Edge Detection Using Integrate and Fire Neuron Model

M. Ozan İNCETAŞ*, Rukiye UZUN ARSLAN**

1Alanya Alaaddin Keykubat University, ALTSO Vocational School, Department of Electric and Energy, 07450, Antalya, Turkey
2Zonguldak Bulent Ecevit University, Engineering Faculty, Department of Electrical & Electronics Engineering, 67100, Zonguldak, Turkey

*Corresponding author: ozan.incetas@alanya.edu.tr

Abstract: Edge detection is one of the most basic stages of image processing and have been used in many areas. Its purpose is to determine the pixels formed the objects. Many researchers have aimed to determine objects’ edges correctly, like as they are determined by the human eye. In this study, a new edge detection technique based on spiking neural network is proposed. The proposed model has a different receptor structure than the ones found in literature and also does not use gray level values of the pixels in the receptive field directly. Instead, it takes the gray level differences between the pixel in the center of the receptive field and others as input. The model is tested by using BSDS train dataset. Besides, the obtained results are compared with the results calculated by Canny edge detection method.

Keywords
Edge detection, Receptive field, Spiking neuronal network

Toplava ve Ateşle Nöron Modeli Kullanılarak Kenar Algılama

Anahtar Kelimeler
Kenar algılama, Alıcı bölge, Işnecıklı sınır ağı


1. Introduction

Edge detection is very important operation for the image processing because it is used in many areas such as feature extraction, segmentation, object recognition and image retrieval. Edge detection can be defined as a sudden change between neighboring pixels. There have been improved many approaches based differences and similarities of neighboring pixels in an image [1-3].

Many image processing techniques, including edge detection algorithms, actually aim to reach the perception of the human eye. Human visual system (HVS) has a quite complex process, which starts within the retina and becomes considerably more complex at other stages (the visual pathways and visual cortex) [4-7]. Many researches by neurologist and computer scientists have been devoted to understand the operation of this complex process and to develop models simulating its behavior [8-13]. In this context, spiking neural networks (SNN) imitate more exactly the biological image processing in HVS. SNNs permit real-time processing which has high speed and computational power due to the usage of temporal coding scheme [14-18]. SNNs use simple neuron models and process the information via encoding by the spikes. In literature, one can find many studies for edge detection based biological neural systems. For example, Wu et al. [19] proposed a network model based on SNNs for edge detection. Meftah et al. [20] developed a SNN model to fulfill segmentation and edge detection. Kerr et al. [21] presented an approach for edge detection using both SNNs and a biologically plausible hexagonal pixel arrangement with hexagonally arranged near-
circular receptive fields. In another study, a bio-inspired model called The Perceptual boundaRy dEtection Neural (PREEN) in the recurrent interactions of the early visual areas was proposed by Diaz-Pernas et al. [22] to detect color natural scenes boundaries. They concluded that the proposed model gives better results as compared with the best algorithms for some images in Berkeley Segmentation Test Dataset. A computational model, named as COF, is developed for the orientation-selective cell in the primary visual cortex - V1 [23]. In a recent study, Yedjour et al. [18] have proposed a SNN using Hodgkin-Huxley (HH) model for edge detection. They analyzed the performance of the model by using five different edge detection methods. Although it is stated that the model is evaluated with BSDS images and gives more successful results than the classical models, it is seen that the techniques used for comparison are more suitable for the test of noise filters. Most of these aforementioned studies took in consideration the simplified neuron models even though there is a more realistic neuron model (HH model) which simulate the activity of a neuron with a high degree of precision. However, the usage of HH model requires long simulation time, powerful and expensive machines, due to its computational complexity. Because of this, it has been preferred to use simplified neuron models. In this study, we used conductance-based integrate-and-fire (IF) neuron model to detect edges in images.

Remainder of this paper is organized as follows. “Conductance-Based Integrate-and-Fire Neuron Model” section gives a brief introduction to integrate-and-fire neuron model. The architecture of the network is presented in ‘Network structure’ section. The simulation results and discussions are presented in “Simulation results” and “Conclusion” sections, respectively.

2. Material and Method
2.1. Conductance based integrate and fire neuron model

HH model proposed by Hodgkin & Huxley [24] is the first mathematical neuron model, which describes the electrical behavior of neuron excellently. The model uses a set of nonlinear differential equations to characterize how action potential (or spike) is initiated and propagated [18, 25]. However, the usage of this model has some drawbacks; such as, the requirement of solving a set of several first-order differential equations induces that the numerical implementations are computationally expensive and the analysis are difficult. Therefore, in literature, there have been proposed more simple neuron model such as integrate-and-fire (IF), FitzHugh–Nagumo (FHN), Izhikevich neuron models etc. [26-29]. Among these, IF neuron model is the much simplified model and captures many of the principal features of neuron dynamics, thus it is quite popular at discussing of the neuronal coding, memory or neuron’s dynamics [28, 30, 31]. The model is also more useful as compared to HH model, if the model is applied to large-scale neuronal networks in terms of computational complexity [19]. It is well known that the variation of ion channels (Na+ and K+) conductance at the HH model have vital effects in spike generation [32]. By taking into this consideration, conductance-based IF model was conceived [33].

In the conductance based integrate-and-fire model, the time evolution of the membrane potential \(v(t)\) is given as follows [19, 28, 34-36]:

\[
\frac{dv(t)}{dt} = g_l(E_l - v(t)) + \sum_j \frac{w_j g_{syn}(t)}{A_{syn}} (E_{syn} - v(t))
\]

(1)

Where \(c_m\) is the capacitance, \(g_l\) represents the conductance and \(E_l\) is reversal potential of the membrane, respectively. \(E_{syn}\) is the reversal potential of inhibitory (i) and excitatory (e) synapses where \(s \in \{i, e\}\), respectively. \(w_j\) represents the strength of the synapse \(j\), and the membrane patch area \(A_{syn}\) is linked to the corresponding synapse. \(g_{syn}\) represents the conductance of synapse \(j\). If \(v(t)\) reaches a certain threshold \(v_{th}\) (spiking threshold), it is instantaneously reset to a lower value \(v_r\) (reset potential) for a time \(t_{ref}\) (refractory time) and a spike occurs. A neuron receives spike trains from three afferent neurons in a receptive field is given in Figure 1.

2.2. Network structure

Figure 2 shows the preferred network model which is inspired from Wu et al.’s study. As in the most of image processing studies based SNN, the network structure has three layer: receptor layer, intermediate layer and output layer. Receptor layer comprises of photoreceptors related to each pixels of the image.

The intermediate layer is constituted with different types of neurons to obtain the receptive fields. The main difference of our study from the existing studies is in the receptive layer where neurons have synaptic connections. The receptive layer given in Figure 2 consists of four different receptive field (RF). In these
fields, gray level values are not directly used, instead the absolute difference between the gray level values of pixels shown with yellow and the gray level value of center pixel is utilized. Besides, the blue pixels are ignored and the gray level values of these pixels are taken as zero. These four different receptive field allow the determination of edges in different directions by making synaptic connections with four different neurons in the intermediate layer.

![Proposed SNN structure](image)

**Figure 2. Proposed SNN structure**

The output of these neurons was summed by each neuron in the output layer to obtain corresponding neuron’s firing rate. Any direction edge within the input image can be obtained by means of the firing rates of the neurons at the output layer. To put a finer point on it, as shown from Figure 2, the preferred network model has four parallel arrays that have the same dimension with the receptor layer and have flagged as $N_1$, $N_2$, $N_3$ and $N_4$ for only one output neuron. These layers linked to receptor layer by changeable weight matrices help one to fulfill the processing of the edges (right, left, up and down). The size of these weight matrices can be changed by taking into consideration the receptive field’s width. For instance, neuron $N_1$ links to the receptive field (RF) through a synaptic connection ($w^{N_1}$) and gives an output if there is an edge. $w^{N_1}$ contains synapses which are enhances the membrane potential.

If all of the pixels within the RF have the same gray level value, the absolute value of the difference between the gray level values of the neighboring pixels and the gray level values of the center pixel will be 0. This will not change the membrane potential of neuron $N_1$, thus neuron $N_1$ will not fire any spike. On the other hand, if the image in the RF has an upper edge, the $N_1$ will fire spike with the help of synaptic connections found just below the center.

One can think that $w^{N_1}$ is like a filter that find the upper-edge within the receptive field. Similarly, within the receptive field, the down-edge can be detected by neuron $N_2$ with the synaptic matrix with $w^{N_2}$; the left-edge can be detected by neuron $N_3$ with the synaptic matrix with $w^{N_3}$ and the right-edge can be detected by neuron $N_4$ with the synaptic matrix with $w^{N_4}$. Finally, the neuron in the output layer sums the all of the outputs of these neurons at the intermediate layer and can elicit any direction edge within the receptive fields (RFs). This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation as well as the experimental conclusions that can be drawn.

As stated before, the conductance based IF neuron model is simple and easy to analyze as compared with HH neuron model. Therefore, aforementioned network model is carried out based on conductance based IF neuron model. In the model, $(x, y)$ represents the pixel coordinate in the RFs and each pixel in RFs can be defined as the absolute difference of gray level values $(G_{x,y})$. For the receptive fields peak conductance value of each pixel is calculated by the following expressions.

$$R_{x,y} = |G_{x,y} - G_{x_c,y_c}|$$  \hspace{1cm} (2)

$$q_{x,y} = \alpha R_{x,y}$$  \hspace{1cm} (3)

where $G_{x_c,y_c}$ is the gray level value of the center pixel in RFs, $R_{x,y}$ is the absolute difference of center pixel gray level $G_{x_c,y_c}$ and its neighbor pixel gray level $G_{x,y}$, $q_{x,y}$ is the peak conductance and $\alpha$ is the normalization coefficient.

Since the synaptic connections are based on the absolute difference between the gray level values, the peak synaptic conductance value of a neighboring pixel having the same value with the center pixel will be 0 (zero). Hence, the all-synaptic connections have been assumed as excitatory synapse. When all the pixels in the RFs had the same value, there would not be found any spike because all of the peak conductance would be equal zero.

For a neuron in the intermediate layer (e.g. $N_1$), the following equations are given:

$$\frac{dg_{x,y}^{syn}(t)}{dt} = -\frac{1}{\tau_{syn}}g_{x,y}^{syn}(t) + q_{x,y}$$  \hspace{1cm} (4)

$$c_m \frac{dv_{N_1}(t)}{dt} = g_l(E_l - v_{N_1}(t)) + \sum_{(x,y)\in RF_1} \frac{w^{N_1}_{x,y}(t)g_{x,y}^{syn}(t)}{A_{syn}} (E_{syn} - v_{N_1}(t))$$  \hspace{1cm} (5)

Where $c_m$ is the membrane capacitance, $g_l$ represents the membrane conductance and $E_l$ is reversal potential of the membrane. $E_{syn}$ is the reversal potential of the synapses connected to RF, $g_{x,y}^{syn}$ represents the conductance of the synapse corresponding to the pixel in RF and $A_{syn}$ is the membrane patch area connected to the synapse. $w_{x,y}^{N_1}$ shows the weight of the synapses and calculates as:

$$w^{N_1}_{x,y} = \begin{cases} 0 & \text{if } (y - y_c) \neq 1 \\ w_{max} & \text{if } (y - y_c) = 1 \end{cases}$$  \hspace{1cm} (6)
Where \((x_c, y_c)\) gives the center of the RF, \((\delta_x, \delta_y)\) are constants and \(w_{\text{max}}\) are the maximum weight for synapses. Similarly, these equations are also valid and should be calculated for other neurons at the intermediate layer (that is \(N_2, N_3, N_4\)). If \(v(t)\) reaches a certain threshold \(v_{\text{th}}\) (spiking threshold), it is instantaneously reset to a lower value \(v_r\) (reset potential) for a time \(\tau_{\text{ref}}\) (refractory time) and a spike occurs. Let \(S_{Ni}(t)\) gives a spike train, which is fired by neuron \(i\):

\[
S_{Ni}(t) = \begin{cases} 1 & \text{if neuron } i \text{ fires a spike at time } t \\ 0 & \text{if there is no spike at time } t \end{cases}
\]

Finally, each neuron in the output layer (\(N(x', y')\)) is defined by the following equations [19]:

\[
g_{x'y'}^{\text{syn}}(t) = \frac{-1}{\tau_{\text{syn}}}g_{x'y'}^{\text{syn}}(t) + (w_{\text{max}}S_{N1}(t) + w_{N2}S_{N2}(t) + w_{N3}S_{N3}(t) + w_{N4}S_{N4}(t))
\]

\[
c_m \frac{dv_{x'y'}(t)}{dt} = g_l \left( E_l - v_{x'y'}(t) \right) + \frac{g_{x'y'}^{\text{syn}}(t)}{A_{\text{syn}}} \left( E_{\text{syn}} - v_{x'y'}(t) \right)
\]

It should be noted that each neuron at the output layer is linked to intermediate neuron merely by excitatory synapses.

### 3. Results and Discussion

The proposed model was performed in MATLAB using the following parameters: \(v_{\text{th}} = -60 \text{ mV}\), \(v_{\text{reset}} = -70 \text{ mV}\), \(E_{\text{syn}} = 0\), \(E_l = -70 \text{ mV}\), \(g_l = 1 \mu \text{S} \text{mm}^2\), \(c_m = 10 \mu \text{F} \text{mm}^2\), \(\tau_{\text{syn}} = 4 \text{ ms}\), \(\tau_{\text{ref}} = 6 \text{ ms}\) and \(A_{\text{ex}} = 0.028953 \text{ mm}^2\). The strength of the synapses are adjusted by the maximal weights relevant synapses to guarantee that the neuron does not fire if the input image has a uniform structure. \(w_{\text{max}}\) is taken as 0.7093. The absolute differences of gray level values are set to be in the range of 0 to 1.

To do this, \(\alpha\) is determined by \(1/255\). \(\delta_x = 6, \delta_y = 2\), and the width and height of the RFs are set to 5. The matrices used for \(w_{N1}^{x'y'}\) and \(w_{N3}^{x'y'}\) are given as follows:

The edges of the original image shown in Figure 3 are determined with the preferred model and the result is shown as a gray level image. Firing rates of each \(N(x', y')\) neuron in the output layer are used as gray level values. The value of the gray level approaches to 255, i.e. to white, at high firing rate pixels; whereas at low firing rate pixels, its values are 0, i.e. black. The preferred model was tested using 200 images on BSDS dataset and the results were calculated as F-scores. The F-score is the harmonic mean of the precision and recall values. The precision value is the ratio of the true edge pixels in all selected edge pixels by algorithm and recall is the ratio of the edge pixels selected by algorithm in image. The results were compared with the Canny edge detector known as the most common edge detection method.

In Figure 4, the first row shows 3 images in BSDS Train dataset. The edges of these images that are determined by the users manually are displayed at the second row of the figure. The results, which were determined by using the proposed model and Candy edge detection method, are given in the third and fourth rows of the figure, respectively. Both of these methods are quite successful for these images. The success of the proposed model as F-score were 0.7681, 0.7369 and 0.9536 by column order, whereas the success of the Candy edge detection method was obtained 0.7372, 0.7135 and 0.9542, respectively.

Figure 5 shows the results obtained for 2 different sample images. The first column is the original images in the BSDS Train dataset, whereas the grand truth images of them is given in the second column. The results obtained from the proposed model and Candy edge detection method are presented in the third and fourth columns, respectively. The recall, precision and F-score values of the proposed model for the face image in the first row of Figure 5, respectively, are 0.7773, 0.7332 and 0.7546, while these values are calculated as 0.8689, 0.7890 and 0.8271 by the Candy edge detection method. The image in the second row is the image in which the worst results are obtained by these two methods. The recall, precision and F-score values for this image
calculated by the proposed model are 0.6006, 0.0955 and 0.1648, whereas they are computed as 0.8718, 0.0864 and 0.1572 by Candy edge detection method.

Table 1 depicts the average values of the edge detection results of all images. It is seen that F-score values obtained from both methods are quite close to each other. But the recall values obtained with the canoe edge detection method are higher as seen from the average values. These shows that Canny edge detection method is more successful to find the right edges. On the other hand, the precision values obtained with the proposed model are higher than the values calculated by the Candy edge detection method. This shows that the proposed method produces more successful results in terms of pixels accidentally marked as an edge.

Table 1. Average Results.

<table>
<thead>
<tr>
<th>Edge Detection Methods</th>
<th>Recall</th>
<th>Precision</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canny</td>
<td>0.8013</td>
<td>0.4344</td>
<td>0.5357</td>
</tr>
<tr>
<td>Proposed model</td>
<td>0.6876</td>
<td>0.4766</td>
<td>0.5381</td>
</tr>
</tbody>
</table>

4. Conclusion

Although SNNs are used for edge detection, it is seen that there are very few studies in the literature. SNNs, which can work similarly to human visual system, are still used with different designs. In this study, a new edge detection technique based on SNN is proposed. The proposed model is tested on 200 images in the BSDS train dataset. Besides, canoe edge detection method, which is one of the most known methods for edge detection, is used on the same images for comparison. The F-score values are calculated for all of them. In addition, this is the first that the results are given as f-score in the edge detection studies using SNNs. It is observed that there are major differences in Recall and precision values obtained by proposed method and Canny edge detection method, although there is a slight difference in the calculated F-score values. For this reason, it is planned to make changes on the network structure in order to increase the Recall values in future studies.

References

Information Sciences, Rochester Institute of Technology.


