International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

September 2021 | Issue 48 | Volume 13 | Number 3 | Pages 1-6

GENERALIZATION ABILITY AUGMENTATION AND REGULARIZATION OF DEEP CONVOLUTIONAL NEURAL NETWORKS USING $l^{1/2}$ POOLING

E.H. Hssayni M. Ettaouil

Modelling and Mathematical Strictures Laboratory, Faculty of Sciences and Techniques, Sidi Mohamed Ben Abdellah University, Fez, Morocco, elhoussaine.hssayni@usmba.ac.ma, mohamed.ettaouil@usmba.ac.ma

Abstract- As one of the most used machine learning techniques, deep convolutional neural networks (DCNNs) have demonstrated remarkable performance in various challenging artificial intelligence and machine learning tasks, furthermore, pulled in extensive interests lately. A pooling mechanism assumes a significant part in CNN, which serves to diminish the dimensionality of prepared data for diminishing computational expense just as for preventing overfitting and improving the network generalization capacity. Whereas classical pooling techniques, including the max and the l^p pooling (where $p \geq 1$) are technically proposed in various studies, we alternatively propose, in this paper, a now pooling technique named $l^{1/2}$ pooling in order to improve the generalization capability of CNNs. Test results on two image benchmarks indicates that $l^{1/2}$ pooling outflanks the current pooling methods in characterization execution just as is efficient for improving the generalization capability of CNNs. Also, we prove that the $l^{1/2}$ pooling combined with other regularization techniques, such as dropout and batch normalization, achieved significant results in terms of improving the generalization capacity.

Keywords: Deep Learning, Convolutional Neural Networks, Pooling Methods, Generalization Capacity, Regularization Methods.

1. INTRODUCTION

Nowadays, deep learning models have evolved rapidly and attracting a great deal of attention because of their achievement in various areas, for example, pattern recognition, object detection, time series classification, and many other domains such as genomics and drug discovery. Its outstanding and high capability to learn extremely complex relationships directly from the data makes DNN consummately appropriate to carry out intelligent tasks successfully, similar to those performed by the human brain.

There are many deep neural network architectures [1] such as, convolutional neural networks and deep restricted Boltzmann machine.

As of late, Convolutional neural organization (CNN), one of the famous and incredible profound learning models which at first introduced for PC vision task [2], has continually indicated confident outcomes and critical improvement in large areas and applications, for example pattern-recognition [3], detection and classification of covid-19 [4], and time series classification (t_{sc}) [5].

It is well known that the feedforward neural networks consist of input, output, and several hidden layers. The CNNs have the same architecture, but with a particular structure. In these models, the input image is convoluted with trained kernels for extracting the features. The size of these layers is reduced using a special layer called a pooling layer. The obtained feature maps are presented as inputs for the next convolution. This process continues until deep features are extracted and presented finally as inputs for a fully connected classifier that can end up with a Softmax output layer.

In this paper, we focus on the feature extraction part and especially on the pooling layer, which plays a crucial role in CNN since it is chiefly answerable for the invariance to data variation and perturbation [6]. The pooling layer is acquired by applying pooling method to assemble data inside every small region of the input feature maps and then down-sampling the results [7].

Some past works have interested on pooling techniques intend to achieve spatial invariance by lessening the feature maps resolution, for enhancing the computational effectuation of CNN, and obtain large increases in performance. As well as reducing the overfitting problem. The most standard and popular pooling technique is the average pooling [10], which takes in consideration uniformly all the units in the pooling locale, and takes the arithmetic mean of the elements in each pooling locale. Be that as it may, when joined with the widely used activation function ReLU, it can take many 0 units into account and down weight strong units [9]. In order to avoid these downsides, various improved pooling techniques have been introduced. These techniques include stochastic pooling [9] which takes the units inside every pooling region dependent on the standardized likelihood of activation values. Indeed, in the training phase, it initially appoints a likelihood to every component in the pooling region. This strategy has demonstrated a superior speculation capability compared with the average and the

max pooling for many datasets. The hybrid pooling [11] is another improved pooling method based on choosing randomly good pooling method within each pooling layer.

A successful variant of pooling methods is l^p pooling [12], which calculates the output of the pooling region using an l^p norm with $p \geq 1$, and can be regarded as a generalized pooling technique counting the average pooling (for $p = 1$) and the max pooling (for $p = +\infty$) as special cases. Moreover, by inspiring to the regularization theory, which affirm that an l^p regularization method produces a sparse solution only when ($0 \leq p \leq 1$), and the smaller the p, the sparser the solution [13], and according to the work [14], we introduce a new polling technique named $l^{1/2}$ pooling, which uses the absolutely homogeneous function $||.||_{1/2}$ to calculate the output of the pooling region. To additionally legitimize the introduced technique, we perform experiments on two classification datasets: MNIST [2] and Fashion-MNIST [15]. The obtained results outline that $l^{1/2}$ pooling beats the current pooling strategies in terms of accuracy and that is viable for improving the speculation capacity of CNN.

In summary, main points of this work are as follows:

• We introduce a new simple, yet effective pooling technique named $l^{1/2}$ pooling to improve the CNNs generalization capability.

• We approve and show the efficiency of our technique on two benchmark image datasets.

• By combining our technique and other regularization techniques, including dropout and batch-normalization, we have achieved the previous accuracy with moderate parameters on two image benchmarks.

The remainder of the present article is organized as follows: Section 2 presents some DNN regularization strategies. The third Section presents the pooling process used in CNN and introduces the most popular pooling methods. In section 4 we introduce our proposed method.

In section 5, details the experiments and shows the effectiveness of $l^{1/2}$ pooling through several numerical results. Furthermore, combining $l^{1/2}$ pooling with other regularization techniques, including dropout and batch-normalization, was adopted to demonstrate the effectiveness of $l^{1/2}$ pooling in terms of generalization capability. The last section presents the conclusion and some perspectives of our study.

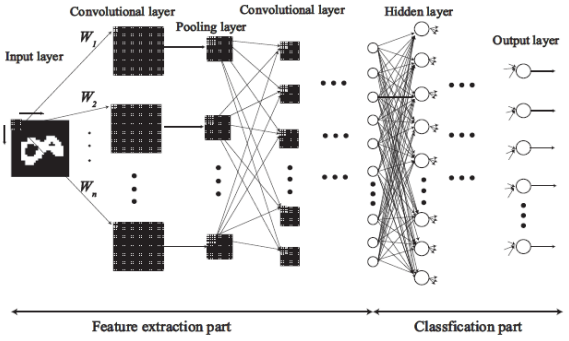

Figure 1. A standard CNN architecture

2. DNNs REGULARIZATION

Recently, the most successful deep neural networks often require a large number of parameters, which leads to a significant amount of memory and a higher computational cost. This may produce some undesirable phenomena, notably overfitting. In this context, several approaches have been adopted to combat this challenging issue. This methods includes Batch Normalization [17], DropConnect [16], and Dropout [2]. In the next paragraphs of this section, we describe these methods.

2.1. Dropout

As one of the famous approaches proposed to combat the overfitting problem, Dropout [2] demonstrated good effectiveness in preventing the overfitting problem in DNNs. This famous approach is based on sampling randomly a vector m using Bernoulli distribution with a given probability p. This latter represents a hyperparameter in the DNNs.

$$m_i \sim Bernoulli(p) \quad (1)$$

The generated vector m is then used to generate the output as follow:

$$a_l = m \odot f(W.a_{l-1} + b_l) \quad (2)$$

where, $\odot$ represents the entrywise product, $a_{l-1} \in \mathbb{R}^m$ and $b_l \in \mathbb{R}^n$ represent the previous layer outputs and the current layer biases respectively, $W \in \mathbb{R}^{n \times m}$ are the dropout layer weight and $f(.)$ represents the non linarity activation function.

2.2. DropConnect

As generally known from the literature, DropConnect [16] is another popular and powerful regularization technique that introduces a dynamic sparsity within the network by dropping out a set of connections.

The expression of a layer output after applying dropconnect technique is given by the following formula:

$$a_l = f((M \odot W).a_{l-1} + b_l) \quad (3)$$

where, each element $m_{i,j}$ of the matrix M is given by the following expression:

$$M_{ij} \sim Bernoulli(p) \quad (4)$$

2.3. Data Augmentation

This technique is well known to a great extent used to enhance the training of CNN. It comprises falsely grow the dataset utilizing label preserving changes. It was achieved better performance and significant results in terms of generalization capability improving [3].

2.4. Batch Normalization

Batch normalization [17] essentially lessens preparing time and improves the speculation capacity by decreasing Internal Covariate Shift. In the learning task, usually, the data are normalized for the purpose of making them comparable in terms of the features. However, there is a change in the distribution of network activations due to the change in network parameters during training.

In this context, batch normalization (BN) is one of the successful approaches that greatly improve convergence during training. It consists in normalizing on average and invariance the outputs of the layers of the network.

Batch normalization has indicated that is viable for expanding the speculation capacity of CNN, just as for keeping away from the overfitting issue.

3. POOLING MECANISM

Pooling is a fundamental part in CNNs which serves to prevent overfitting, just as to improve the network performance [2]. A standard feature map is composed of a set of units and a pooling activity is iterated for every subset of the component map, called a pooling region. The subsampling technique is a significant procedure in CNN for removing important highlights in a given element map. The pooling technique serve to decide the output s_k of a pooling region R_k with $k=1,...,K$, and R_k is given as $\{a_1, a_2, ...a_{|R_k|}\}$ where $|R_k|$ represents the number of activations within the element map R_k. Then the pooling feature map $S=\{s_1, s_2, ..., s_k\}$ is acquired by gathering the yields of the total pooling regions. In the following subsections, we briefly survey the existing subsampling strategies.

3.1. Max Pooling

Max Pooling [8] is the famous subsampling technique, which takes the biggest actuation in the pooling element map as presented in the following equation:

$$s_k = \max_{a_i \in R_k} a_i \qquad \text{for} \quad k=1,...,K \tag{5}$$

Max Pooling is pointed toward extricating nearby highlights, for example, lines, edges, and surfaces in the pooling element map.

3.2. Average Pooling

Average Pooling [10] puts the arithmetic mean value of units in the subsampling element map as presented in the following equation:

$$s_k = \frac{1}{|R_k|} \sum_{a_i \in R_k} a_i \qquad \text{for} \quad k=1,...,K \tag{6}$$

This pooling technique can remove worldwide features by smoothing the subsampling region.

3.3. Stochastic Pooling

Stochastic Pooling [9] is an enhanced subsampling strategy, which takes the units inside every pooling region dependent on the standardized likelihood of activation values.

In the training phase, it initially appoints a likelihood to every component in the pooling region as presented in the following equation:

$$p_j = \frac{a_j}{\sum_{a_i \in R_j} a_i} \qquad \text{for} \qquad j=1,...,|R_k| \tag{7}$$

Then the output is given by randomly sampling from a distribution decided with the probabilities p_j for $j=1,...,|R_k|$ as following:

$$s_k = a_l \qquad \text{where} \qquad l \sim P(p_1,...,p_{|R_k|}) \tag{8}$$

In the testing stage, rather than the ordinary averaging, the probabilistic type of averaging is received, as surrendered in Equation (9).

$$s_k = \sum_{a_j \in R_k} p_j \times a_j \qquad \text{for} \quad k=1,...,K \tag{9}$$

This technique is less inclined to overfitting because of the stochastic methodology utilized and beats the max and average pooling [10].

3.4. Hybrid Pooling

Hybrid-Pooling [11] represents an enhanced subsampling method, which associates a probability to use the adequate pooling method for every input map, where the chosen pooling method is used for all the pooling regions in the same map.

In the training stage, for every output map of the convolutional layer, the authors in [11] choose the adequate pooling method in each pooling layer based on the following formula:

$$S = \begin{cases} S_{avg} & \text{with probability } p \\ S_{\max} & \text{with probability } 1-p \end{cases} \tag{10}$$

where, $S_{avg} = \{s_1^{avg}, ..., s_K^{avg}\}$ with $s_k^{avg} = \frac{1}{|R_k|} \sum_{a_i \in R_k} a_i$ for $k=1,...,K$ and $S_{\max} = \{s_1^{\max}, ..., s_K^{\max}\}$ with $s_k^{\max} = \max_{a_i \in R_k} a_i$.

In the testing phase, every pooling region output is obtained by expecting the value of S in the previous equation, which is given as following:

$$S = S_{hyb} = p\, S_{avg} + (1-p)\, S_{\max} \tag{11}$$

This strategy has demonstrated that is viable for expanding the generalization capacity of CNNs.

3.5. l^p Pooling

The l^p Pooling [12] is a successful variant of pooling which can be observed as a continuous parametrization transition from average pooling to max pooling. This technique is a pooling method using the l^p norm with $p \geq 1$, given in the following expression:

$$s_k = \frac{1}{|R_k|} \left(\sum_{a_i \in R_k} {a_i}^p \right)^{\frac{1}{p}} \qquad \text{for} \quad k=1,...,K \tag{12}$$

Two uncommon instances of l^p pooling are outstanding ($p=1$) presents the average pooling, and ($p=+\infty$) presents the max pooling.

The l^p pooling has appeared to give enormous enhancements in terms of accuracy in computer vision tasks contrasted with max pooling [12].

4. PROPOSED METHOD $l^{1/2}$ POOLING

In order to improve the generalization capability of deep CNN as well as to avoid overfitting on the one hand, what's more, because of the critical role of pooling mechanism in CNN and the successful of l^p pooling method on the second hand, and by inspiring to the regularization theory, which affirm that an l^p regularization method produces a sparse solution only when ($0 \le p \le 1$), and the smaller the p, the sparser the solution [13], and according to the work [14], we propose a new polling method named $l^{1/2}$ pooling, which use the absolutely homogeneous function $||.||_{1/2}$ to calculate the output of the pooling region.

This pooling technique serves to decide the output s_k of the pooling region R_k for $k = 1,...,K$ as follows:

$$s_k = \frac{1}{|R_k|}\left(\sum_{a_i \in R_k} \sqrt{|a_i|}\right)^2 \quad \text{for} \quad k = 1,...,K \tag{13}$$

where, R_k and $|R_k|$ are defined as in the previous section.

5. EXPERIMENTS RESULTS

To legitimize the viability of $l^{1/2}$ pooling strategy, we lead various experiments on two standard datasets: the first one is MNIST [2] and the second is Fashion-MNIST [15]. The structures of these datasets are portrayed beneath.

5.1. Used Dataset

The **MNIST** [2] is a standard and widely adopted dataset for classification tasks on handwritten digits. It is composed totally of 70,000 samples, among which 60,000 samples are utilised for training and the remaining 10,000 images for testing. All pictures comprising of 28×28 pixels, every one of which has 256 pixel esteems relating to grayscale shading powers. Basically, the assignment is to group the pictures into 10 digit classes (0-9). Figure 2 shows a sample of the MNIST database. This database comprises a benchmark of the approval for mostly proposed profound learning approaches. For this reason, we initially justify the validity of our approach using this dataset

Figure 2. Sample handwritten digits of the MNIST database [2]

Fashion-MNIST presents a new image-database, as of late introduced by [15] which is planned to fill in as an immediate drop-in trade for the first MNIST database. The Fashion-MNIST database has all the quality of the first MNIST's, which is made out of 28×28 pixel of grayscale style thumbnail pictures with 60,000 preparing and 10,000 test samples. The assignment is to group the pictures into 10 classifications of design items. Figure.3 indicates some examples from Fashion-mnist database.

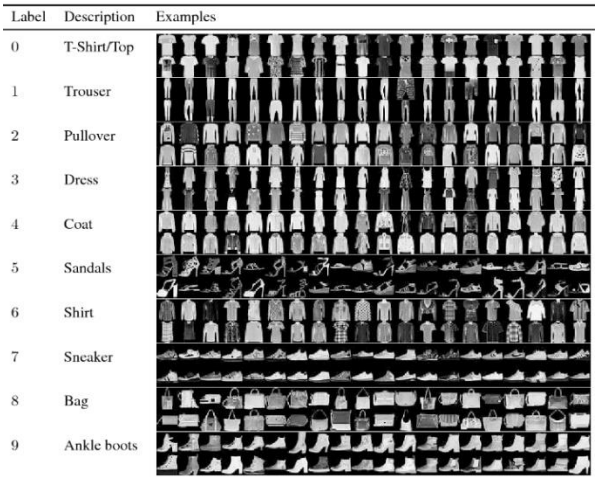

Figure 3. Some examples and their corresponding class names from Fashion-MNIST database [15]

5.2. Simulation Methods

In the practice evaluation of our model, we opt for the framework Caffe [12] to demonstrate the effectiveness of our approach. Regarding the activation function, we have used The ReLU function for all the layers except the output layer in which we have used the softmax function. The training phase was performed using mini-batch stochastic GD algorithm, with the batch size equal 128. The other hyperparameters used in this experiments are cited in the following table (Table 1).

Table 1. hyperparameters values used in the experiments

Hyperparameter	Value
momentum	0.9
weight decay	0.0004
kernel size	5×5
pooling region size	3×3
learning rate	0.01 for the first 130 iterations, 0.001 for the next 40 iterations, 0.0001 for the last 20 iteration.
bias decay	0.0004

5.3. Results

In this subsection, we present and discuss the obtained results. In order to evaluate the performance of $l^{1/2}$ Pooling we have compared the accuracy which represents the proportion of all instances which were classified correctly and the training time using different pooling techniques. Finally, we have combined $l^{1/2}$ Pooling with dropout in case of MNIST classification and with Batch normalization in case of Fashion-MNIST classification in order to achieve better performance.

For the MNIST classification, the got results are introduced in the accompanying table (Table 2).

Table 2. Training-time (S) and performance accuracies (%) on MNIST classification

Pooling technique	Prediction accuracy	Training-time
Max-Pooling	90.09	75.7
Average-Pooling	90.18	75.16
$l^{1/2}$ Pooling	92.16	75.11
$l^{1/2}$ Pooling + Dropout	93.24	78.52

As detailed in Table 2, we see that $l^{1/2}$ pooling accomplishes a high forecast precision of 92.16% on the MNIST dataset. What's more, when it joined with dropout, it accomplishes a high forecast accuracy of 93.24% with a marginal increase in training time.

Table 3. Training-time (S) and performance accuracies (%) on Fashion-MNIST

Pooling method	Prediction accuracy	Training-time
Max Pooling	88.72	164.7
Average Pooling	88.34	164.16
$l^{1/2}$ Pooling	90.16	164.23
$l^{1/2}$ Pooling + Batch Normalization	91.57	167.52

As detailed in Table 3, we see that $l^{1/2}$ pooling accomplishes a high forecast precision of 90.16% on the Fashion-MNIST dataset. What's more, when it joined with Batch Normalization, it accomplishes a high forecast accuracy of 91.57% with a marginal increase in training time.

Experiments show that our method can improve performance in terms of classification efficiency and network training speed.

6. CONCLUSION

Based on the important effect of the pooling mechanism in CNNs and the efficiency due to the previously proposed l^p pooling method (for $p \geq 1$) in improving the generalization capacity of Deep Convolutional Neural Networks, we have alternatively introduced in this paper a new pooling method called $l^{1/2}$ pooling method which uses the absolutely homogeneous function $||.||_{1/2}$ to calculate the output of the pooling region. Then Experimental results show that $l^{1/2}$ pooling method is superior to some existing pooling methods on a range of datasets. Although we have shown the effectiveness of the $l^{1/2}$ pooling using two benchmark image datasets: MNIST and Fashion-MNIST, it would be significant to further verify it in other challenging datasets and application tasks such as time series forecasting.

REFERENCES

[1] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F. E. Alsaadi, "A survey of deep neural network architectures and their applications", Neurocomputing, vol. 234, pp. 11-26, 2017.

[2] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition", Proceedings of IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.

[3] A. Krizhevsky, I. Sutskever, G.E. Hinton, "ImageNet classification with deep convolutional neural networks", Advances in neural information processing systems, pp. 1097-1105, 2012.

[4] S. Thakur, A. Kumar, "X-ray and CT-scan-based automated detection and classification of covid-19 using convolutional neural networks (CNN)", Biomedical Signal Processing and Control, Vol. 69, 102920, 2021.

[5] W. Chen, K. Shi, "A deep learning framework for time series classification using relative position matrix and convolutional neural network", Neurocomputing, vol. 359, pp. 384-394, 2019.

[6] D. Scherer, A. Muller, S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition", International conference on artificial neural networks, Springer, pp. 92-101, 2010.

[7] M. Sun, Z. Song, X. Jiang, J. Pan, Y. Pang, "Learning pooling for convolutional neural network", Neurocomputing, vol. 224, pp. 96-104, 2017.

[8] M. Ranzato, Y.L. Boureau, Y. LeCun, "Sparse feature learning for deep belief networks", in Advances in neural information processing systems, pp. 1185-1192, 2008.

[9] M.D. Zeiler, R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks", arXiv preprint arXiv: 1301.3557, 2013.

[10] Y. LeCun, B.E. Boser, J.S. Denker, D. Henderson, R.E. Howard, W.E. Hubbard, L.D. Jackel, "Handwritten digit recognition with a backpropagation network", in Advances in neural information processing systems, pp. 396-404, 1990.

[11] Z. Tong, G. Tanaka, "Hybrid pooling for enhancement of generalization ability in deep convolutional neural networks", Neurocomputing, vol. 333, pp. 76-85, 2019.

[12] P. Sermanet, S. Chintala, Y. LeCun, "Convolutional neural networks applied to house numbers digit classification", IEEE 21st International Conference on Pattern Recognition (ICPR2012), pp. 3288-3291, 2012.

[13] T. Zhang, "Analysis of multi-stage convex relaxation for sparse regularization", Journal of Machine Learning Research, vol. 11, pp. 1081-1107, 2010.

[14] W. Wu, Q. Fan, J.M. Zurada, J. Wang, D. Yang, Y. Liu, "Batch gradient method with smoothing l1/2 regularization for training of feedforward neural networks", Neural Networks, vol. 50, pp. 72-78, 2014.

[15] H. Xiao, K. Rasul, R. Vollgraf, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms", arXiv preprint arXiv: 1708.07747, 2017.

[16] L. Wan, M. Zeiler, S. Zhang, Y. LeCun, R. Fergus, "Regularization of neural networks using dropconnect", international conference on machine learning, pp. 1058-1066, 2013.

[17] S. Ioffe, C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", arXiv preprint arXiv: 1502.03167, 2015.

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: a simple way to prevent

neural networks from overfitting", The journal of machine learning research, vol. 15, no. 1, pp. 1929-1958, 2014.
[19] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, "Caffe: Convolutional architecture for fast feature embedding", The 22nd ACM international conference on Multimedia, pp. 675-678, 2014.
[20] R. Beaulah Jeyavathana, "Deep Learning Applications and Challenges", International Journal of Control and Automation, Vol. 13, No. 2s, pp. 79-83, 2020.
[21] E.H. Hssayni, M. Ettaouil, "New approach to face recognition using co-occurrence matrix and bayesian neural networks", IEEE 6th International Conference on Optimization and Applications (ICOA), pp. 1-5, 2020.
[22] E.H. Hssayni, M. Ettaouil, "Regularization of deep neural networks with average pooling dropout", Journal of Advanced Research in Dynamical and Control Systems, Vol. 12, No. 04-sp, pp.1720-1726, 2020.
[23] E.H. Hssayni, M. Ettaouil, "A novel pooling method for regularization of deep neural networks", International Conference on Intelligent Systems and Computer Vision (ISCV), pp. 1-6, 2020.
[24] B. Jabir, N. Falih, "Big data analytics opportunities and challenges for the smart enterprise", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 47, Vol. 13, No. 2, pp. 20-26, June 2021.
[25] S. Abdufattokhov, K. Ibragimova, M. Khaydarova, A. Abdurakhmanov, "Data-driven finite horizon control based on gaussian processes and its application to building climate control", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 47, Vol. 13, No. 2, pp. 40-47, June 2021.

BIOGRAPHIES

El Houssaine Hssayni received his Master degree in Mathematics and Applications from Faculty of Sciences and Techniques, University of Sidi Mohamed Ben Abdellah (USMBA), Fez, Morocco. He is a Ph.D. student in Laboratory of Modelling and Mathematical Structures at the same faculty. He works on deep neural networks, regularization problems, hyperparameters and architectures optimization, statistical learning methods and applications.

Mohamed Ettaouil is a Doctorate Status in Operational Research and Optimization, FST University Sidi Mohamed Ben Abdellah (USMBA), Fez, Morocco. He received Ph.D. degree in Computer Science from University of Paris 13, Galilee Institute, Paris, France. He is a Professor at the Faculty of Science and Technology of Fez FST, and he was responsible for research team in modelization and pattern recognition, operational research and global optimization methods. He was the Director of Unit Formation and Research UFR: Scientific computing and computer science, engineering sciences. He is also a responsible for research team in artificial neural networks and learning, modelization and engineering sciences, FST Fez. He is an expert in the fields of the modelization and optimization, engineering sciences.