reduce the computational cost of visual tasks [10]. There is a large number of applications for such artificial attentional systems, e.g. driver assistance [15], retinal prostheses [17] or robotics [9]. Several artificial systems propose an adaptation of attentional process on a neural level, i.e. biologically plausible efficient artificial systems implemented with neural networks [1, 4, 6, 14, 19]. Based on the time scale of the description, one can distinguish two ways of representing encoded information in neural models. In spiking neuron networks (SNN), information is encoded and exchanged between neurons with spikes, i.e. transient voltage pulses. With frequency-based neural network (FNN), information is encoded in the activation of each neuron, a quantity expressing the mean frequency of spiking rate [13]. The choice of the model type has a major influence both on the computational cost of the system and on its possibilities, i.e. mechanisms which can be adapted from biological observations. Which neural model, between SNN and FNN, is the best suited to implement an efficient bio-inspired attentional system? We propose a comparison of the computational complexity of these two type of networks and an evaluation of their performances on artificial and natural images.

The different psychological theories of the human attention, such as [7, 18, 20], agree that several local visual features (e.g. color, orientation, contrast or movement) are detected in a parallel process and then combined on a saliency

map. This saliency map indicates potentially interesting regions, called saliencies. The attentional process then selects the most salient regions.

In bio-inspired attentional systems, the detection of visual is usually achieved with difference of Gaussians (DOG) filters to reproduce the spatial transformation of retinal ganglion cells [8] and Gabor wavelets to detect orientations, as observed in the simple cells of the primary visual cortex [11]. Systems implemented with FNN [1, 14, 19] use neural networks to combine features on the saliency map and to select the most important saliencies. This selection stage is realized with a Winner-Take-All (WTA) mechanism [21], which allow to select only the most salient region. In order to determine the next salient regions, an inhibition of return (IOR) mechanism is used to inhibit the previously selected regions. When relying on spiking neurons to simulate visual processing, it is possible to take advantage of the precise timing of spike firing to implement an anytime system. Anytime systems [2] may be stopped at any instant and return a response. The quality of response depends on the computation time allowed to the system: quality of response increases with the computation time. An anytime SNN for visual attention, as described in [5, 4], is able to extract a number of saliencies which depends on the CPU time allowed to the SNN. The saliencies are detected in the order of their importance, e.g. the first saliency found is the most salient, hence there is no need of a WTA.

In Sect. 2, an architecture of attentional system and its implementation with FNN and SNN are described. A complexity analysis of the FNN and SNN implementation is detailed in Sect. 3.1 and a performance comparison on artificial and natural are proposed respectively in Sect. 3.2 and 3.3. Conclusions and perspectives are detailed in Sect. 4.

2 Models and Implementations

We use a a multiscale and multi-feature attentional architecture similar to architecture proposed by [10]. This attentional architecture uses local contrast of luminance, orientations and colors to extract saliencies. Figure 1 displays the organisation of this architecture, which is composed of 2D neural map. The luminance and colors of an input image are fed in Input maps. Detections of contrasts, orientations and color opponency are realized for a high and a low spatial frequencies. Local luminance contrasts are obtained with a DOG filtering. Orientation information are detected with Gabor wavelets filtering for four distinct orientations (0, $\frac{\pi}{4}$, $\frac{\pi}{2}$ and $\frac{3\pi}{4}$). Color opponency uses a combination of DOG filtering to detect red-green and blue-yellow opponency. . The high and low spatial frequency information are combined on the saliency map.

The SNN implementation of the attentional architecture is described in [4] and uses Leaky Integrate-and-Fire neural model. The LIF model describes the evolution of an internal parameter V and when V exceeds a threshold ϑ, the neuron fires a spike. The LIF model is characterized by the following differential equation:

$$\begin{cases} \frac{dV}{dt} = -\lambda(V(t) - V_{\text{rest}}) + I_{\text{input}}(t), \text{ if } V < \vartheta \\ \text{else fires a spike and } V \text{ is set to } V_{\text{reset}} \end{cases} \quad (1)$$

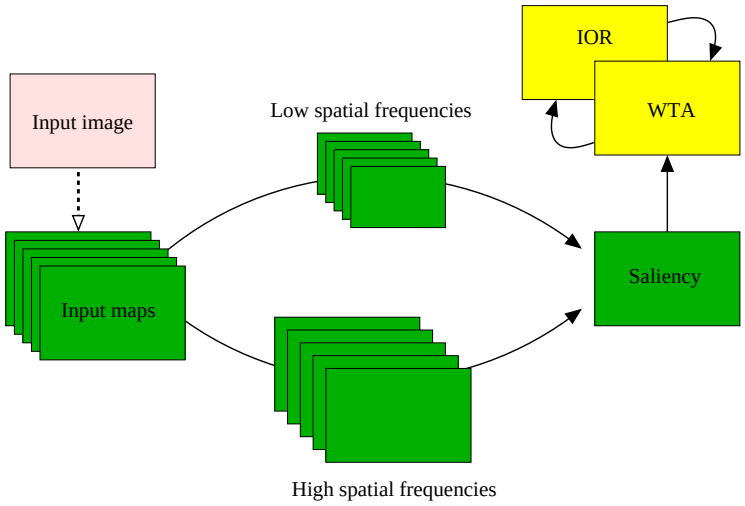

Fig. 1. Preattentive visual architecture, multiscale and multi-features (local luminance contrasts, detection of orientations and color opponency). Neural maps used in the SNN and FNN implementation are displayed in green and neural maps used only with FNN are in yellow.

where λ is the membrane relaxation constant and $I_{\text{input}}(t)$ is an input term. This SNN realizes an anytime neural filtering, leading to a gradual response which get closer to the convolution response as the computation time increases [3]. The early responses of neural filtering exhibit a bias which overvalue filtering responses for high luminance values. The saliency map gathers the neural filtering responses in order to determine the saliencies. Saliencies are thus obtained gradually and the most salient region are detected first. As the SNN extract salient regions already sorted in the order of their importance, there is no WTA. It is important to note that the result of the SNN depends on the simulated network time t.

In FNN implementation, frequency-based neural model are used to implement a classical convolution algorithm. It is computationally equivalent to filter the image with a classical algorithm and then convert the resulting image values in neuronal activity. When the input image is filtered for both spatial frequencies, the resulting activation are summed on the saliency map. To sort the saliencies in the order of their importance, the FNN relies on a WTA map coupled with an inhibition of return map (see Fig. 1). As realistic biological WTA have a high computation cost [21], we use a WTA implementation relying on a ARGMAX function which gives the same results as biologically plausible WTA but with a lower computational cost.

3 FNN and SNN Comparisons

3.1 Complexity Analysis

The most important computational cost for FNN implementation is the image filtering cost. The image filtering is realized with a classical convolution algorithm. As the input image and the filter are relatively small, this is a acceptable choice. Hence, for an architecture processing f features at s spatial scales, with filters of size M and an input image of N pixels, the time complexity is in $\mathcal{O}(f \times s \times M \times N)$. The WTA algorithm used in this FNN as a time complexity of $\mathcal{O}(N)$. The overall time complexity is thus $\mathcal{O}(f \times s \times M \times N)$. The FNN implementation uses Promethe [12], a distributed real-time neural network simulator.

The time and space complexity of a SNN heavily depends on implementation choices. The SNN is implemented on a simulator developed by the authors, which uses a simulation strategy called hybrid-synchronous [16]. Synchronous simulators rely on an internal clock with a time step Δt to update the state variables of every neurons in the network. The choice of the time step value is crucial as it influences the computational and the precision of the obtained results. Smaller Δt value offers more precise results but higher computational cost. Here, $\Delta t = 0.1$ms which is sufficiently small to obtain precise and reproducible results. With a hybrid-synchronous strategy, only the "active" neurons are updated, i.e. neurons with non null input term $I_{\text{input}}(t)$ at instant t.

The computational cost of a simulation can be expressed as the sum of the spike propagation cost and the neuron update cost. Here, it is:

$$c_p \times F \times N \times M + c_u \times \frac{A}{\Delta t} \tag{2}$$

The total propagation cost depends on the unitary propagation cost c_p, the mean number of fired spike which depends of the mean frequency rate F and the number of neurons N and the mean number of connection per neuron (which depends on the filter size M, see [4]). The total update cost rely on the unitary update cost c_u, the mean number of active neurons A and Δt. Here the unitary update cost requires 10 FLOP.

The computational cost is dependent of the input image: a complex image (in the meaning of filter used in the SNN) induced a large number of spikes and the simulation cost is high. To illustrate this fact, we construct test images with various complexity, i.e. with different number of features. These test images are set up by converting impulse response of DOG and Gabor filter in small images (called patches). The amplitude and spatial position of these patches are drawn randomly. Left part of Fig. 2 shows a test image used to evaluate the influence of an image complexity on the required processing CPU time. CPU time (measured in CPU cycles) required to process image of growing complexity (with 1, 10, 50 or 100 patch's) is recorded and shown on the right part of Fig. 2. One can see CPU cycles needed to extract saliencies increases with the image complexity.

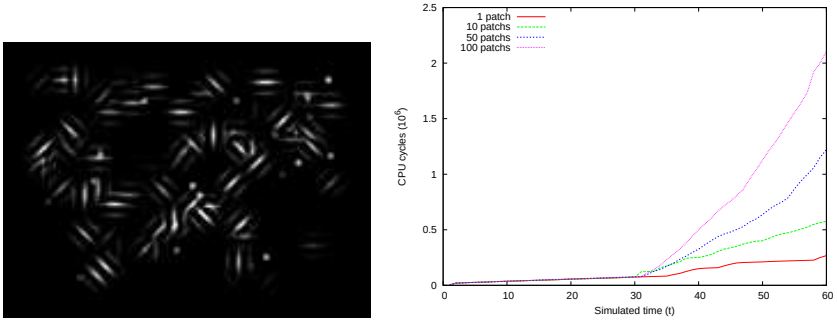

Fig. 2. Left:A test image with 100 patches. Each patch is the impulse response of a DOG or a Gabor filter. Right: CPU cycles needed to extract saliencies on images with growing complexity, i.e. different number of patches. CPU cycles are measured for each simulated time step in the SNN.

3.2 Artificial Images

We propose a comparison of saliency detected on pop-out images [18], where a target among distractors is easily identifiable. This is the case when the target differs from distractors for a given feature. Figure 3 shows the most salient region obtained on two pop-out test images.

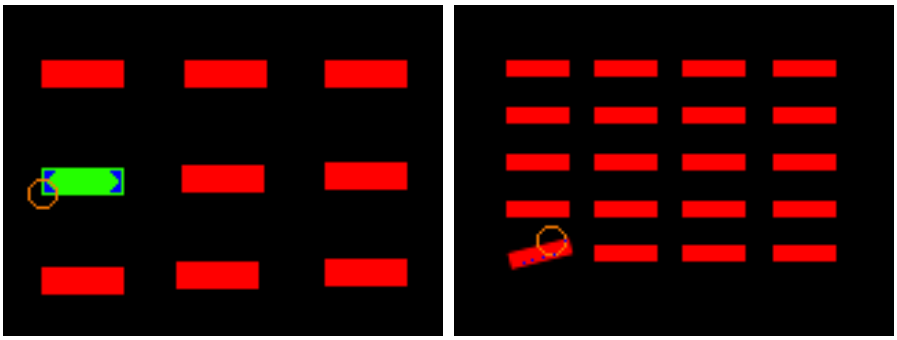

Fig. 3. Pop-out images (160x120 pixels) used for saliency detection. On the left, target differs from distractors by its color and, on the right, by its orientation. The most salient region detected by FNN is represented as an orange circle and for SNN by blue pixels.

The Figure 3 shows that both SNN and FNN are able to detect saliencies, but results take different forms. With FNN, a saliency correspond to the position of the winner neuron in the WTA, i.e. neuron with the highest activation on saliency map. As the winner location and its direct neighborhood is then inhibited by

the IOR, we indicate saliency as a circle centered on the winner location. The SNN extract salient regions already sorted in the order of their importance. The first neurons to fire on the saliency map indicate the most salient regions, so there can be several points with an identical saliency value. On the left image of Figure 3, the edges of the green target are salient and on the right image four salient dots are detected on the bottom part of the target.

3.3 Natural Images

We propose a comparison based on 19 natural images of 160x120 pixels acquired with a standard webcam. Figure 4 shows two of the 19 images and the three most salient region detected by the FNN and the SNN. On few images, salient regions are not extracted in the same order in SNN and FNN (as on the right image). These differences are due to the fact that SNN present a bias toward high luminance value. On the right image, luminance and color contrasts of the blue can (white on light background) are surevaluated compared to contrasts of cans on the bottom (light grey on dark background).

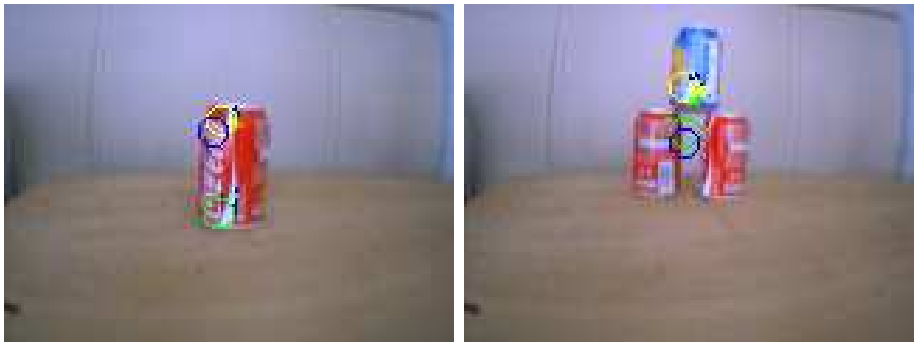

Fig. 4. Examples of saliencies obtained on natural images. For each image, the three most salient regions are respectively indicated in yellow, green and blue. FNN saliencies are indicated with circles and SNN saliencies are indicated by colored pixels.

To evaluate the computational performance of FNN and SNN, we measured the number of CPU cycles needed for each neural networks to find the most important salient region. Each measure is repeated 10 times to compute a mean number of CPU cycle and its standard deviation. CPU cycle measurements are almost constant on the 19 natural images for FNN: it required 2.68414 $\times 10^6$ CPU cycles (with a standard deviation of 0.008 $\times 10^6$) to find the most salient region. As an example, on a computer equipped with 4 AMD Opteron 2.4 GHz, the mean execution time is 0.62 sec, that is 1.6 frame per second.

The Table 1 shows the mean number of CPU cycles measured with the SNN and the difference in percent with the FNN. As previously explained, SNN uses an iterative simulation, so we chose to stop the simulation as soon as the SNN

find the first salient region (usually after 45 time steps). It appears that the mean CPU cycles required to find the most salient region varies from one image to another. This effect is due to the fact that the 19 images have different complexity, in term of the filters used in the architecture. One can see that for one fourth of the images, the SNN find the most salient region before the FNN does. For the image on the right part of Fig. 4 (denoted as image #6 in Table 1), the SNN find the most important saliency, indicated in yellow, before the FNN.

4 Conclusions

This contribution proposes a comparison of two neural model, spike-based and frequency based, to implement an artificial attentional system. FNN have a lesser computational cost than SNN but require a WTA to extract the saliencies. The SNN is an anytime system and saliencies are extracted gradually, in the order of their importance. Both neural networks indicate the same saliencies and the SNN find the first saliency before FNN in one fourth of the natural images used in the evaluation. One can note that if a FNN relies on a biologically realistic implementation of WTA, the computational cost of the FNN will be greatly increased. To implement a bio-inspired attentional system, FNN is an efficient solution. An possible solution to benefit from the advantage of both neural models is to use both FNN and SNN, running in parallel on different computers, to process visual input.

Image	SNN (10^6 CPU cycles)	Difference with FNN
1	3.389 ± 0.041	23.62 %
2	2.359 ± 0.049	**-12.74 %**
3	2.409 ± 0.006	**-11.66 %**
4	3.487 ± 0.010	28.27 %
5	3.682 ± 0.076	38.42 %
6	2.530 ± 0.006	**-3.68 %**
7	2.944 ± 0.005	12.00 %
8	2.830 ± 0.004	7.56 %
9	2.816 ± 0.004	6.01 %
10	3.336 ± 0.107	25.39 %
11	3.520 ± 0.004	32.74 %
12	2.868 ± 0.002	7.80 %
13	4.157 ± 0.006	53.07 %
14	3.994 ± 0.003	46.86 %
15	3.737 ± 0.004	35.43 %
16	4.144 ± 0.036	53.48 %
17	2.992 ± 0.097	12.46 %
18	2.348 ± 0.010	**-12.74 %**
19	2.264 ± 0.011	**-15.77 %**

Table 1. Number of CPU cycles required to find the most salient region on each of the 19 images with the SNN. For the FNN, the mean number of CPU cycles required is 2.68414 10^6. The difference between SNN and FNN is shown in the last column.

References

1. Ahrns, I., Neumann, H.: Space-variant dynamic neural fields for visual attention. In: CVPR. vol. 2, p. 318. IEEE (1999)
2. Boddy, M., Dean, T.: Deliberation scheduling for problem solving in time-constrained environments. Artificial Intelligence 67(2), 245–285 (1994)
3. Chevallier, S., Dahdouh, S.: Difference of gaussians type neural image filtering with spiking neurons. In: IJCCI. pp. 467–472 (2009)
4. Chevallier, S., Tarroux, P.: Covert attention with a spiking neural network. In: ICVS. LNCS, vol. 5008, pp. 56–65 (2008)
5. Chevallier, S., Tarroux, P., Paugam-Moisy, H.: Saliency extraction with a distributed spiking neural network. In: ESANN. pp. 209–214 (2006)
6. de Brecht, M., Saiki, J.: A neural network implementation of a saliency map model. Neural Networks 19(10), 1467–1474 (2006)
7. Duncan, J., Humphreys, G.: Visual search and stimulus similarity. Psychological Review 96(3), 433–458 (1989)
8. Enroth-Cugell, C., Robson, J.: The contrast sensitivity of retinal ganglion cells of the cat. Journal of Physiology 187(3), 517–552 (1966)
9. Frintrop, S., Jensfelt, P.: Attentional landmarks and active gaze control for visual SLAM. IEEE Transactions on Robotics 24(5), 1054–1065 (2008)
10. Itti, L.: Models of bottom-up attention and saliency. In: Itti, L., Rees, G., Tsotsos, J. (eds.) Neurobiology of Attention, pp. 576–582. Elsevier (2005)
11. Jones, J., Palmer, L.: An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. Journal of Neurophysiology 58(6), 1233–1258 (1987)
12. Lagarde, M., Andry, P., Gaussier, P.: Distributed real time neural networks in interactive complex systems. In: CSTST. pp. 95–100 (2008)
13. Maass, W.: Networks of spiking neurons: the third generation of neural network models. Neural Networks 10, 1659–1671 (1997)
14. Maillard, M., Gapenne, O., Gaussier, P., Hafemeister, L.: Perception as a dynamical sensori-motor attraction basin. In: Advances in Artificial Life. LNCS, vol. 3630, pp. 37–46 (2005)
15. Michalke, T., Fritsch, J., Goerick, C.: Enhancing robustness of a saliency-based attention system for driver assistance. In: ICVS. LNCS, vol. 5008, pp. 43–55. Springer (2008)
16. Morrison, A., Mehring, C., Geisel, T., Aertsen, A., Diesmann, M.: Advancing the boundaries of high-connectivity network simulation with distributed computing. Neural Computation 17(8), 1776–1801 (2005)
17. Parikh, N., Itti, L., Weiland, J.: Saliency-based image processing for retinal prostheses. Journal of neural engineering 7(1) (2010)
18. Treisman, A.: Preattentive processing in vision. Computer Vision, Graphics and Image Processing 31, 156–177 (1985)
19. Vitay, J., Rougier, N., Alexandre, F.: A distributed model of spatial visual attention. In: Biomimetic Neural Learning for Intelligent Robots. pp. 54–72. LNAI (2005)
20. Wolfe, J.: Guided Search 2.0: A revised model of visual search. Psychonomic Bulletin and Review 1(2), 202–238 (1994)
21. Yuille, A., Geiger, D.: Winner-take-all mechanisms. In: The Handbook of Brain Theory and Neural Networks, pp. 1056–1060. MIT Press (1998)