RESEARCH ARTICLE OPEN ACCESS

Traffic Sign Recognition and classification using Convolutional Neural Networks

Kenzari Zouleykha [1], Boulekouirat Sabrina [2], Abbas Fayçal [3], Bekhouche Abdelali [4]
Department of Computer Science
University of Abbes Laghrour khenchela
Algeria

ABSTRACT
In the field of computer vision the problem of recognition and classification is imposed, the Convolutional Neural Networks (CNN) are a new concept for the deep learning way especially recognition and classification of images and videos, natural language processing and other applications. In this paper we focus on the recognition and classification of traffic sign, Our model operates in two stages, A data processing step to simplify the feature extraction step, the second step is to apply a Convolutional Neural Network to recognize and classify the traffic signs, our method has achieved a low error rate and reached an accuracy of 99.6%.
Keywords :— Deep learning, convolutional neural networks, recognition and classification of Traffic Signs.

I. INTRODUCTION

The recognition of traffic signs is a very important operation for the help and safety of drivers, especially for autonomous cars, the role of a human being would be reduced to simply providing the travel endpoints and some additional requirements, It is very difficult to recognize traffic sign boards because of a lack of visibility such as illumination, objects overlap as well as difficult meteorological conditions.
Our approach focuses on the use of the technique of convolutional neural network noted CNN, which is currently the best performing model for object recognition and classification in the computer vision domain.
We propose in this work a new approach based on the recognition and classification of traffic signs, it will allow us to identify traffic signs in different meteorological conditions. Our model has proved its efficiency because it reaches a higher precision with a low error rate.
The rest of the research is organized as follows: The first section will discuss related work on the recognition and classification of traffic signs. The description of our model will be presented in the second section. In the third section we will present the results obtained and finally we will conclude this work with a conclusion and we will propose some perspectives for future works.

II. PREVIOUS WORKS

Convolutional networks were introduced for the first time by Fukushima [1] and derived a hierarchical nervous network architecture inspired by Hubel's research [2]. Lecun [3] proposed generalized architectures for classifying digits successfully and for recognizing LeNet-5 handwritten control numbers. Ciresan [4] used convolutional networks and performed best in the literature for multi-object recognition of multiple image databases: MNIST, NORB, HWDB1.0, CIFAR10 and the dataset IMAGEnet. due to the application of the CNN, new methods have been developed by Qian, R. et al [5], To perform an MPPS (max pooling position) recognition as a tool to predict class classifications, MPPS explains the properties between layers based on (GTSRB) database, and improves classification and speed, this has led to an increase in accuracy. Jorge Enrique Zafra et al [6] presents two neural network algorithms for recognition of traffic signs boards using Backpropagation and convolutional neural networks. In training times the Backpropagation network is much faster compared to convolutional neural networks but the accuracy is very high which is equal to 80% and the random precision of Backpropagation which is between 50% and 98%. Lim K, et al [7] presents a real-time traffic sign recognition method based on a graphics processing unit applied to a dataset from Germany and South Korea, They proposed a powerful method against changes of illumination, and they subsequently performed region recognition using a hierarchical model. This method has produced stable results in low illumination environments. The hierarchy of the model was carried out in real time, the proposed method obtained a score of 0, 97 accuracy. A study was presented by Qian R et al [8] to compare four methods, these methods are divided into two categories; the first category is constituted of two descriptors, HP and HOG, the second category is constituted of two classifiers MLP and SVM. The study concluded that the HP-SVM method offers competitive performance in terms of the accuracy and processing time of traffic signs recognition. A recognition system was proposed by Reinders C et al [9] where the advantages of convolutional neural networks and random forests were combined to construct a fully convolutive network for the prediction of inclusive boxes.

III. OUR MODEL

The convolutional neural networks is based on a layer-based architecture, which will undergo a succession of different processes in order to extract the important characteristics of the image. As you cross the network; the layers will be compressed, each convolution layer is followed by an activation layer (RELU), then these characteristics will be transmitted to an integrally connected neuron network in order to perform a recognition phase or image classification.

A. *Pretreatments of data*

This step is essential to improve the performance of our model. Fig. 1 shows the pretreatments applied to our images.
The image processing layer is an optional pretreatment layer of predefined filters that are held fixed during learning. We first apply three pretreatment steps to our images, in the following figure:

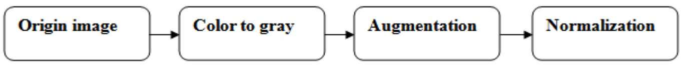


Fig. 1 Data preprocessing.

1) ***Grayscale image:*** image is represented as a matrix of pixels. The grayscale digital image is in the form of a two-dimensional array, while a color image is in the form of a three-dimensional array. RGB color is converted to grayscale to increase accuracy and minimize execution time. In our model we will use an RGB image (32, 32.3) and we will transform thereafter into grayscale (32, 32.1).

Fig. 2 Converting the image to grayscale.

2) ***Normalization:*** Normalization consists in putting the training on the same scale, but there are normalization techniques, namely Min-Max, Zscore [10.11].

The method used for data normalization is based on the following formula:

$$X = \frac{\text{Pixels}-128}{128}$$

• X: the standard variants.
• 128: is a constant to normalize the pixel intensity for images whose pixel intensity value is fixed in the domain [0.255].

3) ***Augmentation:*** consists in increasing the number of images composing the training by carrying out a set of operations which are: translation, scale, rotation and contrast, all this allows our model to be generalized and to give good predictions to the problems due to the operations applied on our images (the traffic signs), the Fig. 3 shows some of the images generated with increase.

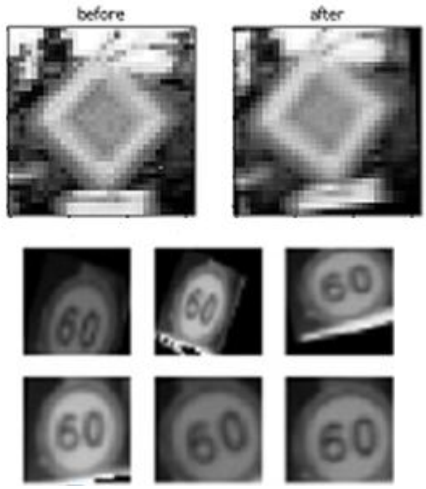
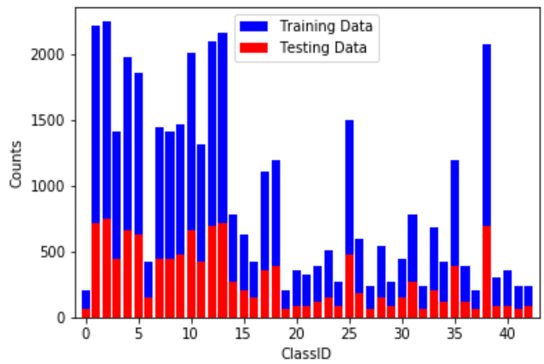
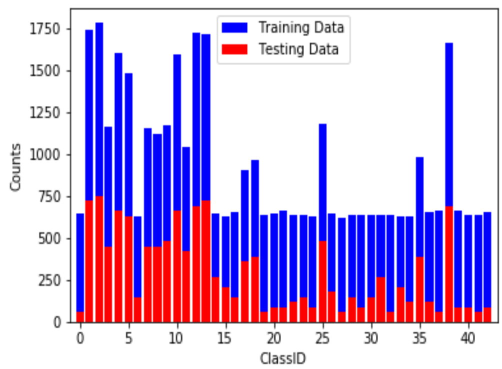


Fig. 3 Data augmentation.

The purpose of the distribution graph is to show the number of continuous images in our dataset. Fig. 4 shows the graphical representation of the distribution of the training game before and after the pretreatment step.



(a)



(b)

Fig. 4 (a) Dataset before the data augmentation, (b) data set after the data augmentation.

B. *Architecture*

The convolutional neural network architecture used in our model is composed of four convolutional layers.

The convolution layer is the first layer; it is parameterized by the size, the number of cards, the kernel sizes and the connection table.
The convolutional layer formula is defined as follows [12]:

$$W_x^n = \frac{W_x^{n-1} - F_x^n + 2P}{S_x^n + 1} + 1$$
$$H_y^n = \frac{H_y^{n-1} - F_y^n + 2P}{S_y^n + 1} + 1$$

- W_x^{n-1}, H_y^{n-1}: the size of the input volume.
- F_y^n: The spatial size of the output volume.
- P: the size of the margin (zero padding).
- S_y^n : stride.
- W_x^n, H_y^n : The size of the output volume and The index *n* indicates the layer.

The latter will be representing an image smaller than the size of the convolution layer.
Fig. 5 shows the calculation of the output volume size in a convolutional layer.

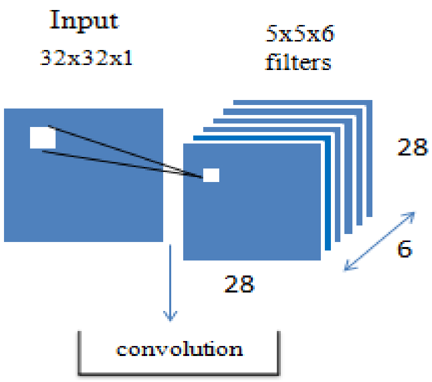


Fig. 5 Calculation of the output volume size in a convolutional layer.

After the convolution layer, we must apply a filter layer (convolution kernels), the latter will represent an image size smaller than the size of the convolution layer; this layer will represent the weights of the neurons. Then we will apply a pooling layer, it allows oversampling the image to minimize its size, it allows us to reduce the calculation time and memory space and the risk of over-fitting. Each neuron in this layer is connected to the neuron of the previous layer.
Fig. 6 shows the over-sampling by applying a 4 × 4 size max-pooling layer.

2	3	5	3
4	7	8	1
3	3	9	8
4	2	4	7

7	8
4	9

Fig. 6 Max-pooling.

The formulas below illustrate the calculation of the pooling size:

$$W_x^n = \frac{W_x^{n-1} - F_x^n}{S_x^n + 1} \quad , \quad H_y^n = \frac{H_y^{n-1} - F_y^n}{S_y^n + 1}$$

- W_x^{n-1}, H_y^{n-1}: the size of the input volume.
- F_y^n: The spatial size of the output volume.
- S_y^n: stride.
- W_x^n, H_y^n : The size of the output volume and the index *n* indicates the layer.

Once the convolution step has been performed, an activation function is applied to all the values of the filtered image. There are many activation functions, for example, the ReLU which is defined as follows:

$$f(\sigma) = max(0, \sigma)$$

- f: The activation function.
- σ : the value of the neuron.

Fig. 7 shows the CNN architecture used:

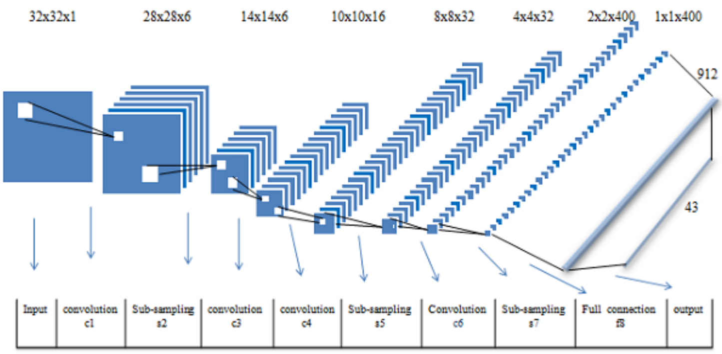


Fig. 7 The architecture of CNN.

The first layer is a convolution layer corresponding to an input image of size 32x32x1 and six filters; each filter is of size 5 × 5x1. At the output we get an image of size 28x28x6, (28 equals 32-5 + 1), It is a two-dimensional filter. Since we are working with a grayscale image, at the exit of a convolution we will apply a RELU activation function, then we will use the Max Pooling layer to reduce the size of the images and to obtain a faster convergence during training.
Then we transmit the output of CNN to a fully connected neuron network (Fully connected), the particularity of this layer is that its neurons are not connected only with the neurons of the previous layer but also with all the neurons of

the next layer. The second part normalizes the results; we connect the last layer fully connected to a softmax. This function is used to produce a probability distribution between the different classes. The table below summarizes the construction steps of the convolutional network.

Table 1 Description of cnn used.

layer	Description	Input	Output
1	Conv (5x5,1x1 stride, valid padding) RELU activation	32x32x1	28x28x6
2	max_Pool(2x2, stride 2)	28x28x6	14x14x6
3	Conv (5x5,1x1 stride, valid padding) RELU activation	14x14x6	10x10x16
4	Conv (3x3,1x1 stride, valid padding) RELU activation	10x10x16	8x8x32
5	Max_Pool(2x2, stride 2)	8x8x32	4x4x32
6	Conv(3x3 ,1x1 stride, valid padding) RELU activation	4x4x32	2x2x400
7	Max_Pool(2x2, stride 2)	2x2x400	1x1x400
8	Flatten	4x4x32	512
9	Concat	512+400	912
10	Fully_conn	912	43

Fig. 8 illustrates the global architecture of our model including the pre-processing step and the recognition and classification step:

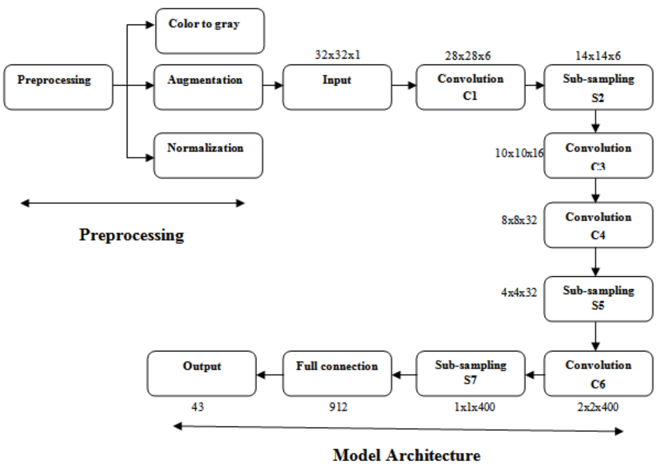


Fig. 8 The global architecture of our model.

IV. RESULT

A. The dataset

The dataset was created from about 10 hours of recorded video while driving through different types of roads in Germany. Sequences were recorded in March, October and November 2010. For data collection, the set contains images of more than 1700 instances of traffic signs. The size of the traffic signs varies between 15 × 15 and 222 × 193 pixels, this set of data is divided into 43 different types of traffic signs; it contains a total of 39,209 images devoted to training, a total of 12,630 images intended for testing and finally 9,902 images used for validation. Each image has a size of 32 × 32 pixels.

B. Evaluation

In this phase, we will present the results of accuracy, the error of our model, we will present new tests and predictions.
From Fig. 9, the precision curves of the training and the validation augment according to the number of periods-, our model offers better results in term of the precision of the training (results is equal to 100 %) and in terms of accuracy of the validation (results is equal to 99.6%).

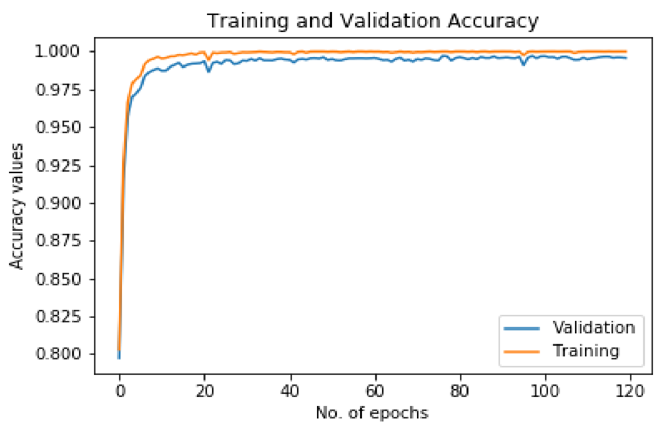


Fig. 9 Training and validation accuracy.

In Fig. 10 the training error and validation curves decrease by a value of 120 at the time.

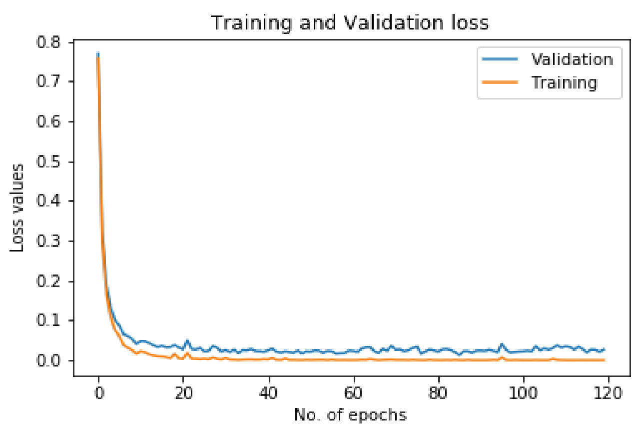


Fig. 10 Training and validation loss.

We evaluated our model using signs. Fig. 11 below shows some test results generated by our model.

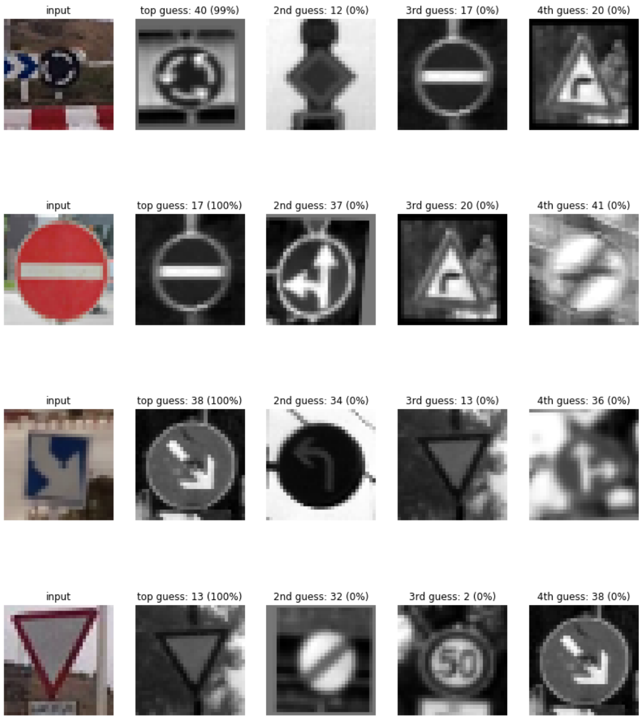


Fig. 11 Test results on new images.

Fig. 12 shows the predictions on new images, we use the softmax probabilities of the model for the certainty of a prediction, the prediction results of our model are very satisfactory.

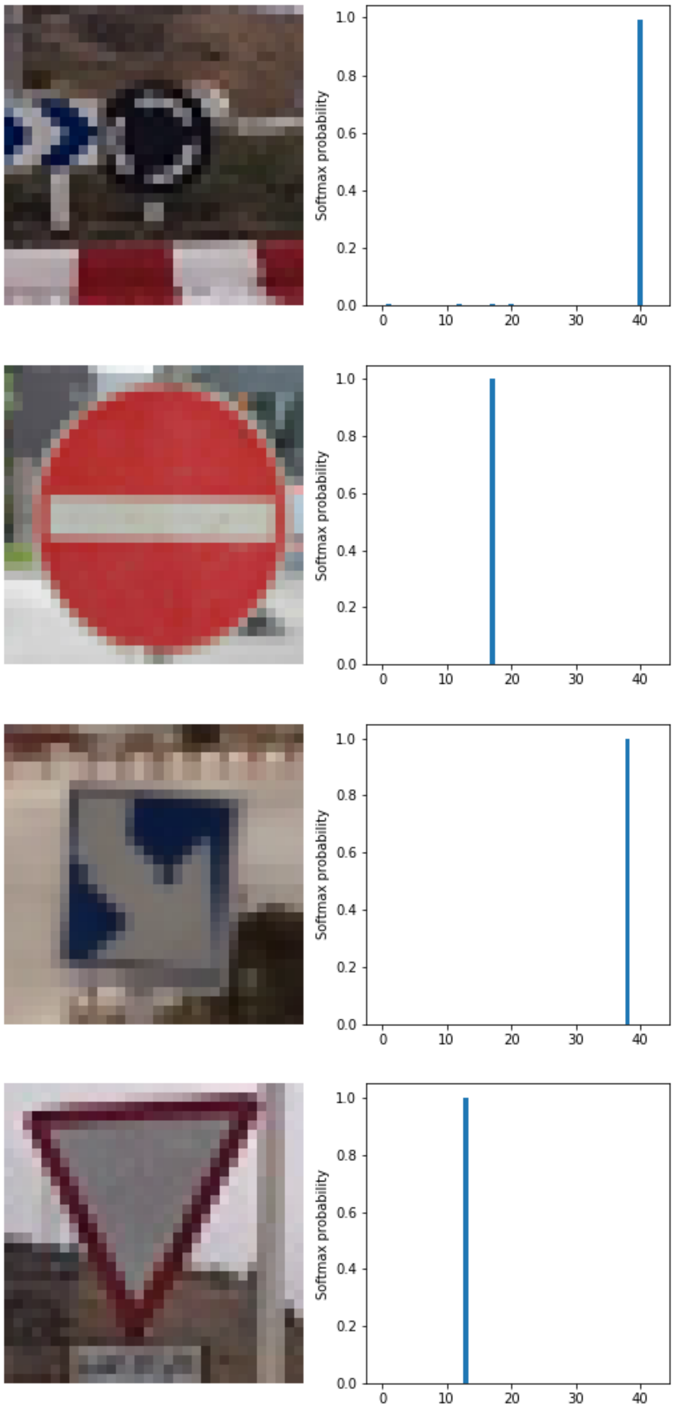


Fig. 12 the prediction results.

V. CONCLUSIONS

In the field of computer vision particularly autonomous driving, the recognition and classification of objects is a crucial problem. In this paper we tend to take up the challenge of recognizing and classifying traffic signs by proposing a model based on convolutional neural networks, we have achieved this goal by dividing the phase of classification and recognition into several stages, the first step is to perform pretreatments on the training that allow us to increase the performance of our model and reduce the risk of over-fitting of the latter, the second step consists of a convolution network to Extract features and an integrally connected neuron network for recognition of traffic signs.
We find that this approach greatly improves the accuracy rate and decreases the error rate. However, despite the use of regularization, normalization and optimization techniques, the training time of our model remains a problem to be raised. As a future extension of our model, we will consider the use of pre-trained neural networks that will allow us to augment the performance of our model, so we will consider taking advantage of the immense computing power of GPU graphics processors by distributing calculations on several GPUs.

REFERENCES

[1] K. Fukushima, "Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological cybernetics*, vol. 36, pp. 193–202, 1980.

[2] D. H. Hubel, and T. N. Wiesel, "Ferrier lecture-Functional architecture of macaque monkey visual cortex," *Proc. R. Soc. Lond. B*, vol.198, pp. 1–59, 1977.

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, " Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, 1998.

[4] D. C. Ciresan, U. Meier, J. Masci, L. Maria Gambardella, and J. Schmidhuber, " Flexible, high performance convolutional neural networks for image classification," *In IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, Vol. 22, pp. 1237, July. 2011.

[5] R. Qian, Y. Yue , F. Coenen, and B. Zhang, "Traffic sign recognition with convolutional neural network based on max pooling positions, " *In Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 12th International Conference on, IEEE*,2016, p. 578–582.

[6] J. E. Zafra, R. J. Moreno, and R. D. Hernández, "Comparison between Backpropagation and CNN for the Recognition of Traffic Signs," *International Journal of Applied Engineering Research*, vol. 12, pp. 6814–6820, 2017.

[7] K. Lim, Y. Hong, Y. Choi, and H. Byun, "Real-time traffic sign recognition based on a general purpose GPU and deep-learning," *PLoS one*, vol. 12, pp. e0173317, 2017.

[8] A. Salhi, B. Minaoui, M. Fakir, H. Chakib, and H. Grimech, (2017). "Traffic Signs Recognition using HP and HOG Descriptors Combined to MLP and SVM Classifiers," *Traffic*, vol. 8, 2017.

[9] C. Reinders, H. Ackermann, M. Y. Yang, and B. Rosenhahn, "Learning a Fully Convolutional Network for Object Recognition using very few Data," arXiv preprint arXiv:1709.05910, 2017.

[10] S.K.PANDA, S.NAG, and P. K.JANA, "A smoothing based task scheduling algorithm for heterogeneous multi-cloud environment," *In Parallel, Distributed and Grid Computing (PDGC), International Conference on. IEEE*, 2014. p. 62–67.

[11] S. K. Panda and P. K. Jana, "A multi-objective task scheduling algorithm for heterogeneous multi-cloud environment," *In Electronic Design, Computer Networks and Automated Verification (EDCAV), 2015 International Conference on. IEEE*, 2015, p. 82–87.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 86,pp. 2278–2324, 1998.