Customized convolutional neural networks for Moroccan traffic signs classification

Fatima Ezzahra Khalloufi, Najat Rafalia, Jaafar Abouchabaka
Faculty of Sciences, University of Ibn Tofail, Kenitra, Morocco

Article Info

Article history:

Received Feb 24, 2024
Revised May 27, 2024
Accepted Jun 5, 2024

Keywords:

CNN
Deep learning
DensNet
ResNet
Traffic signs classification
VGGNet

ABSTRACT

Recognition of traffic signs is a challenging task that can enhance road safety. Deep neural networks have demonstrated remarkable results in numerous applications, such as traffic signs recognition. In this paper, we propose an innovative and efficient system for recognizing traffic signs, based on customized convolutional neural network (CNN) developed through hyperparameters optimization. The effectiveness of the proposed system is assessed using a novel dataset, the Moroccan traffic signs dataset. The results show that the proposed design recognizes traffic signs with an accuracy of 0.9898, outperforming several CNN architectures such as VGGNet, DensNet, and ResNet.

Corresponding Author:

Fatima Ezzahra Khalloufi
Faculty of Sciences, University of Ibn Tofail
Kenitra, Morocco
Email: fatima-ezzahra.khalloufi@uit.ac.ma

1. INTRODUCTION

The continual progress in autonomous vehicles (AV) technology lead to substantial enhancements in the safety and efficiency of road transportation. One of the most challenges facing AVs is the perceiving the landscape and guidance the vehicles in real-world outdoor environments, such as pedestrian detection [1]-[4], vehicle environment perception [5]-[7] traffic sign recognition [8]-[10]. One crucial aspect of AV functionality is the accurate traffic signs recognition, which plays a crucial role in ensuring safe navigation and compliance with traffic regulations. In the context of Morocco, a country with its unique road infrastructure and diverse traffic sign designs, developing a robust traffic sign recognition system becomes imperative to facilitate the successful integration of AVs its roads. Furthermore, proficient recognition of Moroccan traffic signs holds the potential to significantly enhance interaction with surrounding environment, empowering AVs to adapt their behavior and make well informed decisions, thereby fostering smoother traffic flow and reducing travel times.

Several studies on traffic signs recognition have been published, primally focusing on traffic signs in countries outside of Morocco. However, a limited number of benchmark datasets are publicly accessible including the Laboratory for Intelligent and Safe Automobiles (LISA) [11] and widely used German traffic signs benchmark [12]. Numerous machine learning methods are employed for the classification of traffic signs like support vector machines [13], [14], random forest, decision trees [15], [16], nearest neighbors [17], and neural networks [18], [19].

In the last decade, progress made in deep learning techniques especially convolutional neural network (CNN) has played impressive role in image classification tasks, making them well suited for this specific application. Inspired by the exceptional performance of CNN on traffic signs classification. The authors in [20] have used CNN to develop a traffic sign recognition system. They modified the LetNet network by reducing the number of layers aiming to decrease network parameters for faster computation, and test its adaptability in challenging conditions like weather changes and varied backgrounds. The results show significantly high accuracy, outperforming previous studies in this area. Lim *et al.* [21] proposed an ensemble learning deep learning model for traffic sign recognition. They implemented this model using three pre-trained deep learning models: ResNet50, DenseNet121, and VGG16. The results demonstrated that the proposed ensemble learning model outperforms existing traffic sign classification techniques, achieving recognition rates of 98.84% on the GTSRB dataset, 98.33% on the Belgium traffic sign dataset, and 94.55% on the Chinese traffic sign database.

This paper aims to introduce a customized CNN architecture to classify traffic Signs in road of Morocco. By adjusting CNN's architecture, we aim to overcame the challenges associated with other architectures of CNN such as ResNet, VggNet, and DensNet. The successful implementation of customized CNN based traffic sign recognition system can lead to numerous benefits. First, it enhances road safety by providing real time identification of various traffic signs, including speed limits, stops signs, and warning signs.

To address the problem of computational complexity, frequently encountered in well-know CNN models especially in massive data, we reduce the number of layers and number of hyper-pametrs. Additionally, we augment the dataset and utilizes the dropout technique. By adjusting the manipulated parameters, such as stride number, padding, convolution kernel, learning rate was chosen to improve the accuracy of the proposed model. For that, we have developed a customized CNN model with a well-designed architecture. An overview of this paper is as follows: section 2 presents the proposed approach. Section 3 shows the results of the proposed model. Finally, paper conclude in section 4.

2. METHOD

In literature, numerous CNN architectures have been developed with powerful capabilities for learning road signs [22]-[25]. In this study, we propose a customized CNN to recognize traffic signs in Morocco. Figure 1 illustrates the steps of our proposed system. After collecting various Moroccan traffic signs, we perform a preprocessing stage involving several operations. After, we apply the data augmentation techniques to prevent overfitting. Then, we apply a customized CNN and determine the optimal hyper-parameters for our CNN method, based on model's effectiveness. The following paragraphs detail the steps used to define our proposed method:

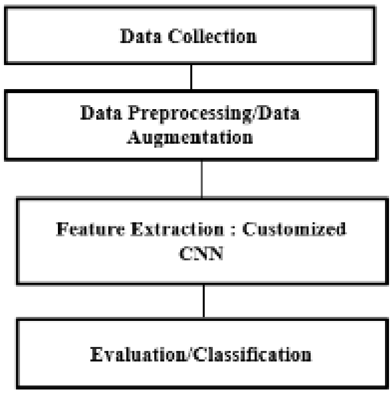

Figure 1. Proposed system overview

2.1. Data collection

In this paper, we utilize GTSDB datasets, as its road signs largely match those found in Morocco, with a few exceptions like the stop sign. We remove these unwanted images and supplement the dataset with images taken from Moroccan road scenes under various weather conditions and lighting. This adaptation ensures our database includes images of diverse signs commonly found on Moroccan roads. This Moroccan traffic signs dataset contains 35 classes that are categorized in four categories (warning 12 classes, regulatory 13 classes, obligatory 8 classes and priority 2 classes. Figure 2 shows examples of Moroccan traffic signs collected.

Figure 2. Examples of Moroccan traffic signs

2.2. Data preprocessing and augmenation

Preprocessing step important in artificial intelligence algorithms, as it accelerates the training process and improves overall performance. It involves preprocessing raw data prior to inputting it into the machine learning architecture. We preprocess our dataset, by resizing all images to 35×35 dimensions, and converting them in one dimensional format to decrease the computational time required for CNN training. Additionally, we observed the limitation of the dataset. To address this, we apply data augmentation techniques, such as rotating images, applying Gaussian blur and noise, scaling images, flipping horizontally, and cropping images to expand the dataset. To prevent early saturation of non-linear activation, we standardize the images to follow the standard normal distribution.

2.3. CNN overview

A CNN is a type of deep learning neural network architecture that is well-suited for images and video analysis. CNNs are specifically used for image recognition and tasks that involve the processing of data such as images. CNNs are composed of three types of layers: the Figure 3 depicts the basic architecture of CNN.

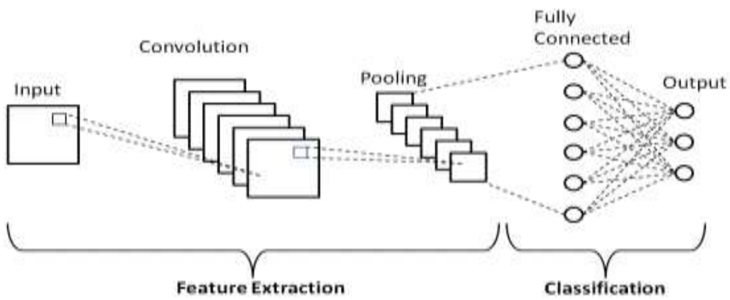

Figure 3. Basic CNN architecture

2.3.1. Convolutional layer

CNNs are composed of multiple layers, with the initial ones typically being convolutional layers. Convolution is a mathematical operation that extracts features from input data. In CNNs, this involves applying a set of learnable filters to the input data in a sliding window fashion. These filters capture various features such as edges, textures, and more complex patterns in the input image.

2.3.2. Pooling layer

After convolution, pooling layers are often used to down sample the feature maps produced by the convolutional layers. Common pooling operations include max-pooling and average pooling. Pooling reduces the spatial dimensions of the feature maps, making subsequent layers computationally more efficient and helping to retrain the most essential information.

2.3.3. Fully connected layer

At the end of the CNN, there are typically one or more fully connected layers. These layers are traditional artificial neural network layers, where each neuron is connected to every neuron in the previous and subsequent layers. Fully connected layers combine high-level features extracted from the convolutional and pooling layers to make final predictions, commonly used in tasks like image classification.

After convolutional and fully connected layer, activation functions such as rectified linear unit (ReLU) are applied to introduce non-linearity into the network, enabling it to learn complex patterns. CNNs are trained through backpropagation and optimization algorithms like gradient decent, allowing the network to adjust the weights of its layers to minimize the difference between predicted and actual outputs during training.

There are several well-known CNN architectures that have been developed over the years, each with its own unique design and performance characteristics. There are some of the most notable CNN architectures such as LeNet-5, AlexNet, VGG, Inception, ResNet and MobileNet. These architectures vary in terms of depth, the number of layers, and the arrangement of convolutional and pooling layers.

CNNs are a type of neural network that are particularly well-suited for image classification tasks. Some of the advantages of CNNs for image classification include dimensionality reduction, automatic feature extraction, spatial and hierarchical structure, accuracy weight sharing, and computation minimization. Overall, CNNs are powerful tool for image classification due to their ability to automatically extract features from raw pixel data, reduce dimensionality, and exploit the spatial and hierarchical structure of images.

2.4. Training dataset with customised CNN

After completing the data preprocessing step, which includes resizing, augmentation and normalization, the data is ready for training and enters the feature extraction stage. The extracted features are then flattened to create a one dimensional vector. This feature vector serves as the input for the classifier, enabling precise predictions. The final step involved evaluating the model's performance on validation data. We iterated this process with various hyper-parameters until achieving the best results. The CNN model consists of three principal types of layers: convolutional layer, fully connected layer, and pooling layer. The convolutional layer comprises a set of filters that convolve across the input data to extract features. In (1) illustrates the output of convolution for the next layer, where each pixel is calculated. The output of the next layer is represented by $net\ (i,j)$, with x as the input image, w was the matrix of the filter and is the convolution operation.

$$net(i,j) = (x * w)[i,j] = \sum\sum x[m,n]\ \ w[i-m,j-n] \tag{1}$$

After two convolutional layers, a max-pooling layer follows, which reduces the computational costs required for data processing. This is achieved by decreasing the spatial size of the resulting activation maps. The mathematical expression of Max-pooling is:

$$Pi = \max(\ (I_i\ )) \tag{2}$$

The fully connected layer provides the probability for each input data by using the SoftMax function as its activation function, the SoftMax function [26] is given by:

$$Softmax(z)_i \ = \ \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} \tag{3}$$

Where:

- z is the vector of raw outputs from the neural network.
- The $i-th$ entry in SoftMax output vector, denoted as $softmax(z)$,indicates the predicted probability that the test input belongs to class i.

As presented in Table 1, our architecture's model consists of four convolutional layers with a 3×3 filter size. Following each convolutional layer, a ReLU activation function is applied to improve non-linearity. A max-pooling layer with 2×2 is introduced after every two convolutional layers to reduce computational complexity and control overfitting. The fully connected layer employs the SoftMax function. Dropout is employed to prevent overfitting and reduce training time by randomly removing a certain number of weak neurons. All the hyperparameters used in our architecture are detailed in Table 2.

Table 1. Customized CNN architecture

Layer	Output Shape	Parameters
conv2d(Conv2D)	(None,35,35,64)	1792
Activation	(None,35,35,64)	0
conv2d_1(Conv2D)	(None,35,35,64)	36928
activation	(None,35,35,64)	0
max_poolling2d(MaxPooling2D)	(None,17,17,64)	0
conv2d_2(Conv2D)	(None,17,17,64)	36928
activation	(None,17,17,64)	0
conv2d_3(Conv2D)	(None,17,17,64)	36928
activation	(None,17,17,64)	0
max_poolling2d-1(MaxPooling2D)	(None,8,8,64)	0
flatten (Flatten)	(None,4096)	0
Dropout	(None,4096)	0
dense (Dense)	(None,250)	1024250
dense_1(Dense)	(None,35)	8785

Table 2. Optimal values of the hyperparameters

Hyperparameter	Value
Input image	35×35
Number of filters	64
Filter size	3×3
Train/test	80%/20%
Loss function	Categorical Crossentropy
Optimizer	Adam
Padding	same
Dropout	0.25
Batch size	32
Learning rate	0.001
Number of epochs	20

3. RESULTS AND DISCUSSION

In this paper, we propose a self built CNN architecture for classification of Moroccan traffic signs datasets into 35 classes. The customized CNN method consists of these steps:

- Preprocessing and augmentation: we resize our images into 35×35. After that, we augment the data by applying data augmentation techniques including rotation, gaussian blur, gaussian noise, scaling, flipping and cropping. Then, we normalized our data.
- Training the data: before train our data, we split it into train set (80%) and test set (20%). Then, we implement our approach with the suitable hyperparameters.

To evaluate the performance of our proposed method, we compared the results of the customized CNN architecture with other architecture of CNN such as ResNet, DensNet, and VGGNet.

3.1. Evaluation metrics

We assessed the effectiveness of our proposed CNN approach using a range of metrics, such as accuracy, recall, precision, and F1-score. The mathematical notation for each performance metric is shown in (4)-(7) respectively. In these equations, "TP" denotes true positive, "TN" represents true negative, "FP" signifies false positive, and "FN" indicates false negative.

$$Accuracy = \frac{TP+TN}{TP=FP+FN+TN} \tag{4}$$

$$Precision = \frac{TP}{TP+FP} \tag{5}$$

$$Recall = \frac{TN}{TN+FP} \tag{6}$$

$$F1-score = \frac{2\times Recall\times Precision}{Recall+Precision} \tag{7}$$

3.2. Evaluation of proposed system

The proposed architecture of CNN is trained and tested in our Moroccan traffic signs dataset. At the same time, we train and test the different famous architectures such as ResNet50, ResNet101, ResNet34, VGG19, VGG16, DensNet121, DensNet69 and DensNet201 to the same dataset. In order to compare the results obtained by the proposed architecture of CNN with the other architectures and demonstrate the good performance of the proposed approach.

Table 3 presents a comparison result of our customized CNN architecture with other architectures of CNN such as ResNet, DensNet, and VGGNet. On the same dataset. The proposed CNN architecture has a higher accuracy (0.9898), precision (0.9696), recall (0.9685) and f1-score (0.9686) comparing with other architectures. It is obvious from this table that our customized architecture provides good and accurate results comparing with other architectures, it can be clearly taking this architecture as a suitable architecture of our dataset. Reducing overfitting and selecting optimal parameters, our model surpassed other models across the majority of metrics values.

The learning process of our model over the 20 epochs is depicted in Figure 3. The curves illustrate a well-fitted model, where both training and testing loss decrease to a point of stability, and a training and validation accuracy increase until stability is reached.

To assess the efficacy of a classification model, machine learning often utilizes confusion matrices. In the scenario of traffic sign classification employing customized CNN model, as depicted in Figure 4, the confusion matrix proves to be a crucial tool for evaluating our model's effectiveness. Specifically, for 35 classes, the confusion matrix is a 35×35 matrix that displays the counts of correctly and incorrectly classified instances for each class, as demonstrated in Figure 5.

Table 3. Comparison between customized CNN and other architectures of CNN

Model	Accuracy	Precision	Recall	F1-score
ResNet50	0.9277	0.8409	0.8015	0.8022
ResNet101	0.8882	0.9205	0.9162	0.9167
ResNet34	0.9464	0.9660	0.9635	0.9639
VGG19	0.0622	0.0040	0.0633	0.0075
VGG16	0.0624	0.0037	0.06127	0.0070
DensNet121	0.9818	0.9615	0.9580	0.9584
DensNet169	0.9784	0.9474	0.9437	0.9430
DensNet201	0.9557	0.9257	0.9131	0.9128
Customized CNN	**0.9898**	**0.9696**	**0.9685**	**0.9686**

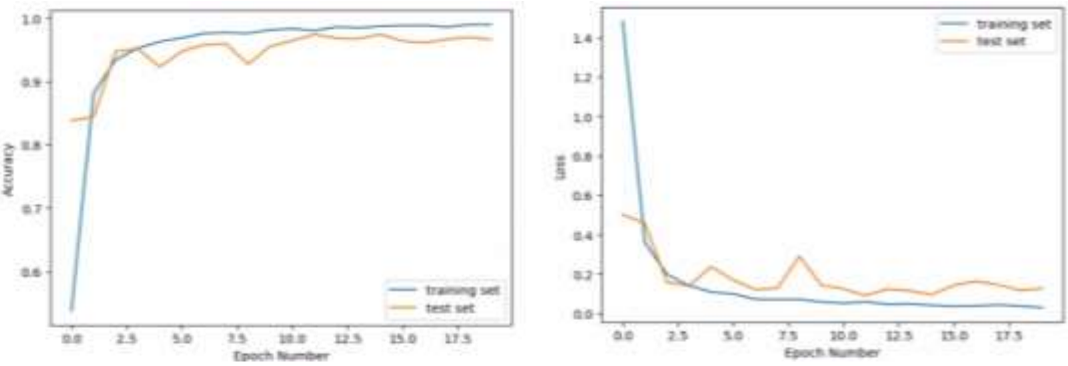

Figure 4. Plots of accuracy and loss models

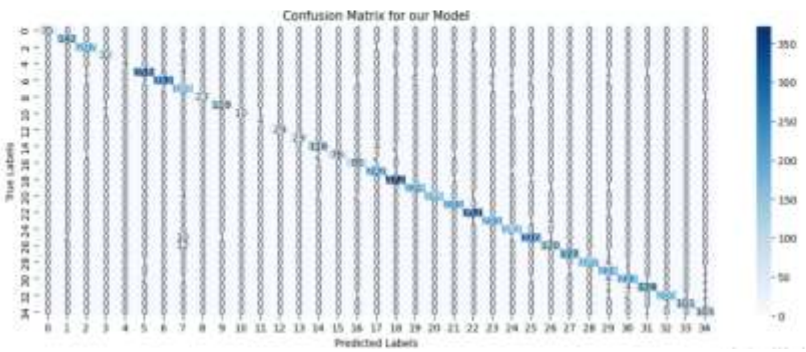

Figure 5. Confusion matrix for our model

4. CONCLUSION

Recognition of traffic sign is an essential component of Avs system, as it can significantly reduce accidents on roads. Therefore, accurate results are crucial, particularly for real time responsiveness. In this work, we have designed a new architecture of CNN to classify traffic signs in Moroccan roads. We propose a customized CNN system to address the problem of computational complexity encountered in well-CNN architectures by reducing number of layers. Additionally, we use the data augmentation and dropout techniques to avoid the overfitting. Then, we choose the optimal parameters to improve the accuracy of the model. The results demonstrated an accuracy of 0.9898 validating the effectiveness of the model. Furthermore, a comparative study highlighted the superiority of our approach over CNN architectures such as ResNet, VGGNet, and DensNet. In our future work, we aim to extend our work to detect the Moroccan traffic signs using Yolo, faster R-CNN and transformers vision techniques to develop our recognition traffic signs system in roads of Morocco.

REFERENCES

[1] M. S. Farooq *et al.*, “A conceptual multi-layer framework for the detection of nighttime pedestrian in autonomous vehicles using deep reinforcement learning,” *Entropy*, vol. 25, no. 1, Jan. 2023, doi: 10.3390/e25010135.

[2] K. R. Sri Preethaa and A. Sabari, “Intelligent video analysis for enhanced pedestrian detection by hybrid metaheuristic approach,” *Soft Computing*, vol. 24, no. 16, pp. 12303–12311, Aug. 2020, doi: 10.1007/s00500-020-04674-5.

[3] R. Lahmyed, M. El Ansari, and A. Ellahyani, “A new thermal infrared and visible spectrum images-based pedestrian detection system,” *Multimedia Tools and Applications*, vol. 78, no. 12, pp. 15861–15885, Jun. 2019, doi: 10.1007/s11042-018-6974-5.

[4] U. Gawande, K. Hajari, and Y. Golhar, “Real-Time deep learning approach for pedestrian detection and suspicious activity recognition,” in *Procedia Computer Science*, vol. 218, pp. 2438–2447, 2023, doi: 10.1016/j.procs.2023.01.219.

[5] F. Walden, S. Dasgupta, P. D. Student, M. Rahman, and M. Islam, “Improving the environmental perception of autonomous vehicles using deep learning-based audio classification”, *arXiv*, 2022, doi: 10.48550/arXiv.2209.04075.

[6] D. Sudha and J. Priyadarshini, “An intelligent multiple vehicle detection and tracking using modified vibe algorithm and deep learning algorithm,” *Soft Computing*, vol. 24, no. 22, pp. 17417–17429, 2020, doi: 10.1007/s00500-020-05042-z.

[7] A. Gupta, A. Anpalagan, L. Guan, and A. S. Khwaja, “Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues,” *Array*, vol. 10, p. 100057, Jul. 2021, doi: 10.1016/j.array.2021.100057.

[8] Y. Zheng and W. Jiang, “Evaluation of vision transformers for traffic sign classification,” *Wireless Communications and Mobile Computing*, vol. 8, pp. 1-14, 2022, doi: 10.1155/2022/3041117.

[9] Y. Liu, J. Ling, Q. Wu, and B. Qin, “Scalable privacy-enhanced traffic monitoring in vehicular ad hoc networks,” *Soft Computing*, vol. 20, no. 8, pp. 3335–3346, Aug. 2016, doi: 10.1007/s00500-015-1737-y.

[10] J. Cao, C. Song, S. Peng, F. Xiao, and S. Song, “Improved traffic sign detection and recognition algorithm for intelligent vehicles,” *Sensors*, vol. 19, no. 18, Sep. 2019, doi: 10.3390/s19184021.

[11] A. Møgelmose, M. M. Trivedi, and T. B. Moeslund, “Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1484–1497, 2012, doi: 10.1109/TITS.2012.2209421.

[12] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, “Detection of traffic signs in real-world images: The German traffic sign detection benchmark,” in *Proceedings of the International Joint Conference on Neural Networks*, 2013, doi: 10.1109/IJCNN.2013.6706807.

[13] Y. Liu and W. Zhong, “A Novel SVM network using HOG feature for prohibition traffic sign recognition,” *Wireless Communications & Mobile Computing*, vol. 2022, 2022, doi: 10.1155/2022/6942940.

[14] S. Maldonado-Bascón, S. Lafuente-Arroyo, P. Gil-Jiménez, H. Gómez-Moreno, and F. López-Ferreras, “Road-sign detection and recognition based on support vector machines,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 264–278, Jun. 2007, doi: 10.1109/TITS.2007.895311.

[15] A. J, G. R, A. K, and S. R, “Traffic Sign Detection using HOG and GLCM with Decision Tree and Random Forest,” in *2022 International Conference on Automation, Computing and Renewable Systems (ICACRS)*, IEEE, Dec. 2022, pp. 879–883, doi: 10.1109/ICACRS55517.2022.10029118.

[16] A. J, G. R, A. K, and S. R, “Traffic Sign Detection using HOG and GLCM with Decision Tree and Random Forest,” in *2022 International Conference on Automation, Computing and Renewable Systems (ICACRS)*, 2022, pp. 879–883, doi: 10.1109/ICACRS55517.2022.10029118.

[17] A. Ruta, Y. Li, and X. Liu, “Real-time traffic sign recognition from video by class-specific discriminative features,” *Pattern Recognition*, vol. 43, no. 1, pp. 416–430, Jan. 2010, doi: 10.1016/j.patcog.2009.05.018.

[18] A. Kerim and M. Ö. Efe, "Recognition of Traffic Signs with Artificial Neural Networks: A Novel Dataset and Algorithm," *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, Jeju Island, Korea (South), 2021, pp. 171-176, doi: 10.1109/ICAIIC51459.2021.9415238.

[19] H. Wan, L. Gao, M. Su, Q. You, H. Qu, and Q. Sun, “A Novel Neural Network Model for Traffic Sign Detection and Recognition under Extreme Conditions,” *J Sensors*, vol. 2021, 2021, doi: 10.1155/2021/9984787.

[20] H. Ben Fredj, A. Chabbah, J. Baili, H. Faiedh, and C. Souani, “An efficient implementation of traffic signs recognition system using CNN,” *Microprocessors and Microsystems*, vol. 98, Apr. 2023, doi: 10.1016/j.micpro.2023.104791.

[21] X. R. Lim, C. P. Lee, K. M. Lim, and T. S. Ong, “Enhanced traffic sign recognition with ensemble learning,” *Journal of Sensor and Actuator Networks*, vol. 12, no. 2, Apr. 2023, doi: 10.3390/jsan12020033.

[22] D. C. Santos *et al.*, “Real-Time Traffic Sign Detection and Recognition using CNN,” *IEEE Latin America Transactions*, vol. 18, no. 03, pp. 522–529, Mar. 2020, doi: 10.1109/TLA.2020.9082723.

[23] A. Vennelakanti, S. Shreya, R. Rajendran, D. Sarkar, D. Muddegowda, and P. Hanagal, “Traffic Sign Detection and Recognition using a CNN Ensemble,” *2019 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, USA, 2019, pp. 1-4, doi: 10.1109/ICCE.2019.8662019.

[24] X. Qiao, “Research on Traffic sign recognition based on CNN deep learning network,” in *Procedia Computer Science*, 2023, pp.826-837, doi: 10.1016/j.procs.2023.11.102.

[25] M. A. Khan, H. Park, and J. Chae, “A lightweight convolutional neural network (CNN) architecture for traffic sign recognition in urban road networks,” *Electronic*, vol. 12, no. 8, Apr. 2023, doi: 10.3390/electronics12081802.

[26] P. Domingos, “A few useful things to know about machine learning,” *Communications of the ACM*, vol. 55, no. 10. pp. 78–87, Oct. 2012, doi: 10.1145/2347736.2347755.

BIOGRAPHIES OF AUTHORS

Fatima Ezzahra Khalloufi received his master’s degree in applied mathematics, from Hassan Premier University, Settat, Morocco. She is a Ph.D. student in computer research Laboratory (LaRI) at Ibn Tofail. His research interests include big data, data mining, machine and deep learning. She can be contacted at email: fatima-ezzahra.khalloufi@uit.ac.ma.

Pr. Najat Rafalia obtained three doctorates in Computer Sciences from Mohammed V University, Rabat, Morocco by collaboration with ENSEEIHT, Toulouse, France, and Ibn Tofail University, Kenitra, Morocco. Currently, she is a Professor at Department of Computer Sciences, Ibn Tofail University, Kenitra, Morocco. Her research interests are in distributed systems, multi-agent systems, concurrent and parallel programming, communication, security, big data, and cloud computing. She can be contacted at email: najat.rafalia@uit.ac.ma.

Pr. Jaafar Abouchabaka obtained two doctorates in Computer Sciences applied to mathematics from Mohammed V University, Rabat, Morocco. Currently, he is a Professor at Department of computer Sciences, Ibn Tofail University, Kenitra, Morocco. His research interests are in concurrent and parallel programming, distributed systems, multi agent systems, genetics algorithms, big data and cloud computing. He can be contacted at email: jaafar.abouchabaka@uit.ac.ma.