International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2019): 7.583

Images Classification by Pulse Coupled Neural Networks

Rafidison Maminiaina Alphonse[1], Ramafiarisona Hajasoa Malalatiana[2]

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

[2]Telecommunication-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 U_j 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 α. 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 modelis given by

$$F_{ij}[n] = \exp(-\alpha_F \delta_n) F_{ij}[n-1] + S_{ij} + V_F \sum_{kl} M_{ijkl} Y_{kl}[n-1] \quad (1)$$

$$L_{ij}[n] = \exp(-\alpha_L \delta_n) L_{ij}[n-1] + V_L \sum_{kl} W_{ijkl} Y_{kl}[n-1] \quad (2)$$

where M and W are the constant synaptic weights, and S is the external stimulus. V_F and V_L are normalizing constants. α_F and α_L are the time constants, $\alpha_F < \alpha_L$. The linking modulation (the internal state) is given by

$$U_{ij}[n] = F_{ij}[n](1 + \beta . L_{ij}[n]) \quad (3)$$

where β is the linking weight parameter. The pulse generator is the step function as

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

$$\Theta_{ij}[n] = \Theta_{ij}[n-1]\exp(-\alpha_\Theta) + V_\Theta Y_{ij}[n-1] \quad (5)$$

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

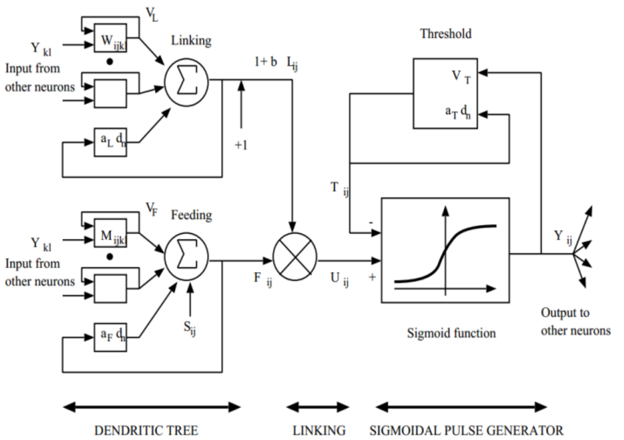


Figure 1: The outline of PCNN with the sigmoidal pulse generator

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

$$Y_{ij}[n] = \frac{1}{1+exp[-\Upsilon(U_{ij}[n]-\Theta_{ij}[n-1])]} \quad (6)$$

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

Wavelet 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:

$$\begin{aligned} & f \to (a^L | d^L) \\ & a^L = (a_1, a_2, \dots, a_{N/2}) \\ & d^L = (d_1, d_2, \dots, d_{N/2}) \end{aligned} \quad (7)$$

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:

$$a_m = \frac{f_{2m} + f_{2m-1}}{\sqrt{2}} form = 1,2, \dots, N/2 \quad (8)$$

$$d_m = \frac{f_{2m} - f_{2m-1}}{\sqrt{2}} form = 1,2, \dots, N/2 \quad (9)$$

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].

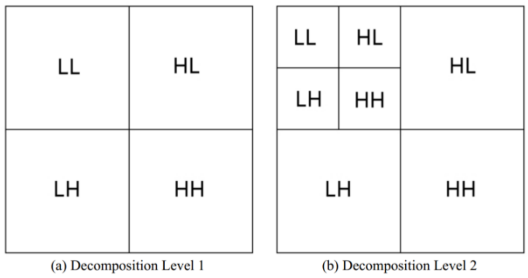


Figure 2: Pyramid decomposition using Haar wavelet filter

4. Fully Connected Neural Network

FCNN is composed by input layer, hidden layer(s) and output layer. The number of hidden layers depends on the project purpose. Each neuron must be connected to all neuron in next layer. A calculation between input, weight and bias should be done before going through activation function.

The activation function in which, we interest is the sigmoid and softmax function described in equation (10) and (11).

$$f_{sigmoide}(x_i) = \frac{1}{1+e^{-x_i}} \quad (10)$$

$$f_{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}} \quad (11)$$

where x_i is the sum inputs multiplied by weights plus bias and N the number of neuron in output layer.The value of sigmoid function is between 0 and 1 and for softmax

International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2019): 7.583

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.

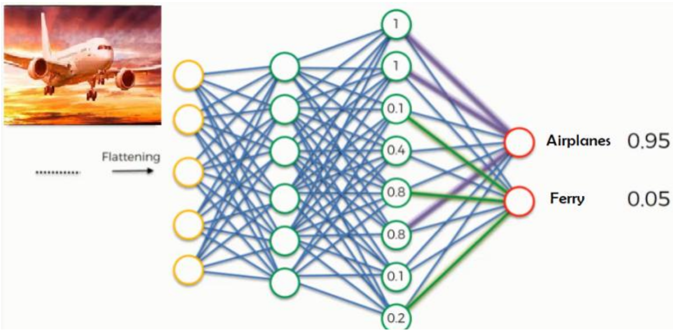


Figure 3: Example of CNN

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 $I_1 \to I_2$

Step 3: Apply Canny filter for edge detection $I_2 \to I_3$

Step 4: Apply blurring filter for $I_3 \to I_4$

Step 5: I_4 is going through PCNN with sigmoidal pulse generator to obtain I_5

Step 6: Calculate foveation of I_5

$$I_6 \begin{cases} 1 \; if \;\; I_5 > 0.2 \\ 0 \; Otherwise \end{cases} \tag{12}$$

Step 7: Apply HWT level 1 $I_6 \to I_7$

I_7 is LL described in Figure 2. (a)

Step 8: Apply HWT level 2 $I_7 \to I_8$

I_8 is LL described in Figure 2. (b)

Step 9: Reshape I_8 to single vector to form image signature I_9

Step 10: Set up FCNN composed by input layer (number of neuron same as I_9length), 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

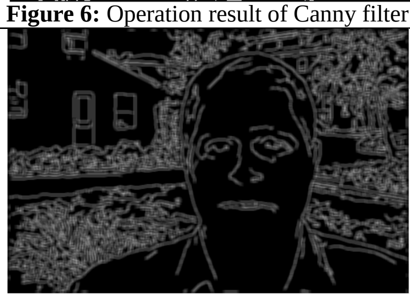
Caltech 101 is the image dataset that we use for application. It consists of pictures of objects belonging to 101 classes, plus one background clutter class. Each image is labelled with a single object. Each class contains roughly 40 to 800 images, totaling 9146 images.

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 1: Step 1 to Step 4

Comments	Images	Comments	Images
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.	 **Figure 4:** Original image	We need to reduce the quantity of information to accelerate the processing then we apply directly Canny filter for edge detection.	 **Figure 6:** Operation result of Canny filter
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.	 **Figure 5:** Grayscale image	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.	 **Figure 7:** Operation result of blurring filter

6.2 PCNN processing

We consider the following parameters for PCNN iteration:

Weights matrix

$$M = W = \begin{bmatrix} \sqrt{2}/2 & 1 & \sqrt{2}/2 \\ 1 & 1 & 1 \\ \sqrt{2}/2 & 1 & \sqrt{2}/2 \end{bmatrix} \qquad (13)$$

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 RxC and weights matrix initiates the output value Y of PCNN. The first value of dynamic thresholdΘ is an R-by-C matrix of two.

Delay constants

$$\alpha_F = 0.1, \alpha_L = 1 \text{ and } \alpha_\theta = 1 \qquad (14)$$

Normalizing constants

$$V_F = 0.5, V_L = 0.2, V_\Theta = 20, \beta = 0.1 \text{ and } \Upsilon = 0.9 \qquad (15)$$

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.

Figure 8: PCNN image output

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.

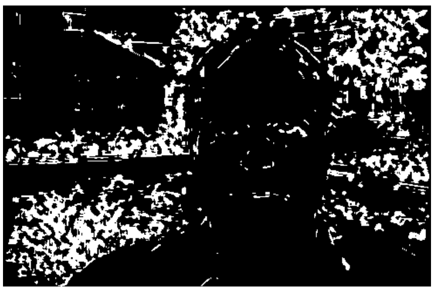


Figure 9: Foveation image

Figure 10: LL of Level 1 HWT

Figure 11: LL of Level 2 HWT

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

accordion	airplanes	anchor	…	windsor_chair	wrench	Yin_Yang
1	0	0	0	0	0	0
0	1	0	0	0	0	0
0	0	1	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	0	0	0	1	0	0
0	0	0	0	0	1	0
0	0	0	0	0	0	1

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.

International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2019): 7.583

Table 3: Output format

Class	100% dataset	70% dataset for training	30% dataset for testing	Real result	Accuracy
accordion	55	39	16	16	100.0000
airplanes	800	560	240	231	96.2500
anchor	42	29	13	13	100.0000
ant	42	29	13	13	100.0000
BACKGROUND_Google	468	328	140	136	97.1429
barrel	47	33	14	14	100.0000
bass	54	38	16	16	100.0000
beaver	46	32	14	14	100.0000
binocular	33	23	10	10	100.0000
bonsai	128	90	38	38	100.0000
brain	98	69	29	28	96.5517
brontosaurus	43	30	13	12	92.3077
buddha	85	60	25	23	92.0000
butterfly	91	64	27	25	92.5926
camera	50	35	15	15	100.0000
cannon	43	30	13	13	100.0000
car_side	123	86	37	36	97.2973
ceiling_fan	47	33	14	14	100.0000
cellphone	59	41	18	18	100.0000
chair	62	43	19	18	94.7368
chandelier	107	75	32	31	96.8750
cougar_body	47	33	14	14	100.0000
cougar_face	69	48	21	21	100.0000
crab	73	51	22	22	100.0000
crayfish	70	49	21	21	100.0000
crocodile	50	35	15	15	100.0000
crocodile_head	51	36	15	15	100.0000
cup	57	40	17	16	94.1176
dalmatian	67	47	20	19	95.0000
dollar_bill	52	36	16	16	100.0000
dolphin	65	46	19	19	100.0000
dragonfly	68	48	20	19	95.0000
electric_guitar	75	53	22	22	100.0000
elephant	64	45	19	18	94.7368
emu	53	37	16	16	100.0000
euphonium	64	45	19	19	100.0000
ewer	85	60	25	25	100.0000
Faces	435	305	130	128	98.4615
Faces_easy	435	305	130	130	100.0000
ferry	67	47	20	20	100.0000
flamingo	67	47	20	20	100.0000
flamingo_head	45	32	13	13	100.0000
garfield	34	24	10	10	100.0000
gerenuk	34	24	10	10	100.0000
gramophone	51	36	15	15	100.0000
grand_piano	99	69	30	30	100.0000
hawksbill	100	70	30	28	93.3333
headphone	42	29	13	13	100.0000
hedgehog	54	38	16	16	100.0000
helicopter	88	62	26	24	92.3077
ibis	80	56	24	24	100.0000
inline_skate	31	22	9	9	100.0000
joshua_tree	64	45	19	19	100.0000
kangaroo	86	60	26	25	96.1538
ketch	114	80	34	33	97.0588
lamp	61	43	18	18	100.0000
laptop	81	57	24	24	100.0000
Leopards	200	140	60	56	93.3333
llama	78	55	23	22	95.6522
lobster	41	29	12	12	100.0000
lotus	66	46	20	20	100.0000
mandolin	43	30	13	13	100.0000
mayfly	40	28	12	12	100.0000
menorah	87	61	26	24	92.3077
metronome	32	22	10	10	100.0000
minaret	76	53	23	21	91.3043
Motorbikes	768	538	230	225	97.8261
nautilus	55	39	16	16	100.0000
octopus	35	25	10	10	100.0000
okapi	39	27	12	12	100.0000
pagoda	47	33	14	14	100.0000
panda	38	27	11	11	100.0000
pigeon	45	32	13	13	100.0000
pizza	53	37	16	15	93.7500
platypus	34	24	10	10	100.0000
pyramid	57	40	17	17	100.0000
revolver	82	57	25	24	96.0000
rhino	59	41	18	15	83.3333
rooster	49	34	15	15	100.0000
saxophone	40	28	12	12	100.0000
schooner	63	44	19	16	84.2105
scissors	39	27	12	12	100.0000
scorpion	84	59	25	25	100.0000
sea_horse	57	40	17	17	100.0000
snoopy	35	25	10	9	90.0000
soccer_ball	64	45	19	19	100.0000
stapler	45	32	13	13	100.0000
starfish	86	60	26	26	100.0000
stegosaurus	59	41	18	18	100.0000
stop_sign	64	45	19	19	100.0000
strawberry	35	25	10	10	100.0000
sunflower	85	60	25	25	100.0000
tick	49	34	15	15	100.0000
trilobite	86	60	26	23	88.4615
umbrella	75	53	22	21	95.4545
watch	239	167	72	70	97.2222
water_lilly	37	26	11	11	100.0000
wheelchair	59	41	18	16	88.8889
wild_cat	34	24	10	10	100.0000
windsor_chair	56	39	17	16	94.1176
wrench	39	27	12	11	91.6667
Yin_Yang	60	42	18	17	94.4444
TOTAL	9115	6389	2726	2658	97.5055

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

[1] T. Lindblad, J.M. Kinser, “Image Processing Using Pulse-Coupled Neural Networks,” 2nd ed., Springer Berlin Heidelberg New York, 2005, pp. 11–23.

[2] M. Tanaka, T. Watanabe, Y. Baba, T. Kurita and T. Mishima, "Autonomous foveating system and integration of the foveated images," IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.99CH37028), Tokyo, Japan, 1999, pp. 559-564 vol.1, doi: 10.1109/ICSMC.1999.814153.

[3] E. H. Houssein, M. A. S. Ali and A. E. Hassanien, "An image steganography algorithm using Haar Discrete Wavelet Transform with Advanced Encryption

System," 2016 Federated Conference on Computer Science and Information Systems (FedCSIS), Gdansk, 2016, pp. 641-644.
[4] W.N. Ibrahem,"Wavelets and Multiresolution Processing," Image processing, Lecture 8, pp.1–12.

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.

Paper ID: SR201126194736 DOI: 10.21275/SR201126194736