Article

Implementation of the Canny Edge Detector Using a Spiking Neural Network

Krishnamurthy V. Vemuru †

Riverside Research, 2900 Crystal Dr., Arlington, VA 22202, USA; kvv5cf@virginia.edu; Tel.: +1-937-912-6188
† Current Address: BrainChip Inc., 23041 Avenida de la Carlota, Laguna Hills, CA 92653, USA.

Abstract: Edge detectors are widely used in computer vision applications to locate sharp intensity changes and find object boundaries in an image. The Canny edge detector is the most popular edge detector, and it uses a multi-step process, including the first step of noise reduction using a Gaussian kernel and a final step to remove the weak edges by the hysteresis threshold. In this work, a spike-based computing algorithm is presented as a neuromorphic analogue of the Canny edge detector, where the five steps of the conventional algorithm are processed using spikes. A spiking neural network layer consisting of a simplified version of a conductance-based Hodgkin–Huxley neuron as a building block is used to calculate the gradients. The effectiveness of the spiking neural-network-based algorithm is demonstrated on a variety of images, showing its successful adaptation of the principle of the Canny edge detector. These results demonstrate that the proposed algorithm performs as a complete spike domain implementation of the Canny edge detector.

Keywords: edge detection; segmentation; spiking neural networks; bio-inspired neurons

check for updates

Citation: Vemuru, K.V. Implementation of the Canny Edge Detector Using a Spiking Neural Network. *Future Internet* **2022**, *14*, 371. https://doi.org/10.3390/fi14120371

Academic Editors: Remus BRAD and Arpad Gellert

Received: 6 June 2022
Accepted: 7 December 2022
Published: 11 December 2022

1. Introduction

Artificial neural networks (ANNs) have become an indispensable tool for implementing machine learning and computer vision algorithms in a variety of pattern recognition and knowledge discovery tasks for both commercial and defense interests. Recent progress in neural networks is driven by the increase in computing power in data centers, cloud computing platforms, and edge computing boards. In size, weight, and power (SWaP)–constrained applications, such as unmanned aerial vehicles (UAVs), augmented reality headsets, and smart phones, more novel computing architectures are desirable. The state-of-the-art deep learning hardware platforms are often based on graphics processing units (GPUs), tensor processing units (TPUs) and field programmable gate arrays (FPGAs). The human brain is capable of performing more general and complex tasks at a minute fraction of the power required by deep learning hardware platforms. Spiking neurons are regarded as the building blocks of the neural networks in the brain. Moreover, research in neuroscience indicates the spatiotemporal computing capabilities of spiking neurons play a role in the energy efficiency of the brain. In addition, spiking neurons leverage sparse time-based information encoding, event-triggered plasticity, and low-power inter-neuron signaling. In this context, neuromorphic computing hardware architecture and spike domain machine learning algorithms offer a low-power alternative to ANNs on von Neumann computing architectures. The availability of neuromorphic processors, such as IBM's TrueNorth [1], Intel's Loihi [2], and event-domain neural processors, for example, BrainChip's Akida [3,4], which offers the flexibility to define both artificial neural network layers and spiking neuron layers, are motivating the research and development of new algorithms for edge computing. In the present work, we have investigated how one can program an algorithm for Canny type edge detection using a spiking neural network and spike-based computing.

2. Background

An edge detection algorithm is widely used in computer vision to locate object boundaries in images. An edge in an image shows a sharp change in image brightness, which is a result of a sharp change in pixel intensity data. An edge detector computes and identifies the pixels with sharp changes in intensity with respect to the intensity of neighboring pixels. There are several edge detection image processing algorithms.

The three stages in edge detection are image smoothing, detection, edge localization. There are mainly three types of operators in edge detection. These are (i) gradient-based, (ii) Laplacian-based and (iii) Gaussian-based. The gradient-based edge detection method detects the edges by finding the maximum and the minimum in the first derivative of the image using a threshold. The Roberts edge detector [5], Sobel edge detector [6], and Prewitt edge detector [7] are some of the examples of gradient-based edge detectors. These detectors use a 3×3 pattern grid. A detailed discussion on these edge detectors and a comparison of their advantages and disadvantages can be found in [8]. The Roberts edge detection method is built on the idea that a difference on any pair of mutually perpendicular directions can be used to calculate the gradient. The Sobel operator uses the convolution of the images with a small, separable, and integer-valued filter in horizontal and vertical directions for edge detection. The Prewitt edge detector uses two masks, each computing the derivate of the image in the x-direction and the y-direction. This detector is suitable to estimate the magnitude and orientation of the edge. Laplacian-based edge detectors find the edges by searching for zero crossings in the second derivative of the image. The Laplacian of the Gaussian algorithm uses a pre-smoothing step with a Gaussian low-pass filter on an image followed by a second-order differential, i.e., Laplacian, which finds the image edge. This method needs a discrete convolutional kernel that can approximate the second derivative for the image which consists of discrete pixels. The Marr–Hildreth edge detector is also based on the Laplacian of the Gaussian operator [9]. The Gabor filter edge detector [10] and Canny edge detector [11] are Gaussian-based edge detectors. The Gabor filter is a linear filter with its impulse response function defined by the product of a harmonic function with a Gaussian function and is similar to the human perception system.

The Canny edge detector provides excellent edge detection, as it meets the three criteria for edge detection [12]: (i) detection with low error rate, (ii) the edge point should localize in the center of the edge, and (iii) an edge should only be marked once and image noise should not create edges. Canny edge detection uses the calculus of variations to optimize a functional which is a sum of four exponential terms, which approximates the first derivative of a Gaussian. A Canny edge detector is a multi-step algorithm designed to detect the edges of any analyzed image. The steps of this process are: (1) removal of noise in the image using a Gaussian filter, (2) calculation of the gradient of the image pixels along x- and y-directions, (3) non-maximum suppression to thin out edges, (4) double-threshold filtering to detect strong, weak and non-relevant pixels, and (5) edge tracking by hysteresis to transform weaker pixels into stronger pixels if at least one of their neighbors is a stronger pixel. The Canny edge detection algorithm is highly cited ($\sim$36,000 citations) and the most commonly used edge detection algorithm [11].

Edge detection is a primary step in identifying an object and further research is strongly desirable to expand these methods to event-domain applications. The Canny edge detector has a better performance as an edge detector compared to Roberts, Sobel and Prewitt edge detectors, but at a higher computational cost [8]. An alternate implementation of the Canny edge detection algorithm is for edge computing event-domain applications, where low-power and real-time solutions can be attractive to target applications, in view of its tunable performance using the standard deviation of the Gaussian filter.

3. Related Work

In the human vision system, the photoreceptors in the retina convert the light intensity into nerve signals. These signals are further processed and converted into spike trains by the ganglion cells in the retina. The spike trains travel along the optic nerve for further processing in the visual cortex. Neural networks that are inspired by the human vision system have been introduced to improve image processing techniques, such as edge detection [13]. Spiking neural networks, which are built on the concepts of spike encoding techniques [14], spiking neuron models [15] and spike-based learning rules [16], are biologically inspired in their mechanism of image processing. SNNs are gaining attraction for biologically inspired computing and learning applications [17,18]. Wu et al. simulated a three-layer spiking neural network (SNN), consisting of a receptor layer, an intermediate layer with four filters, respectively, for up, down, left, and right directions, and an oputput layer with Hogkin–Huxley-type neurons as the building blocks for edge detection [19].

Clogenson et al. demonstrated how a SNN with scalable, hexagonally shaped receptive fields performs edge detection with computational improvements over rectangular shaped pixel-based SNN approaches [20]. The digital images are converted into a hexagonal pixel representation before being processed by the SNN. A spiking neuron integrates the spikes from a group of afferent neurons in a receptive field. The network model used by the authors consists of an intermediate layer with four types of neurons corresponding to four different receptive fields, corresponding to up, down, right and left orientations. Yedjour et al. [21] demonstrated the basic task of contour detection using a spiking neural network based on the Hodgkin–Huxley neuron model. In this approach, the synaptic weights are determined by the Gabor function to describe the receptive field's behaviors of simple cells in the visual cortex. Vemuru [22] reported the design of a SNN edge detector with biologically inspired neurons and demonstrated that the edge detector detects edges in simulated low-contrast images. These studies focused on defining SNNs using an array of Gabor filter receptive fields in the edge detector. In view of the success of SNNs in edge detection, it is desirable to develop a spike domain implementation of an edge detector with the Canny edge detector algorithm because it has the potential to offer a high performance alternative for edge detection.

4. Methods

Network models of the visual cortex are simulated with spiking neurons using Hodgkin and Huxley equations [23]. Retinal ganglion cells convey the visual image from the eye to the brain [24,25]. Receptive fields exist in the visual cortex; however, an accurate representation of the neuron circuits for the visual cortex is still unclear. Neural network models have been proposed explaining how the visual system is able to process an image efficiently, and more research is desired to further our understanding of the visual cortex [26]. As ANNs grow in complexity, their associated energy consumption becomes a challenging problem. Such challenges also exist for computing edges in images, where the computing devices are resource-constrained while operating on a limited energy budget. Therefore, specialized optimizations for deep learning have to be performed at both software and hardware levels. Edge detection can be achieved using a spiking neuron model [19]. Spiking neural networks offer a low-energy computational alternative with only a few layers, while maintaining edge features. Our solution for spike-based edge detection uses only one layer of Hodgkin–Huxley-type neurons (1 neuron/pixel) with five spike processing layers, one conductance calculation layer and a synaptic current update layer. The simple form of Hodgkin–Huxley neurons used in the network are similar to the conductance-based leaky integrate-and-fire neurons, which are frequently used in neuromorphic hardware implementation [1,2].

To implement a neuromorphic analogue of the Canny detector, we invented spike-based computation using the five key steps introduced earlier and implemented in MATLAB. Figure 1 illustrates the flowchart of the algorithmic steps in the spike domain computation of Canny edge detection. The image $I(x,y)$, where (x,y) are the coordinates of the pixels, is first converted into grayscale $I(grayscale)(x,y)$ and scaled such that $I(grayscale)_{max}$ = 0.01 to match the units of the model parameters of the Hodgkin–Huxley neuron model. Then, it is assigned as peak conductance for excitatory synapse q_{ex} and peak conductance for inhibitory synapse q_{in}. The peak synapses are then converted into time-dependence conductances, g_{ex} and g_{in}, for excitatory and inhibitory synapses, respectively, using the equations

$$g_{ex} = q_{ex}(\tau_{ex} \times dt)/(\tau_{ex} + dt) \tag{1}$$

$$g_{in} = q_{in}(\tau_{in} \times dt)/(\tau_{in} + dt), \tag{2}$$

where τ_{ex} = 4 ms and τ_{in} = 7 ms are the time constants for excitatory, inhibitory synapses, respectively [27]. Then, the conductances are processed using a Gaussian kernel of 5 × 5, to calculate the synaptic current at each time step t [28]:

$$I_z(t) = g_{ex}(V - V_{ex}) + g_{in}(V - V_{in}), \tag{3}$$

where V_{ex} and V_{in} are the reverse potentials for excitatory and inhibitory synapses, respectively. Note that the kernel size was smaller than 5 × 5 for edge pixels.

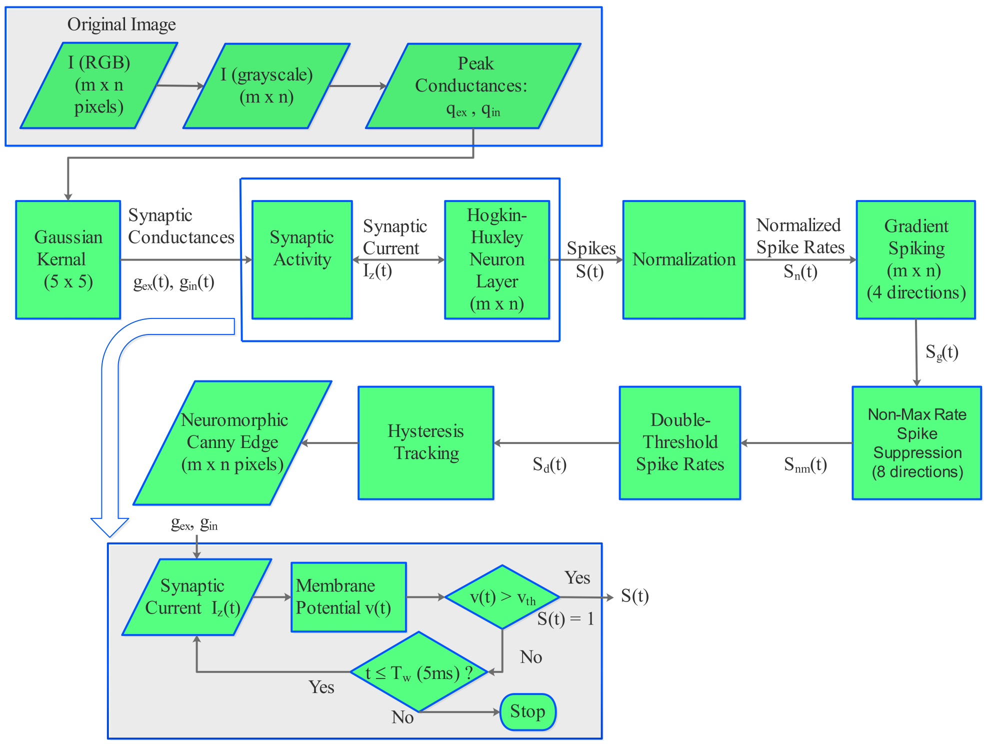

Figure 1. Processing steps in the algorithm for neuromorphic computation of Canny edge detector using conductance-based Hodgkin–Huxley neurons in a neural network.

The synaptic current is then used as the input to the Hodgkin–Huxley neurons where the membrane potential in the Hodgkin–Huxley (HH) neuron model [23] is governed by the differential equation:

$$C\frac{dV}{dt} = I_z(t) - G_K n^4(V(t) - V_K) - G_{Na}m^3h(V(t) - V_{Na}) \\ -G_L(V(t) - V_L), \tag{4}$$

where C is the capacitance per unit area of the lipid bi-layer membrane, $I_z(t)$ represents the total membrane current per unit area, G_K is the potassium conductance per unit area, G_{Na} is the sodium conductance per unit area, V is the membrane potential, V_K is the potassium reverse potential, V_{Na} is the sodium reverse potential, G_L is the leak conductance per unit area, V_L is leak reverse voltage, m is the activation variable for Na$^+$ channels, h is the inactivation variables for the Na$^+$ channels, and n is the K$^+$ inactivation variable. The dynamics of m, h and n are independently governed by additional differential equation for each of these variables [23] .

The exact solution to the HH model, given by the equation for the membrane potential,

$$\begin{aligned} v(t) = (1/G_L)\{(-exp(G_L t/C))(I_z(t) + 70G_L + G_L V_L) \\ + I_z + G_L V_L\} \end{aligned} \tag{5}$$

is reported by Aaby [29] and compared with numerical solutions by Siciliano [30] for the case of $G_K = 0$ and $G_{Na} = 0$, i.e., somewhat similar to the case of the leaky integrate-and-fire neuron model. The use of conductance terminology from the HH model is intended to maintain generality. Recently, Vemuru [22] used this solution to calculate the membrane potential in a SNN of HH neurons and showed that it is effective in evaluating neuron dynamics for edge detection applications.

We set the parameters of the neuron model as follows: $G_L = 0.003$ mS/cm^2, $V_L = -44.4$ mV, $C = 0.01$ μF/cm^2, $V_{in} = -72.1$ mV and $V_{ex} = 55.2$ mV. We followed the approach reported in references [22,30] for using HH neurons in a spiking neural network for edge detection. The reset membrane potential, V_{reset} is set to -70 mV and threshold voltage, V_{th} as -55 mV. The time constant τ_{reset} is set to 3 ms.

The membrane potential v is evaluated using Equation (5) and is compared with the threshold voltage V_{Th} to determine the spiking activity of the network at each time step t. When $v > V_{Th}$, the neuron emits a spike, i.e., $S = 1$ and v is reset to v_{reset}. When this condition is not met, the neuron will not spike, i.e., $S = 0$.

The output spikes of the HH model are then processed to calculate normalized spiking rates $S_n(t)$ which are used to calculate the gradient spikes in two to four directions, depending on the location of the corresponding pixel as illustrated in the flow-chart shown in Figure 1. The rates are offset by adding a constant of 0.25, which is introduced to maintain non-zero spike rates in most of the pixels. This constant plays the role of one of the hyper-parameters. Then, the non-maximum spike rates are suppressed by comparing with the spike rates of nearest neighbors in eight or fewer directions, depending on the coordinates of the central pixel. The next step of the Canny algorithm, i.e., the double threshold, is calculated with two parameters, a lower threshold = 0.20 and an upper threshold = 0.23. The values for the lower threshold and the upper threshold are chosen so as to maximize the edge features. The edge features are quantified by the number of edge pixels in the image. As such, there is no metric that requires the difference between the lower-threshold and the upper-threshold to be in a specific range. They are considered two parameters in the edge detection model that optimize the edge features in the images. In many image processing algorithms, including the Canny edge detection algorithm, which is the goal of the present work, the model parameters are chosen heuristically. This is not a disadvantage when one is not learning the edge detection by training the model.

The final step, which is hysteresis tracking, is also computed using the spike rates. The final output of the spikes and spike-rates-based computation results in an edge map. The limitation of the current approach is that the hyperparameters are heuristically determined, as the main goal is to demonstrate the principle of Canny edge detection using spike-based computing. Hyperparameter optimization without significantly increasing the computational burden will be desirable in a future extension of the algorithm. Because of the close similarity in algorithmic steps between conventional computation and spike domain computation, we refer to the new detector as a neuromorphic Canny edge detector. The neuromorphic Canny edge detection algorithm can be attractive for implementation with potential gains in speed or a decrease in power requirements with a suitable neuromorphic hardware that can enable the definition of the conductance-based neuron model used in our algorithm. Intel's Loihi 2 processor [31], which comes with the option to program the neuron models, is one of the suitable processors to explore the algorithm reported in the present work. Machine learning style hyperparameter tuning is not feasible because the network is not designed as a learning system in the present form. In image processing, it is considered reasonable to use model parameters that are tuned via an experiment to test with randomly selected images from benchmark datasets. In addition, a comparison with the reference methods or state-of-the-art methods gives as idea of the validity from a general context.

5. Results and Discussion

Figure 2 displays the original image (No. 61060) from BSD500 dataset [32,33] (license: Not found) and the intermediate output images of the spiking detector network after the Gaussian kernel, gradient calculation, non-maximum suppression, double-threshold filtering, and edge-tracking by hysteresis. The intermediate features show the spike domain implementation is effective in performing all five steps involved in the conventional Canny edge detection algorithm. The edge map resulting from the spike-based computation, i.e., the output of the network after the edge-tracking hysteresis step, can be referred to as the neuromorphic Canny edge detector or SNN-based Canny edge detector. We find that the resolution of the images from BSD500 dataset is sufficient to evaluate the qualitative and quantitative performance of edge detection. In Figure 3, we compare the neuromorphic Canny detector with the Sobel detector and the conventional Canny detector for a set of four images and their ground truth edges from the BSD500 dataset [32,33]. We find the neuromorphic Canny detector detects edges similarly to the edges generated by both the conventional detectors. The spike-based detection reported here results in relatively more highlighted edges in the objects while retaining some of the structural features that seem to have been lost in the conventional edge detectors. This can be attributed to the difference in the range of thresholds used in the spike domain implementation compared to the conventional Canny edge detector. With an implementation on a low size, weight and power (SWaP) hardware, for example, by emulating spikes on a field programmable gate array (FPGA) [34] or edge computing devices, such as NVIDIA Jetson TX2, together with a neuromorphic camera, the spike domain image processing algorithm will be attractive for new applications, for example, in handheld computer vision systems and for low-Earth satellite-based object detection.

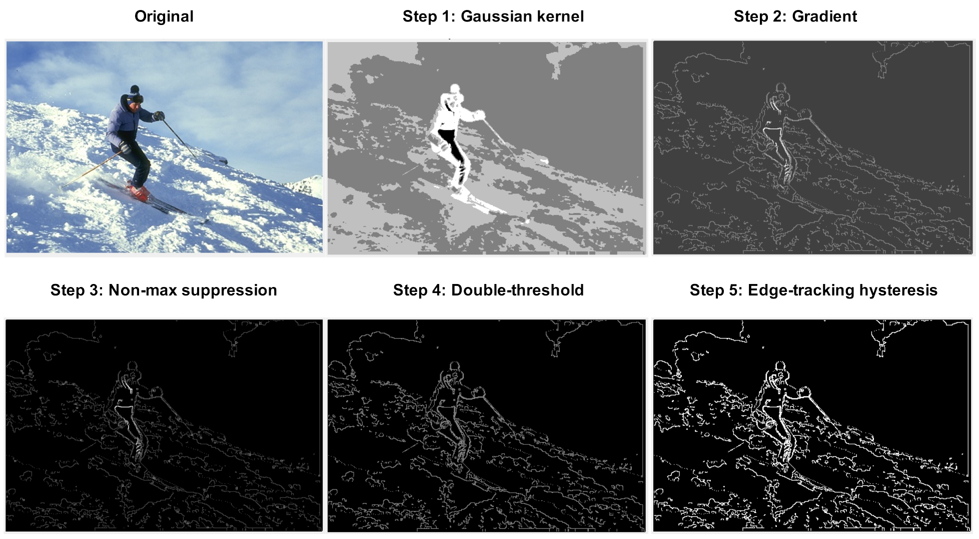

Figure 2. The original and the output images from all 5 stages of neuromorphic Canny edge detection.

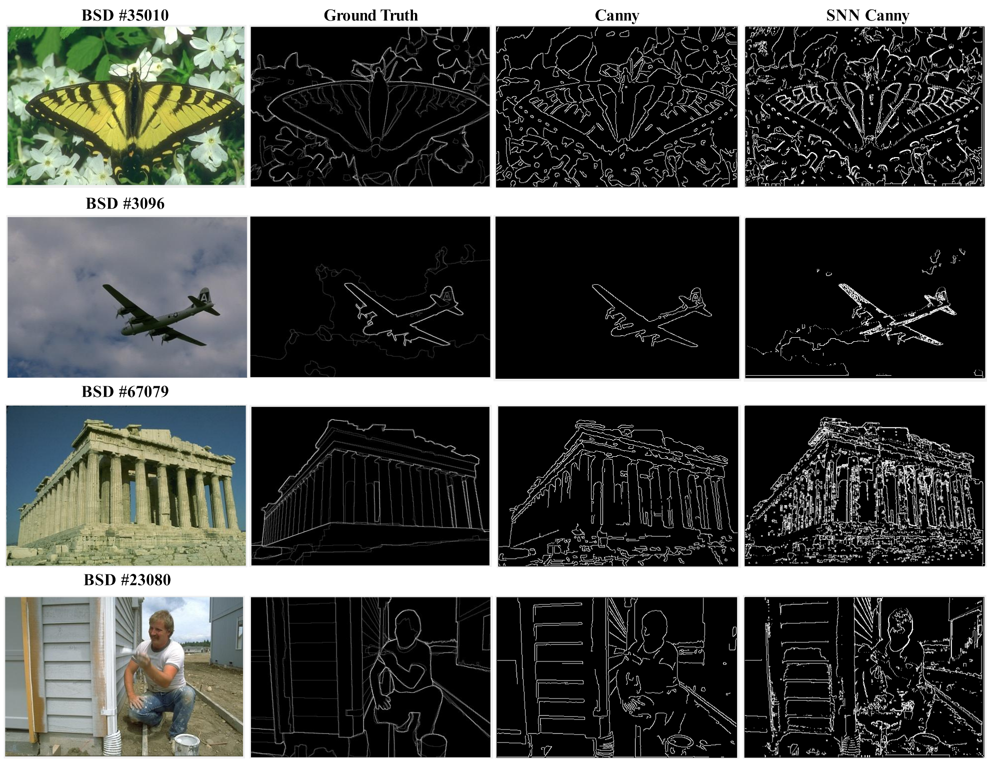

Figure 3. A comparison of the ground truth with the results from Canny edge detector and the SNN analogue of Canny edge detector for selected images from BSD500 dataset.

In image processing, it is common to select a few images and demonstrate how the method works. Statistics on training sets and test sets usually applies to machine learning experiments, which is the not the case in the present study. All the images that are pictures of natural scenes with edges and none of the examples are synthesized. From visual inspection, the SNN edge detector appears to render wider edges and detect more background information. It is possible that the difference in the SNN Canny edge map compared to the conventional Canny edge map is also related to the choice of the threshold used or the smoothing parameters. The SNN edge maps are realized in a narrower threshold range, and this makes it difficult to perform an ablation study, which is typically done in machine learning experiments. We would like to note that the primary goal of the present work is to demonstrate a spike-based implementation of Canny edge detection, not the superior computational efficiency. Currently, there is no hardware that can implement the exact version of the neuron model used in the present work to evaluate the computational time for the targeted neuromorphic domain. The context of this work is that it addresses the question of whether it is possible to compute the steps of the Canny edge algorithm by exclusively using spikes. The first step of smoothing with the Gaussian kernel is performed by using the HH neurons. The rest of the steps involve the algebraic computations in the spike domain without introducing the neurons to match the method used for the conventional Canny edge detection algorithm. We do not claim that the spike-based algorithm can be faster. The research is conducted to test if Canny-type edge maps can be generated if the data are collected with an event camera rather than using an image taken by a conventional camera.

The neuromorphic Canny edge detection network implemented in the present work uses 1 neuron/pixel with the layers arranged in a sequential fashion and introduces a Gaussian kernel in the first layer. This indicates a significant improvement compared to the state-of-the art implementation of the spiking edge detector based on an architecture with a parallel arrangement of Gabor filters, which require six neurons/pixel with the filtering layer [22]. Both works use Hodgkin–Huxley neurons for bio-inspired simulation as a common feature.

To compare the edge detectors with the ground truth, we define the performance ratio (PR) as our first metric [35]:

$$PR = \frac{True\ Edges}{False\ Edges + False\ Non - Edges} \tag{6}$$

where *True Edges* are edge pixels identified as edges, *False Edges* are non-edge pixels identified as edges, and *False Non-Edges* are edge pixels identified as non-edges. For PR, 0 is the worst score, and ∞ is the best score. Our second metric is F_1 score or F-measure, defined as

$$F_1 = \frac{2TP}{(2TP + FP + FN)}$$

where TP is true positives, FP is false positives and FN is false negatives. For F_1, 0 is the worst score and 1 is the best score. Note that F_1 is the same as the Dice similarity coefficient and it can be expressed as $F_1 = 2J/(1 + J)$, where J is the Jaccard similarity index.

Table 1 compares the metrics PR and F_1-score of the Canny detector and the neuromorphic Canny detector for the set of images displayed in Figure 3. This table also compares the two metrics for three additional images. The ground truth, Canny edge and SNN Canny edge features of these later three images are displayed for a comparison in Figure 4. We find that the F_1-score for the neuromorphic Canny detector is higher than the Canny detector for all seven images. The SNN-based Canny detector has a relatively higher PR metric than the Canny detector for the images from Figure 3 which corresponds to the top four entrees in Table 1 and a relatively lower PR metric for the images in Figure 4. A detailed observation of the edge maps of the SNN based Canny detector in Figure 4 shows several edge features that appear in the background portion of the image, mostly associated with the variation in the intensity due to texture. These additional features can be helpful to characterize the type of background in the images, and may have an advantage over conventional edge detectors for object detection in edge maps using deep learning.

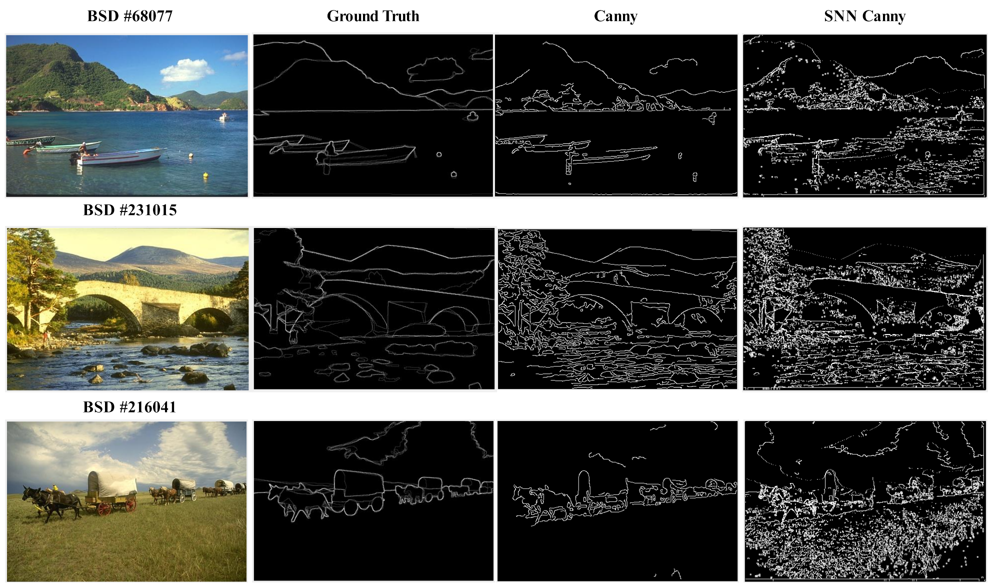

Figure 4. A comparison of the ground truth with the results from Canny edge detector and the SNN analogue of Canny edge detector for selected images from BSD500 dataset. For these images, the metrics for SNN Canny edge detector come out lower compared to the Canny edge detector.

Object detection in infrared images is another key application for edge detectors, for example in night vision gadgets or for autonomous driving. Novel edge detectors, especially those that can extract edges in low-resolution infrared images with low-SWaP requirements for hardware implementation, are sought across commercial and defense applications. In this context, we evaluated our SNN edge detector with a few infrared images from the Thermal Road Dataset [36] (license: Not found). Figure 5 shows a comparison of edge detection with SNN-based Canny detector, conventional Canny detector and Sobel detector. Ground truth edge maps are not available for this dataset to perform a quantitative comparison similar to the one presented for the RGB images in Table 1. A visual comparison of the results from the three edge detectors in Figure 5 indicates that the SNN-based Canny edge detector is able to generate edge maps very similar to the ones generated by the conventional Canny edge detector.

Biomedical image processing is a field with an increasing demand for advanced algorithms that automate and accelerate the process of segmentation. In a recent review, Wallner et al. reported multi-platofrm performance evaluation of open-source segmentation algorithms, including the Canny-edge-based segmentation method, for cranio-maxillofacial surgery [37]. In this context, it is desirable to test newer algorithms such as the one developed here using spiking neurons on medical images. To this end, we performed an edge detection experiment with a few representative medical images from the computed tomography (CT) emphysema dataset [38] (this database can be used free of charge for research and educational purposes). This dataset consists of 115 high-resolution CT slices as well as 168 square patches that are manually annotated in a subset of the slices. Figure 6 illustrates a comparison of the three edge detectors, Sobel, Canny and SNN-based Canny detectors, for a few example images from the CT emphysema dataset. Ground truth for edge maps is not available for this dataset to perform a quantitative comparison. A visual comparison of the edges generated from the three edge detectors, presented in Figure 6, shows that the SNN-based Canny edge detector is competitive with the other two edge detectors and offer an algorithmically neuromorphic alternative to the conventional Canny edge detector.

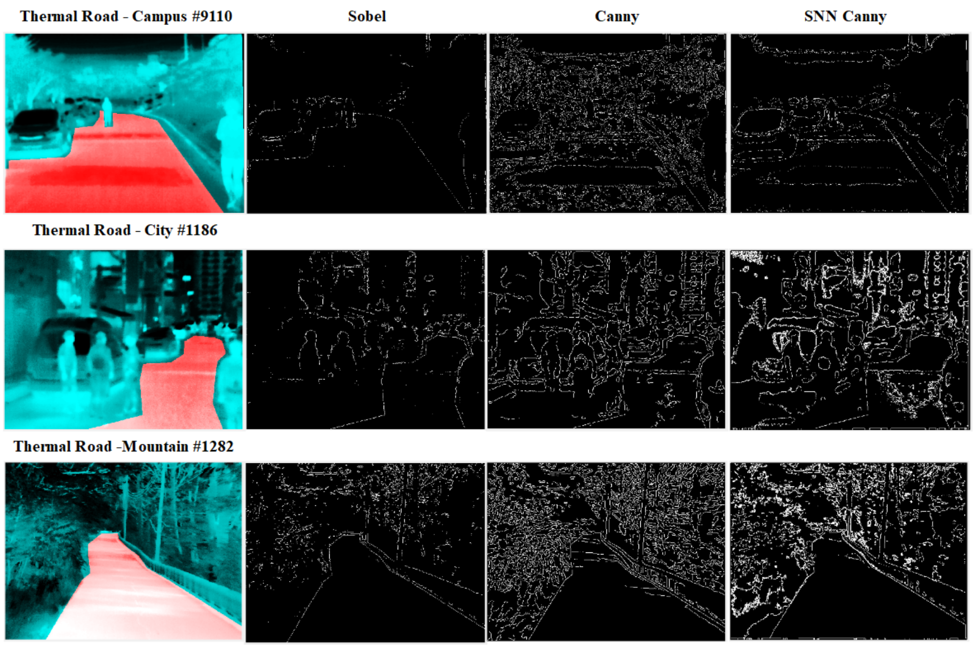
Thermal Road - Campus #9110
Sobel
Canny
SNN Canny
Thermal Road - City #1186
Thermal Road -Mountain #1282

Figure 5. A comparison of edge detection by Sobel, Canny, and SNN Canny edge detectors for images from thermal road dataset.

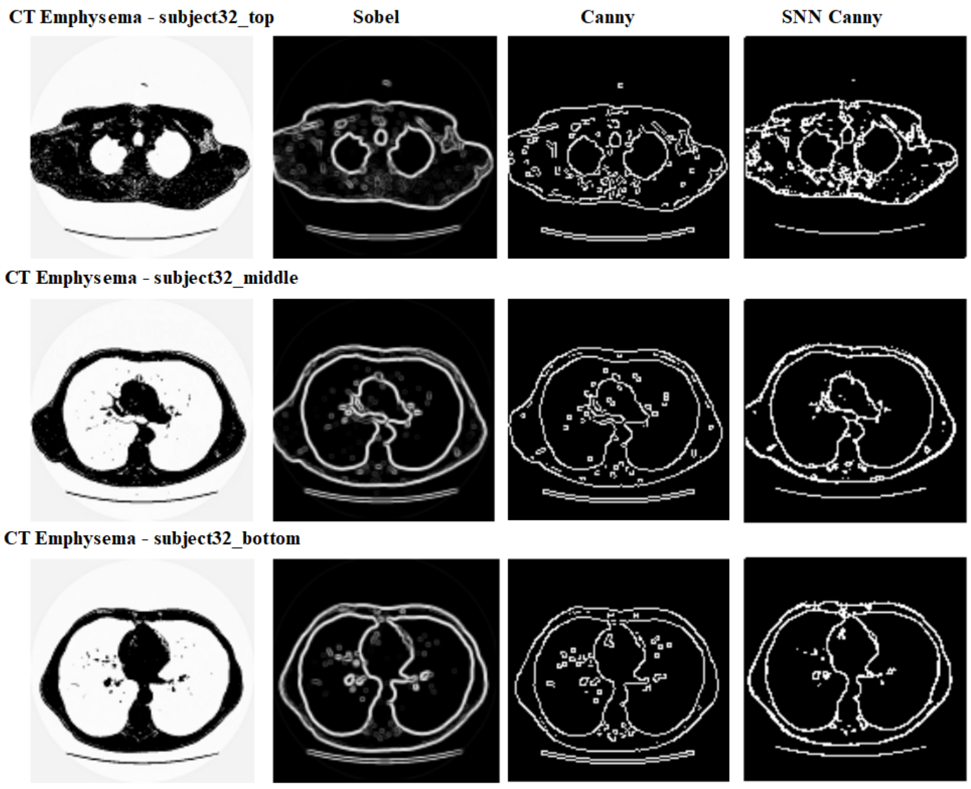
CT Emphysema - subject32_top
Sobel
Canny
SNN Canny
CT Emphysema - subject32_middle
CT Emphysema - subject32_bottom

Figure 6. A comparison of edge detection by Sobel, Canny, and SNN Canny edge detectors for medical images from CT emphysema dataset.

Table 1. Comparison of performance ratio, PR, and F_1-score for selected images.

Image	Canny	SNN Canny	Canny	SNN Canny
	PR	PR	F_1-Score	F_1-Score
35,010	15.7	17.5	0.009	0.018
3096	7.3	11.6	0.002	0.013
67,079	12.7	17.4	0.014	0.034
23,080	12.6	13.8	0.014	0.023
68,077	11.1	10.7	0.015	0.016
231,015	15.5	14.2	0.015	0.022
216,041	13.4	9.9	0.088	0.102

6. Conclusions

In conclusion, we present a spiking neural network (SNN) implementation of the Canny edge detector as its neuromorphic analogue by introducing algorithms for spike based computation in the five steps of the conventional algorithm with the conductance-based Hodgkin–Huxley neuron as the building block. Edge detection examples are presented for RGB and infrared images with a variety of objects. A quantitative comparison of the edge maps from the SNN-based Canny detector, conventional Canny detector and a Sobel detector using the F_1-score as a metric, shows that the neuromorphic implementation of the Canny edge detector achieves better performance. The SNN architecture of the Canny edge detector also offers promise for image processing and object recognition applications in the infrared domain. The SNN Canny edge detector is also evaluated with medical images, and the edge maps compare well with the edges generated with the conventional Canny detector. Future work will focus on the implementation of the algorithm on an FPGA or on a neuromorphic chip for hardware acceleration and testing in an infrared object detection task potentially with edge maps as features together with a pre-processing layer to remove any distortions, enhance contrast, remove blur, etc., and a spiking neuron layer as a final layer to introduce a machine learning component. An extension of the SNN architecture of the Canny edge detector with additional processing layers for object detection in LiDAR point clouds would be another interesting new direction of research [39].

Funding: This work was internally funded by Riverside Research's Open Innovation Center.

Data Availability Statement: Not applicable.

Conflicts of Interest: The author declares no conflict of interests.

References

1. Akopyan, F.; Sawada, J.; Cassidy, A.; Alvarez-Icaza, R.; Arthur, J.; Morella, P.; Imam, N.; Nakamura, Y.; Dutta, P.; Nam, G.-J.; et al. TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2015**, *34*, 1537–1557. [CrossRef]
2. Davies, M.; Srinivasa, N.; Lin, T.-H.; Chinya, G.; Cao, Y.; Choday, S.H.; Dimou, G.; Joshi, P.; Imam, N.; Jain, S.; et al. Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. *IEEE Micro* **2018**, *38*, 82–99. [CrossRef]
3. Posey, B.M. What Is the Akida Event Domain Neural Processor? 2022. Available online: https://brainchipinc.com/what-is-the-akida-event-domain-neural-processor/ (accessed on 17 January 2022).
4. Vanarse, A.; Osseiran, A.; Rassau, A.; van der Made, P. A Hardware-Deployable Neuromorphic Solution for Encoding and Classification of Electronic Nose Data. *Sensors* **2019**, *19*, 4831. [CrossRef] [PubMed]
5. Roberts, L.G. Machine Perception of three-Dimensional Solids. Doctoral Dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 10 May 1963.
6. Sobel, I.; Feldman, G. A 3 × 3 Isotropic Gradient Operator for Image Processing, a talk at the Stanford Artificial Intelligence Project, bf 1968, 271–272. Available online: https://www.researchgate.net/publication/281104656_An_Isotropic_3x3_Image_Gradient_Operator (accessed on 5 June 2022).
7. Prewitt, J.M. Object enhancement and extraction. *Pict. Process. Psychopictorics* **1970**, *10*, 15–19.
8. Shrivakshan, G.T.; Chandrasekar, C. A comparison of various edge detection techniques used in image processing. *Int. J. Comput. Sci. Issues (IJCSI)* **2012**, *9*, 269.
9. Marr, D.; Hildreth, E. Theory of edge detection. *Proc. R. Soc. Lond. Ser. Biol. Sci.* **1980**, *207*, 187–217.
10. Mehrotra, R.; Namuduri, K.R.; Ranganathan, N. Gabor filter-based edge detection. *Pattern Recognit.* **1992**, *25*, 1479–1494. [CrossRef]
11. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *6*, 679–698. [CrossRef]
12. Jain, R.; Rangachar, K.; Schunck, B.G. *Machine Vision*; McGraw-Hill (Publishers) Inc.: New York, NY, USA, 1995.
13. Egmont-Petersen, M.; de Ridder, D.; Handels, H. Image processing with neural networks—A review. *Pattern Recognit.* **2002**, *35*, 2279–2301. [CrossRef]
14. Auge, D.; Hille, J.; Mueller, E.; Knoll, A. A survey of encoding techniques for signal processing in spiking neural networks. *Neural Process. Lett.* **2021**, *53*, 4693–4710. [CrossRef]
15. Brette, R.; Rudolph, M.; Carnevale, T.; Hines, M.; Beeman, D.; Bower, J.M.; Diesmann, M.; Morrison, A.; Goodman, P.H. Simulation of networks of spiking neurons: A review of tools and strategies. *J. Comput. Neurosci.* **2007**, *23*, 349–398. [CrossRef]
16. Frémaux, N.; Gerstner, W. Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules. *Front. Neural Circuits* **2016**, *9*, 85. [CrossRef]
17. Maass, W. Networks of spiking neurons: The third generation of neural network models. *Neural Netw.* **1997**, *10*, 1659–1671. [CrossRef]
18. Ponulak, F.; Kasinski, A. Introduction to spiking neural networks: Information processing, learning and applications. *Acta Neurobiol. Exp.* **2011**, *71*, 409–433.
19. Wu, Q.; McGinnity, M.; Maguire, L.; Belatreche, A.; Glackin, B. Edge Detection Based on Spiking Neura Network Model. In *Lecture Notes in Computer Science, Proceedings of the Advanced Intelligent Computing Theories and Applications, With Aspects of Artificial Intelligence, ICIC 2007, Qingdao, China, 21–24 August 2007*; Huang, D.-S., Heutte, L., Loog, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4682.
20. Clogenson, M.; Kerr, D.; McGinnity, T.M.; Coleman, S.A.; Wu, Q. Biologically inspired edge detection using spiking neural networks and hexagonal images. In Proceedings of the International Conference on Neural Computation Theory and Applications, Paris, France, 24–26 October 2021; SciTePress: Setúbal, Portugal, 2011; pp. 381–384.
21. Yedjour, H.; Meftah, B.; Lézoray, O.; Benyettou, A. Edge detection based on Hodgkin–Huxley neuron model simulation. *Cogn. Process.* **2017**, *18*, 315–323. [CrossRef]
22. Vemuru, K.V. Image edge detector with Gabor type filters using a spiking neural network of biologically inspired neurons. *Algorithms* **2020**, *13*, 165. [CrossRef]
23. Hodgkin, A.L.; Huxley, A.F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.* **1952**, *117*, 500–544. [CrossRef]
24. Hosoya, T.; Baccus, S.A.; Meister, M. Dynamic predictive coding by the retina. *Nature* **2005**, *436*, 71–77. [CrossRef]

25. Kandel, E.R.; Shwartz, J.H.; Jessell, T.M.; Seigelbaum, S.; Hudspeth, A.J.; Mack, S. *Principles of Neural Science*; Edward Amold (Publishers) Ltd.: London, UK, 1981.
26. Lindsay, G.W. Convolutional Neural Networks as a Model of the Visual System: Past, Present, and Future. *J. Cogn. Neurosci.* **2021**, *33*, 2017–2031. [CrossRef]
27. Curtis, D.R.; Eccles, J.C. The time courses of excitatory and inhibitory synaptic actions. *J. Physiol.* **1959**, *145*, 529–546. [CrossRef]
28. Destexhe, A.; Rudolph, M.; Fellous, J.-M.; Sejnowski, T.J. Fluctuating synaptic conductances recreate in vivo-like activity in neocortical neurons. *Neuroscience* **2001**, *101*, 13–24. [CrossRef] [PubMed]
29. Aaby, D. *A Comparitive Study of Numerical Methods for the Hodgkin-Huxley Model of Nerve Cell Action Potentials*; University of Dayton: Dayton, OH, USA, 2009.
30. Siciliano, R. The Hodgkin-Huxley Model, Its Extensions, Analysis and Numerics. 2012. Available online: https://www.math.mcgill.ca/gantumur/docs/reps/RyanSicilianoHH.pdf (accessed on 17 November 2021).
31. Service, R.F. Learning Curve: New brain-inspired chips could provide the smarts for autonomous robots and self-serving cars. *Science* **2021**, *374*, 24–25. [CrossRef] [PubMed]
32. Arbelaez, P.; Maire, M.; Fowlkes, C.; Malik, J. Contour Detection and Hierarchical Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 898–916. [CrossRef] [PubMed]
33. Martin, D.; Fowlkes, C.; Tal, D.; Malik, J. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In Proceedings of the Eighth IEEE International Conference on Computer Vision. ICCV 2001, Vancouver, BC, Canada, 7–14 July 2001; Volume 2, pp. 416–423.
34. Glackin, B.; Harkin, J.; McGinnity, T.M.; Maguire, L.P.; Wu, Q. Emulating spiking neural networks for edge detection on FPGA hardware. In Proceedings of the 2009 International Conference on Field Programmable Logic and Applications, Sydney, Australia, 31 August–2 September 2009; pp. 670–673.
35. Khaire, P. Measures of Edge Detection, MATLAB Central File Exchange. Available online: https://www.mathworks.com/matlabcentral/fileexchange/52205-measures-of-edge-detection (accessed on 17 November 2021).
36. Yoon Jae, S.; Park, K.; Hwang, S.; Kim, N.; Choi, Y.; Rameau, F.; Kweon, I.S. Thermal-infrared based drivable region detection. In Proceedings of the 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, Sweden, 19–22 June 2016; pp. 978–985.
37. Wallner, J.; Schwaiger, M.; Hochegger, K.; Gsaxner, C.; Zemann, W.; Egger, J. A review on multiplatform evaluations of semi-automatic open-source based image segmentation for cranio-maxillofacial surgery. *Comput. Methods Programs Biomed.* **2019**, *182*, 105102. [CrossRef]
38. Sorensen, L.; Shaker, S.B.; De Bruijne, M. Quantitative analysis of pulmonary emphysema using local binary patterns. *IEEE Trans. Med. Imaging* **2010**, *29*, 559–569. [CrossRef]
39. Choi, Y.; Kim, N.; Hwang, S.; Park, K.; Yoon, J.S.; An, K.; Kweon, I.S. KAIST Multi-Spectral Day/Night Data Set for Autonomous and Assisted Driving. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 934–948. [CrossRef]