MODIFIED CONVOLUTIONAL NEURAL NETWORK FOR ARIARY BANKNOTES AUTHENTICATION

RAFIDISON MAMINIAINA ALPHONSE
Telecommunication-Automatic-Signal-Image-Research Laboratory/Doctoral School in Science and Technology of Engineering and Innovation/University of Antananarivo Antananarivo 101, Madagascar
mamynyaina@gmail.com

RAMAFIARISONA HAJASOA MALALATIANA
Telecommunication-Automatic-Signal-Image-Research Laboratory/Doctoral School in Science and Technology of Engineering and Innovation/University of Antananarivo Antananarivo 101, Madagascar
mhramafiarisona@yahoo.fr

ABSTRACT
This paper presents a new algorithm named modified convolutional neural network (MCNN) to authenticate current existing banknotes in Madagascar. At the end of processing, we are able to identify the money face value and fake notes according to the input. Convolutional neural network (CNN) is the promising technical to solve such problem although feature extraction layer takes time due of convolution and spooling operations. The solution that we propose here is to swap this layer with pulse coupled neural network (PCNN) which has a capability to collect easily input image characteristics before going through fully connected layer. So, we have three images as input because one banknote has recto, verso and watermarked faces which are converted to gray. Our pulse couple neural network extracts all features to form a single vector called image signature. This vector is presented to full connected neural network constituted by few hidden layers, one output layer using softmax function as neural activation function. It means that we have authentic' s probability.

INTRODUCTION
Ariary (ISO 4217 code MGA) is the currency of Madagascar. Malagasy government decided to enhance banknotes security in 2017 by changing 100, 200, 500, 1000, 2000, 5000, 10000 notes and created a new one 20000 Ariary. The system that we implement here will recognize money face value and judge if fake note is placed in entrance. To archive this goal, we capture banknote's image in three positions: recto, verso, watermarked. They will be treated by pulse coupled neural network (PCNN) for features extraction then fully connected neural network takes over. In the next paragraph, we will talk about PCNN followed by convolutional neural network (CNN) then modified convolution neural network (MCNN) and terminate with results performance discussion and application.

PULSE COUPLED NEURAL NETWORK
A PCNN neuron shown in Figure 1 contains two main compartments: The Feeding and Linking compartments. Each of these communicates with neighbouring neurons through the synaptic weights M and W respectively. Each retains its previous state but with a decay factor. Only the Feeding compartment receives the input stimulus, S. The values of these two compartments are determined by,

$$F_{ij}(n) = S_{ij} + F_{ij}(n-1).\exp(-\alpha_F) + V_F.(M * Y(n-1))_{ij} \quad (1)$$

$$L_{ij}(n) = L_{ij}(n-1).\exp(-\alpha_L) + V_L.(W * Y(n-1))_{ij} \quad (2)$$

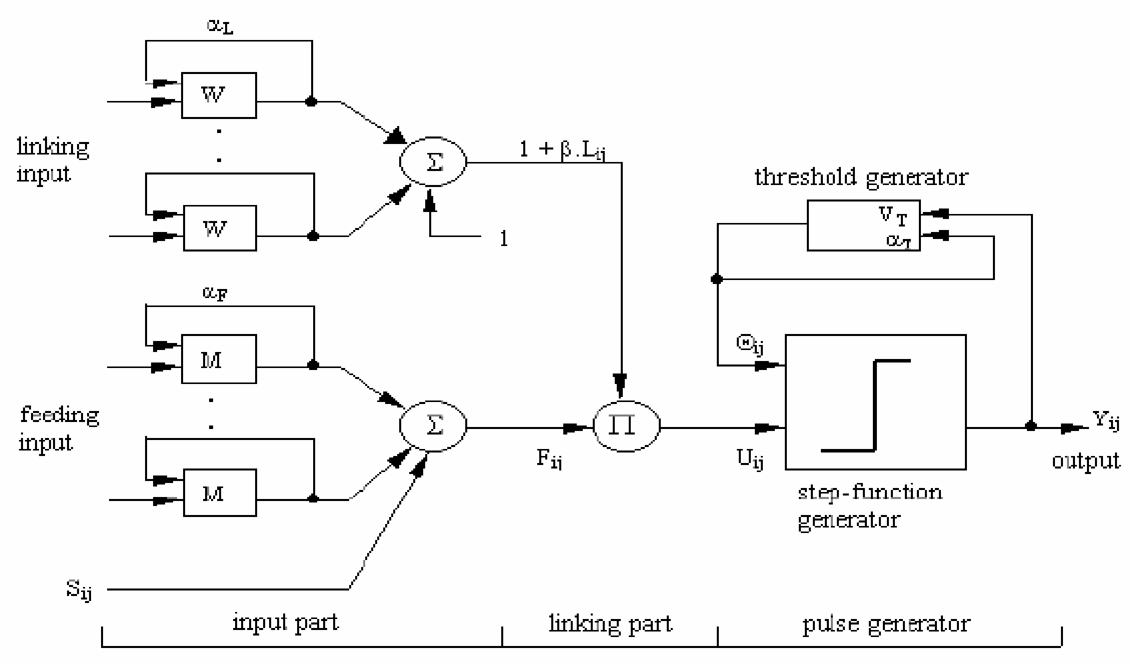

Figure 1: Schematic representation of a PCNN processing element

Where F_{ij} is the Feeding compartment of the (i, j) neuron embedded in a 2D array of neurons, and L_{ij} is the corresponding Linking compartment. Y_{ij}'s are the outputs of neurons from a previous iteration $[n - 1]$. Both compartments have a memory of the previous state, which decays in time by the exponent term. The constants V_F and V_L are normalizing constants. If the receptive fields of M and W change then these constants are used to scale the resultant correlation to prevent saturation [1].
The state of these two compartments are combined in a second order fashion to create the internal state of the neuron, U. The combination is controlled by the linking strength, β. The internal activity is calculated by,

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

The internal state of the neuron is compared to a dynamic threshold, Θ, to produce the output, Y, by

$$Y_{ij}(n) = \begin{cases} 1, & \text{if } U_{ij}(n) > \Theta_{ij}(n) \\ 0, & \text{else} \end{cases} \tag{4}$$

The threshold is dynamic in that when the neuron fires ($Y > \Theta$) the threshold then significantly increases its value. This value then decays until the neuron fires again. This process is described by,

$$\Theta_{ij}(n) = \Theta_{ij}(n-1).\exp(-\alpha_\Theta) + V_\Theta.Y_{ij}(n) \tag{5}$$

Where V_Θ is a large constant that is generally more than an order of magnitude greater than the average value of U [2][3].

CONVOLUTIONAL NEURAL NETWORK

CNNs are feedforward networks composed by:

- Convolution layers which extract the feature of input image
- Pooling layers which reduce the information quantity
- Fully connected layer which classifies the original input using the image signature provided by the previous layer [5].

For more clarification, Figure 2 illustrate CNN structure

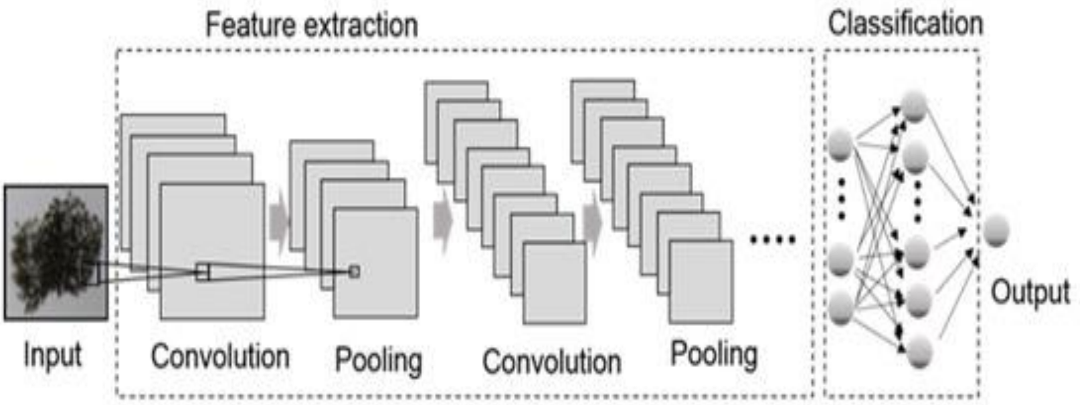

Figure 2: CNN structure

MODIFIED CONVOLUTIONAL NEURAL NETWORK

In this paragraph, we will focus on the new method and the explanation of each module. Three images of single note pass via filter noise then convert them to gray. PCNN extracts feature by iteration. First part of iteration is dedicated for segmentation and the second one for edge detection. Once we have a good quality result for each part, we proceed to calculate entropy, energy entropy, logarithm and energy logarithm. The result of each calculation will be transferred to a vector (image signature) which is the input layer of fully connected neural network. This neural network output provides the result after training.

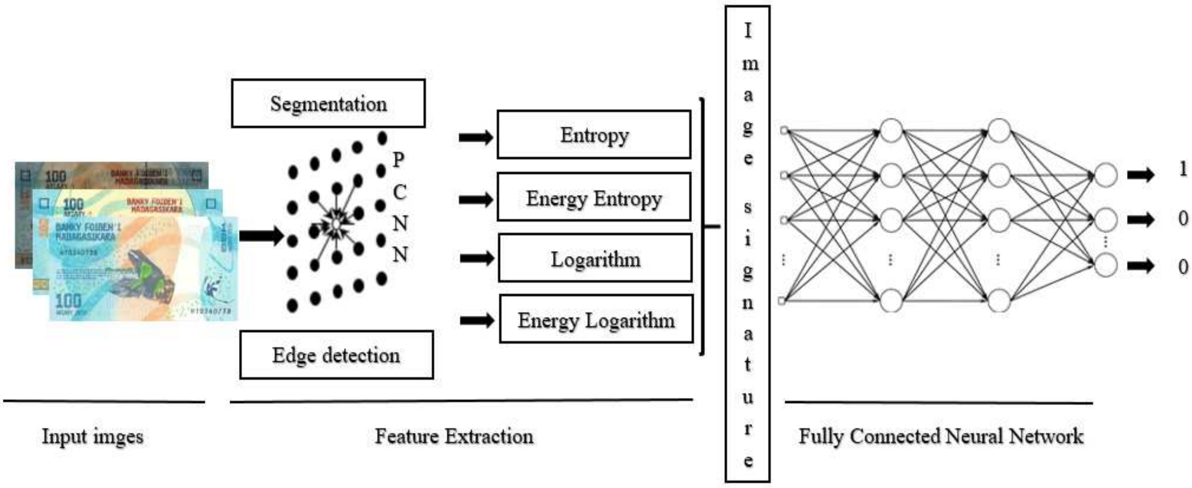

Figure 3: New approach architecture

PCNN MODULE

Before starting iteration, we initiate all variable. Once segmentation performance is reached, we save three images and continue with edge detection. We consider the following parameters:

Weights matrix

$$M = W = \begin{bmatrix} \sqrt{2}/_2 & 1 & \sqrt{2}/_2 \\ 1 & 1 & 1 \\ \sqrt{2}/_2 & 1 & \sqrt{2}/_2 \end{bmatrix} \qquad (6)$$

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 threshold Θ is an R-by-C matrix of two [4].

Delay constants

$\alpha_F = 0.1$, $\alpha_L = 1$ and $\alpha_\theta = 1$ (7)

Normalizing constants

$V_F = 0.5$, $V_L = 0.2$, $V_\Theta = 20$ and $\beta = 0.1$ (8)

Calculating the misclassified pixel rate is the way to fix the iteration for segmentation and edge detection. First low value of this parameter corresponds to segmentation and the second one to edge detection.

IMAGE SIGNATURE

Image signature is a vector containing the characteristics of given image. In our case, we will use some mathematics approach to define it. The aim is to build one vector to identify an image.

Each image has its own signature: first to fourth row correspond to entropy, entropy energy, logarithm and logarithm energy for segmented image. Sixth to eighth dedicated for edge detection. So we have a matrix 8x3. We transform this last matrix to column matrix G by moving the two columns to the first one.

We describe below the way how to calculate these four parameters:

Entropy:

Entropy is a kind of representation of the image statistical feature, which reflects the amount of information contained in the image. Similarly, the entropy of output binary images is a one-dimension entropy series, as shown in equation (9), where E is the information entropy of binary image; P_1 is the probability of 1's in a binary image, and P_0 is that of 0's in the binary image.

$$E = -P_1 log_2(P_1) - P_0 log_2(P_0) \tag{9}$$

Another parameters:

Some derived feature extraction methods based on entropy are presented, including energy entropy (EE), logarithm (L) and energy logarithm (EL) as shown in equation (10) – (12) [3].

$$EE = -P_1^2 log_2 P_1^2 - P_0^2 log_2 P_0^2 \tag{10}$$

$$L = -log_2 P_1 - log_2 P_0 \tag{11}$$

$$EL = -log_2 P_1^2 - log_2 P_0^2 \tag{12}$$

Single vector:

As explained earlier, a single vector S presents a banknote so to understand clearly, the equations (13), (14) show the evidence:

$$S_{n\in\{7,10\}} = \begin{pmatrix} E_n \\ EE_n \\ L_n \\ EL_n \end{pmatrix} \tag{13}$$

Where S_n is the image signature of iteration n and $S_n r$, $S_n v$, $S_n w$ are recto, verso and watermarked signature

$$S = \begin{pmatrix} S_n r \\ S_n v \\ S_n w \end{pmatrix} \tag{14}$$

Single vector S size is 24x1

Fully connected neural network:

We have 24 neurons constitute input layer. For our case, three hidden layers are enough with 8 final output which presents the face value of banknotes in ascending order. The function activation that we use between hidden layer is sigmoid function (15) and softmax (16) for output layer. The initial weight is taken randomly with size 12x24, 12x12, 12x12, 8x12 for w_1, w_2, w_3, w_4 and after deep learning, we get the correct weight to test our new system. We will see in the next paragraph the results of our algorithm.

$$f_{sigmoid}(x_i) = \frac{1}{1+e^{-x_i}} \qquad (15)$$

$$f_{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{8} e^{x_j}} \qquad (16)$$

x_i is weighted sum (sum inputs multiplied by weights)

RESULTS AND DISCUSSION

Ariary Banknotes

This document doesn't show in image all existing banknotes in Madagascar except Ariary 100, 20000 shown in Figure 4 for reference only. They are available in internet in case a reader of this paper needs.

Figure 4: Ariary 100 and 20000 notes recto, verso, watermarked

Dataset

Different type of real notes photos was taken to form our data set: fresh, old and dirty notes in total 240 whose 30 per face value. 50% for each class are used for neural network training and 50% for testing. As part of our objective is to detect fake notes, so after getting the correct weight, we place in entrance some printed scan photo notes.

Data extraction

PCNN ensure this role of extraction by segmentation and edge detection Figure 5 and 6. Low misclassified pixel rates are obtained when n=7 and 10. We have a binary image and the application of equations (9) to (12) is not complicated to constitute the image signature.

n=7

n=10

Figure 5: Ariary 100 note recto, verso, watermarked treated by PCNN

n=7

n=10

Figure 6: Ariary 2000 note recto, verso, watermarked treated by PCNN

We don't show here all image signature of training dataset; one sample is enough presented in Table 1 for dirty notes.

Table 1. Sample MCNN input

Notes / Input	Ariary 100	Ariary 200	Ariary 500	Ariary 1000	Ariary 2000	Ariary 5000	Ariary 10000	Ariary 20000
X_1	0.4515	0.4313	0.5266	0.4461	0.5430	0.6699	0.3650	0.4348
X_2	0.2958	0.2768	0.3708	0.2907	0.3881	0.5329	0.2179	0.2801
X_3	3.5459	3.6325	3.2534	3.5686	3.1947	2.7895	3.9459	3.6171
X_4	7.0918	7.2649	6.5069	7.1373	6.3894	5.5791	7.8918	7.2341
X_5	0.4349	0.4187	0.5170	0.4230	0.5335	0.6628	0.3640	0.4216
X_6	0.2802	0.2652	0.3609	0.2691	0.3780	0.5242	0.2170	0.2678
X_7	3.6167	3.6886	3.2883	3.6693	3.2286	2.8103	3.9513	3.6756
X_8	7.2335	7.3772	6.5766	7.3386	6.4572	5.6206	7.9026	7.3512

Notes / Input	Ariary 100	Ariary 200	Ariary 500	Ariary 1000	Ariary 2000	Ariary 5000	Ariary 10000	Ariary 20000
X_9	0.3958	0.3782	0.3650	0.3133	0.3109	0.5081	0.5637	0.5688
X_{10}	0.2446	0.2292	0.2180	0.1758	0.1739	0.3517	0.4104	0.4160
X_{11}	3.7943	3.8795	3.9457	4.2303	4.2445	3.3216	3.1230	3.1057
X_{12}	7.5886	7.7589	7.8914	8.4606	8.4890	6.6432	6.2461	6.2114
X_{13}	0.3909	0.3763	0.3596	0.3114	0.2982	0.4983	0.5513	0.5273
X_{14}	0.2402	0.2275	0.2133	0.1743	0.1641	0.3417	0.3970	0.3716
X_{15}	3.8178	3.8891	3.9740	4.2411	4.3214	3.3588	3.1655	3.2507
X_{16}	7.6356	7.7782	7.9480	8.4822	8.6429	6.7177	6.3310	6.5014

Notes / Input	Ariary 100	Ariary 200	Ariary 500	Ariary 1000	Ariary 2000	Ariary 5000	Ariary 10000	Ariary 20000
X_{17}	0.9340	0.9221	0.9081	0.9608	0.6955	0.4182	0.9869	0.9528
X_{18}	0.8961	0.8779	0.8567	0.9377	0.5645	0.2647	0.9789	0.9251
X_{19}	2.1363	2.1617	2.1922	2.0798	2.7165	3.6908	2.0264	2.0966
X_{20}	4.2725	4.3235	4.3844	4.1595	5.4331	7.3816	4.0528	4.1931
X_{21}	0.9641	0.9455	0.9685	0.9943	0.9238	0.8440	0.9509	0.9808
X_{22}	0.9429	0.9138	0.9498	0.9909	0.8805	0.7630	0.9222	0.9693
X_{23}	2.0729	2.1118	2.0639	2.0114	2.1581	2.3370	2.1006	2.0387
X_{24}	4.1459	4.2237	4.1278	4.0228	4.3161	4.6739	4.2011	4.0774

After signature calculation, we have the following two graphs in Figure 7 which presents E, EE, L, LE of watermark Ariary 100 and 20000 notes. As we see, the trend becomes the same from seven to ten iteration. It means that a good quality of segmentation is reached when n=7 and edge detection for n=10.

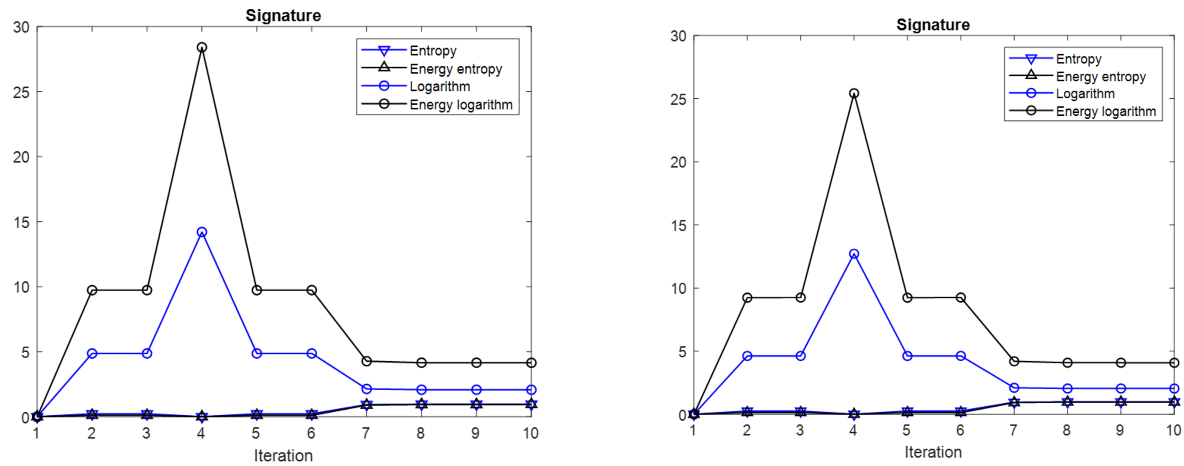

Figure 7: Watermarked Ariary 100 note signature/ Watermarked Ariary 2000 note signature

System output

For network training, we fix the output detailed in Table 2. During testing, the value is not necessarily 1, it may be approximatively value due of softmax function as activation function.

Table 2. Ideal output

Input / Output	Ariary 100	Ariary 200	Ariary 500	Ariary 1000	Ariary 2000	Ariary 5000	Ariary 10000	Ariary 20000
Y_1	1	0	0	0	0	0	0	0
Y_2	0	1	0	0	0	0	0	0
Y_3	0	0	1	0	0	0	0	0
Y_4	0	0	0	1	0	0	0	0
Y_5	0	0	0	0	1	0	0	0
Y_6	0	0	0	0	0	1	0	0
Y_7	0	0	0	0	0	0	1	0
Y_8	0	0	0	0	0	0	0	1

Accuracy

120 images are available for testing including 15 images per class. Table 3 shows the result. Once the output is not equal 1, we classify it as wrong. The note is not obligatorily fake but it is our own way to calculate the accuracy. With this principle, we reach 96.6% of accuracy.

Table 3. Accuracy

Class	Training dataset	Testing dataset	Real result	Accuracy
Ariary 100	15	15	15	100.0000
Ariary 200	15	15	14	93.3333
Ariary 500	15	15	14	93.3333
Ariary 1000	15	15	15	100.0000
Ariary 2000	15	15	14	93.3333
Ariary 5000	15	15	14	93.3333
Ariary 10000	15	15	15	100.0000
Ariary 20000	15	15	15	100.0000
Total	120	120	116	96.6667

Real notes vs fake notes

In each class, we take one sample of real dirty note and we present the outcome as below in Table 4.

Table 4. Sample MCNN output

Input / Output	Ariary 100	Ariary 200	Ariary 500	Ariary 1000	Ariary 2000	Ariary 5000	Ariary 10000	Ariary 20000
Y_1	1.0000	0.0047	0.0000	0.0000	0.0000	0.0021	0.0000	0.0000
Y_2	0.0000	0.9952	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000
Y_3	0.0000	0.0000	0.9990	0.0000	0.0008	0.0000	0.0000	0.0000
Y_4	0.0000	0.0000	0.0000	1.0000	0.0029	0.0000	0.0000	0.0000
Y_5	0.0000	0.0000	0.0000	0.0000	0.9963	0.0000	0.0000	0.0000
Y_6	0.0000	0.0000	0.0000	0.0000	0.0000	0.9971	0.0000	0.0000
Y_7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0008	1.0000	0.0000
Y_8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000

Table 5 had the result of fake notes. We remark that we don't have a value more than 45% for each output.

Table 5. Fake notes detection

Input / Output	Ariary 100	Ariary 200	Ariary 500	Ariary 1000	Ariary 2000	Ariary 5000	Ariary 10000	Ariary 20000
Y_1	0.3191	0.3151	0.3108	0.0000	0.0000	0.2112	0.0000	0.0000
Y_2	0.2005	0.3238	0.3917	0.0000	0.0000	0.1002	0.0000	0.0000
Y_3	0.1687	0,3611	0.2975	0.0000	0.2353	0.0501	0.0000	0.0000
Y_4	0.0001	0.0000	0.0000	0.3869	0.394	0.0000	0.0000	0.0000
Y_5	0.0000	0.0000	0.0000	0.4101	0.3707	0.0000	0.0000	0.0000
Y_6	0.3116	0.0000	0.0000	0.203	0.0000	0.2894	0.249	0.2438
Y_7	0.0000	0.0000	0.0000	0.0000	0.0000	0.3491	0.3689	0.3701
Y_8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.3821	0.3861

We can consider if maximum output is less than accuracy, we can judge directly that a fake note is placed in input.

MOTIVATION AND APPLICATIONS

Since 20000 notes are in circulation, the number of fake banknotes detected in market increase because it is the higher new face value existing in Madagascar. The motivation is coming from this sad situation and the aim is to differentiate a real and fake banknote used by Malagasy people in daily life basis. We know that existing CNN can do the job such as LeNet, AlexNet, VGG, GoogLENet, ResNet and more …. However, the implementation is so heavy then we should search a new approach to create a light version by introducing PCNN neural network which is an efficient tool to manage image feature extraction.

Mobile money becomes part of Malagasy people necessity and malicious person profits to produce fake banknotes to transform in e-money for money laundering, so our new approach may help mobile money cash point, bank, etc. We provide here the method/algorithm and each entity can develop his own application using the same then integrate it with machine banknotes authentication. For us, the programming was done with Matlab and the input collection was performed by scan.

CONCLUSION

Until now, CNN is classified as best tool to recognize or classify an image. However, the disadvantage is waste time for extracting image feature with convolution and pooling operations layers. PCNN is strong neural network inspired of human vision to collect a pertinent information present in image. So, we replace feature extraction layer in standard CNN by PCNN. Apart of this change, we invent the implication of entropy and logarithm in aim of forming image signature. We are able to classify Ariary note per face value and judge if a banknote is fake or not. May be you ask this question: "What's happen if Central Bank of Madagascar change the current banknotes?". The answer is very simple; the algorithm is still valid however you should extract the characteristics of the new money using PCNN then train the fully connected neural network and you will get a new weight for usual authentication.

This method called MCNN has a good performance with 96.6% of accuracy. It cannot be limit only on Ariary authentication but can be used in different domain such as image classification, fingerprint recognition, handwriting recognition, steganalysis, medical imaging, etc.

REFERENCES

1) T. Lindblad, J.M. Kinser. Image Processing Using Pulse-Coupled Neural Networks. 2nd ed., Springer Berlin Heidelberg New York, 2005, pp. 11–16.
2) T. Lindblad, J.M. Kinser. Image Processing Using Pulse-Coupled Neural Networks. 2nd ed., Springer Berlin Heidelberg New York, 2005, pp. 99–106.
3) Y. Ma, K. Zhan, Z. Wang. Application of Pulse-Coupled Neural Networks. Higher Education Press, Beijing, May 2010, pp. 111–114.
4) M.A. Rafidison. Eyes Detection by Pulse Coupled Neural Networks. IJCSN, vol. 2, no. 5, pp. 11-28, Oct. 2013.
5) W. Rawat, Z. Wang. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. Neural Comput. 2017 Sep;29(9):2352-2449, doi: 10.1162/NECO_a_00990, Epub 2017 Jun 9. PMID: 28599112.