Images Classification by Pulse Coupled Neural Networks

Rafidison Maminiaina Alphonse¹, Ramafiarisona Hajasoa Malalatiana²

¹Telecommunication-Automatic-Signal-Image-Research, Laboratory/Doctoral School in Science and Technology of Engineering and Innovation/University of Antananarivo Antananarivo 101, Madagascar
mamynaina[at]gmail.com

²Telecommunication-Automatic-Signal-Image-Research, Laboratory/Doctoral School in Science and Technology of Engineering and Innovation/University of Antananarivo Antananarivo 101, Madagascar
mhramafiarisona[at]yahoo.fr

Abstract: The purpose of this paper is a presentation of new method of images classification. When we talk about this subject, the first reflex is thinking on convolutional neural network (CNN) such as LeNet, AlexNet, GoogLeNet, ResNet, etc. They have a good performance however another way to improve always exists. We introduce the notion of foveation which consists of collecting all pertinent information in different region of an image. Pulse coupled neural networks (PCNN) is a strong tool to accomplish this foveation task. Once, essential information is extracted, we cannot forward directly to fully connected neural network (FCNN) due of large data quantity so we compress them with Haar wavelet transform. Reshape compressed picture will be presented to FC. This neural network ensures the images classification as per the input. The singularity of this approach is the minimum time response and high accuracy percentage. Output's total value is one because softmax function is the activation function for last layer. The neuron which has higher value indicates the corresponding class of the image.

Keywords: blurring filter, foveation, pulse coupled neural networks, wavelet transform, fully connected neural network, softmax, accuracy.

1. Introduction

Searching image in database by keyword is always the subject of research in image analysis area. We never can replace an intelligence human but develop a new approach is never waste time also. We cannot delegate a lot of human resource to accomplish a task then automation is required even the performance ideal cannot be reached. So the solution is translating mammalian visual system into mathematics.

The mammalian visual system is considerably more elaborate than simply processing an input image with a set of inner products. Many operations are performed before decisions are reached as to the content of the image. A mathematical operation is thus performed on the image before it even leaves the eye. The eye also receives feedback information. We humans do not stare at images, we foveate.

Pulse coupled neural networks (PCNN) is a suitable neural network to perform this foveation operation and fully connected neural network (FCNN) assume images classification. To understand the new method, we should know beforehand some technical subject so in the next paragraph, we will talk about PCNN, Haar wavelet transform and FCNN. After having a clear understanding of them, we present our related works followed by testing and conclusion.

2. Pulse Coupled Neural Network

In this section, we will give a brief review of PCNN. It was originally presented by Eckhorn et al in order to explain the synchronous neuronal burst phenomena in the cat visual cortex. The model neuron consists of three parts: the dendritic tree, the linking modulation, and the pulse generator.

The role of the dendritic tree is receiving inputs from two kinds of receptive fields. Depending on the type of receptive field, it is subdivided into two channels, the linking and the feeding. The linking receives local stimulus. On the other hand, the feeding receives external stimulus as well as local stimulus. Each channel has the state and change its state depending on its current state. In the linking modulation, it gathers the outputs from two channels. The linking modulation is made by adding a bias to the linking and multiplying this by the feeding. The resultant quantity Uj is called the internal state of a neuron. The pulse generator generates the pulse dep ending on the internal state and the threshold. The threshold changes state depending on the output of the neuron. Whole behavior of PCNN is explained as follows. If the pulse is generated now, it makes the threshold higher suddenly. Then the pulse is not generated, but the threshold is getting smaller exponentially depending on its time constant a. As a result, the pulse can be generated again. Of course the pulse influences the initial states of the other neuronseach other.

2.1 PCNN by T.Lindblad and J.M.Kinser

T.Lindblad and J.M.Kinser [1] have applied PCNN to the image processing. The dendritic tree of their models given by

\[ F_{ij}[n] = \exp(-\alpha \delta_{ij}) F_{ij}[n - 1] + S_{ij} + V_p \sum_{kl} M_{ijkl} Y_{kl}[n - 1] \]  
\[ L_{ij}[n] = \exp(-\alpha \delta_{ij}) L_{ij}[n - 1] + V_l \sum_{kl} W_{ijkl} Y_{kl}[n - 1] \]
where $M$ and $W$ are the constant synaptic weights, and $S$ is the external stimulus. $V_F$ and $V_L$ are normalizing constants. $\alpha_F$ and $\alpha_L$ are the time constants, $\alpha_F < \alpha_L$. The linking modulation (the internal state) is given by

$$U_{ij}[n] = F_{ij}[n](1 + \beta \cdot L_{ij}[n])$$  \hspace{1cm} (3)

where $\beta$ is the linking weight parameter. The pulse generator is the step function as

$$Y_{ij}[n] = \begin{cases} 1 & \text{if } U_{ij}[n] > \Theta_{ij}[n-1] \\ 0 & \text{Otherwise} \end{cases}$$  \hspace{1cm} (4)

$$\Theta_{ij}[n] = \Theta_{ij}[n-1]\exp(-\alpha_0) + V_0 Y_{ij}[n-1]$$  \hspace{1cm} (5)

where $V_0$ is a normalizing constant. $\alpha_0$ is the time constant. Note that three time constants are chosen such that they satisfy the constraint: $\alpha_F < \alpha_G < \alpha_L$. Then output of the neuron takes the binary value, 0 or 1. Such pulse generator is called the binary pulse generator [2].

![Figure 1: The outline of PCNN with the sigmoidal pulse generator](image)

**2.2 PCNN with sigmoidal pulse generator**

In the model of T.Lindblad and J.M.Kinser, the output of the neuron takes binary value, so PCNN would select a lot of candidates for the foveation points. To avoid such confusing situation, we adapt the sigmoidal pulse generator

$$Y_{ij}[n] = \frac{1}{1 + \exp[-\gamma(U_{ij}[n] - \theta_{ij}[n-1])]}$$  \hspace{1cm} (6)

where is the parameter for the sigmoid function. Figure 1 shows the outline of our model. The output of the neuron takes the analogue value from 0 to 1 [2].

**3. 2D-Haar-Wavlet Transform**

Wavlet transform has the capability to offer some information on frequency-time domain concurrently. In this transform, time domain is passed through high-pass and low-pass filters to extract high and low frequencies respectively. This process is repeated for a number of times and each time a section of the signal is drawn out. DWT analysis splits signal into two classes (i.e. Approximation and Detail) by signal decomposition for different frequency bands and scales. DWT employs two function sets: scalingand wavelet which associate with low and high pass filters orderly. Decomposition follows the manner of dividing time separability. Meanly, only half of the samples in a signal are sufficient to represent the whole signal, doubling the frequency separability [3].

Haar wavelet operates on data by calculating the sums and differences of adjacent elements. If we consider a function $f$, the Haar Wavelet Transform (HWT) is defined as:

$$f \rightarrow (a^l, d^l)$$

$$a^l = (a_1, a_2, ..., a_{N/2})$$

$$d^l = (d_1, d_2, ..., d_{N/2})$$  \hspace{1cm} (7)

where $L$ is the decomposition level, $a$ is the approximation subband, $d$ is the detail subband and $N$ is the length of time signal. Each coefficient is calculated as below:

$$a_m = \frac{f_{2m} + f_{2m-1}}{\sqrt{2}} \text{ form } 1,2, ..., N/2$$  \hspace{1cm} (8)

$$d_m = \frac{f_{2m} - f_{2m-1}}{\sqrt{2}} \text{ form } 1,2, ..., N/2$$  \hspace{1cm} (9)

To apply HWT on images, we first apply a one level Haar wavelet to each row and secondly to each column of the resulting "image" of the first operation. The resulted image is decomposed into four subbands: LL, HL, LH, and HH subbands. (L=Low, H=High). The LL-subband contains an approximation of the original image while the other subbands contain the missing details. The LL-subband output from any stage can be decomposed further. The figure below shows the result of one and two level HWT based on the pyramid decomposition [4].

![Figure 2: Pyramid decomposition using Haar wavelet filter](image)
function, it’s a probability value to define the object class. An image signature of a particular object composes the input layer and the output layer classifies the image. We illustrate this explanation with the following figure. Airplanes picture is placed as input data and after feature extraction, we get image signature. This information is compiled by FCNN and finally provides the probability of membership class.

5. Related Work

Now, you understand all subject that we will discuss for this new approach. To be very clear and short, we present the method in algorithm format.

5.1 Algorithm of images classification

Step 1: Create an image dataset from object classification and separate in two parts whose 70% for data training and 30% percent for testing. \( I_i \) denotes the image issue from step \( i \).

Step 2: Convert all images in dataset to grayscale \( I_1 \rightarrow I_2 \).

Step 3: Apply Canny filter for edge detection \( I_2 \rightarrow I_3 \).

Step 4: Apply blurring filter for \( I_3 \rightarrow I_4 \).

Step 5: \( I_4 \) is going through PCNN with sigmoidal pulse generator to obtain \( I_5 \).

Step 6: Calculate foveation of \( I_5 \)

\[
I_6 = \begin{cases} 
1 & \text{if } I_5 > 0.2 \\
0 & \text{Otherwise}
\end{cases}
\]

Step 7: Apply HWT level 1 \( I_6 \rightarrow I_7 \).

\( I_7 \) is LL described in Figure 2. (a)

Step 8: Apply HWT level 2 \( I_7 \rightarrow I_8 \).

\( I_8 \) is LL described in Figure 2. (b)

Step 9: Reshape \( I_9 \) to single vector to form image signature \( I_9 \).

Step 10: Set up FCNN composed by input layer (number of neuron same as \( I_9 \) length), six hidden layers and one output layer (number of neuron same as the number of class).

Step 11: Start deep learning by initializing all weights randomly.

Step 12: Save all weights once the desired output is reached and stop training.

Step 13: Test the trained neural network with 30% of dataset training and calculate the accuracy.

6. Experience

Caltech 101 is the image dataset that we use for application. It consists of pictures of objects belonging to 101 classes, plus one background clutter class. Each image is labelled with a single object. Each class contains roughly 40 to 800 images, totaling 9146 images.

For testing, we take one image from faces class because we cannot present here all existing image in dataset for each processing steps.

6.1 Pre PCNN processing

From original image to be classified to blurring filter step, we illustrate the operation in Table 1.

<table>
<thead>
<tr>
<th>Comments</th>
<th>Images</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>In case, the image is already in grayscale, no need to perform the conversion. This original image has triplet color (red, green, bleu), so we need only one matrix. It is the reason of conversion.</td>
<td>Figure 4: Original image</td>
<td>We need to reduce the quantity of information to accelerate the processing then we apply directly Canny filter for edge detection.</td>
</tr>
<tr>
<td>The grayscale image value is between 0 and 255. PCNN supports simple matrix. So one color channel can be processed however we need the 3 colors then we must do such conversion.</td>
<td>Figure 5: Grayscale image</td>
<td>PCNN output is a binary image same as Canny filter output. Figure 6. has a large information and we should reduce then we apply blurring filter in the aim to accept it as PCNN image input.</td>
</tr>
</tbody>
</table>

Table 1: Step 1 to Step 4.

Figure 4: Original image

Figure 5: Grayscale image

Figure 6: Operation result of Canny filter

Figure 7: Operation result of blurring filter

DOI: 10.21275/SR201126194736

1672
6.2 PCNN processing

We consider the following parameters for PCNN iteration:

Weights matrix

\[ M = W = \begin{bmatrix} \sqrt{\frac{2}{2}} & 1 & \sqrt{\frac{2}{2}} \\ 1 & 1 & 1 \\ \sqrt{\frac{2}{2}} & 1 & \sqrt{\frac{2}{2}} \end{bmatrix} \]  \hspace{1cm} (13)

Initial values of matrix

The initial values of linking \( L \), feeding \( F \) matrix and stimulus \( S \) are the same as the input image. The convolution between null matrix which has the same size as the input image \( R \times C \) and weights matrix initiates the output value \( Y \) of PCNN. The first value of dynamic threshold \( \theta \) is an \( R \)-by-\( C \) matrix of two.

Delay constants

\[ \alpha_F = 0.1, \quad \alpha_L = 1 \quad \text{and} \quad \alpha_\theta = 1 \]  \hspace{1cm} (14)

Normalizing constants

\[ V_F = 0.5, \quad V_L = 0.2, \quad V_\theta = 20, \quad \beta = 0.1 \quad \text{and} \quad Y = 0.9 \]  \hspace{1cm} (15)

At the end of iteration, we have the image in Figure 8. with output grayscale level between 0 and 1 because we used PCNN with sigmoidal pulse generator.

6.3 Post PCNN processing

PCNN ensure the extraction of essential information. Now we need foveation operation to collect the data sensitive by human eyes. In this case, we apply the equation (12) and the Figure 9 shows the result.

Currently, image contents only the important information but we should find a solution to reduce the size. So, we apply HWT in 2 levels as shown in Figure 10 and 11.

6.4 Image classification

CNN input is the reshape of LL level 2 HWT into vector that we called image signature.

Training

70% of image in Caltech 101 dataset will be processed same as above face image processing. At least, six hidden layers are required and for our testing, we fix it to six. The activation function for them is sigmoid function and all weights are initialized randomly. Concerning, output layer, we have 101 neurons which present each class. The presentation is in Table 2 below.

Table 2: Output format

<table>
<thead>
<tr>
<th>accordeon</th>
<th>airplanes</th>
<th>anchor</th>
<th>…</th>
<th>windsor_chair</th>
<th>wrench</th>
<th>Yin_Yang</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

We train our network and after training, we get the fix weights.

Testing

30% (2726 images) of remaining dataset will pass through of our network for testing purpose. It means that 30% for each class we test. The global accuracy is 97.5%. We illustrate in Table 3.
7. Conclusion

Feature extraction layer is a weakness of CNN. It caused the processing delay, so the introduction of foveation is a good solution because it works like human vision and concentrates in important information in image only. The dataset of these information constitutes the image signature that we place as input of CNN. If you search the accuracy of actual favorite CNNs and compare with this new approach, 97.5% can be qualified as best performance using Caltech 101 dataset.

PCNN remains an unmissable step for image data mining and foveation is one application of this smart neural network. Apart searching image in database, we can apply this algorithm for example in face recognition, finger print recognition, etc. The improvement axe of this research can be oriented to use PCNN light version like intersecting cortical model (ICM) or modified pulse coupled neural networks (MPCNN).

References


Volume 9 Issue 11, November 2020

www.ijsr.net

Licensed Under Creative Commons Attribution CC BY

Paper ID: SR201126194736
DOI: 10.21275/SR201126194736

1674


Author Profile

**Rafidison Maminiaina Alphonse**, he received his telecommunication engineer degree in 2007, M.Sc. in 2011 and the Ph.D. degree in sciences engineering from Antananarivo university, in 2014. Apart of technical degree, he has a management degree from Normandy Management Institute France and CFD Belgium. Currently, he works as OSS/BSS for a telecommunication company. In parallel, he is continuing his research after Ph.D presentation. Until now, his focus is on image processing using neural network but it will be expanded to image steganography and steganalysis. Of course, signal processing will be part of next paper publication.

**Ramafiarisona Hajasoa Malalatiana**, she is lecturer at Antananarivo Madagascar university after receiving her telecommunication engineer degree, M.Sc. and the Ph.D. Recently, she gets HDR degree (Ability to Supervise Research) and continue her research in image and signal processing domain.