American Journal of Intelligent Systems 2014, 4(2): 21-31
DOI: 10.5923/j.ajis.20140402.01

Digital Watermarking and Signing by Artificial Neural Networks

Hajji Tarik[1,*], Jaara El Miloud[2]

Laboratoire de recherche en informatiques LaRI, Faculté des sciences FSO, Oujda, 6000, Maroc (LaRI/FSO/UMP1)

Abstract The objective of this work is the description of a new approach based on artificial neural networks to signing the digital images using the principle of watermarking. The idea of this approach is to watermark the image to be protected by the compressed image previously generated by the artificial neural network. This approach provided, on the one hand, a very efficient approach to digital signature for protecting the integrity of the image and by the other hand; it provides an easy and immediate mechanism for verification. The role of artifices neural networks in this approach is to generate the compressed image that will be used in the sequel to the watermarking of the image by using an algorithm for optimal positioning and that does not change the visual appearance of the image and also to verify the signature. This work also includes a comparative study to select the structure and parameters of the artificial neural network are performing for the problem of the compression and decompression of digital images. Such as, it contains a description of an application that allows for the generation of an artificial neural network with a simple Meta model description.

Keywords Digital Watermarking, Digital Signature, Digital image, Artificial Neural Networks, Meta model, IT security

1. Introduction

1.1. Digital Signature

The digital signature is an operation that ensures all the documents signed. In other words, the mechanism protects the document against falsification and the operations to change the documents at the time of transmission of this document. We can distinguish between two types of signatures:

1. The first type is exhibited in the RSA cryptographic scheme [1]. In this method, the signature of a document is another document comparable with the document signed (in this case, you must send two documents: the signed document and the signature of the document). The size of the document signature is a function of the size of the signed document. As we have a characteristic property of such signature is that only subjects can verify the signature because it is them that they have the verification keys.

2. The second type of signature is the classical notion of signature, when we have a public signing algorithm (eg a hash function) accessible by everyone. This type does not depend on a key signature, even for the size of the signature is in general independent of the size of the document and the signature is not more than a reduced information amount compared to the signed information.

* Corresponding author:
hajji-tarik@hotmail.com (Hajji Tari)
Published online at http://journal.sapub.org/ajis

We have a very large collection of methods invented to do the signature of digital images and we can be further subdivided into two types:

External signatures provide an alternative to conventional watermarking techniques under service integrity check in the pictures. Unlike image watermarking techniques, the trade mark is not inserted in the image itself, but transmitted with it in an encrypted form. The technique of "row-column hash function" is to calculate a hash value for each row and each column of the original image. When it is desired to check the integrity of an image, it recalculates the hash values of the rows and columns of the image to be tested and compared with those of the original image. Another algorithm also uses hash functions; it is the function Hash Block-Based (BBH). The principle is similar to that described above, except that it no longer operates on the rows or columns of the image, but on blocks. Thus when there are differences in the hash values, simply refer to the relevant blocks to locate areas of the image that has been manipulated [3, 13].

Unlike techniques using hash functions for generating a fingerprint image, some authors, such as Lin and Chang or Queluz offer to extract the intrinsic characteristics of the image, such as edges, and encrypting using an asymmetric encryption algorithm to transmit simultaneously image [15, 16].

1.2. Digital Watermarking

The basic idea of "watermarking" is to hide in a digital document subliminal information (inaudible or invisible depending on the nature of the document) to ensure security

service (copyright, integrity, traceability, non-repudiation, etc.). One of the peculiarities of digital watermarking compared to other techniques, such as a simple storage of information in the header of the file, is that the brand is linked intimately and resistant to data manner.

Therefore, the Watermarking is theoretically independent of the file format and it can be detected or extracted even if the document has been modified or if it is incomplete. The problem of the integrity is still little addressed by the "Watermarking" community and many questions remain open. One can for example ask whether it is preferable to use a delicate watermarking rather than a robust watermark, or even opt for a different solution. We can say that we have four watermarking approach:

The first methods proposed to ensure service integrity were based on the use of a fragile watermark, as opposed to the robust watermark conventionally used to protect copyrights. The principle of these approaches is to insert a mark or a binary logo (usually predefined and independent of the data to be protected) in the original image so that the smallest changes to the image also affect the trade mark inserted. To verify the integrity of an image, then just locally check for this brand [2, 3].

One of the first techniques used for verifying the integrity of an image was to insert values of "checksum" in the least significant bit (LSB) of the pixels of the image. The proposed algorithm by Walton in 1995 consists to selecting pseudo-randomly (according to a key), groups of pixels and calculates, for each, a value of "checksum". These values are obtained from the numbers formed by the seven most significant bits (MSB) of the selected pixels, and are then inserted in binary form at the lower bits. Fridrich and Goljan have, meanwhile, also developed a technique using LSB as support, but in the end, this time, to hide enough information to be able to not only detect possible manipulations, and also enable partial reconstruction of damaged areas. Faced with this semi-failure, research is currently moving towards semi-fragile say approaches. Methods using a semi-fragile watermark stand fragile methods insofar as they offer increased robustness to image manipulations. The objective is to discriminate malicious operations, such as adding or deleting an important element of the image of global transformations shall not affect the semantic content of the image [4, 9, 10].

Watermarking by region is to crop the image that you want to protect relatively large blocks (about 64 × 64 pixels) and insert in each, a "relatively strong" brand. When one wishes to verify the integrity of the image, we test the presence of the brand in different blocks. The basic idea of this method is to extract some features of the original image and then hide in the image in the form of a robust and invisible watermarking the classical sense of copyright. When it is desired to check the integrity of an image, we simply compare the characteristics of the image with those of the original image contained in the watermarking. If the characteristics are the same, it means that the image has not been manipulated, if the differences indicate regions that have been affected [11, 14, 18, 19].

1.3. Artificial Neural Networks

The neural network is inspired by biological neural system. It is composed of several interconnected elements to solve a collection of varied problems.

The brain is composed of billions of neurons and trillions of connections between them. The nerve impulse travels through the dendrites and axons, and then treated in the neurons through synapses.

This results in the field of artificial neural networks in several interconnected elements or belonging to one of the three marks neurons, input, output or hidden. Neurons belonging to layer n are considered an automatic threshold. In addition, to be activated, it must receive a signal above this threshold, the output of the neuron after taking into account the weight parameters, supplying all the elements belonging to the layer n+1. As biological neural system, neural networks have the ability to learn, which makes them useful.

The artificial neural networks are units of troubleshooting, capable of handling fuzzy information in parallel and come out with one or more results representing the postulated solution. The basic unit of a neural network is a non-linear combinational function called artificial neurons. An artificial neuron represents a computer simulation of a biological neuron human brain. Each artificial neuron is characterized by an information vector which is present at the input of the neuron and a non-linear mathematical operator capable of calculating an output on this vector. The following figure shows an artificial neuron [20]:

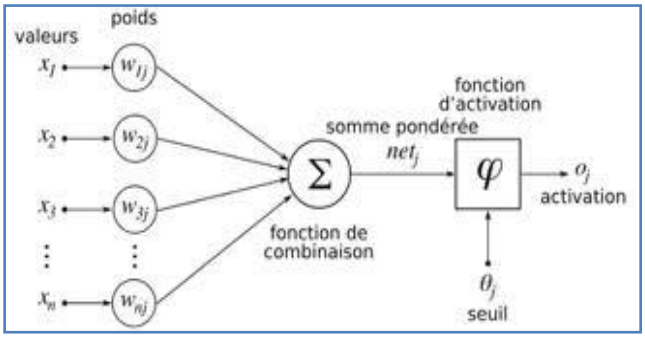

Figure 1. Artificial neuron

The synapses are Wij (weights) of the J neurons; they are real numbers between 0 and 1. The function is a summation of combinations between active synapses associated with the same neuron. The activation function is a non-linear operator to return a true value or rounded in the range [0 1]. In our case we use the sigmoid function [21];

$$f(x) = 1/(1 + exp(-x)) \quad (1)$$

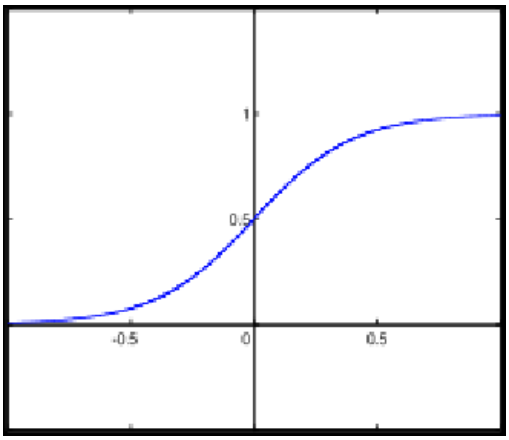

Figure 2. Sigmoid function

An artificial neural network is composed by a collection of artificial neurons interconnected among them to form a neuronal system able to learn and to understand the mechanisms.

Each artificial neural network is characterized by its specific architecture; this architecture is denoted by the number of neurons of the input layer, the number of hidden layers, the number of neurons in each hidden layer and the neurons number in the output layer. A layer of neurons in a neural network is a group of artificial neurons, with the same level of importance, as is shown in the following figure [22]:

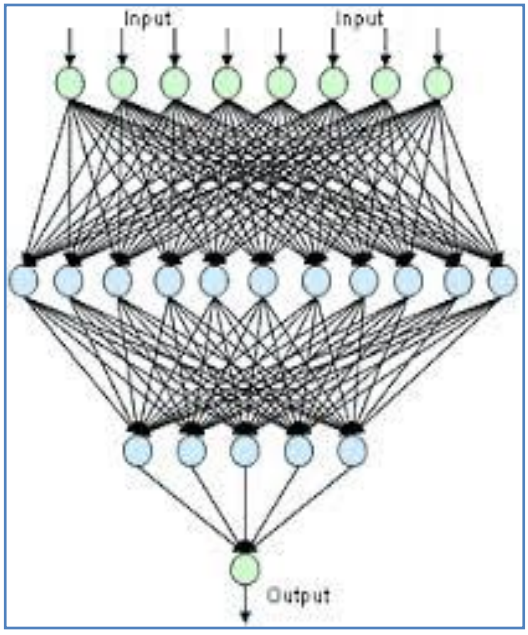

Figure 3. Artificial neural network

The operating principle of artificial neural networks is similar to the human brain; first, it must necessarily pass on the learning phase to record knowledge in the memory of the artificial neural network. The storage of knowledge is the principle of reputation and compensation to a collection of data that forms the basis of learning.

We have several algorithms that can teach an artificial neural network as backpropagation. The backpropagation is a method of calculating the weight for network supervised learning is to minimize the squared error output. It involves correcting errors according to the importance of the elements involved in fact the realization of these errors: the synaptic weights that help to generate a significant error will be changed more significantly than the weights that led to a marginal error. In the neural network, weights are, first, initialized with random values. It then considers a set of data that will be used for learning.

1.4. Compression of Digital Images by Artificial Neurons Networks

It is noteworthy that networks of artificial neurons are very effective methods for compression and decompression of an image. For this operation, a multilayer neural network is used (at least three layers) [8, 9]. With a hidden layer formed by a number of artificial neurons below those of the input and output layer. In this case, the advantage of using artificial neural network in two modes: the compressor and decompresses, as the following figure shows:

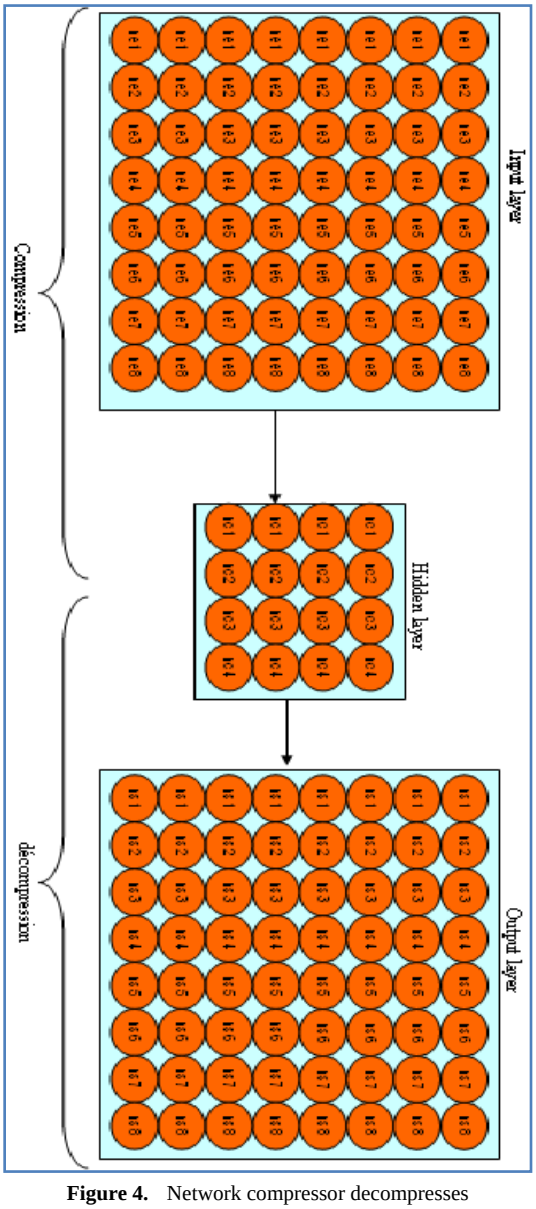

Figure 4. Network compressor decompresses

Compression: Layer entered → hidden layer.
Decompression: hidden layer → Output layer.

But the problem is finding the right architecture of artificial neural network gives better results (number of layers and number of neurons in each layer).

It should be noted that architecture of such a neural network is considered good if the network combines converges in a minimum of time and minimizes the loss of information on time of decompression.

1.5. Proposed Method

We propose in this work, a new approach that combines between the signature and watermarking images with the use of artificial neural networks. This new approach allows protecting the integrity of the images and offers a robust and simple verification mechanism. The principle of the method is to watermarking the image of the compressed image obtained by an artificial neural network for supervised learning. The verification mechanism is to retrieve previously saved in the compressed image and use the same network of artificial neurons to regenerate the image.

The idea that we will propose in this work is to watermarking and saves without a trace in the images of the administrative documents the compressed of these images in a way that does not change the visual structure of pictures. The verification process is therefore to retrieve the information watermarking of each image and regenerate their compressed and finally use the artificial neural network to regenerate the original image. If the pictures look like pictures regenerated then we can deduce that the document is clean if not then the document is forged. The role of artificial neural networks in this process is to do the compression and decompression of digital images and also verify the signature of the signed images. The following example illustrates the principle of this approach; it has an image composed by 64 pixels:

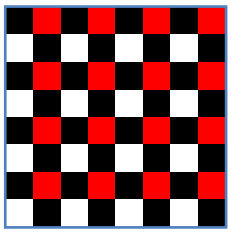

Figure 5. Original bloc

The original block is composed by 8 * 8 = 64 pixels. We will use the artificial neural network to compress for the tablet shown in the following figure:

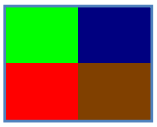

Figure 6. Compressed block

The compressed block is composed of 2 * 2 = 4 pixels; it represents the result of the compression of the original block by network artificial neurons of (64 * 4 * 64) architecture. Then we'll take the 4 pixels of the compressed block. And we will inject them into the 4 sides of the original block to have tattooed block like the following block:

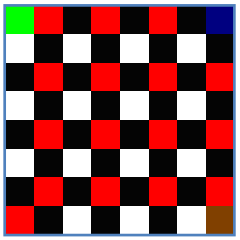

Figure 7. Block watermarked

2. Materials and Methods

In this section, we will describe the steps of our methodology:

Step 1. It begins with the analysis and design of an automatic generator of artificial neural networks, for instantiating a collection of networks characterized by different architectures, the goal is to make a comparative study to choose the most performing network for the compression and decompression of digital images. The backpropagation algorithm is used for learning these artificial neural networks.

Step 2. Then we select the most effective artificial neural network, for generating the compressed image to digitally sign.

Step 3. We make the insertion of the compressed image in the original image with a watermarking algorithm.

Step 4. To verify the integrity of the signed image, then back out our brand watermarking and finally using our artificial neural network for the restoration of the original image; if the restored image looks like the original image then we can say that our image has not been manipulated, if not, then we can say that our image has been manipulated. The following figure explains the principle of our approach:

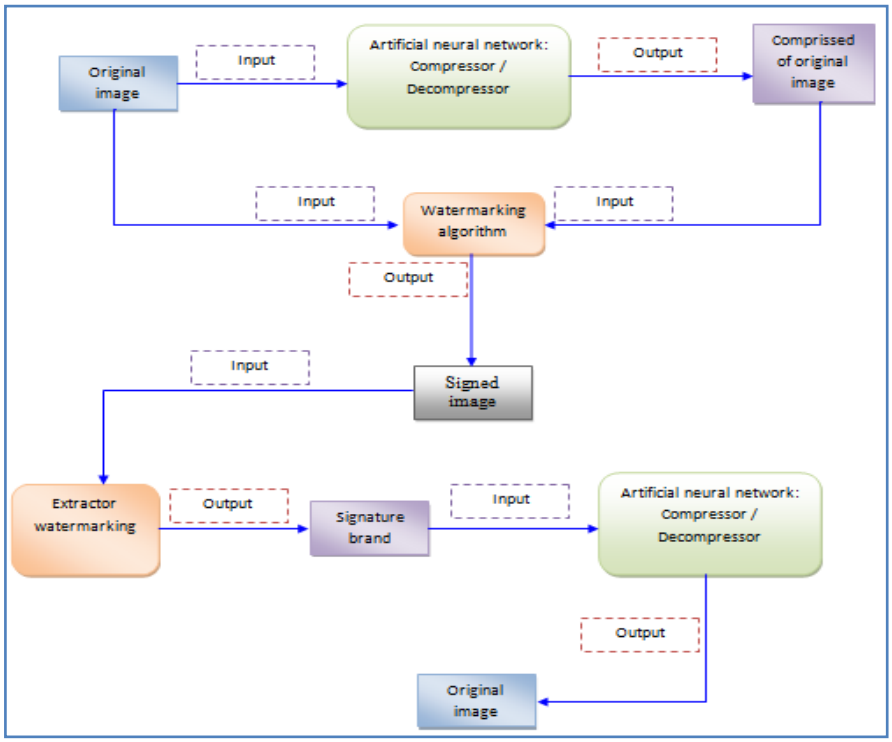

Figure 8. Original bloc

3. Application and Results

3.1. Automatic Generator of Artificial Neural Networks

We started our work by analysing and developing an application able to automatically generate of the programs represent the artificial neural networks in an object oriented language such as Java and C + +.

This application uses a Meta model description of the characteristics of artificial neural network we want to instantiate. Let us know that each artificial neural network is characterized by a specific architecture. We designed a language that allows us to give all specifications for artificial neural network as the number of hidden layers and the number of neurons in each layer. Our meta-model is under following extensible XML form:

```
<RNA> </RNA>
<PARAMETRAGE> </PARAMETRAGE>
<NEURON> </NEURON>
<COUCHENNTREE></COUCHENNTREE>
<COUCHECACHEE></COUCHECACHEE>
<COUCHESORTIE></COUCHESORTIE>
```

The following description shows an example of use for the instantiation of an artificial neural network characterized by the following specifications:

1. An input layer formed by 8 artificial neural.
2. A hidden layer formed by 4 artificial neurons.
3. An output layer formed by 8 artificial neurons.

We can use the write (8 * 4 * 8) as an abbreviation of this architecture:

```
<RNA nom = Reconnaissance des formes>
<PARAMETRAGE>
<ACTIVER> 0,5</ACTIVER>
<APPRENTISSAGE> OUI
</APPRENTISSAGE>
<AGREGATION> sigmo ïde
</AGREGATION>
<ERREURQUADRATIQUE > 0; 000001
</ERREURQUADRATIQUE >
<NOMBREITERATION > 1000000
</NOMBREITERATION >
<APRRENTISSAGE> r étro-propagation
</APRRENTISSAGE >
</PARAMETRAGE>
<COUCHENNTREE>
<NEURON> 8 </NEURON>
</COUCHENNTREE>
<COUCHECACHEE>
<NEURON activer=0.5> 2 </NEURON>
<NEURON activer=0.8> 2 </NEURON>
</COUCHECACHEE>
<COUCHECACHEE>
<NEURON activer=d éfaut> 6 </NEURON>
</COUCHECACHEE>
<COUCHECACHEE>
<NEURON> 4 </NEURON>
</COUCHECACHEE>
<COUCHESORTIE>
<NEURON> 8 </NEURON>
</COUCHESORTIE>
</RNA>
```

The generation application is composed by 4 programs:

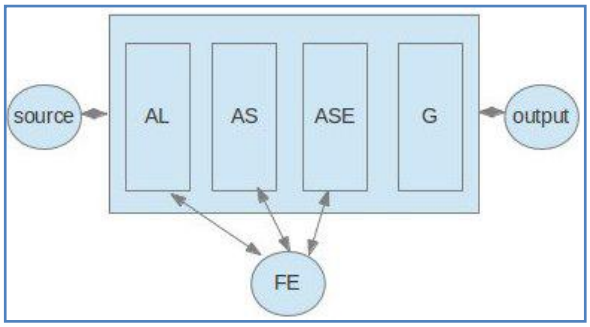

Figure 9. General architecture of the generator

Lexical analyser: who makes the verification of the lexicon and the recognition of all lexical units and fills the error file if any errors ever met. If the lexical analyser completes his execution without encountering any error then it passes to parser the program, it should be noted that this latter is generated by the Flex tool that provides a C program that must be compiled to give an executable. The parser can recognize syntax errors then recorded them in the error log and evaluate the expressions using the units generated by the lexical analyser program. The semantic analyser allows for the semantic validation of sources according to all specifications of artificial neural networks. If we encounter errors, we have to save them in the errors file and stop the build operation.

Syntactic, lexical and semantic analysers represent the control phase of the generator while the generator program do the automatic generation of a program written in an object oriented language. It should be noted that this application can generate an artificial neural network program directly in C + + and java or in an XML form that must be exploited to expand the set of target languages.

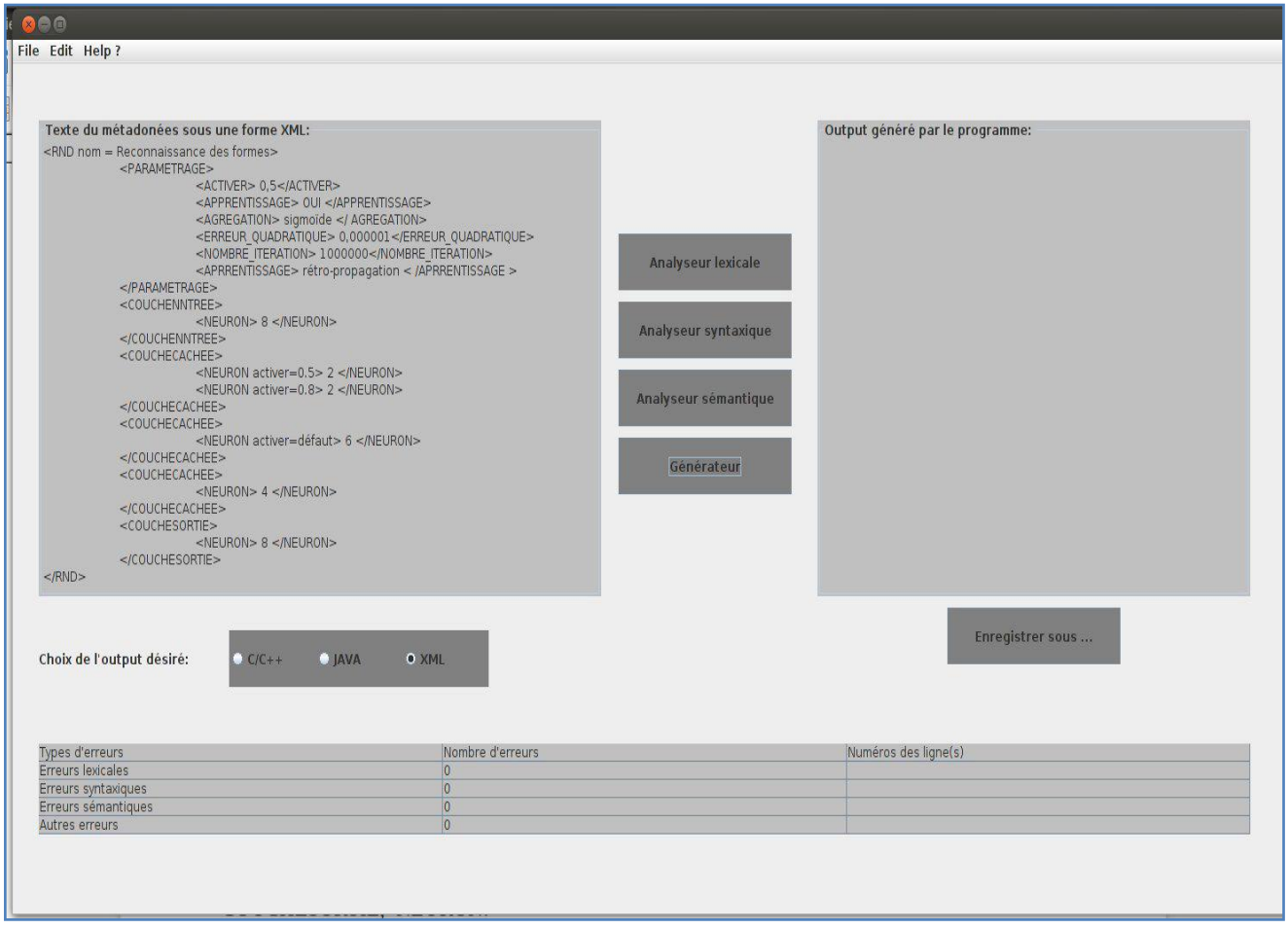

Figure 10. Automatic generator artificial neural network

The program generated by the application meets the following generic class diagram:

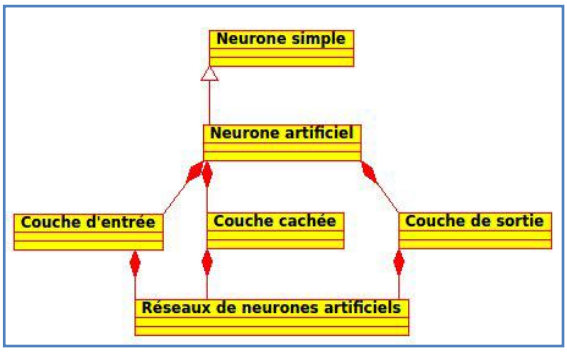

Figure 11. Class diagram

The main class of the network is composed by the class of input layer, one or more classes' hidden layer and an output layer. The input layer, hidden layer and the output layer are composed of a collection of artificial neurons. The following figure shows all Java classes in Eclipse SDK developed.

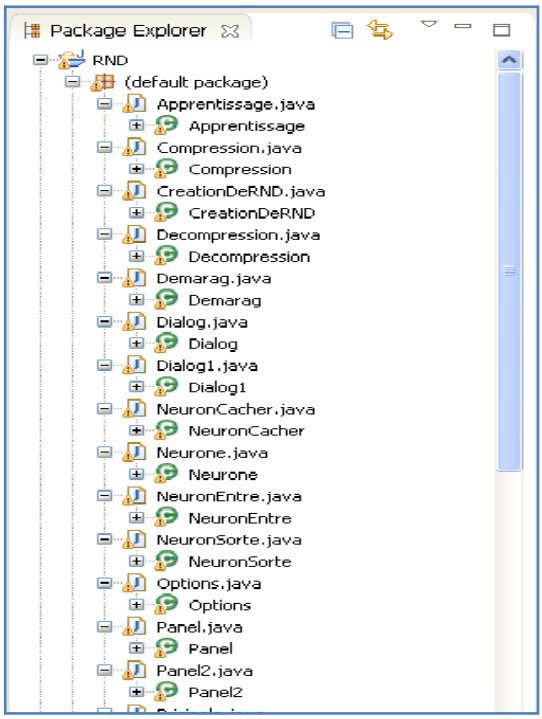

Figure 12. Java code developed

We used this application to generate a collection of the artificial neural networks of different architectures able to learn the mechanism of compression and decompression of digital images. It should be noted that the learning process is done with the backpropagation algorithm and a learning base composed by blocks of carefully selected images. The images are subdivided into several blocks to fit the inlet of the artificial neural network. We used the table of results achieved to choose the artificial neural network to be used for tattooing and signature. With this neural network, it was the constriction of the compressed image of the picture we want to digitally sign. It should be noted that the compression mechanism is on block, so we will do the same for the tattooing process.

3.2. Learning Algorithm

Step 1: Initialize the connection weights (weights are taken randomly).

Step 2: Propagation entries entered the E_i are presented to the input layer:

$$X_i = E_i \quad (2); \tag{2}$$

The spread to the hidden layer is made using the following formula:

$$Y_i = f\left(\sum_{i=1}^{7} X_i * W_{ji} + X_0\right) ; \tag{3}$$

Then from the hidden layer to the output layer, we adopt:

$$Z_k = f\left(\sum_{i=1}^{3} Y_i * W_{kj} + Y_0\right) ; \tag{4}$$

X_0 and Y_0 are scalar;

$f(x)$ Is the activation function:

$$f(x) = \frac{1}{1+e^{-x}} ; \tag{5}$$

Step 3: Back propagation of error at the output layer, the error between the desired output S_k and Z_k output is calculated by:

$$E_k = Z_k(Z_k - 1)(S_k - Z_k) ; \tag{6}$$

The error calculation is propagated on the hidden layer using the following formula:

$$Y_j = f_j(1 - Y_j)\sum_{k=1}^{7} W_{kj} \; E_k ; \tag{7}$$

Step 4: Fixed connection weights the connection weights between the input layer and hidden layer is corrected by:

$$DW_{ji} = n X_i F_j; \tag{8}$$

And

$$DY_0 = n F_j ; \tag{9}$$

Then, we change the connections between the input layer and the output layer by:

$$DW_{kj} = n Y_j E_k; \tag{10}$$

N is a parameter to be determined empirically.

Step 5 loops: Loop to step 2 until a stop to define criterion (error threshold, the number of iterations)

3.3. Watermarking Algorithm

For each block of the original image do:

Calculate the compressed block of the current block with the neural network.

It should be noted that the size of the compressed block is less than the original block size, this property is very important, it allowed us to tattoo the original block.

We replace the sub-blocks in the original block by block in the compressed block.

The original block is composed by 8 * 8 = 64 pixels.

We will use the artificial neural network to compress for the tablet.

The compressed block is composed of 2 * 2 = 4 pixels; it represents the result of the compression of the original block by network architecture of artificial neurons (64 * 4 * 64).

It will take 4 pixels of the compressed block, and then we will inject them into the 4 sides of the original block to have the watermarked block.

And this process must repeat this algorithm wholes blocks up the image.

It should be noted that a sequence of experiments was conducted to determine a replacement algorithm blocks based on their visual asp.

For checking the signature, is collected in a single block in the blocks previously saved on image to form the compressed block.

Then use the artificial neural network for decompression.

Finally, if the decompressed image looks like the original image then the image is signed. If not, then the image is not signed or has been falsified.

3.4. Results of Compression by Artificial Neurons Networks and Watermarking

In the following table, we pooled the results obtained through the comparative study to choose the structure of the artificial neural network adopted for the problem of digital image compression:

1. The first column: to indicate the number of neurons in the input layer.
2. The second column: is composed of three sub columns are reserved to indicate the number of neurons in the first three hidden layers.
3. The third column: indicating the number of neurons in the output layer.
4. The fourth column: to give a test stop for the learning algorithm and the number of iterations.
5. The fifth column used the quadratic error. The last column represents the rate of convergence for each experiment.

nbr_ne_ce	Couche cachées			nbr_ne_cs			
	nbr_ne_cc1	nbr_ne_cc2	nbr_ne_cc3				
8*8	1*1	**	**	8*8	100 000	0,0001	30
					1 000 000	0,0001	35
	2*2	**	**		100 000	0,0001	50
					1 000 000	0,0001	52
	3*3	**	**		100 000	0,0001	20
					1 000 000	0,0001	25
	4*4	**	**		100 000	0,0001	90
					1 000 000	0,0001	95
	3*3	2*2	**		100 000	0,0001	32
					1 000 000	0,0001	40
	4*4	3*3	**		100 000	0,0001	50
					1 000 000	0,0001	56
	4*4	2*2	**		100 000	0,0001	80
					1 000 000	0,0001	85
	6*6	4*4	**		100 000	0,0001	60
					1 000 000	0,0001	65
	6*6	4*4	2*2		100 000	0,0001	70
					1 000 000	0,0001	76
10*10	1*1	**	**	10*10	1 000 000	0,0001	25
					10 000 000	0,0001	30
	2*2	**	**		1 000 000	0,0001	35
					10 000 000	0,0001	38
	3*3	**	**		1 000 000	0,0001	15
					10 000 000	0,0001	20
	4*4	**	**		1 000 000	0,0001	89
					10 000 000	0,0001	92
	3*3	2*2	**		1 000 000	0,0001	30
					10 000 000	0,0001	38
	4*4	3*3	**		1 000 000	0,0001	48
					10 000 000	0,0001	53
	4*4	2*2	**		1 000 000	0,0001	80
					10 000 000	0,0001	85
	6*6	4*4	**		1 000 000	0,0001	62
					10 000 000	0,0001	65
	6*6	4*4	2*2		1 000 000	0,0001	68
					10 000 000	0,0001	72

Figure 13. Table of results

An example of compression and decompression of image Linda with a neural network (64 * 8 * 64):

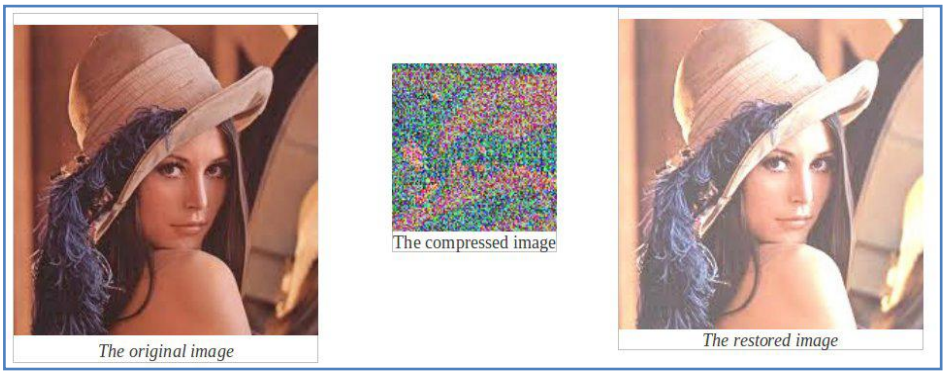

Figure 14. Compression and decompression with an artificial neural network

In this figure, we have the result of an operation for compressing and decompressing image Linda. It should be noted that the compressed digital image with the artificial neural networks is an image consisting of information that is unrelated to the original image.

The following photo of Linda is a highly used in many applications for image processing picture. In this work, we will use this image to explain our approach.

Figure 15. Image original Linda

An example of a watermarking image of Linda with artificial neural network (64 * 8 * 64) and with an artificial neural network (64 * 4 * 64):

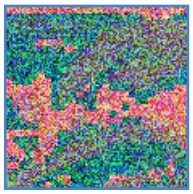

Figure 16. Linda compressed with RNA (64 * 16 * 64)

Linda compressed represents an amount of incomprehensible information, it occupies space that Linda month. For this architecture, each block of 64 pixels is reduced to a block of 16 pixels.

Figure 17. The compressed of Linda with artificial neural network (64 * 4 * 64)

This tablet is the result of compression of the image with a Linda network of artificial neurons of different architecture. We have here; each block of 64 pixels is reduced to a block of 4 pixels.

Figure 18. Image signed with artificial neural network (64*16*64)

In the following, we will present the results relating to the operation of signature by the watermarking and different architecture of network naturals networks:

1. Figure 19: This picture shows the result of the signature operation with the image compressed of Linda with an artificial neural network (64 * 16 * 64).

Figure 19. Image signed with artificial neural network (64*8*64)

2. Figure 20: Here, we used the same picture of Linda and we used the compressed generated by a neural network (64 * 8 * 64).

Figure 20. Image signed with artificial neural network (64*4*64)

3. Figure 21: In this picture we used a compressed of Linda on artificial neural network (64 * 4 * 64), but we changed the algorithm for distributing of the information on the tattoo picture.

Figure 21. Image signed with artificial neural network (64*2*64)

The use of an artificial neural network (64 * 2 * 64) reduces the amount of watermarking data.

4. Interpretation and Discussion

From the table of results of the comparative study to deduce the structure of the artificial neural network that maximizes the rate of convergence for the problem of compression and decompression of images, we concluded that the most suitable architecture (64 * 4 * 64) hat is to say, a neural network with an input layer with 64 artificial neurons, a hidden layer with 16 of artificial neurons and an output layer with 64 of artificial neurons. The use of this architecture is ideal for compression problem because it keeps more than 96% of the information at the time of decompression. Signing with this architecture is possible and it gives very good results as it helps to recognize the signature with the network of artificial neurons quickly. But the disadvantage of this architecture is that it generates a lot of spots on the image. To solve this problem, you can choose specific locations in the image, as can for example use the next closest architecture (64 * 4 * 64). And if the image is very large, we can sign with the compressed of the compressed of the image.

Figure 22. Linda image tattoo with the compressed second

It should be noted that two neural networks with the same architecture cannot give the same compression's results. This assumption we prove experimentally reinforces the importance of this method. That is to say, as we do not have the artificial neural network used for signature, then we cannot create equivalent signatures with other artificial neural networks.

5. Conclusions and Perspective

We have confirmed in this work the use of artificial neural networks to sign administrative documents by tattoo pictures that make up the document. We went through the development of an application that allows for the automatic generation of artificial neural networks and simplify the task of programming the one hand and by the other, it allows us to customize and change the settings of artificial neural networks specifications. It should be noted that you can use this application in other work that uses artificial neural networks.

Then we made a comparative study to choose the structure of the artificial neural network which retains the maximum information at the time of compression and decompression of digital images. And we conclude that good architecture is (64 * 16 * 64).

And finally we passed to make the signature of the images with the best artificial neural networks to the end to conclude that this is the architecture (64 * 4 * 64) which is more efficient for the tattoo.

And the prospect of this work, we seek to develop this approach to using the recursive compressions for digitally signing of the administrative documents.

REFERENCES

[1] R. Rivest, A. Shamir, L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems]. Communications of the ACM, Vol. 21 (2), pp.120–126. 1978.

[2] C. Rey & J.-L. Dugelay. Blind Detection of Malicious Alterations on Still Images Using Robust Watermarks. IEE Secure Images and Image Authentication colloquium, London, UK, Apr. 2000.

[3] M. Kutter, S. Voloshynocskiy and A. Herrigel. The Watermark Copy Attack. In Proceedings of SPIE Security and Watermarking of Multimedia Content II , vol. 3971, San Jose, USA, Jan. 2000.

[4] J. Fridrich, M. Goljan & N. Memon. Further Attacks on Yeung-Minzer Fragile Watermarking Scheme. SPIE International Conf. on Security and Watermarking of Multimedia Contents II , vol. 3971, No13, San Jose, USA, Jan. 2000.

[5] Julien Chable. Emmanuel Robles Programmation Android, de la conception au déploiement avec le SDK Google Android, New York, France, 2009.

[6] UML Infrastructure Final Adopted Speci_cations, version 2.0, Sebtembre 2003. http://www.omg.org/cgi-bin/doc?ptc/03-09-15.pdf.

[7] UML Infrastructure Final Adopted Speci_cations, version 2.0, Sebtembre 2003. http://www.omg.org/cgi-bin/doc?ptc/03-09-15.pdf.

[8] S. Bhattacharjee and M. Kutter. Compression Tolerant Image Authentication. IEEE International Conf. on Image Processing (ICIP'98), Chicago, USA, Oct. 1998.

[9] V. Prasad Vaddella, K. Rama, 2010, Artificial Neural Networks For Compression Of Digital Images: A Review, International Journal of Reviews in Computing, ISSN: 2076-3328, E-ISSN: 2076-3336.

[10] J.-L. Dugelay & S. Roche. Introduction au tatouage d'images. Annales des Télécommunications, 54, no 9-10, pp. 427-437, 1999.

[11] J.-L. Dugelay. Procédé de dissimulation d'informations dans une image numérique. Brevet INPI FR 98-04083 (EURECOM 09-FR), March 1998.

[12] J.-L. Dugelay & S. Roche. Process for marking a multimedia document, such an image, by generating a mark. Pending patent EP 99480075.3 (EURECOM 11/12 EP), July 1999.

[13] J. Fridrich. Robust Bit Extraction from Images. ICMCS'99, Florence, Italy, june 1999.

[14] J. Fridrich and M. Goljan. Protection of Digital Images using Self Embedding. The Symposium on Content Security and Data Hiding in Digital Media, New Jersey Institute of Technology, Mar. 1999.

[15] J. Fridrich. Image Watermarking for Tamper Detection. Proceedings IEEE Int. Conf. on Image Processing (ICIP'98), Chicago, USA, Oct. 1998.

[16] J. Fridrich. Methods for detecting changes in digital images. Proceedings IEEE Int. Conf. on Image Processing (ICIP'98), Chicago, USA, Oct. 1998.

[17] Kodak. Understanding and Intergrating KODAK Picture Authentication Cameras. http://www.kodak.com/US/en/digital/software/imageAuthentication/.

[18] D. Kundur and D. Hatzinakos. Towards a Telltale Watermarking Technique for Tamper-Proofing. IEEE International Conf. on Image Processing (ICIP'98), Chicago, USA, Oct. 1998.

[19] C.-Y. Lin and S.-F. Chang. A Watermark-Based Robust Image Authentication Using Wavelets. ADVENT Project Report, Columbia University, Apr. 1998.

[20] C.-Y. Lin and S.-F. Chang. Generating Robust Digital Signature for Image/Video Authentication. Multimedia and Security Workshop at ACM Multimedia 98, Bristol, UK, Sep 1998.

[21] http://upload.wikimedia.org/wikipedia/commons/thumb/5/58/ArtificialNeuronModel_francais.png/420px-ArtificialNeuronModel_francais.png.

[22] http://upload.wikimedia.org/wikipedia/commons/thumb/9/96/SigmoidFunction.svg/220px-SigmoidFunction.svg.png.

[23] http://underflow.fr/wp-content/uploads/2012/08/neuralt_naetverk.png.