ICISAT'2022

12th International Conference on Information Systems and Advanced Technologies

Certificate of Attendance

Awarded to

Aida CHEFROUR

For attending:

12th International Conference on Information Systems and Advanced Technologies "ICISAT'2022"

and presenting the paper entitled: CAE-CNN: Image Classification Using Convolutional AutoEncoder Pre-Training

Authors: Aida Chefrour and Samia Drissi

ICISAT'2022 General Chair

CAE-CNN: Image Classification Using Convolutional Autoencoder Pre-Training

Aida Chefrour [1,2], Samia Drici[1,3]

[1] Computer Science Department , Mohamed Cherif Messaadia University, Souk Ahras, Algeria
[2] LISCO Laboratory, Badji Mokhtar University, Annaba, Algeria
[3] LiM Laboratory, Faculty of Science and Technology, University of Souk Ahras, P.O. Box 1553, 41000 Souk-Ahras, Algeria
{aida.chefrour, s.drici}@univ-soukahras.dz

Abstract. The work presented in this paper is in the general framework of classification using deep learning and, more precisely, that of convolutional neural networks (CNN). In particular, the convolutional autoencoder proposes an alternative for the processing of high-dimensional data, to facilitate their classification**.** In this paper, we propose the incorporation of convolutional autoencoders as a general unsupervised learning data dimension reduction method for creating robust and compressed feature representations in order to improve CNN performance on image classification tasks. For prediction reasons, we applied the two methods to the MNIST image databases. The use of CNN with the convolutional autoencoder gives better results compared to the individual use of each of them in terms of accuracy, to obtain a good classification of the data high-dimensional entrance.

Keywords: Deep Learning, Convolutional Neural Network, Convolutional AutoEncoder, MNIST.

1 Introduction

Computers are a very important part of this society, but there are still so many things that a human does better, despite their limited storage and computational capacity. Probably one of the most intriguing areas of study is learning, which can be described in many ways, including acquiring new knowledge, improving existing knowledge, representing knowledge, organizing knowledge, and discovering facts through experiments. In addition, the continuous growth of data volume contributes to the improvement of techniques that seek the implicit knowledge of these data[1].

Machine learning (ML) is the application of underlying computational methods to experience-based decision making. ML is one of the most studied tasks nowadays. It is a very important part of Artificial Intelligence and should be one of the main characteristics of intelligent systems. By learning, we can exploit and build models of

reality based on experiences, either by creating a model completely or by modernizing a partially built model. The goals of machine learning are to provide greater solution accuracy, greater problem coverage, greater economy in obtaining solutions, and greater simplicity in representing knowledge.

Machine learning tasks are divided into three types: supervised, unsupervised, and reinforcement learning [2].

The goals of Machine Learning are to provide greater solution accuracy, greater problem coverage, greater economy in obtaining solutions, and greater simplicity in knowledge representation.

Classification techniques represent a very active topic in machine learning. They are frequently found in many fields of application and have become the basic tool for almost all pattern recognition tasks. Several structural and statistical approaches have been proposed to build classification systems from data. Traditionally, the classification system is trained using a training dataset under the supervision of an expert who controls and optimizes the training process. The performance of the system is fundamentally related to the training algorithm and the training dataset used. The latter contains labeled samples of the different classes that must be recognized by the system. In almost all learning algorithms, the training dataset is visited several times in order to improve the classification performance, which is usually measured using a separate test dataset.

In the last decade, there has been a growth in related research on machine learning and a subtype of this, deep learning (DL). This has been achieved by improvements in computational capabilities, storage capacity, and data availability [3]. Several studies have been conducted using deep learning for image recognition, with the architecture of convolutional neural networks being the most promising [4]. CNNs learn important features from images automatically, allowing them to perform classifications with great discriminatory efficiency. The goal of a CNN is to discover the kernel coefficient that minimizes the classification task's error [3].

Due to the emergence of these powerful neural networks, methods based on them have been introduced to learn better data representations and obtain very interesting performance improvements for clustering algorithms. A simple method is to use a Convolutional AutoEncoder (CAE) to learn the representation. Specifically, the high-dimensional features of the original input are fed into the encoder, which generates the low-dimensional output. This output is then passed to the decoder, which attempts to recover as much of the original input data as possible. These methods use images as input and thus use convolutional neural networks in the learning. For this objective, our research subscribes to this context and aims to reduce the high-dimensional image using a CNN before its classification.

The contributions of this paper include:

- The proposition of the Convolutional Autoencoder (CAE), which is a simple but more general representation learning framework, allows us to reduce the dimension of the database (inputs) and to generate a feature vector (minimum dimensional data) before performing the clustering phase by the CNN algorithm (second part) to obtain better results.

- We make use of the well-known and widely used database in the field of image processing: MNIST, in order to evaluate our architecture, which consists of integrating the CNN with the convolutional autoencoder in terms of accuracy.
- The choice of the CNN clustering algorithm was made because it produces compact clusters and processes Convolutional Neural Networks (CNNs) more quickly than other methods. With a large number of training images and several categories, DCNNs trained using backpropagation can perform well on image classification tasks. This is why the CNN classification technique is used to describe image classification.

2 Literature review

Several algorithms for the incorporation of CAE in the CNN exist in the literature. In this section, we outline the best-known and most recent ones. We noticed that all of these algorithms have shown good results in the last few years. However, no one of them could be said to be the best, as they all depend on the content of input parameters and their application domain:

The authors of [5] described the problem of painter classification. They proposed a novel method based on deep convolutional autoencoder neural networks. They trained first a deep convolutional autoencoder on dataset of paintings to find features. Having trained a CAE, they removed the decoder components and they used it for initializing a supervised CNN. We adopt a similar embedding with a convolutional autoencoder based on the CNN classification algorithm, but with a different goal from CAE which is feature extraction in this work.

In the study of [6], the researchers evaluated the impact of CNN pre-training on the network's accuracy outside of samples using CAE. In order to do this, they first trained a convolutional autoencoder on a particular dataset, and then they used its convolution layer weights to initialize the convolution layers of a CNN in a process stage. They compared the CNN's test set accuracy to that of a reference network that performed the same training process but used convolution weights that were initially initialized a random.

[7] proposed a new architecture combining a convolutional autoencoder with a convolutional neural network CAE-CNN. They specifically used a convolutional autoencoder to identify useful features from the positive samples in DNA nucleotides, which was inspired by the image reconstruction. The convolutional neural network use the learned features during the training phase. In order to more effectively capture the features of DNA nucleotides through a gated unit, then also used a highway connecting layer.

[8] proposed a novel method CNN-AE, to predict the survival chance of COVID-19 patients. The dataset's properties were fully investigated to identify essential features and calculate their correlations. To balance the dataset, an autoencoder based data augmentation method was proposed. Thus, the embedding of CNN and AE differs from their work in that it aims to augment data while out performing CNN classification.

In [9], the authors presented a novel deep learning framework for linear systems with time invariant parameters that detects the type and location of faults in sensor data and reconstructs the correct sensor data for fault detection, fault classification, and fault reconstruction. In their approach, the presence of a fault and its type are initially detected using a CNN. Then for construction, a group of individually trained CAE networks corresponding to each type of error is used. In contrast of our approach, we have applied the CAE before CNN.

3 Background

Our proposed approach comprises two modules: dimension reduction and classification. The dimension reduction is realized using CAE. The classification is carried out using a CNN. In this section, we briefly review the main concepts of CAE and CNN.

3.1 CNNs

CNNs are massively used in image-based learning applications. Due to their autonomous feature extraction technique, CNNs can extract useful data from training samples. CNNs are usually created with several convolutional, pooling and fully connected layers. In order to extract features, the input is convolved with convolutional kernels, as shown in Fig. 1. Without significantly changing the feature map's resolution, the pooling layer actually reduces the network's computational complexity. In CNN's, as the number of layers increases, the size of the pooling layers typically falls. Max pooling and average pooling are two of the most used forms of pooling layers [10].

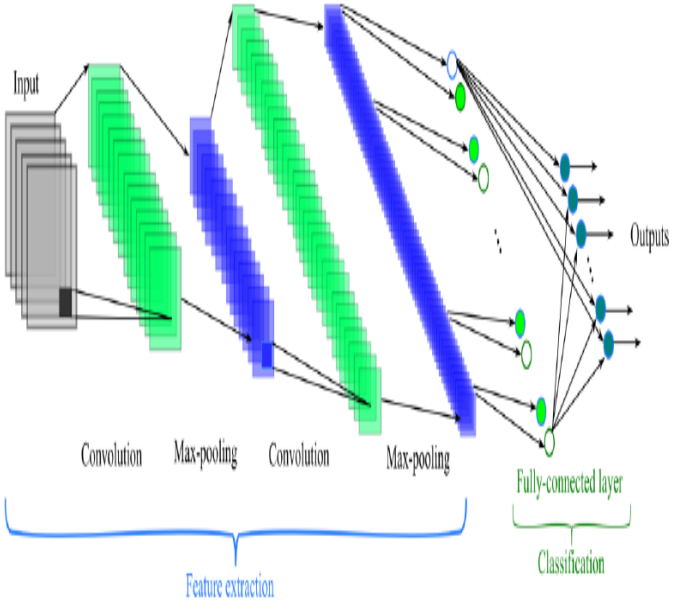

Fig. 1. CNN architecture [8].

3.2 AEs

Although they don't require a training dataset, AEs fit into the category of unsupervised learning. An AE creates a compressed latent space representation of the input data, which then decompresses it to reconstruct the data. In the compression step, AEs carry out dimensionality reduction, which is similar to principal component analysis (PCA) But unlike PCA, which uses linear transformation, AEs use deep neural networks to do linear transformation

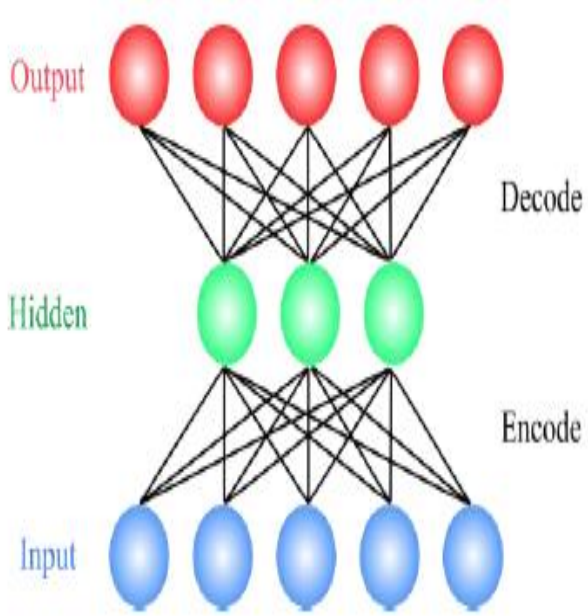

Fig. 2. AE architecture [8].

3.2 CAEs

Convolutional AutoEncoders are unsupervised dimensionality reduction models composed by convolutional layers capable of creating compressed image representations.

In general, CAEs are used to extract robust features, reduce and compress the size of the input dimension, and remove the noise while simultaneously preserving all necessary information.

The use of convolutional layers is the main difference between CAE and traditional AE. It is important to note that these layers are distinguished by their desirable capability of knowledge extraction and internal representation of image data learning.

More specifically, as shown in fig. 3, CAEs are composed of 2 CNN models, the encoder and the decoder. The encoder's principal function is to convert the initial input image into a latent representation with a reduced dimensionality. The decoder, on the other hand, is responsible for rebuilding the compressed latent representation and producing an output image that is as similar to the original as possible.

Fig. 3. CAE architecture.

4 Proposed CAE-CNN classification algorithm

To overcome the limitation of the data representation and the high dimensionality of the dataset and feature extraction, we have developed in this work an embedding of the CAE and CNN (CAE-CNN).

The objective of this proposed approach is the application of deep learning to learn models in order to transform the input data into more user-friendly representations and to reduce the dimensions for classification. The CAE is based on a set of successive transformations that amplify the features of the input data that discriminate against them and attenuate their variations.

The proposed CAE-CNN architecture is shown in fig. 4. The initial training dataset is used to train a CAE. The decoder component is eliminated one the CAE has completed its training process, and the encoder is employed to reduce the size of the original high dimensional image dataset into a compressed image dataset. Finally, the compressed image dataset produced by the CAE's encoder is utilized to feed and train a CNN classification model.

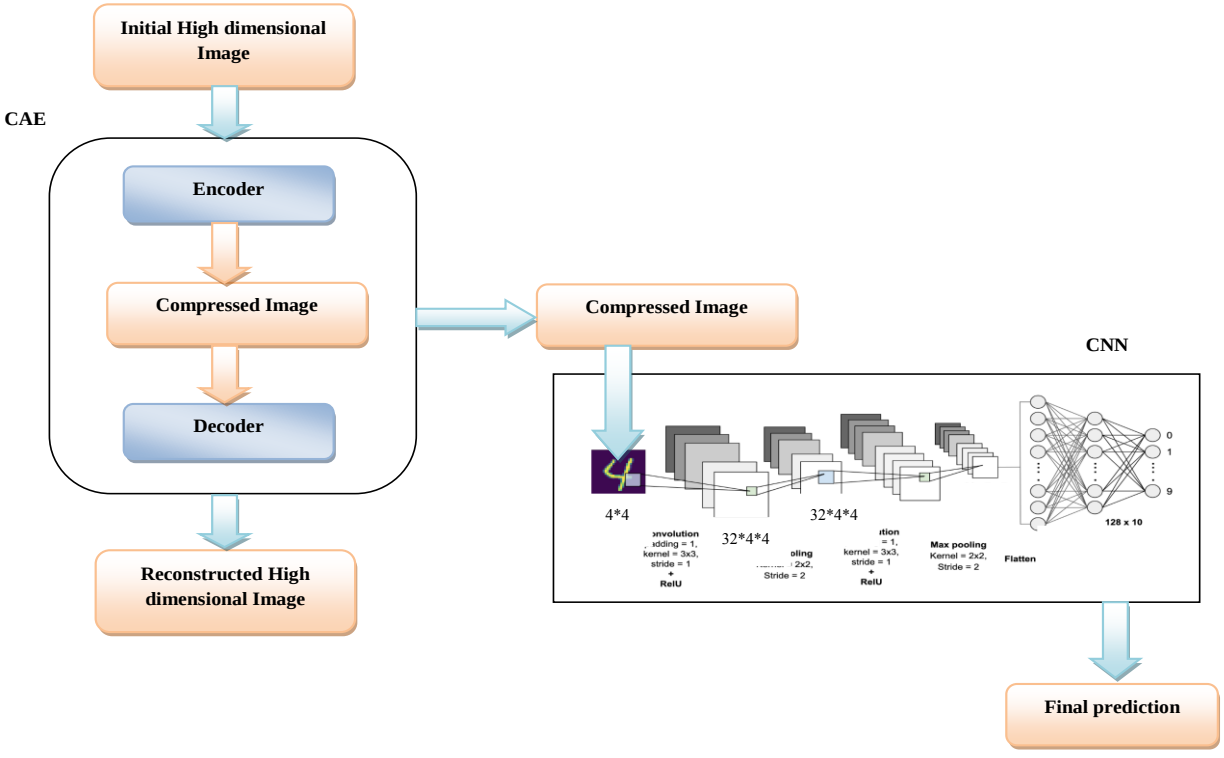

Fig. 4. The proposed architecture of CAE-CNN.

5 CAE-CNN for handwritten digit recognition

For unsupervised training of CAE, we use the MNIST handwritten digit image dataset from the UCI machine repository to validate the accuracy and efficiency of our proposed approach.

The MNIST database has a training set of 60000 examples, and a test set of 10000 examples, with varying resolutions averaged approximately at 28*28 pixels.

The goal is to train the CAE to find features, but here we use the encoder part for compressing the initial high dimensional image dataset into a compressed image dataset.

Our CAE contains the following layers:

1. The input layer consists of the raw image (28*28 pixels);
2. Convolutional layer of size 128*28*28;
3. Maxpooling layer of size 2*2;
4. Convolutional layer of size 64*14*14;
5. Maxpooling layer of size 2*2;
6. Convolutional layer of size 32*7*7;
7. Maxpooling layer of size 2*2;
8. Output Encoder of size 32*4*4;
9. Unpooling layer of size 2*2;
10. Deconvolutional layer of size 32*4*4;
11. Unpooling layer of size 2*2;
12. Deconvolutional layer of size 64*8*8;
13. Unpooling layer of size 2*2;
14. Deconvolutional layer of size 128*14*14;
15. Unpooling layer of size 2*2;
16. Deconvolutional layer of size 128*28*28;

After a CAE has been trained, the decoder components (items 9 to 16 in the list above) can be removed, and the CAE can then be used to initialize a supervised CNN. The softmax activation function is applied.

The following layers are present in the full CNN:

1. The input layer consists of the raw image (28*28 pixels);
2. Convolutional layer of size 32*4*4;
3. Maxpooling layer of size 1*1;
4. We take the dropout to be 0.5 to avoid overfitting. Every hidden unit (neuron) is set to 0 with a probability of 0.5;
5. We use a flatten layer, which reduces the data to a single dimensional array by adding each new row to the previous one. We'll apply one hidden layer with 512 neurons per layer and an output layer with 10 neurons, one for each of the 10 possible digits;
6. Fully connected layer of size 512.

5.1 Performance evaluation parameters

In this subsection, we validate the efficiency and robustness of the proposed approach by performing comprehensive experimental simulations. The measurement of quality is based on the well-known widely used evaluation metrics: Accuracy (Acc), Precision , Recall, and F1-score. These parameters can be calculated using Equations (1, 2, 3, and 4):

$$\text{Precision}=\frac{TP}{TP+FP} \quad (1)$$

$$\text{Recall}=\frac{TP}{TP+FN} \quad (2)$$

$$\text{F1\_score}=\frac{2\times Precision\ \times Recall}{Precision\ +Recall} \quad (3)$$

$$\text{Acc}=\frac{TP+TN}{TP+FP+TN+FN} \quad (4)$$

5.2 Experimental results and discussion

The classification process consists of two steps: the first one performs the dimension reduction with CAE, and the second stage represents the decision-making process for classification using CNN.

We train CAE on the MNIST dataset. We obtain the following model (as shown in the Fig. 5):

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 28, 28, 1)]       0

 conv2d (Conv2D)             (None, 28, 28, 128)       1280

 max_pooling2d (MaxPooling2D  (None, 14, 14, 128)      0
 )

 conv2d_1 (Conv2D)           (None, 14, 14, 64)        73792

 max_pooling2d_1 (MaxPooling  (None, 7, 7, 64)         0
 2D)

 conv2d_2 (Conv2D)           (None, 7, 7, 32)          18464

 CODE (MaxPooling2D)         (None, 4, 4, 32)          0

 conv2d_3 (Conv2D)           (None, 4, 4, 32)          9248

 up_sampling2d (UpSampling2D  (None, 8, 8, 32)         0
 )

 conv2d_4 (Conv2D)           (None, 8, 8, 64)          18496

 up_sampling2d_1 (UpSampling  (None, 16, 16, 64)       0
 2D)
```

Fig. 5. Illustration of CAE

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_16 (Conv2D)          (None, 2, 2, 32)          9248

 max_pooling2d_4 (MaxPooling  (None, 1, 1, 32)         0
 2D)

 dropout (Dropout)           (None, 1, 1, 32)          0

 flatten (Flatten)           (None, 32)                0

 dense (Dense)               (None, 512)               16896

 dense_1 (Dense)             (None, 10)                5130

=================================================================
Total params: 31,274
Trainable params: 31,274
Non-trainable params: 0
```

Fig. 6. Illustration of CNN

In the first part, the model generated by CAE present in Fig. 5 is composed of seven convolution layers, two maxpooling layers, and two fully connected layers.

The input image is of size 28*28. It goes first to the convolution layer, that is composed of 14 filters. Each of our layers of convolution is followed by a function of activation called ReLU, which forces the neurons to return positive values.

The output of the CAE is a reduced-size image of 4*4.

In this training process of CAE, the data is divided into training and test sets. Therefore, 60,000 samples were used to train the CAE model, and the remaining 10,000 samples were used for testing purposes to calculate the accuracy error (as described above). We obtain an accuracy of 81.44% after training the CAE model for 50 epochs. There is still a modest result.

To improve this result:

In the second part, we apply the CNN classification algorithm (see Fig. 6) to the results of the encoder and the feature vector resulting from the previous step to determine which of the images are similar to each other and group them into one of the 10 classes.

The division of the database into learning and testing remains the same. We obtain an accuracy of 96.22 % after training the CAE model for 50 epochs.

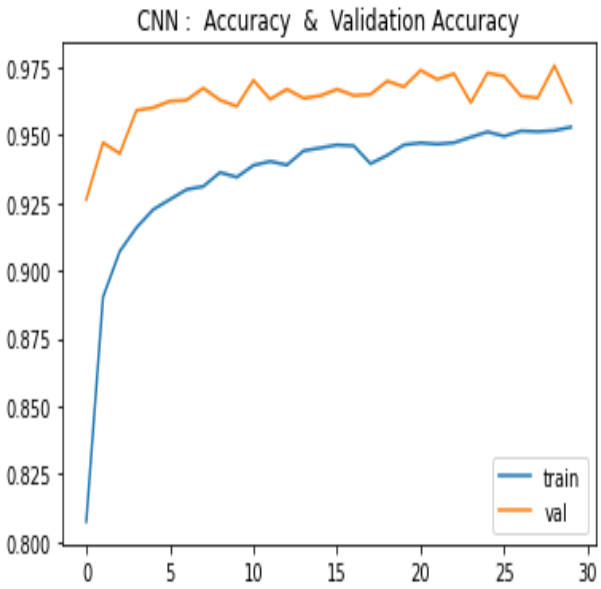

Fig. 7. Accuracy rate obtained from CAE-CNN

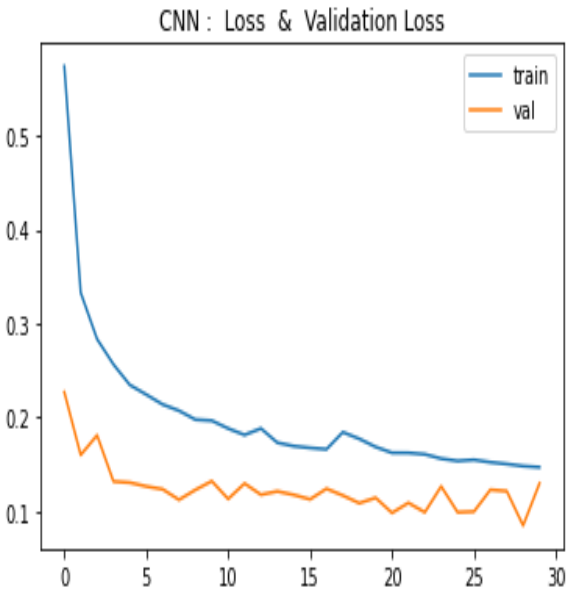

Fig. 8. Error rate obtained from CAE-CNN

We discovered (see Fig.7, Fig. 8) that the number of epochs (epoch=50) increases the accuracy of learning and testing. The result found is good and the model learns more information. Conversely, the error (loss) of learning and testing decreases with the increase of the number of epochs (epoch).

Table 1 summarizes the performance of the proposed approach regarding the MNIST dataset for the 10 classes in terms of the evaluation performance measures.

Table 1. The results of the CAE-CNN.

Classes	Precision	Recall	F1_score	Accuracy
0	0.98	0.99	0.98	
1	0.99	0.98	0.99	
2	0.96	0.98	0.97	
3	0.97	0.95	0.96	
4	0.98	0.96	0.97	**0.96**
5	0.99	0.93	0.96	
6	0.99	0.96	0.98	
7	0.99	0.93	0.96	
8	0.84	0.99	0.91	
9	0.95	0.94	0.95	

To validate the performance of our proposed method, we compared the results with some other partially modified approaches. Table 2 summarizes the performance of the evaluated approaches regarding the MNIST dataset. Notice that each approach has been evaluated utilizing accuracy and the embedding of CAE with a CNN classification model.

Table 2. The results of the evaluation parameters of five different approaches on the MNIST dataset.

Methods	Classification Accuracy (%)
Pre-Training CNNs Using CAE [6]	92.5%
CAE	81.44%
CNN	94.56%
Proposed CAE-CNN	**96.22%**

From the results obtained, we can conclude that the incorporation of convolutional autoencoders as an image preprocessing technique (dimension reduction) could improve the performance of CNN models, leading to robust and accurate results. Therefore, it can be considered as a promising tool for high-dimensional and noisy dataset applications.

6 Conclusion and perspectives

Sometimes, the dimensionality of the input data is very high, and classical learning algorithms cannot provide better performance. To overcome this problem, deep learning algorithms can reduce the dimensionality of the data, such as convolutional neural networks based on the multilayer perceptron.

In this work, we proposed and suggested the incorporation of convolutional autoencoders as a general unsupervised learning data dimension reduction method for creating robust and compressed feature representations in order to improve CNN performance on image classification tasks.

The results presented in this paper show that deep learning methods can be effectively employed for image classification. Our results show that CAEs are capable of extracting meaningful information from digits by dimension reduction, and when combined with supervised CNN, we were able to significantly improve classification accuracy from 96.22%.

Our work opens the way to many perspectives that can be incorporated in the future. Among which we can cite:

- We will also use other classification algorithms, such as: k-means, density-based algorithms, etc;
- With the use of large data sets, we will introduce the notion of incrementality into the database provided to the autoencoder;
- This architecture can also be used in certain application domains, such as handwriting recognition, with very large datasets.

Acknowledgments. The authors would like to thank the DGRSDT (General Directorate of Scientific Research and Technological Development) - MESRS (Ministry of Higher Education and Scientific Research), ALGERIA, for the financial support of LISCO Laboratory, and the LiM Laboratory.

References

1. Jost, I., Valiati, J. F.: Deep Learning Applied on Refined Opinion Review Datasets. Inteligencia Artificial,Vol. 21. (2018) 91-102
2. Abudalfa, S., Mikki. M.: K-means algorithm with a novel distance measure. Turk. J. of. Elect. Eng. Comput. Sci, Vol. 21. (2013) 1665-1684
3. López-Cabrera, J. D., Rodríguez, L. A. L., Pérez-Díaz, M.: Classification of breast cancer from digital mammography using deep learning. Inteligencia Artificial, Vol. 23. (2020) 56-66
4. Dal Prá, B. R., de Mesquita, R. N., de Menezes, M. O., de Andrade, D. A.: Nutritional Evaluation of Brachiaria brizantha cv. marandu using Convolutional Neural Networks. Inteligencia Artificial, Vol. 23. (2020) 85-96
5. David, O. E., Netanyahu, N. S.: Deeppainter: Painter classification using deep convolutional autoencoders. In Int. conf. on. art. neural. net. Springer, Cham (2016) 20-28
6. Kohlbrenner, M., Hofmann, R., Ahmmed, S., Kashef, Y.: Pre-training cnns using convolutional autoencoders. TU Berlin, Berlin, Germany (2017)
7. Zhang, Y., Qiao, S., Zeng, Y., Gao, D., Han, N., Zhou, J.: CAE-CNN: Predicting transcription factor binding site with convolutional autoencoder and convolutional neural network. Exp. Syst. with. Appli, Vol. 183. (2021) 115404
8. Khozeimeh, F., Sharifrazi, D., Izadi, N. H., Joloudari, J. H., Shoeibi, A., Alizadehsani, R., ... Islam, S. M. S.: Combining a convolutional neural network with autoencoders to predict the survival chance of COVID-19 patients. Sci. Reports, Vol. 11. (2021) 1-18
9. Jana, D., Patil, J., Herkal, S., Nagarajaiah, S., Duenas-Osorio, L.: CNN and Convolutional Autoencoder (CAE) based real-time sensor fault detection, localization, and correction. Mech. Syst. Sign. Proces, Vol. 169. (2022) 108723
10. Khodatars, M., Shoeibi, A., Sadeghi, D., Ghaasemi, N., Jafari, M., Moridian, P., ...;Berk, M.: Deep learning for neuroimaging-based diagnosis and rehabilitation of autism spectrum disorder: a review. Comp. in. Bio. and. Med., Vol. 139. (2021) 104949