Information Sciences with Applications, Vol. 1, 2024
https://doi.org/10.61356/j.iswa.2024.19673

A Deep Learning Framework for Timely Bone Fracture Detection and Prevention

Ahed J Alkhatib [1], Mohamad Alharoun [2], and Areej AlZoubi [3,*]

[1] Legal Medicine, Toxicology of Forensic Science and Toxicology Department Jordan University of Science and Technology Irbid, Jordan; ajalkhatib@just.edu.jo
[2] Medical Laboratory Science Department Hashemite University Irbid, Jordan.
[3] Computer Information Systems Department Jordan University of Science and Technology Jerash, Jordan; azalzoubi19@cit.just.edu.jo.

* Correspondence: azalzoubi19@cit.just.edu.jo.

Abstract: Accurate bone fracture identification remains critical in medical diagnostics, motivating researchers to investigate deep-learning systems to address this difficulty. The performance of convolutional neural networks (CNN) and region-based convolutional neural networks (RCNN) in fracture diagnosis is investigated in this paper. Algorithmic efficacy was investigated using a complete set of experiments involving various epochs, batch sizes, and optimization techniques (Adam and SGD). As discussed in the discussion, the results continuously highlight the superiority of the RCNN algorithm, which demonstrated amazing performance across many experimental settings. Algorithms trained using the Adam optimizer regularly demonstrated high levels of accuracy, precision, recall, and F1 score. Nonetheless, the effect of epoch counts and batch size on performance variability was seen, necessitating careful consideration to avoid overfitting and ensure generalization. These findings support careful algorithm selection guided by optimization techniques and fine-tuned hyperparameters. The RCNN algorithm's impressive results, proven by its constant superiority, underline its potential to revolutionize bone fracture diagnosis. Further research should focus on hyperparameter tuning and comprehensive validation across several datasets, fostering accurate and efficient solutions for bone fracture diagnoses in medical practice.

Keywords: Deep learning; Medical Imaging; Predictive Analytics; Healthcare Technology; Diagnostic Tools; Convolution Neural Networks; Fracture Prevention.

1. Introduction

Bone fractures are currently one of the most common types of injuries. Every year, 2.7 million fractures occur [1]; a staggering number of people are affected by this disorder, and the consequences of an untreated fracture can result in lifelong damage or even death [2]. Falls, crashes, fights, and other mishaps frequently result in fractures in infants, the elderly, and young people [3] [4]. Many clinicians rely on medical pictures to determine whether bone fractures occur. With the advancement of medically advanced technology, there are numerous methods for obtaining high-quality medical images, including X-ray, computer tomography (CT), and magnetic resonance imaging (MRI) [5].

Doctors bear a tremendous deal of responsibility for this, as they must examine tens of thousands of X-ray images every day. An experienced doctor must spend a significant amount of time inspecting where a bone fracture occurred in an X-ray image. However, many hospitals lack competent radiologists to handle these medical images. Computer-aided diagnosis (CAD) has been widely utilized in medical image analysis to assist doctors throughout bone fracture detection, and it has garnered growing attention in recent years [6].

As a result, a more practical approach would be beneficial in detecting bone fractures on various types of bones in the human body. Due to the wide variances in bone types, developing such a system is a difficult task [7]. Deep learning, specifically the convolution neural network (CNN) and the recurrent convolution neural network (RCNN), has attained results comparable to those of humans in bone fracture classification in recent years. We suggest a system to accomplish this. In this paper, we propose a system that detects bone fractures by combining R-CNN and Convolutional Neural Networks. Doctors rely heavily on X-ray scans to determine where bone fractures occur.

R-CNN and CNN are used to detect bone fractures in X-ray images and to classify bones. Moreover, compare the performance of these two methods to determine the best approach for detecting bone fractures in the human body's upper and lower limbs.

2. Related Work

Lin et al. [8] detected skull fractures using R-CNN (Region-based Convolutional Neural Network). To improve classification performance, they used prior clinical knowledge in quicker R-CNN. Kitamura et al. [9] created an ensemble-based deep CNN model for detecting ankle fractures. For feature extraction, they use Xception, InceptionV3, and ResNet. The ensemble-based approach correctly detects healthy and fractured individuals 81% of the time. Yang et al. [10] created two deep CNN models for segmenting and identifying intertrochanteric fractures. They divided the dataset into two portions, training, and testing, with 32,045 and 11,465 photos utilized in each. First, using cascade architecture-based CNN, the area of interest (ROI) is extracted. Following that, a different CNN is employed for segmentation and identification. Ma et al. [7] discovered bone fracture in two steps: first, a quicker R-CNN was used to identify 20 fracture locations, and then, a unique CrackNet for bone fractured classification was used. Their approach correctly diagnoses healthy and damaged bone with a 90.14% accuracy. On a small dataset, Yahalomi et al. [12] and Abbas et al. [13] used a pre-trained Faster R-CNN model and achieved accuracies of 96% and 97%, respectively. Luo et al. [14] developed a decision tree-based technique for identifying broken bones, with an accuracy of 86.57%. Deep Convolutional Neural Networks (CNNs) were used by Beyaz et al. [15] and Jones et al. [16] to extract features from bone scans, with validation accuracies of 83% and 97.4%, respectively. Hardalaç et al. [17] detected wrist fractures from healthy bone with 86.39% accuracy using ensemble-based deep CNN models. By combining deep and SURF characteristics, Pranata et al. [18] focused on calcaneus fractures in CT images. When VGG16 and ResNet, two pre-trained deep CNN models, were compared, ResNet performed best, with a classification accuracy of 98%. Mutasa et al. [19] used Generative Adversarial Networks (GANs) and Digitally Reconstructed Radiographs (DRRs) approaches to create a picture dataset and classify it using deep CNN models.

Weikert et al. [20] used a deep learning-based CNN model to identify rib fractures with a 90.2% accuracy. Tanzi et al. [21] used the InceptionV3 model to classify proximal femur X-ray images, with 86% accuracy in discriminating between healthy and damaged femurs. The goal of the study is to enhance sternum fracture diagnosis through the application of cutting-edge computer vision technology. If a fracture is not treated right away, serious consequences may result. The authors [23] created an automated detection model using deep convolutional neural networks (CNNs) and methods like cascade R-CNN, attention mechanisms, and atrous convolution. Their method dramatically improved detection, particularly for small and subtle fractures in intricately detailed X-ray pictures. Their model outperformed other cutting-edge approaches in comparison tests, reaching a mean Average Precision (mAP) of 0.71, demonstrating its efficacy in sternum fracture identification. The authors [24] highlight the rising incidence of bone illnesses, which calls for a precise and prompt diagnosis, especially in cases of fractures. Missed fractures can cause complications and hold up treatment. Deep learning in particular is gaining popularity as a tool to help radiologists identify fractures on radiographs. The review emphasizes CNN-based models as being especially good in this aspect, particularly InceptionNet and XceptionNet. Although the potential of DL is acknowledged, a considerable obstacle still exists due to the lack of labeled training data. Instead of trying to replace radiologists, this essay tries to increase their effectiveness and maintain their place in the diagnostic process. Despite the potential of DL, radiologists' worries about their jobs' security prevent its broad incorporation into the field of radiology

3. Methodology

This section includes data collection, augmentation of data using transformations of the image, and finally classification of healthy and bone using deep CNN and RCNN. The experiment has been performed on the bone X-ray image data sets, collected from different sources publicly available for research from the Kaggle. The fracture of the healthy bone, and dataset information are shown in Table I.

Table I. Dataset Specifications.

Class	Train	Test	Total
Fractured	4480	360	4840
Not Fractured	4383	240	4623
Total	8863	600	9463

CNN and RCNN Models Architecture Proposal a deep convolution neural network model has been created in our suggested work. Convolution, pooling, flattening, and dense layers are included [22]. CNN automatically extracts features from the input image and uses a fully connected layer to identify them as fractured or healthy bone. The features of the image are extracted using the pooling and convolution layers [23]. To reduce noise, a suitable size of 3x3 is applied to each of the convolution and pooling layers. The RCNN model is a computer vision model used for object recognition and localization. It was one of the first architectures to use convolutional neural networks (CNNs) in conjunction with region proposal approaches to recognize objects in images. The image is initially processed with a selective search algorithm or a similar region suggestion technique. This provides a set of region recommendations, which are potential bounding boxes for objects. A fixed-size image patch is retrieved and scaled to a consistent input size for each suggested location. Following that, a pre-trained CNN is utilized to extract high-level features from each region. This stage aids in the collection of discriminative information about the objects contained inside the proposed regions. Each region's collected features are put into a network of fully linked layers that perform object classification. To forecast the existence of different classes within each region, these layers are often followed by a softmax activation function. RCNN predicts the precise coordinates of the bounding box for each detected object in addition to classification. This is accomplished by training a regressor to change the coordinates of the initial region proposal to better fit the object. The projected bounding boxes may have overlapped after categorization and bounding box regression. To reduce duplicate detections and choose the most confident ones, non-maximum suppression is used.

4. Experimental Results

This section presents the evaluation of the proposed models based on widely used metrics in DL to provide comparable results with other models. To reach better results, we built other codes based on different deep learning algorithms, which have been reported in other studies to give high accuracy in multi-classification images. Initially, we run these codes with different parameters to compare their results. Our model can only produce two right or erroneous predictions when dealing with binary classification jobs when there are only two classes. The counts of test records that the classification model correctly and erroneously predicted are used to assess the performance of the model. A more accurate view of a predictive model's performance, including which classes are being forecasted correctly and erroneously and what kinds of errors are being made, can be obtained from the confusion matrix. Accuracy, Recall, Precision, and F1-Score evaluation measures are used to estimate the performance output as follows:

Accuracy is the ratio of True Positives to all the predictions.

$$\text{Accuracy}=(TP+TN)/(TP+TN+FP+FN) \quad (1)$$

The recall is the ratio of True Positives to all the positives in the Dataset.

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

The Precision is the ratio of True Positives to all the positives predicted by the model.

$$\text{Precision}=TP/(TP+FP) \quad (3)$$

F1-Score provides a single score that balances both the concerns of precision and recall in one number. A good F1 score means that you have low false positives and low false negatives.

$$\text{F1-Score}=2/(1/\text{Recall}+1/\text{Precision}) \quad (4)$$

Where:

TP: True Positive is the observation is positive, and the prediction is positive. Label depressed and prediction depressed.

TN: True Negative is the observation is negative, and the prediction is negative. The label is not depressed and the prediction is not depressed.

FP: False Positive is that the observation is negative but positive for the prediction. The label is not depressed and the prediction is depressed.

FN: A False Negative is an observation that is positive but negative for a prediction. The label is depressed and the prediction is not depressed.

Using multiple optimization methods in distinct trials, such as Adam and SGD (Stochastic Gradient Descent), is a popular approach in machine learning and deep learning research. These techniques are used to adjust a neural network's weights and biases during training to minimize the loss function and improve the model's performance. Adam (Adaptive Moment Estimation): Adam is a learning rate optimization approach that combines the advantages of AdaGrad and RMSProp. It keeps an exponentially declining average of previous squared gradients and previous gradients, modifying the learning rate for each parameter separately. Adam is popular because of its effectiveness and efficiency in a wide range of jobs. To train a neural network on a specific dataset, run an experiment with the Adam optimizer. when working on an image classification problem with a convolutional neural network (CNN), Adam is the optimization method, and measuring the model's accuracy and loss during training. This experiment may include hyperparameter adjustment to determine the best learning rate, beta1, and beta2 values for your particular task.

SGD (Stochastic Gradient Descent): SGD is a basic optimization technique used in neural network training. It makes modest steps in the opposite direction of the gradient of the loss function to update the model's parameters. It employs a preset learning rate that must be manually configured. In a different experiment, the same neural network may be trained on the same dataset using the SGD optimizer. need to decide on an acceptable learning rate and maybe experiment with different learning rate plans, such as learning rate decline. To get better results for our model we have changed the parameters of the algorithms; Where we used Adam and SGD optimizers and changed the batch size and number of epochs. We compared the results obtained from each experiment as follows:

A. *Experiment I*

Table II compares the performance of various deep learning algorithms under diverse training conditions, specifically using different optimizers (Adam and SGD) and maintaining the number of epochs constant at 5. When using the Adam optimizer and a batch size of 32, the RCNN algorithm consistently outperformed the others. The RCNN algorithm achieved remarkable scores of 98%, 96%, 96%, and 96% in the Accuracy, Precision, Recall, and F1-score parameters, respectively. Interestingly, even after switching the optimizer to SGD, the RCNN algorithm retained its advantage over the other algorithms, producing favorable results of 93%, 91%, 89%, and 88% for the same criteria. This demonstrates the RCNN architecture's robustness across different optimization methodologies.

Table II. The Results of Experiment I With Optimizer Adam And Sgd

Algorithms	Adam				SGD			
	Accuracy	*Precision*	*Recall*	*F1-score*	*Accuracy*	*Precision*	*Recall*	*F1-scor*
CNN	97	91	90	89	25	6	25	10
RCNN	98	96	96	96	93	91	89	88

B. *Experiment II*

When we increased the number of epochs to ten and the batch size to 32, the RCNN with the Adam optimizer, continued to outperform other algorithms. It achieved impressive results of 98%

Accuracy, 97% Precision, 97% Recall, and 97% F1 score. When the optimizer was changed to SGD, the RCNN algorithms outperformed the others, with significant results of 98% Accuracy, 97% Precision, 98% Recall, and 97% F1-scor, as Table III.

Table III. The Results of Experiment Ii with Optimizer Adam And Sgd

Algorithms	**Adam**				**SGD**			
	Accuracy	***Precision***	***Recall***	***F1-scor***	***Accuracy***	***Precision***	***Recall***	***F1-scor***
CNN	97	94	94	94	25	6	25	9
RCNN	98	97	97	97	97	99	98	98

C. Experiment III

In the following experiment, we investigated the influence of a larger batch size of 64 with the same number of epochs 5. The RCNN algorithm maintained its supremacy over the Adam optimizer, achieving 99% Accuracy, 91% Precision, 87% Recall, and 87% F1 score. Even after switching to the SGD optimizer, the RCNN algorithm outperformed its competitors, with scores of 85%, 84%, 78%, and 75% as Table IV.

Table IV. The Results of Experiment Iii with Optimizer Adam And Sgd

Algorithms	**Adam**				**SGD**			
	Accuracy	***Precision***	***Recall***	***F1-scor***	***Accuracy***	***Precision***	***Recall***	***F1-scor***
CNN	97	99	99	99	24	6	25	10
RCNN	99	91	87	87	85	84	78	75

D. Experiment IV

Finally, by extending the experiment to 10 epochs while keeping the batch size at 64, the RCNN algorithm's prowess with the Adam optimizer remained pronounced, yielding impressive results of 99% Accuracy, 97% Precision, 97% Recall, and 97% F1-scor. Notably, while using the SGD optimizer, both the RCNN algorithm maintained their better performance trends, providing results of 98% Accuracy, 97% Precision, 98% Recall, and 97% F1-scor, as Table V.

Table V. THE RESULTS of EXPERIMENT IV WITH OPTIMIZER Adam and SGD

A	**l**	**Adam**	**SGD**

	Accuracy	*Precision*	*Recall*	*F1-scor*	*Accuracy*	*Precision*	*Recall*	*F1-scor*
CNN	97	95	95	95	25	6	25	10
RCNN	99	97	97	97	98	97	98	97

E. Results Analysis

The supplied results show a comprehensive assessment of numerous deep learning algorithms in diverse training circumstances, illuminating their performance across various optimization techniques, batch sizes, and epochs. Here is a debate to put these results into context:

I. Impact of optimization: The study thoroughly evaluated the impact of various optimizers, in particular Adam and SGD, on the effectiveness of deep learning algorithms. It is noteworthy that the RCNN algorithm continuously beats its rivals while using either optimization technique. This suggests that the RCNN design has inherent properties that enable it to adapt to various optimization techniques. This useful quality of adaptability implies that the RCNN model might be suitable for a range of applications and datasets.

II. *Robustness of the RCNN:* A notable finding is the RCNN model's resilience. The RCNN method continued to outperform the other models even after the optimizer was changed from Adam to SGD. This suggests that the RCNN design is not unduly dependent on a particular optimization technique, which is advantageous in real-world situations where the optimizer of choice may vary.

III. Effect of Batch Size and Epochs: The RCNN's superiority was further demonstrated by increasing the number of epochs to ten while retaining a batch size of 32, especially when combined with the Adam optimizer. High ratings were routinely attained by the RCNN in all performance metrics. This implies that the RCNN gains from extensive training, enabling it to more effectively recognize complicated patterns in the data. The RCNN's performance was also unaffected by the batch size selection, showing that it can handle bigger batches without reducing accuracy.

IV. Batch Size Susceptibility: The RCNN algorithm continued to rule, especially with the Adam optimizer, even with a greater batch size of 64. This is a critical realization since higher batch sizes might occasionally result in worse model performance. A notable feature of the RCNN is its capacity to handle bigger batch sizes while retaining excellent accuracy.

V. Performance Consistency: The RCNN consistently displayed outstanding performance throughout all experiments, particularly in terms of accuracy, precision, recall, and F1 score. The model's dependability and suitability for the task at hand are strongly suggested by this level of consistency.

VI. Practical Implications: The results indicate that the RCNN model is a good competitor in situations where adaptability to various optimization approaches and batch sizes is essential. It is therefore a flexible option for practical applications when the optimum training environment might not always be available.

As a result, the study's thorough assessment of the RCNN model under various training circumstances demonstrates its resilience and adaptability. These findings emphasize the significance of taking into account model performance under various training setups and offer useful insights for practitioners and researchers working on related deep-learning problems.

5. Conclusions

The results of the experiments described in the table provide important insights into the performance of various CNN, and RCNN algorithms when applied to the problem of detecting bone fractures. These findings can be used to help guide the development of algorithms and optimization strategies for accurate and dependable bone fracture detection systems. After incorporating an easy-to-use user interface, it should be able to work well in healthcare institutions to identify bone fractures. Finally, the findings of this study highlight the need to choose proper optimization algorithms and hyperparameters for deep learning-based bone fracture detection systems. The improved performance of algorithms trained with the Adam optimizer shows that it is appropriate for this purpose. More study and experimentation, however, are required to fine-tune hyperparameters and corroborate identified trends across bigger and more diverse datasets. Deep learning algorithms, with a refined approach, have the potential to provide reliable and efficient solutions for bone fracture identification, contributing to enhanced medical diagnostics and patient care.

Supplementary Materials

Not applicable.

Author Contributions

All Authors contributed equally to this work.

Funding

This research was conducted without external funding support.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Conflicts of Interest

The authors declare that there is no conflict of interest in the research.

References

[1] International Osteoporosis Foundation. "Broken Bones, Broken Lives: A Roadmap to Solve the Fragility Fracture Crisis in Europe." International Osteoporosis Foundation, 2018. Available online: http://share.iofbonehealth.org/EU-6-Material/Reports/IOF%20Report_EU.pdf (accessed on 21 February 2020).

[2] L. Tanzi, E. Vezzetti, R. Moreno, and S. Moos, "X-ray bone fracture classification using Deep Learning: A baseline for designing a reliable approach," Applied Sciences, vol. 10, no. 4, p. 1507, 2020. doi:10.3390/app10041507

[3] D. B. Burr, "Introduction - bone turnover and fracture risk," J. Musculoskelet. Neuronal Interact, vol. 3, no. 4, pp. 408-409, 2003.
[4] D. P. Yadav and S. Rathor, “Bone Fracture Detection and classification using Deep Learning Approach,” 2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), 2020. doi:10.1109/parc49193.2020.236611
[5] Y. D. Pranata, K. Wang, J. Wang, I. Idram, J. Lai, J. Liu, and I. Hsieh, "Deep learning and SURF for automated classification and detection of calcaneus fractures in CT images," Comput Methods Progr Biomed, vol. 171, pp. 27-37, 2019.
[6] T. Urakawa, Y. Tanaka, S. Goto, H. Matsuzawa, K. Watanabe, and N. Endo, "Detecting intertrochanteric hip fractures with orthopedist-level accuracy using a deep convolutional neural network," Skeletal Radiol, vol. 48, no. 2, pp. 239-244, 2019.
[7] Y. Ma and Y. Luo, “Bone fracture detection through the two-stage system of crack-sensitive convolutional neural network,” Informatics in Medicine Unlocked, vol. 22, p. 100452, 2021. doi:10.1016/j.imu.2020.100452
[8] X. Lin, Z. Yan, Z. Kuang, H. Zhang, X. Deng, and L. Yu, "Fracture R-CNN: An anchor-efficient anti-interference framework for skull fracture detection in CT images," Med. Phys., 2022. [CrossRef]
[9] G. Kitamura, C.Y. Chung, and B.E. Moore, "Ankle fracture detection utilizing a convolutional neural network ensemble implemented with a small sample, de novo training, and multiview incorporation," J. Digit. Imaging, vol. 32, pp. 672-677, 2019.
[10] L. Yang, S. Gao, P. Li, J. Shi, and F. Zhou, "Recognition and Segmentation of Individual Bone Fragments with a Deep Learning Approach in CT Scans of Complex Intertrochanteric Fractures: A Retrospective Study," J. Digit. Imaging, vol. 2022, pp. 1-9.
[11] E. Yahalomi, M. Chernofsky, and M. Werman, "Detection of distal radius fractures trained by a small set of X-ray images and Faster R-CNN," in Intelligent Computing-Proceedings of the Computing Conference, Springer, 2019, pp. 971–981.
[12] W. Abbas, S. M. Adnan, M. A. Javid, W. Ahmad, and F. Ali, "Analysis of tibia-fibula bone fracture using deep learning technique from X-ray images," Int. J. Multiscale Comput. Eng., vol. 19, pp. 25–39, 2021.
[13] J. Luo, G. Kitamura, E. Doganay, D. Arefan, and S. Wu, "Medical knowledge-guided deep curriculum learning for elbow fracture diagnosis from X-ray images," in Medical Imaging 2021: Computer-Aided Diagnosis, International Society for Optics and Photonics, 2021, vol. 11597, p. 1159712.
[14] S. Beyaz, K. Açıcı, and E. Sümer, "Femoral neck fracture detection in X-ray images using deep learning and genetic algorithm approaches," Jt. Dis. Relat. Surg., vol. 31, p. 175, 2020.
[15] R. M. Jones et al., "Assessment of a deep-learning system for fracture detection in musculoskeletal radiographs," NPJ Digit. Med., vol. 3, pp. 1–6, 2020.
[16] F. Hardalaç et al., "Fracture Detection in Wrist X-ray Images Using Deep Learning-Based Object Detection Models," Sensors, vol. 22, p. 1285, 2022.
[17] Y. D. Pranata et al., "Deep learning and SURF for automated classification and detection of calcaneus fractures in CT images," Comput. Methods Programs Biomed., vol. 171, pp. 27–37, 2019.
[18] S. Mutasa, S. Varada, A. Goel, T. T. Wong, and M. J. Rasiej, "Advanced deep learning techniques applied to automated femoral neck fracture detection and classification," J. Digit. Imaging, vol. 33, pp. 1209–1217, 2020.
[19] T. Weikert et al., "Assessment of a deep learning algorithm for the detection of rib fractures on whole-body trauma computed tomography," Korean J. Radiol., vol. 21, p. 891, 2020.
[20] L. Tanzi et al., "Hierarchical fracture classification of proximal femur X-ray images using a multistage deep learning approach," Eur. J. Radiol., vol. 133, p. 109373, 2020.
[21] D. C. Ciresan, U. Meier, J. Masci, L. Gambardella, and J. Schmidhuber, "Flexible high performance convolutional neural networks for image classification," in Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Catalonia, Spain, July 16–22, 2011, vol. 22, p. 1237.

[22] R. Ebsim, J. Naqvi, and T. F. Timothy, "Automatic Detection of Wrist Fractures From Posteroanterior and Lateral Radiographs: A Deep Learning-Based Approach," in International Workshop on Computational Methods and Clinical Applications in Musculoskeletal Imaging, Springer, Cham, 2018, pp. 114-125.
[23] Y. Jia, H. Wang, W. Chen, Y. Wang, and B. Yang, "An attention-based cascade R-CNN model for sternum fracture detection in x-ray images," CAAI Transactions on Intelligence Technology, vol. 7, no. 4, pp. 658–670, 2022. doi:10.1049/cit2.12072
[24] T. Meena and S. Roy, "Bone fracture detection using deep supervised learning from Radiological Images: A paradigm shift," Diagnostics, vol. 12, no. 10, p. 2420, 2022. doi:10.3390/diagnostics12102420

Received: 02 Sep 2023, **Revised:** 01 Jan 2024,
Accepted: 27 Jan 2024, **Available online:** 31 Jan 2024.