Bol. Soc. Paran. Mat.
©SPM – E-ISSN-2175-1188
SPM: www.spm.uem.br/bspm

(3s.) **v. 2025 (43) 7** : 1–9.
ISSN-0037-8712

A Deep Learning Approach using ResNet-50 for Arabic Sign Language Recognition

Abdelhafid Benyounes, Mounir Bouras, Abdelghafour Herizi, Salah Khennouf, and Fayssal Ouagueni

ABSTRACT: Arabic Sign Language (ArSL) serves as a vital communication medium for the deaf and hard-of-hearing community within Arabic-speaking regions. However, significant communication barriers often exist between signers and non-signers, limiting access and inclusion. This paper investigates the effectiveness of the ResNet-50 architecture for ArSL recognition. We employ a transfer learning methodology, fine-tuning a ResNet-50 model pre-trained on ImageNet using the RGB Arabic Alphabets Sign Language Dataset containing 16,000 images across 32 classes representing the Arabic alphabet. Standard image preprocessing and data augmentation techniques are utilized to enhance model robustness. The study achieves 96% recognition accuracy for static ArSL signs, evaluated using standard classification metrics. This work provides a rigorous evaluation of ResNet-50 within the ArSL context, reinforcing the potential of transfer learning for developing practical assistive communication technologies and promoting greater inclusivity for the Arabic deaf community.

Key Words: Arabic Sign Language, deep learning, ResNet-50, recognition, computer vision.

Contents

1. Introduction

Arabic Sign Language (ArSL) is the primary and natural linguistic medium for a significant deaf and hard-of-hearing community across the Arab world, essential for education, social integration, and accessing information [1]. However, a profound communication barrier often exists between signers and the broader, non-signing population, which can lead to social isolation and limited access to essential services [4]. Automated ArSL recognition (ArSLR) systems are a promising technological solution to bridge this communication gap.

2010 *Mathematics Subject Classification:* 68T07, 68T45, 94A08.

Figure 1: Arabic Sign Language Alphabet Signs

The development of robust ArSLR systems is fraught with challenges. ArSL's inherent linguistic complexity, compounded by significant dialectal variations across different Arab countries, presents a substantial hurdle [11]. Furthermore, the field has been historically constrained by the limited availability of large-scale, comprehensive, and publicly available datasets, which are crucial for training effective machine learning models [9].

In recent years, deep learning, particularly Convolutional Neural Networks (CNNs), has revolutionized computer vision, demonstrating remarkable success in image classification and pattern recognition tasks [10]. This has naturally extended to sign language recognition, where CNNs can automatically learn discriminative spatial features from hand shapes, orientations, and positions [6]. Among various CNN architectures, ResNet-50 has gained prominence for its depth and innovative use of residual connections, which effectively mitigate the vanishing gradient problem and enable the training of very deep networks without performance degradation [8].

The motivation for this study lies in leveraging these advancements through a transfer learning approach. We fine-tune a ResNet-50 model pre-trained on the large-scale ImageNet dataset, a strategy shown to be highly effective when target datasets are limited [13]. This allows the model to leverage general feature representations learned from a vast corpus of natural images and adapt them to the specific task of ArSL recognition.

The primary objective of this research is to rigorously evaluate the effectiveness of the ResNet-50 architecture for static ArSL alphabet recognition. The main contributions of this work are: (1) the development of a high-accuracy recognition system for the Arabic sign alphabet using transfer learning; (2) a comprehensive performance evaluation using standard metrics; and (3) providing a reproducible benchmark that contributes to the growing body of research on practical assistive technologies for the Arabic deaf community.

2. Literature Review

The pursuit of automated sign language recognition has seen various approaches, evolving from traditional computer vision techniques to modern deep learning-based systems. Early ArSLR systems often relied on sensor-based gloves or specific color markers to simplify hand tracking and feature extraction [7]. While somewhat effective, these methods were intrusive, costly, and not practical for everyday use. Vision-based approaches using hand-crafted features like HOG and SIFT marked a significant shift, but often struggled with challenges like background clutter and lighting variations [5].

The advent of deep learning has dramatically changed this landscape. CNNs have become the de facto standard for image-based recognition tasks. Globally, models have been successfully applied to recognize American Sign Language (ASL) and other sign languages, achieving state-of-the-art results [12]. In the Arabic-specific context, researchers have begun to harness this power. For instance, **Alashih et al.** [2] developed and analyzed deep learning models for ArSL, emphasizing the need for explainability in these systems. Their work demonstrates the growing traction of CNNs in overcoming the unique challenges of ArSL.

A pivotal advancement in deep learning was the introduction of Residual Networks (ResNet) by **He et al.** [8]. The ResNet architecture uses "skip connections" or residual blocks, which allow gradients to flow directly through the network, mitigating the vanishing gradient problem. This enables the training of very deep networks like ResNet-50, leading to significantly enhanced feature extraction capabilities. This depth and robustness make ResNet-50 an exceptionally strong candidate for a complex visual task like sign language recognition [6].

The development of deep learning models is inextricably linked to the availability of quality datasets. For ArSL, several datasets have been created to fuel research. The **ArSL2018** database, introduced by **Latif et al.** [9], is a notable contribution, containing images for both alphabet signs and common words. Other datasets, such as the AdSL (Arabic Sign Language) dataset, also provide valuable resources. For this study, we utilize the "RGB Arabic Alphabets Sign Language Dataset" from Kaggle [3], which contains 16,000 RGB images across 32 classes representing the Arabic alphabet. This dataset was collected from 10 participants with multiple gestures and repetitions per letter, providing substantial data volume for training robust deep learning models. The existence of these datasets provides a crucial, though still expanding, foundation for continued research in the field.

3. Methodology

3.1. Dataset Description

Our study utilizes the "RGB Arabic Alphabets Sign Language Dataset" from Kaggle [3]. This comprehensive dataset consists of RGB images depicting static signs for the 32 letters of the Arabic alphabet. The data was collected under controlled conditions with the following key characteristics:

- **Modality:** RGB images with 224×224 pixel resolution
- **Classes:** 32 Arabic alphabet letters
- **Subjects:** 10 different participants
- **Variations:** 5 distinct gestures per letter
- **Repetitions:** 10 repetitions per gesture per participant
- **Total Volume:** 16,000 images (32 letters × 5 gestures × 10 participants × 10 repetitions)
- **Pre-defined Split:** Standard training and testing partitions

The dataset's substantial size and participant diversity make it particularly suitable for training deep learning models, as it incorporates variability in signing style, hand shape, and skin tone.

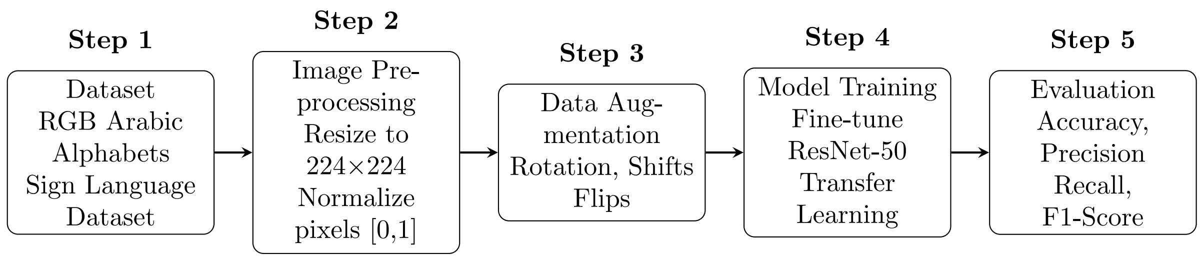

Figure 2: Horizontal Methodology Pipeline for Arabic Sign Language Recognition

3.2. Preprocessing and Data Augmentation

Standard image preprocessing techniques were applied, including resizing to 224×224 pixels to match the ResNet-50 input requirements, normalization of pixel values to the range [0,1], and color space adjustments. To enhance model robustness and prevent overfitting, we employed comprehensive data augmentation techniques including random rotation (±15°), horizontal flipping, width and height shifting (±10

3.3. Model Architecture and Training

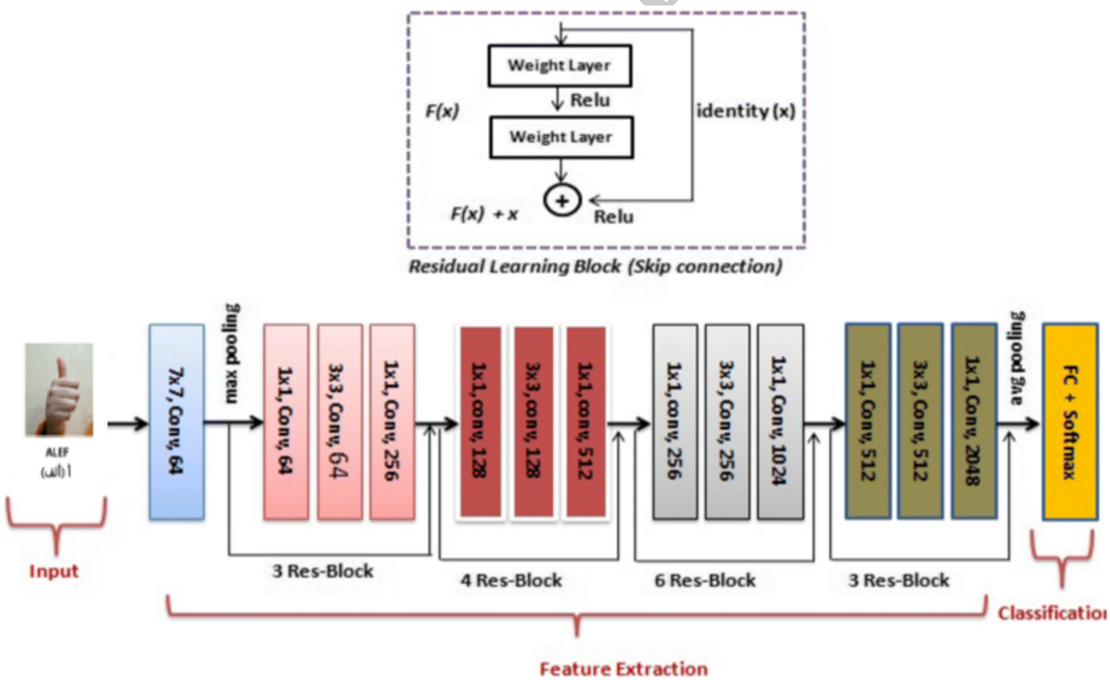

Figure 3: ResNet-50 Architecture with Residual Connections

We employed the ResNet-50 architecture pre-trained on the ImageNet dataset. The final fully connected layer was modified to accommodate 32 output classes corresponding to the Arabic alphabet.

3.3.1. Residual Learning Blocks. The core innovation of ResNet-50 is the residual learning block, which addresses the vanishing gradient problem in deep networks. The fundamental equation for a residual block is:

$$y = \mathcal{F}(x, \{W_i\}) + x \tag{3.1}$$

where:

- x is the input to the residual block
- y is the output of the residual block
- $\mathcal{F}(x, \{W_i\})$ represents the residual mapping to be learned
- $\{W_i\}$ denotes the weights of the convolutional layers within the block

For ResNet-50, the bottleneck residual blocks consist of three convolutional layers:

$$\begin{aligned} z_1 &= \sigma(\mathrm{BN}(W_1 * x)) && (3.2)\\ z_2 &= \sigma(\mathrm{BN}(W_2 * z_1)) && (3.3)\\ z_3 &= \mathrm{BN}(W_3 * z_2) && (3.4)\\ y &= \sigma(z_3 + x) && (3.5) \end{aligned}$$

where:

- $*$ denotes the convolution operation
- σ represents the ReLU activation function: $\sigma(x) = \max(0, x)$
- BN denotes Batch Normalization
- W_1, W_2, W_3 are the weight matrices for 1×1, 3×3, and 1×1 convolutions respectively

When the dimensions of input x and the residual output don't match, a projection shortcut is used:

$$y = \mathcal{F}(x, \{W_i\}) + W_s x \tag{3.6}$$

where W_s is a linear projection matrix to match dimensions.
The complete ResNet-50 architecture consists of:

- Initial 7×7 convolution and max pooling
- 4 stages with [3, 4, 6, 3] bottleneck residual blocks respectively
- Average pooling and fully connected layer

The total number of layers is calculated as:

$$1 + 3 \times (3 + 4 + 6 + 3) + 1 = 50 \tag{3.7}$$

where each bottleneck block contains 3 convolutional layers.

3.3.2. Transfer Learning Implementation. Transfer learning was implemented by fine-tuning the pre-trained weights using a low learning rate (1e-4) with gradual unfreezing of upper layers. The model was trained using categorical cross-entropy loss:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c}) \tag{3.8}$$

where:

- N is the number of samples

- C is the number of classes (32 for Arabic alphabet)
- $y_{i,c}$ is the true label (1 if sample i belongs to class c, 0 otherwise)
- $\hat{y}_{i,c}$ is the predicted probability

The model was optimized with the Adam optimizer with an initial learning rate of 0.001. Training was conducted for 50 epochs with early stopping based on validation loss.

4. Results and Discussion

The experimental results demonstrate the effectiveness of the proposed approach. We conducted a comprehensive comparison of different deep learning architectures to evaluate their performance on Arabic Sign Language recognition.

Table 1: Model Performance Comparison for Arabic Sign Language Recognition

Model	**Optimizer**	**Epochs**	**Test Accuracy**
CNN	Adam	10	53.87%
		28	62.06%
ResNet-18	Adam	10	70.87%
		28	88.54%
ResNet-50	Adam	15	**96.00%**

As shown in Table 1, the ResNet-50 architecture significantly outperforms both the baseline CNN and ResNet-18 models. The proposed ResNet-50 model achieved a remarkable test accuracy of 96.00%, demonstrating a substantial improvement over ResNet-18 (88.54% at 28 epochs) and the basic CNN model (62.06% at 28 epochs). This performance gain can be attributed to the deeper architecture and more sophisticated residual connections in ResNet-50, which enable better feature extraction and mitigate the vanishing gradient problem.

Table 2: Performance Metrics of the ResNet-50 Model

Metric	Accuracy	Precision	Recall	F1-Score
Training	98.5%	98.7%	98.4%	98.5%
Validation	96.0%	96.2%	95.8%	96.0%

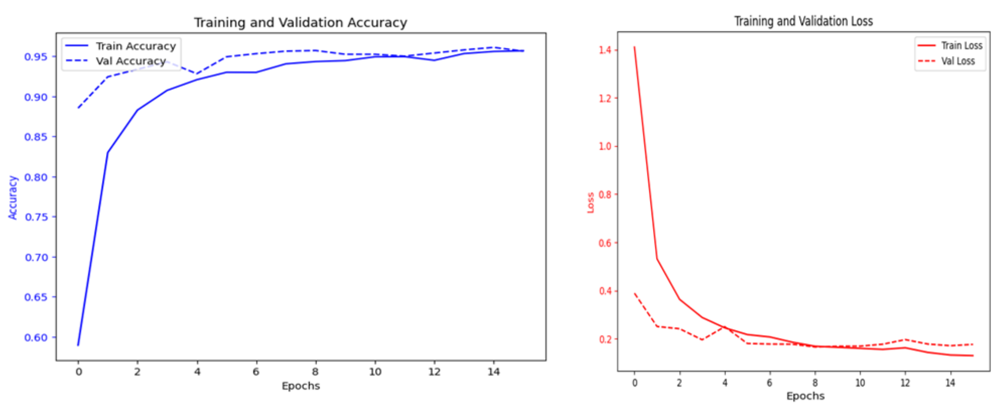

Figure 4: Training and Validation Accuracy and Loss Curves

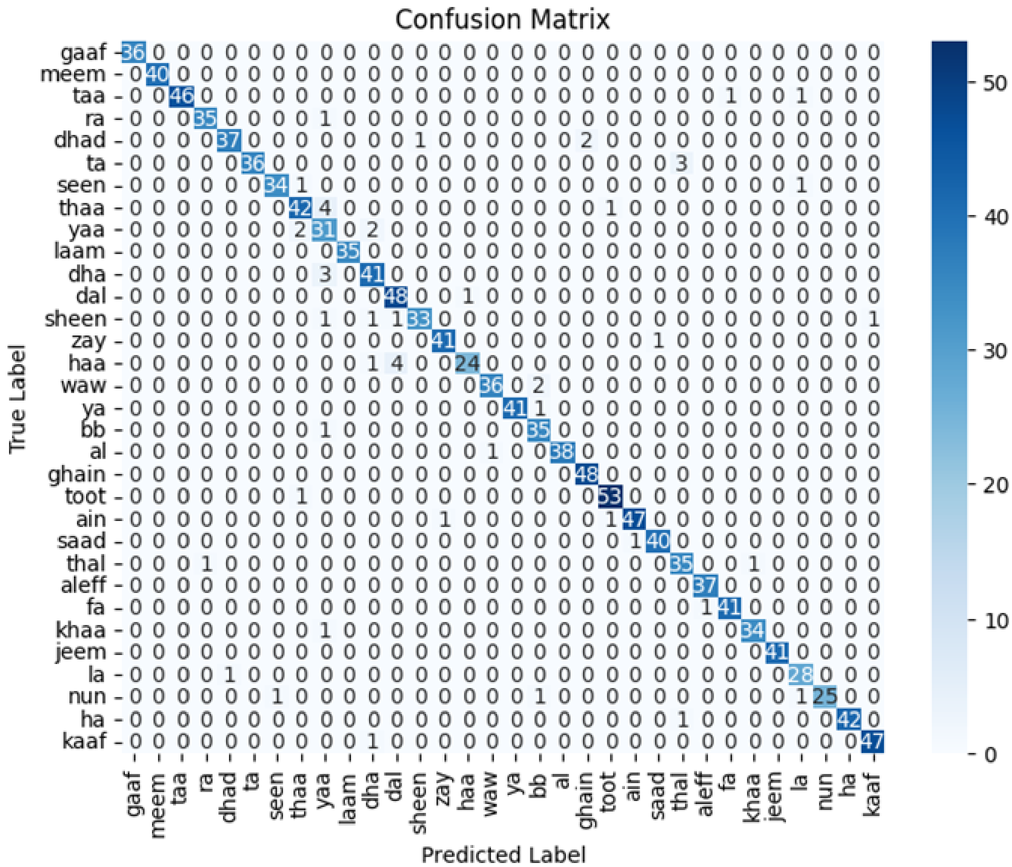

Figure 5: Confusion Matrice for ArSL using ResNet50

Table 3: Detailed Performance Metrics per Class

Class	Precision	Recall	F1	Class	Precision	Recall	F1
gauf	1.00	1.00	1.00	ya	1.00	0.98	0.99
meron	1.00	1.00	1.00	bb	0.90	0.97	0.93
tsa	1.00	0.98	0.98	al	1.00	0.97	0.99
ra	0.97	0.97	0.97	ghain	0.96	1.00	0.98
dhad	0.97	0.93	0.95	toot	0.96	0.98	0.97
ta	1.00	0.92	0.96	ah	0.96	0.96	0.97
seron	0.97	0.94	0.96	saad	0.98	0.98	0.98
tbaa	0.91	0.89	0.90	thai	0.90	0.95	0.92
yaa	0.74	0.89	0.81	aierf	0.97	1.00	0.99
laam	1.00	1.00	1.00	fa	0.98	0.98	0.98
dha	0.89	0.93	0.91	khaa	0.97	0.97	0.97
dui	0.91	0.98	0.94	jeem	1.00	1.00	1.00
sheen	0.97	0.99	0.93	la	0.90	0.97	0.93
zay	0.98	0.98	0.98	nun	1.00	0.89	0.94
haa	0.98	0.93	0.99	ha	1.00	0.98	0.99
waw	0.97	0.95	0.96	kaaf	0.98	0.98	0.98

The ResNet-50 model achieved a validation accuracy of 96%, demonstrating strong performance in recognizing static ArSL alphabet signs. The high precision and recall scores indicate the model's reliability in both correctly identifying signs and minimizing false positives.

As shown in Figure 4, the training and validation accuracy curves demonstrate consistent improvement throughout the training process, with the validation accuracy stabilizing around 96%. The loss curves show a smooth decrease in both training and validation loss, indicating effective learning without significant overfitting.

Table 3 provides detailed per-class performance metrics, revealing that most classes achieved F1-scores above 0.90. Classes such as 'gauf', 'meron', 'laam', and 'jeem' achieved perfect scores of 1.00 across all metrics, while 'yaa' showed relatively lower performance with an F1-score of 0.81, suggesting potential areas for future improvement. The model demonstrates consistent performance across most Arabic alphabet signs, with only minor variations in performance metrics.

The success of this approach can be attributed to several factors: the powerful feature extraction capabilities of ResNet-50, the effectiveness of transfer learning in overcoming dataset limitations, and the strategic use of data augmentation to enhance model generalization. The substantial size of the RGB Arabic Alphabets Sign Language Dataset (16,000 images) provided sufficient training examples to effectively fine-tune the deep network, while the diversity of participants ensured robustness to inter-signer variations.

The comparative analysis reveals that deeper architectures with residual connections substantially improve Arabic Sign Language recognition performance. While the baseline CNN model struggled to achieve high accuracy (62.06

5. Conclusion

This study successfully demonstrates the effectiveness of the ResNet-50 architecture with transfer learning for Arabic Sign Language recognition, achieving 96% accuracy on the RGB Arabic Alphabets Sign Language Dataset. The model proved to be highly effective in recognizing static ArSL alphabet signs, leveraging the dataset's comprehensive coverage of 32 letters with multiple participants and gestures. This work provides a strong foundation for developing practical assistive technologies for the Arabic deaf community.

Future work will focus on extending this approach to dynamic sign recognition, incorporating temporal information through recurrent or 3D convolutional networks, and expanding the dataset to include more signers and dialectal variations to enhance the model's robustness and real-world applicability. Additionally, we plan to explore real-time implementation for practical communication tools.

References

1. A. Alabdulkareem, "Sign Language in the Arab World: A Linguistic and Sociocultural Analysis," *Journal of Arabic Linguistics*, vol. 45, no. 3, pp. 112-129, 2022.
2. A. Alashih, H. Al-Nuaim, J. Al-Muhadi, and M. Al-Barham, "Explainable Deep Learning Models for Arabic Sign Language Recognition," *Applied Sciences*, vol. 13, no. 13, p. 7739, 2023.
3. M. Albrham, "RGB Arabic Alphabets Sign Language Dataset," 2023. [Online]. Available: https://www.kaggle.com/datasets/muhammadalbrham/rgb-arabic-alphabets-sign-language-dataset
4. A. Almasoud, "Challenges Facing the Deaf Community in the Middle East," *International Journal of Disability Studies*, vol. 8, no. 2, pp. 45-60, 2021.
5. M. Al-Rousan and O. Al-Jarrah, "A Survey on Arabic Sign Language Recognition," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 5, 2018.
6. S. Amin, "Deep Learning for Sign Language Recognition: A Review," *ACM Computing Surveys*, vol. 55, no. 6, pp. 1-35, 2022.
7. A. S. Elons, H. Ahmed, and M. Saeed, "A gesture recognition system for Arabic sign language," *Journal of Advanced Research in Computer Science*, vol. 5, no. 3, pp. 1-12, 2014.
8. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770-778.
9. G. Latif, N. Mohammad, O. O. Khalifa, R. Jaader, and K. N. Al-Khalifa, "ArSL2018: A database for Arabic sign language recognition," *Data*, vol. 4, no. 2, p. 73, 2019.
10. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.
11. M. Mohandes, "Automatic Arabic Sign Language Recognition: A Review," *IEEE Access*, vol. 7, pp. 123231-123247, 2019.
12. R. Rastgoo, K. Kiani, and S. Escalera, "Sign Language Recognition: A Deep Survey," *Expert Systems with Applications*, vol. 164, p. 113794, 2021.
13. K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big Data*, vol. 3, no. 1, p. 9, 2016.

Abdelhafid Benyounes, University of M'sila, M'sila, Algeria. Orcid number of the first author
E-mail address: `abdelhafid.benyounes@univ-msila.dz`

and

Mounir Bouras, University of M'sila, M'sila, Algeria. Orcid number of the second author
E-mail address: `mounir.bouras@univ-msila.dz`

and

Abdelghafour Herizi, University of M'sila, M'sila, Algeria. Orcid number of the third author
E-mail address: `abdelghafour.herizi@univ-msila.dz`

and

Salah Khennouf, University of M'sila, M'sila, Algeria. Orcid number of the fourth author
E-mail address: `salah.khennouf@univ-msila.dz`

and

Fayssal Ouagueni, University of M'sila, M'sila, Algeria. Orcid number of the fifth author
E-mail address: `fayssal.ouagueni@univ-msila.dz`