

# Decision Support Through Deep Learning: Application To Image Classification and Recognition

Glory Ndele Wum Edmond <sup>(1)</sup>, Simon Ntumba Badibanga <sup>(2)</sup>.

<sup>(1)</sup> Senior lecturer, Department of Mathematics and Computer Science

<sup>(2)</sup> Professor, Department of Mathematics and Computer Science.

Department of Mathematics and Computer Science, Faculty of Sciences, University of Kinshasa, Kinshasa, Democratic Republic of Congo. February 2022

**Abstract:-**By extracting the relevant characteristics of a multilayer perceptron with four layers of convolutions, we were able to create our Convolutional Neural Network model, for facial recognition, using an existing CNN architecture whose parameters are driven by the gradient retro-propagation algorithm. In a database of 1672 images of which 80% (or 1337 learning images) and 20% (or 335 images of tests); The model predicted very strongly at a coefficient of 99.8% reliability while minimizing the risks of 0.02%. This model is tested with 20 epochs.

**Keywords :-** PMC, CNN, Neurones, Deep Learning, ReLU, Flatten, Softmax.

## I. INTRODUCTION

Deep learning has revolutionized machine learning in recent years. While the first striking results were obtained mainly in image analysis, current work in deep learning now focuses on all types of data and almost all types of processing. Its application has an impact in the field of data science and the extraction of knowledge is considerable.

However, bases of observations characterize a particular domain (animals, fruit, sick, genes, . . .), which are grouped into several classes. Automatic image classification is an application of pattern recognition, which consists of automatically assigning an image to a class using a classification system. [2, 6, 8]

The problem of this research is to recognize the images, restructured them, and to apply the techniques of searching for learning images and tests to facilitate decision-making, using a model of a multilayer perceptron of a convolutional neural network.

This problem formulates the hypothesis that images, can be classified into several classes, to allow better processing, which will help optimization in deep learning to search for images; the recognition of images similar to a query image, as well as the achievement of good performance on average on all images.

The objective of this research is to facilitate the task of searching for images in automatic classification thanks to automatable methods that allow a machine to evolve through a learning process and thus, perform tasks that are difficult or impossible to be performed by more conventional algorithmic means.

Thanks to the Analytical method that made it possible to analyze the different models of convolutional neural networks supported by techniques based on the (CNN) that are part of the types of deep neural network (Deep Neural Network) and the documentary technique; this model, which is a multilayer perceptron driven by the gradient retro-propagation algorithm using an existing architecture of the Convolutional Neural Network, in acronym (CNN) could be realized

## II. MACHINE LEARNING

Machine learning is a technique through which a machine acquires new knowledge for future use. Machine learning is formed of two types of learning, it is a field of data science, which creates a machine  $f$  such as  $y_i = f(x_i)$  defined by :  $\beta = \{(x_i, y_i) : x_i \in \mathbb{R}^n, y_i \in C\}$ . This machine can be created through: (Decision Tree, Neural Network, Vector Machine Support (SVM), Bayesian Network, Random Forest). [2,3,5,6]

### 2.1. Unsupervised learning

Unsupervised learning, also known as automatic classification, consists of highlighting information hidden by the large volume of data, in order to detect hidden trends in this data. The most appropriate techniques for this type of learning are: Clustering, Principal Component Analysis and Correspondence Factor Analysis. This automatic classification concerns a partition of individuals that obeys the law of intra-class homogeneity and inter-class heterogeneity. This classification is done by two main families of methods namely: classification by partitioning and hierarchical classification. [3,10,11,12,14]

#### 2.1.1. Classification par partitionnement

Partitioning classification consists of segmenting a heterogeneous population into homogeneous subgroups in such a way as to minimize intraclass inertia and maximize interclass inertia. In the mathematical sense this classification consists in partitioning a heterogeneous population  $X$  into  $K$  homogeneous classes  $C_1, C_2, \dots, C_k$  such as :  $X = \{x_i \in \mathbb{R}^p, 1 \leq i \leq m\}$ . Soit  $x_1, x_2, \dots, x_k$  a partition of  $X$ .

$$a) C_i \neq \emptyset ; 1 \leq i \leq k$$

$$b) C_i \cap C_j = \emptyset \quad \forall i \neq j$$

$$c) \bigcup_{i=1}^k C_i = X$$

Inertia reduces dispersion around the mean, that is, the distance between individuals ( $X_i$ ) and the center of gravity ( $\bar{g}$ ). According to **Hyughens** Total Inertia = Intraclasse Inertia + Interclasse Inertia.. [7, 8,9]

$$I_{intraclasse} = \sum_{j=1}^k \sum_{i=1}^{n_j} P_j d^2(x_i, g_j) \quad ; \quad P_j = \frac{|C_j|}{n}$$

With  $P_j$  Represents the weight of the class, it is the ratio of the number of individuals in the class to the total number; and  $g_j$ : it is the distance of the individual from the center of gravity of his class.

$$I_{interclasse} = \sum_{j=1}^k P_j d^2(g_j, \bar{g}) \quad ; \quad g_{(j)} = \frac{\sum_{i=1}^m X_{ij}}{n} \quad , 1 \leq j \leq p$$

The methods (Algorithms) most used in segmentation are:

- The K-Means Algorithm (K-medium or Mobile Center);
- Dynamic Clouds (K-nuclei) which generalizes K-Means;
- Fuzzy classification;
- Gaussian mixing model. [12,14]

### 2.1.2. Hierarchical classification.

Hierarchical classification is part of the Unsupervised Learning methods, which includes two main families of methods namely: hierarchical Ascending Classification and Hierarchical Descending Classification. Individuals are represented by a tree structure called the **Dendrogram** or by the **Nested Partition**. [1,5,7]

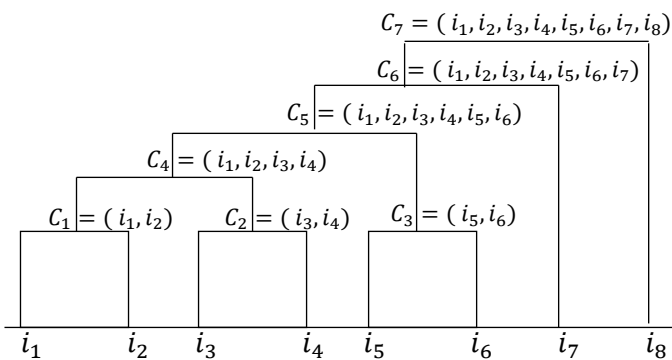


Figure 1 : Dendrogram

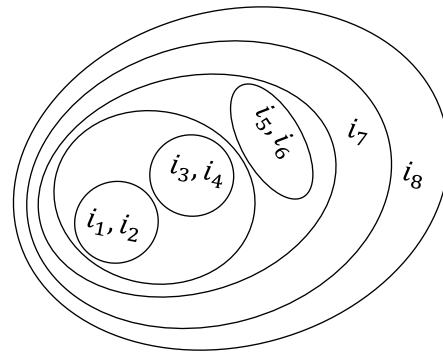


Figure 2 Nested Partition

## 2.2. Supervised learning

Supervised learning involves extrapolating new knowledge from a representative sample from unsupervised learning. This learning is done on labeled data; and in the end, it must take into account its classifier. [6,9,13]

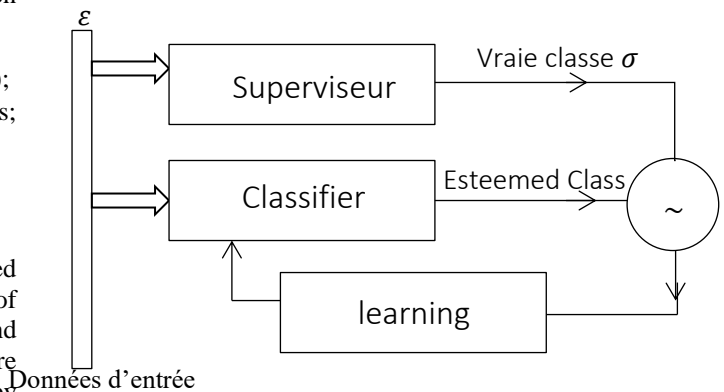


Figure 4 : Supervised learning

## III. DEEP LEARNING (CNN)

Deep learning or convolutional neural networks are very similar to neural networks. They exploit one of the important characteristics of images, namely the spatial distribution of sampling. They consist successively of layers of convolutions, layers of groupings, and connected layers.

The term deep learning refers to the many layers that need to be learned as you train. Convolutional neural networks are not directly inspired by biology and rely on learning algorithms that can fundamentally differ from biological brains. However, they learn internal representations that strongly resemble the ideas that one imagines of representations of the visual cortex.

Considering the classical architecture of a convolutional neural network. An image is provided as input and is convoluted with filters (first layer of convolution) and whose activation cards are grouped and concatenated.

### 3.1. Layers of Deep Learning

The major layers in a convolutional neural network are: (Convolution layer, Pooling layer, and Fully Connected Layers). [10,12,14]

#### a) Convolutional Layers

Convolution layers are a set of filters that are learned during training. The size and number of these filters are defined a priori. [14]

#### b) Pooling Layers

Pooling layers are predefined functions and reduce the number of parameters to learn for later layers while expanding the receptive field. They operate independently at different depths of the network and do not require any weight to drive. One of the classic operations performed is the maximum function, where in the vicinity of N pixels only the maximum is retained in the grouping layer.

#### c) Fully Connected Layers

The neurons in these layers are all connected to all the neurons in the previous activation maps.

In short, in general, the components of a convolutional neural network are:

- Layer of filters convolved on the different channels;
- Pooling: maximum value (max pool) or average (avg pool) in a certain convoluted window;
- Transfer functions: ReLU, etc.
- Near the output, fully connected layers (as with multilayer perceptron)

### 3.2. Contribution of Deep Learning

Deep Learning has come to solve situations that the previous classical algorithms are incapable of, it is or:

- Improve the development of traditional algorithms in the processing of Artificial Intelligence data;
- Develop a large amount of data such as Big Data;
- Adapt to any type of problem;
- Extract characteristics automatically [3,11,13]

### 3.3. Diagram of Deep Learning.

To properly realize a Deep Learning application, it is necessary to respect the following scheme :

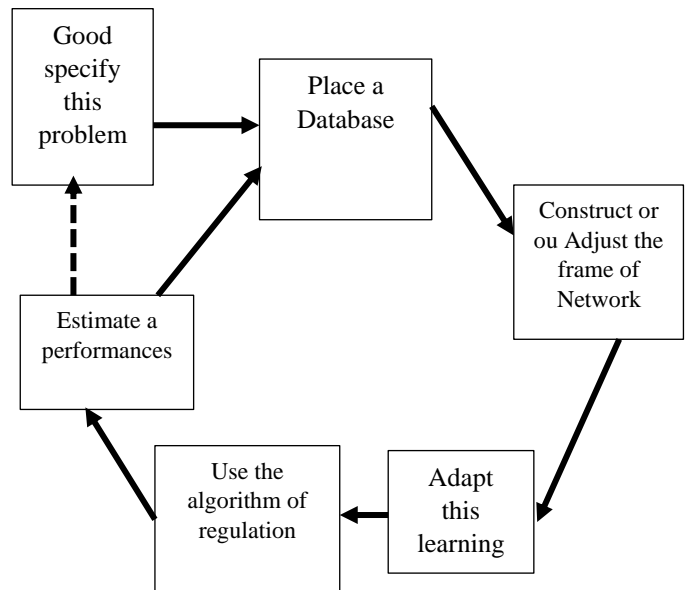


Figure 5 : Diagram of Deep learning

### 3.4. Convolutional operation

We note  $X$  an array  $n * m$ ,  $f$  a filter  $n_f * m_f$  with  $n_f \leq n$ ,  $m_f \leq m$ , the convolution of  $X$  by  $f$  a pixel of coordinates  $(i, j)$ ,  $i \in [0, n - 1]$ ,  $j \in [0, m - 1]$  is defined by :

$$(X * f)_{[i,j]} = \sum_{k=-\frac{n_f}{2}}^{\frac{n_f}{2}} \sum_{l=-\frac{m_f}{2}}^{\frac{m_f}{2}} X[i+k, j+l] * f\left[k + \frac{n_f}{2}, l + \frac{m_f}{2}\right]$$

In this case, if  $i + k$  or  $j + l$  comes out of the image of  $X$ , (that is,  $i + k \geq n$  for example) then we take  $(i + k, j + l) = 0$ , and we talk about « Zero padding » to calculate other padding it is possible by taking the value of the nearest pixel. We can also perform the calculation that on the pixel  $[i, j]$  such that  $(i + k, j + l)$  are always in the image; there will then be the reduction of the image of the output dimension. [7,8,10]

The filter or kernel  $f$  is called Kernel or Filter in English. In a convolution neuron, we do not choose the **Filter**, we learn them, because these are trainable parameters of the network.

For Input  $X \in \mathbb{R}^{n*m*c}$ , with  $c$  : numbers of channels or channels ; When  $c = 1$ , the image is gray level, if  $c = 3$  it is a color channel; And, in the intermediate layers  $C$  corresponds to the number of the previous layer. The convolution neuron  $a = \sigma(z = X * f)$  ; avec  $f \in \mathbb{R}^{n_f*m_f*c}$  et  $\sigma = \mathbb{R} \rightarrow \mathbb{R}$  ; avec  $a \in \mathbb{R}^{n*m}$

Pooling: operation used to reduce the dimension, searches for coarser details of larger structures in the image. (MaxPooling of size  $l$ : we take the maximum element of each sub array of  $l$ ; Sum pooling of size  $l$ : we sum all the elements of each sub-array of size  $l$ ).

Pooling can be used for the frame of square matrices, for rectangular matrices, it is necessary to exploit an algorithm of the staggering to make it square.

**3.4.1. Matrix scaling algorithm (Reduction to reduced staggered form).**

In general,

Let  $M \in Mm, n(K)$ , there is a staggered matrix  $H \in Mm, n(K)$  and a matrix  $V \in GLn(K)$  such that  $H = M.V$

This reduced staggered form is obtained by elementary operations on the columns, more precisely by the following algorithm :

1. Input :  $M \in Mm, n(K)$
  2. Initialization :  $H = M, V$  matrix unit of order  $n, j = 1$
  3. Main loop : for  $i = 1$  to  $m$  :
    - a) Finding a pivot :
 

If  $H_{i,j} = 0$   
 If there is  $s \geq j$  such as  $H_{i,s} \neq 0$

{

      - i.  $X = P^{(n)}$
      - ii.  $V = V.X$
      - iii.  $H = H.X$

}

Else next step.
    - b) Set the pivot to 1
      - i.  $X = D_j^{(n)} \left( \frac{1}{H_{i,j}} \right)$
      - ii.  $V = V.X$
      - iii.  $H = H.X$
    - c) Reduce and stagger: set to 0 the letters coefficients of the pivot line for  $s$  ranging from  $1$  à  $j - 1$  then from  $j + 1$  to  $n$ , loop {
      - i.  $X = X_{j,s}^{(n)} (-H_{i,s})$
      - ii.  $H = H.X$
      - iii.  $V = V.X$
  - d)  $j = j + 1$
  - e) Exit when  $j = n$
- End loop.
4. Output H staggered from  $V \in GLn(K)$  such that  $H = M.V$

In the case of our research, we will apply Max Pooling for square matrices as defined in point 3.4.

The illustrative case or example of Max pooling is given below.

**Le Max Pooling**, (Max Subsampling Operation) takes a region and gives the maximum output. [12,13,14]

$$\begin{bmatrix} 20 & 74 & 17 & 52 \\ 15 & 35 & 21 & 30 \\ 26 & 34 & 15 & 25 \\ 12 & 60 & 40 & 20 \end{bmatrix} \rightarrow \begin{bmatrix} 74 & 52 \\ 60 & 40 \end{bmatrix}$$

By applying Max Pooling, the Starting Matrix is divided into a region of square matrices of order 2, for each region we have retrieved the maximum values to form our Max Pooling matrix.

**Le Mean Pooling**, it calculates the sum of all the values and divides it by the number of values to obtain the representative average of this batch of pixels

$$\begin{bmatrix} 20 & 74 & 17 & 52 \\ 15 & 35 & 21 & 30 \\ 26 & 34 & 15 & 25 \\ 12 & 60 & 40 & 20 \end{bmatrix} \rightarrow \begin{bmatrix} 36 & 40 \\ 34 & 25 \end{bmatrix}$$

By applying mean pooling, the starting matrix is divided into a region of square matrices of order 2, for each region we have retrieved the minimum values to form our Mean Pooling matrix.

**Le Sum Pooling**, realizes the sum of all the values obtained

$$\begin{bmatrix} 20 & 74 & 17 & 52 \\ 15 & 35 & 21 & 30 \\ 26 & 34 & 15 & 25 \\ 12 & 60 & 40 & 20 \end{bmatrix} \rightarrow \begin{bmatrix} 144 & 120 \\ 132 & 100 \end{bmatrix}$$

Applying Sum Pooling, the Starting Matrix is divided into a region of square matrices of order 2, for each region we calculated the sum of all the values to form our Sum Pooling matrix..

**3.5. Flattening**

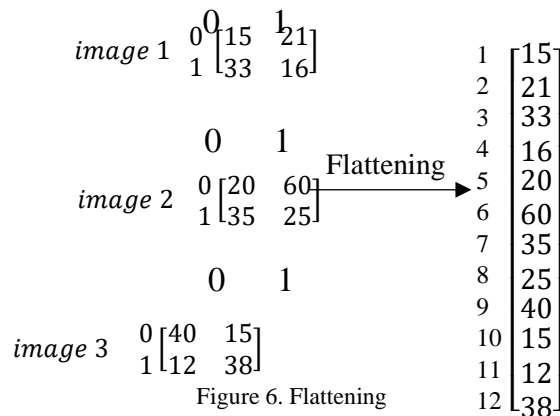


Figure 6. Flattening

By applying flattening by operation Flatten, the Initial Matrices form a column vector.

**IV. IMAGE RECOGNITION**

The layer of neurons (ConvNet) is a stacking of these layers induces local properties of invariance by translation. These properties are essential for the purpose of recognizing characters and more generally images that can be seen from different angles. It is in this area that the most spectacular results were obtained while the name deep learning was advanced in order to accompany the growing success and the associated hype.

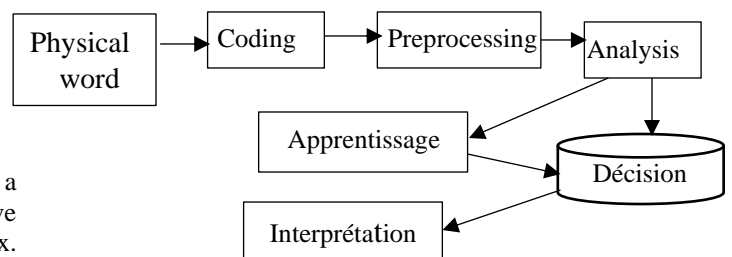


Figure 7. Shape recognition step.

- a) **The physical world** : allows to present the object in its real environment, and in its normal form before any possible treatment.
- b) **The coding** : it consists in observing the shape of the environment in analog form. In order to observe or represent it in discrete form to be processed in the system. The object is codified by binary sequences.
- c) **The Preprocessing** : consists of the standardization of coded information to keep only the information essential to the system.
- d) **Analysis** : it therefore makes it possible to extract the clues that characterize the object represented, in order to establish the parameters on which the learning will be based.
- e) **Learning**: it consists in memorizing and exploiting the knowledge resulting from the parameters of the analysis. Since the model will already be trained by training, it is then that the new data will be used for the test that leads to optimization (prediction). It is during this stage that we will practically know at what coefficient the new model predicts.
- f) **The decision** : this is the stage of image recognition, that is, the stage by which the system will establish a definitive classification. This step is also the step of optimization, because it is during this that we will be exactly defining the object.
- g) **Interpretation** : this is prediction on the image obtained, that is, accurately predict the object. [3,6,11,14]

**V. Application.**

**5.1. Face Database.**

Generally, databases are adapted to the needs of a few specific recognition algorithms. As far as we are concerned, our database consists of 1672 images of faces that we have distinguished into two classes : [2,4,13,14]

- The class of the person "**untouched targets**" containing 1113 images.
- The class of the "**non-target**" person containing 559 images of faces of people different from Intouche.

**5.2. Separation of Databases.**

The implementation of our facial recognition system required to have two image databases: one to perform the learning and the other to test the effectiveness of the data trained. [4,11]

**1. Learning images**

Of the 1672 images of the base, we reserved 80% (1337 images) for training.

**2. Tests images.**

Here on the 1672 of the base we took the 20% (335 images) remaining at the service of the test.

**5.3. Principles and experiments facial recognition system.**

The problem of facial recognition is understood as an image of a face whose identity of the corresponding person is to be found. Face recognition is part of the field of pattern recognition. The purpose of pattern recognition is to classify objects of interest into a number of categories or classes. Objects of interest are usually called models or patterns and in our case they are vectors of characteristics. The classes here represent the different people. Since the classification procedure in our case will be applied on vectors of characteristics.

Recognition is the core of this system, and it is the comparison of the vector code of the face in input with those of the database, and starting from the fact that we want to model a function of the human brain and that we have a classification problem, we chose to implement a Neural Network which is a simulator of the biological neural network. However, in the recognition phase of our system, we used two types of neural network, the first is a Multi Layer Perceptron (MLP) neural network, and the second is a convolutional neural network.

**5.3.1. Structure de l'application.**

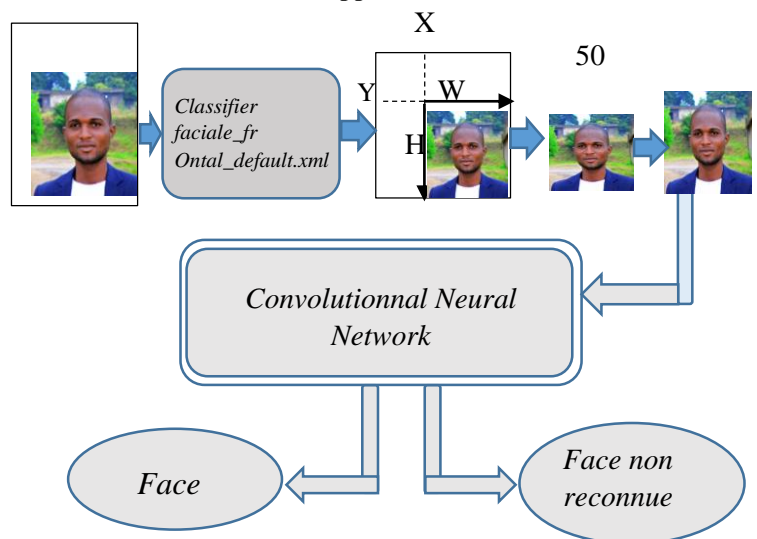


Figure 8. Structure of image recognition

La figure 8 shows the image used in our application to test our Convolutional Neural Network.

**5.3.2. Application architecture.**

The architecture we used a multilayer Perceptron of our convolutional neural network.

This architecture uses four convolution layers, with which the image is filtered before it is vectorized (flattened) of the pixel matrix. This allowed a good prediction, because with these four convolutional layers the model predicts at 99.8% reliability and minimizes 0.02% of the error, these results will be presented in the prediction curve. [3,10,13,14]

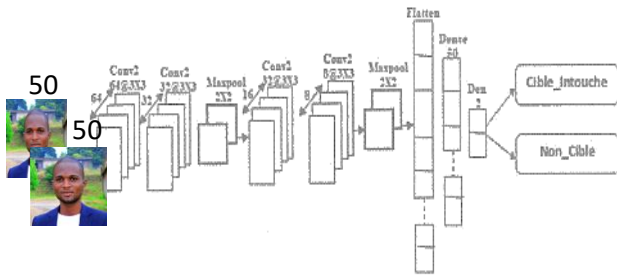


Figure 9. Architecture of our CNN

### 5.3.3. Retro-propagation Algorithm.

The feedback algorithm used in this application is executed in three necessary steps which are :

1. Resizing input images to 50x50x1 Format ;
2. The construction of a CNN structure with four convolutional layers by associating a ReLU correction layer in each convolutional layer whose first layer uses a depth of 64 neurons, the second layer uses a depth of 32 neurons, the third layer uses a depth of 16 neurons and the last one uses a depth of 8 neurons with the nucleus of size 3x3. However, max pooling of size 2x2 is applied after two convolutional layers.
3. After extracting all the features, **the Flatten** operation will flatten the images..
  - **Dense** : denotes the number that wants it on a channel, it always takes as input the result of the flattened image of the previous layer. [13,14]
  - **Softmax** : is a classifier that designates the distribution of probability or that the sum of all probabilities of the output must be equal to 1.

### 5.3.4. Sequential model testing

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 48, 48, 64)	640
activation (Activation)	(None, 48, 48, 64)	0
conv2d_1 (Conv2D)	(None, 46, 46, 32)	18464
activation_1 (Activation)	(None, 46, 46, 32)	0
max_pooling2d (MaxPooling2D)	(None, 23, 23, 32)	0
conv2d_2 (Conv2D)	(None, 21, 21, 16)	4624
activation_2 (Activation)	(None, 21, 21, 16)	0
conv2d_3 (Conv2D)	(None, 19, 19, 8)	1160
activation_3 (Activation)	(None, 19, 19, 8)	0
max_pooling2d_1 (MaxPooling2D)	(None, 9, 9, 8)	0
flatten (Flatten)	(None, 648)	0
dropout (Dropout)	(None, 648)	0
dense (Dense)	(None, 50)	32450
dense_1 (Dense)	(None, 2)	102

Total params: 57,440  
 Trainable params: 57,440  
 Non-trainable params: 0

Figure 10. Structure of the implementation of the CNN model architecture sequentially.

As we see, the sequential model is convoluted to four layers, using the ReLU activation function, the total parameters of the model is 57,440. Prediction with training data to see how the model will predict with 20 epochs.

```

Train on 1337 samples
Epoch 1/20
1337/1337 [=====] - 17s 13ms/sample - loss: 0.4680 - accuracy: 0.8010
Epoch 2/20
1337/1337 [=====] - 16s 12ms/sample - loss: 0.2128 - accuracy: 0.9073
Epoch 3/20
1337/1337 [=====] - 15s 11ms/sample - loss: 0.1271 - accuracy: 0.9536
Epoch 4/20
1337/1337 [=====] - 15s 12ms/sample - loss: 0.0970 - accuracy: 0.9611
Epoch 5/20
1337/1337 [=====] - 16s 12ms/sample - loss: 0.0707 - accuracy: 0.9753
Epoch 6/20
1337/1337 [=====] - 16s 12ms/sample - loss: 0.0602 - accuracy: 0.9768
Epoch 7/20
1337/1337 [=====] - 16s 12ms/sample - loss: 0.0536 - accuracy: 0.9783
Epoch 8/20
1337/1337 [=====] - 16s 12ms/sample - loss: 0.0397 - accuracy: 0.9873
Epoch 9/20
1337/1337 [=====] - 16s 12ms/sample - loss: 0.0470 - accuracy: 0.9835
Epoch 10/20
1337/1337 [=====] - 16s 12ms/sample - loss: 0.0296 - accuracy: 0.9918
Epoch 11/20
1337/1337 [=====] - 16s 12ms/sample - loss: 0.0334 - accuracy: 0.9895
Epoch 12/20
1337/1337 [=====] - 16s 12ms/sample - loss: 0.0250 - accuracy: 0.9903
Epoch 13/20
1337/1337 [=====] - 15s 12ms/sample - loss: 0.0198 - accuracy: 0.9940
Epoch 14/20
1337/1337 [=====] - 15s 11ms/sample - loss: 0.0205 - accuracy: 0.9918
Epoch 15/20
1337/1337 [=====] - 16s 12ms/sample - loss: 0.0117 - accuracy: 0.9940
Epoch 16/20
1337/1337 [=====] - 16s 12ms/sample - loss: 0.0300 - accuracy: 0.9858
Epoch 17/20
1337/1337 [=====] - 16s 12ms/sample - loss: 0.0151 - accuracy: 0.9933
Epoch 18/20
1337/1337 [=====] - 16s 12ms/sample - loss: 0.0090 - accuracy: 0.9963
Epoch 19/20
1337/1337 [=====] - 16s 12ms/sample - loss: 0.0089 - accuracy: 0.9970
Epoch 20/20
1337/1337 [=====] - 16s 12ms/sample - loss: 0.0168 - accuracy: 0.9925
    
```

Figure 11. The result found when evaluating the CNN model with 20 epochs.

As we said before, with 1337 training images, the model to use 20 epochs to predict at a coefficient of 99.8%, so we concluded that our optimizes our learning.

### 5.3.5. Overview of the GUI



Figure 12. Testing the application with a Java interface..

The GUI is created in Java script to test the model.

With this interface, we confirm the result of our model, this is a certainty that validates the search result.

### 5.3.6. Presentation of results.

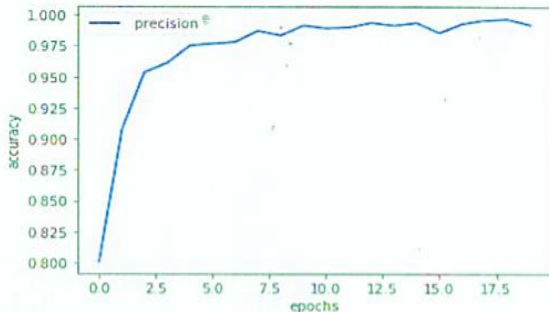


Figure 13. Increased accuracy with the number of epochs.

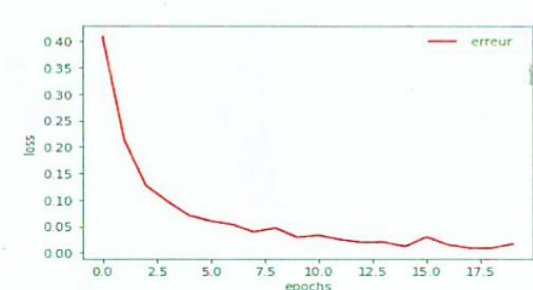


Figure 14. Reduction of errors with the number of epochs.

By analyzing the results obtained, in Figures 13. and 14, we find that the accuracy of learning and validation increases with the number of epochs and after that, it falls again, which means that with each epoch where the accuracy accumulates, the model no longer learns information. If the accuracy is decreased, then we will need more information to make our model learn and therefore we must increase the number of epoch and vice versa. Similarly, the learning and validation error decreases with the number of epochs.

## VI. CONCLUSION.

In summary, image classification is an important task in the field of computer vision, object recognition and machine learning. The objective of this work is to carry out a classification application of an image database into a set of classes in order to recognize the objects in the images, if necessary was that of facial recognition. To operationalize this classification in deep learning; We used the learning method that has shown its performance in recent years and we chose the Gradient Retro-propagation Algorithm as the classification

method, this choice is justified by the simplicity and efficiency of the method.

Thanks to the PYTHON language under the environment of the Anaconda (Tensorflow), we were able to implement our model, this also works thanks to an application created in Javascript on NetBeans 8.4. For each time train the parameters of the model with the new data. The result obtained during the test phase confirms the effectiveness of our approach.

## REFERENCES.

- [1] Antoine Cornéjols, Laurent Miclet, Yves Kodratoff, Apprentissage Artificiel : Concepts et Algorithmes, Deuxième tirage Eyrolles, 2003.
- [2] Achraf Cherti, Jargon Informatique, Logiciel, Version 1.3.6, Avril 2006.
- [3] Christian Gagné, Réseau de neurones à Convolution, éd. Laval, 2008
- [4] Gerard Swinnen, Apprendre à Programmer en Python 3, 2010.
- [5] Karpathy A., Convolutional neural networks for visual recognition. Neural networks, 2016.
- [6] Philippe Preux, Fouille de données, Lille, 2009 ;
- [7] Richard O. Duda, Peter E. Hart, David G. Stork, Pattern Classification, Wiley Interscience, 2001.
- [8] Thibault Allauçon, Introduction à l'apprentissage artificiel, Paris, 2016.
- [9] V. Vapnik. The Nature of Statistical Learning Theory. Springer-Verlag (200)
- [10] Lee, H., Grosse, R., Ranganath, R., and Ng, A.Y. (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In Proceedings of the 26th annual international conference on machine learning, P. 609 – 616. ACM.
- [11] Marc Bidan, *Intelligence artificielle : les défis de l'apprentissage profond*, Universités Nantes 2018 -2019.
- [12] Samuel, A. L. (2000). Some studies in machine learning using the game of checkers. Journal of research and development, 44 (1.2) : 206 – 226.
- [13] S. DIB, Identification des individus multimodals : application sur les images du visage, Thèse de magister, Université de Mohamed Boudiaf, Oran, 2015.
- [14] M. LEMMOUCHI, Identification des visages Humains par réseaux de neurones. Thèse de magister, Université de Batna 2, 2013.