

# Developing Convolutional Neural Network for Recognition of Bone Fractures in X-ray Images

Aymen Saad<sup>1,2\*</sup>, Usman Ullah Sheikh<sup>2</sup>, Mortada Sabri Moslim<sup>3</sup>

<sup>1</sup> Department of Information Technology, Technical College of Management, Al-Furat Al-Awsat Technical University, Kufa, Iraq

<sup>2</sup> Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Johor, Malaysia

<sup>3</sup> Department of Medical Physics Sciences, Al-Mustaqbal University, Hilla, Iraq

\* Corresponding author's e-mail: [aymen.abdalameer@atu.edu.iq](mailto:aymen.abdalameer@atu.edu.iq)

## ABSTRACT

In the domain of clinical imaging, the exact and quick identification proof of bone fractures plays a crucial part in a pivotal role in facilitating timely and effective patient care. This research tends to this basic need by harnessing the force of profound learning, explicitly utilizing a convolutional neural network (CNN) model as the foundation of our technique. The essential target of our study was to improve the mechanized recognition of bone fractures in X-ray images, utilizing the capacities of deep learning algorithms. The use of a CNN model permitted us to successfully capture and learn intricate patterns and features within the X-ray images, empowering the framework to make exact fracture detections. The training process included presenting the model to a various dataset, guaranteeing its versatility to an extensive variety of fracture types. The results of our research show the excellent performance of the CNN model in fracture detection, where our model has achieved an average precision of 89.5%, an average recall of 87%, and an overall accuracy of 91%. These metrics assert the vigour of our methodology and highlight the capability of deep learning in medical image analysis.

**Keywords:** convolutional neural network, X-ray images, deep learning algorithm, medical image analysis.

## INTRODUCTION

As of late, the intersection of medical imaging and AI has introduced a new period in medical care, promising huge headways in diagnostic accuracy [1]. One basic area of concentration inside this field is the automatic detection of bone fractures in X-ray images. As a crucial part of medical diagnosis, X-ray assessments give priceless bits of knowledge into the inner structures of the human body. Nonetheless, the interpretation of X-ray images, especially the detection of bone fractures, is a complex and tedious errand that frequently requires the mastery of talented radiologists. The integration of deep learning techniques in X-ray image analysis offers a promising answer for upgrading the effectiveness and exactness of bone fracture detection [2]. This research

attempts to add to the continuous endeavours in utilizing deep learning methods for the acknowledgment of bone fractures in X-ray images.

The pervasiveness of breaks as an outcome of injury, osteoporosis, and other ailments requires an effective framework for their early identification. Manual assessment of X-rays, however completed via seasoned professionals, can be inclined to human blunder and is restricted by the requirements of time and resources [3]. The use of deep learning models aims to mitigate these difficulties via automating the fracture detection process.

The essential inspiration for this study lies in the possibility to change the area of radiology by harnessing the capacities of deep learning. Convolutional neural networks (CNNs) have exhibited excellent performance in image recognition,

and their application to clinical imaging, including the understanding of X-rays, has shown promising outcomes [4]. By training these neural networks on assorted datasets of X-ray images portraying different sorts of fractures, the model can learn complex patterns and features related to various fracture types [5]. The goal is to develop a model that distinguishes fractures precisely as well as generalizes well to deal with a large number of scenarios, across different patient populations. The utilisation of deep learning for the detection of bone fractures in X-ray images addresses a huge step towards working on the effectiveness of diagnostic processes in radiology.

## BACKGROUND

The researchers in [6] discussed a fracture detection method using a new deep learning method called dilated convolutional feature pyramid network (DCFPN). The method outperform FPN with 82.1% average precision in identifying thigh breaks. The DCFPN uses dilated convolutions in the backbone network and a feature pyramid network for multi-scale feature extraction. The network design incorporates a region-based convolutional neural network for exact detection. The proposed method shows solid potential for practical clinical applications.

The proposed methodology described in [7] for the detection of arm fracture in X-ray images is based on an improved two-stage R-CNN method. The methodology incorporates image pre-processing, a novel backbone network, region proposal network, receptive field adjustment, RoI pooling layer, and final classification. The image pre-processing includes morphological operations and pixel value transformation to improve image quality. As far as results, the proposed deep learning methodology achieved an average precision (AP) of 62.04%. The system used a 3392 images for training and 612 images for testing.

The framework portrayed in [8] is called ParallelNet, which is intended for identifying thigh bone breaks in X-ray images. ParallelNet comprises of different backbone networks with dilated convolution in every pathway. The backbone networks create include maps with various reception fields, and a backward connection is utilized to interface feature maps from various stages. Moreover, a pyramid network structure is used to produce features including maps for fracture

detection in different scales. The region proposal network (RPN) is utilized to detect ROIs (regions of interest) which contain fractures. As far as results, the analyses on the thigh fracture dataset showed promising results. The ParallelNet accomplished a normal accuracy (AP) of 87.8% for AP50 and 49.3% for AP75, outperforming the performance of other algorithms such as DCFPN.

The framework proposed in [9] comprises a two-stage approach for identifying bone fractures in X-ray images. The principal stage includes utilizing Faster R-CNN to distinguish various kinds of bones in the X-ray images by segmenting and classifying them. The second stage uses a crack-sensitive convolutional neural network (CrackNet) to distinguish conceivable break districts inside the identified bones. This two-stage framework means to ease the weight on specialists by giving possible fracture areas in X-ray images. As far as results, the proposed framework accomplished an accuracy of 88.39%, a recall of 87.5%, and a precision of 89.09% on the bone fracture detection task.

The framework described in [10] depends on the ResNeXt+FPN structure. The model comprises an initial encoder (ResNeXt) trailed by FPN that gathers feature maps in various scales to create the features. The feature maps delivered by FPN are then utilized by the Region Proposal Decoder (RPN) to make predictions on the entire image. Subsequently, the RoI-Align operator crops the global features in the global box predictions to produce local features for detection. The local features are then processed by RCNN. As far as results, the feature ambiguity mitigate operator (FAMO) model was introduced to enhance the system performance by moderating feature ambiguity in bone fracture detection. The results showed improvements in different metrics with FAMO. For example, the average precision increased by 0.6%. Also, FAMO led to enhancements in sensitivity, specificity, and AUC, with improvements ranging from 1.7% to 2.6%.

The framework described in [11] includes the assortment of a Femoral X-Ray Image Dataset and the improvement of an object detection strategy for classification types of fracture. An object detection model is prepared to find fractures inside the images, accomplishing a mean Average Precision of 68.8 and 71.5% accuracy. The framework structure in [12] includes the utilization of a Faster R-CNN based on CNN for fracture location on wrist radiographs. The study utilized an Inception-ResNet Faster R-CNN model for training and testing on a dataset of wrist

radiographs. The results showed that the model accomplished a high degree of accuracy in identifying fractures on both front-facing and lateral-facing, with sensitivity going from 91.2% to 98.1% and specificity going from 72.9% to 86.4%.

## CONVOLUTION NEURAL NETWORKS

CNN is considered one of the most important DL algorithms and is very similar to the multi-layer perceptron network [13]. It is capable of evaluating images, identifying objects in them, as well as classifying them. It is also capable of performing tasks related to analysing pixel information [14]. The CNN performance changes depending on the layers used to build the model, their characteristics, and the way they are distributed, in addition to the parameters of each layer [15]. A CNN model can consist of a different set of layers, including the following:

1. Convolutional layers – this process involves applying a mathematical operation to two functions,  $f$  and  $g$ , resulting in a modified function, denoted as  $o$ , which highlights the overlapping region between the original functions. In image processing, convolutional layers apply this process on the input using a set of filters or convolution kernels. The convolution process is executed on each group of input elements with a specific filter size, generating individual values for feature maps. This operation is then repeated for each filter, producing a distinct set of feature maps. These convolutional layers play a crucial role in image classification by extracting relevant features from images, facilitating the recognition and classification of diverse visual patterns. Following the completion of the convolution process, an activation function is applied to the values within the feature maps [16].
2. Subsampling layers – the model's architecture allows for the addition of these optional layers, placed after each convolutional layer if present. These layers are utilized for reducing the number of neurons, as it will reduce each group of input neurons of a certain size to one neuron. The size of the group is determined dependence on the experimental results, but in a manner. In general, the smaller value of the group size,  $2 \times 2$ , is considered the best value because increasing the group size leads to a loss of information, and at the same time reducing the group size consumes a greater

response time. Therefore, experimental results are relied upon to balance the model's accuracy and the response time related to the group size [17]. The number of neurons within each group is reduced to one neuron in one of two ways:

- max-pooling – it selects the highest weight value between all neurons of one group;
  - average-pooling – it selects the arithmetic average of the weights for the neurons of one group.
3. Fully connected layers – after building a layer or several layers of the previous two types, this type of layer connects all the neurons from the previous layer, regardless of their type, and makes them input to each neuron, as in regular neural networks such as Multilayers Perceptron. This type of layer is often added at the end of the layers, and usually two successive layers of this type are added as the last layers, as this layer cannot come before a convolution-type layer [18].

The key to success for CNNs is that each layer contains fewer parameters than the layer before it, so when the algorithm reaches the end it can learn as much information about the data as if this data were taken all at once [18]. Instead, gradual analysis of parts is followed. Smaller amount of data at each step. This allows the model to catch the complex and detailed relationships within the dataset with greater accuracy. This allows the model to learn by extracting finer features of images or videos.

## METHODOLOGY

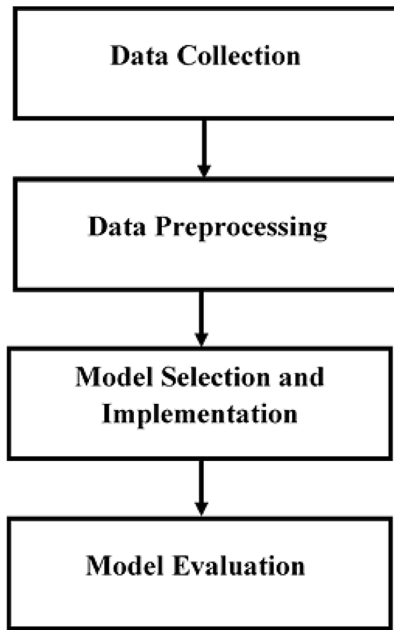
The proposed methodology is based on building an artificial neural network of the CNN type to detect bone fractures. Figure 1 represents the general structure of the proposed methodology.

### Data collection

We used a dataset including 9103 X-ray images. The data set contains two classes – the first is Fractured class, and the second is Not Fractured class.

### Data pre-processing

This stage involves a set of image processing operations that are applied to the images within the dataset. Used in the context of image data augmentation, these operations are used to artificially increase the diversity of the training data set by



**Figure 1.** The general structure of the proposed methodology

applying different transformations to the images [19]. The goal is to improve the generalizability and robustness of the model. Below we explain the pre-processing operations we applied to the images:

- rescale – this parameter scales the pixel values of images. It is common practice to rescale pixel values to a range between 0 and 1 by dividing each pixel value by 255. This normalization helps the neural network converge faster;
- rotation – randomly rotates the image by a specific angle, in this case, up to 40 degrees;
- width-shift and height-shift – randomly changes the width and height dimensions of the image by a small fraction of the total width or height, in this case, up to 20%;
- shea – applies the shear transform with a maximum shear intensity of 20%;
- zoom – randomly zooms the image by a factor of up to 20%;
- horizontal-flip – randomly flips images horizontally;
- fill-mode = nearest – this parameter defines the filling strategy for newly created pixels that may appear after rotation or transformation. “nearest” means it will use the value of the nearest pixel to fill the empty space.

In general, these augmentation techniques aim to expose the model to a variety of transformations

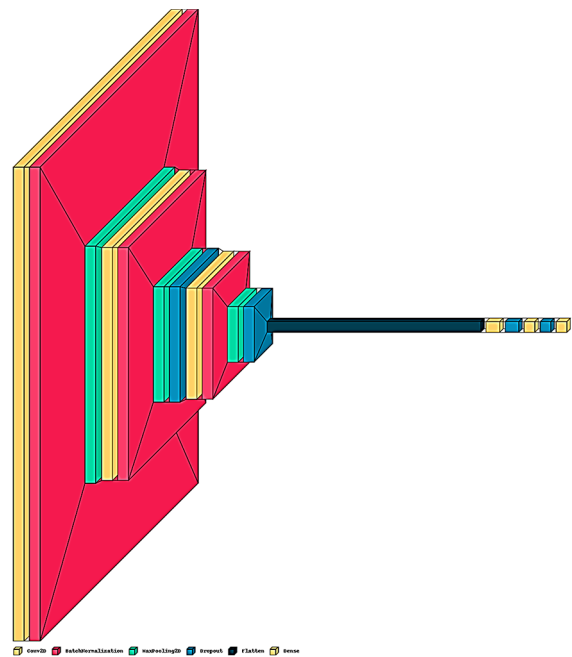
that mimic real-world scenarios, making it more robust and better able to handle different shapes of input data during training.

### Model design

In this research, a model was developed to detect bone fractures using deep learning techniques. The Keras libraries in the Python programming language were relied upon to design the proposed model. The experimental approach was adopted to arrive at the structure of the proposed model. Figure 2 shows the structure of the proposed model. Figure 3 shows the detailed structure of the proposed network This typically follows the typical architecture shown in Figure 3 for image classification tasks using CNNs, with convolutional layers and pooling layers except for feature extraction, and then fully differentiating layers for classification. The forward-looking view with a single unit and sigmoid activation function suggests a binary classification, where the probability of belonging to one of the two categories is predicted.

### Model evaluation

We employed K-fold cross-validation which is a resampling method utilized for assessing the model performance. This method divides the data set into k folds, the model is then trained on  $k-1$



**Figure 2.** The structure of the proposed model

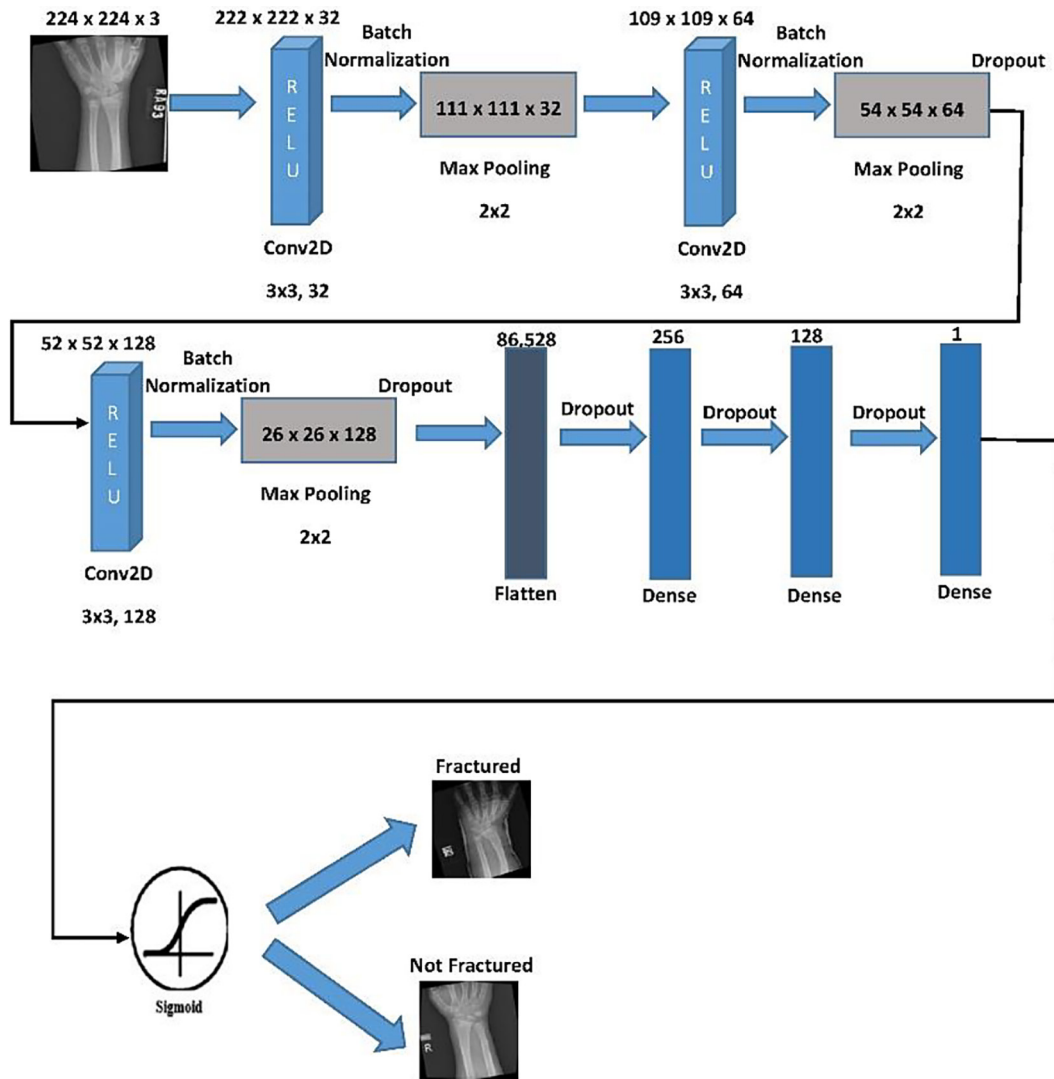


Figure 3. The detailed structure of the proposed network

folders, and evaluated on the remaining fold [20]. The confusion matrix (CM) which referred to an error matrix [21] is a table layout designed to represent the performance of a model, often one employed in supervised learning [22]. The architecture of the CM is illustrated in Figure 4. These parameters are used to calculate recall, precision and accuracy.

$$Recall = \frac{TP}{TP+FN} \tag{1}$$

Equation 1 represents for all classes which are positive, how many correctly predicted classes [23].

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

Equation 2 represents from all predicted positive classes, how many are positive actually [24]. Precision should be as high as possible. The

accuracy is defined as the percentage of correct predictions that a trained ML model achieves [25]. The accuracy is given in Eq. 3.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{3}$$

The F1\_Score serves as a metric that amalgamates precision and recall, offering a well-rounded assessment of a model's performance. The formula is as in Eq. 4.

$$F1\_Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

The Matthews correlation coefficient (MCC) is a measure of the quality of binary classifications, particularly when dealing with imbalanced data sets. It takes into account true positives, true negatives, false positives, and false negatives [26]. The MCC can be calculated using the formula:

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

**Figure 4.** Confusion matrix, where: TP (true positive): is when the model makes correct positive prediction, TN (true negative) – is when the model makes correct negative prediction, FP (false positive) – when the model makes an incorrect positive prediction and FN (false negative) – when the model makes an incorrect negative prediction

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (5)$$

The MCC ranges from -1 to 1, where 1 indicates perfect prediction, 0 indicates random prediction, and -1 indicates total disagreement between prediction and observation [26].

## RESULTS AND DISCUSSION

A learning curve is a graphical representation showing the advancement of a specific metric throughout the training stage of a machine learning model. These curves show the time or progress on the x-axis and error or performance on the y-axis. These graphs are instrumental in checking the improvement of a model during the learning stage, empowering the detection of issues and the enhancement of prediction performance. A prominent example of a learning curve is the plot of loss over time, where loss measures the model’s error, reflecting its effectiveness. As the loss decreases, the model’s performance improves.

Another widely employed learning curve is the accuracy curve, which, like loss, gauges model performance. Higher values on these curves signify enhanced model capabilities. Training loss assesses the model’s fit to the training data, while validation loss evaluates its performance on new, unseen data. Two prevalent types of learning

curves are optimization learning curves, derived from metrics optimizing the model’s parameters (e.g., loss), and performance learning curves, based on metrics used for model evaluation and selection (e.g., accuracy). Figure 5 shows the accuracy and loss curves for both the training and validation processes.

We note that Figure 5 shows that our model has become smarter. Both training and validation metrics are on an upward trajectory, meaning that the model learns from the data and does well on both familiar (training) and unfamiliar (validation) examples. Positive gradients of both loss and accuracy functions across epochs indicate successful convergence toward a minimum and improved generalization, respectively. Both reduced validation loss and increased validation accuracy indicate that the model successfully transfers its knowledge from the training data to new instances. Therefore, we avoid a common pitfall called “overfitting,” where the model becomes so good at simulating the training data that it has difficulty handling anything new. Overall, Figure 5 paints a positive picture: our model learns effectively and can handle unseen data, paving the way for reliable predictions. The training process is considered to have stopped at epoch 11, which achieved the highest accuracy of 95% for the training data and 98% for the validation data.

Now we will review the results achieved by applying the proposed model to the third set of data, which represents the test data, which is considered new data that is not visible to the model. Figure 6 shows the CM results.

The results of the CM indicate that the model correctly predicted 694 samples from the ‘Fractured’ class (0), and the model correctly predicted 195 samples from ‘Not-Fractured’ class (1). In addition, the model incorrectly predicted 54 samples as being from class 0 while they were from class 1, it also incorrectly predicted 32 samples as being from class 1 while they were from class 0. Table 1 shows the values of Precision, Recall, and F1-Score for each category.

We note from Table 1 that concerning bone fractured, for precision (0.93): The model is good at correctly identifying fractures when it predicts them. There’s a 93% chance that a positive prediction (fracture) by the model is actually a true fracture. For recall (0.96): The model captures a high percentage of actual fractures. It correctly identifies 96% of all the

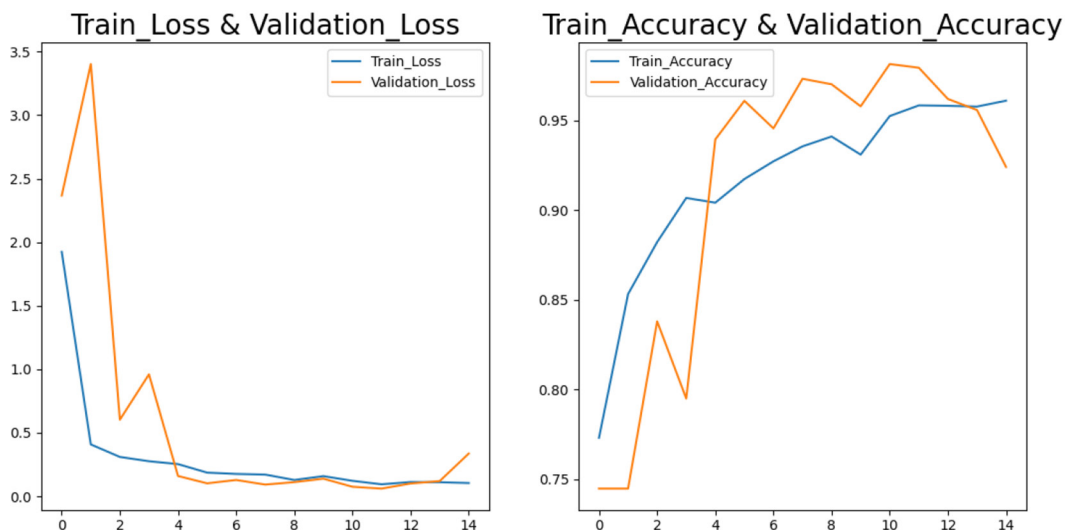


Figure 5. The accuracy and loss curves for both the training and validation processes

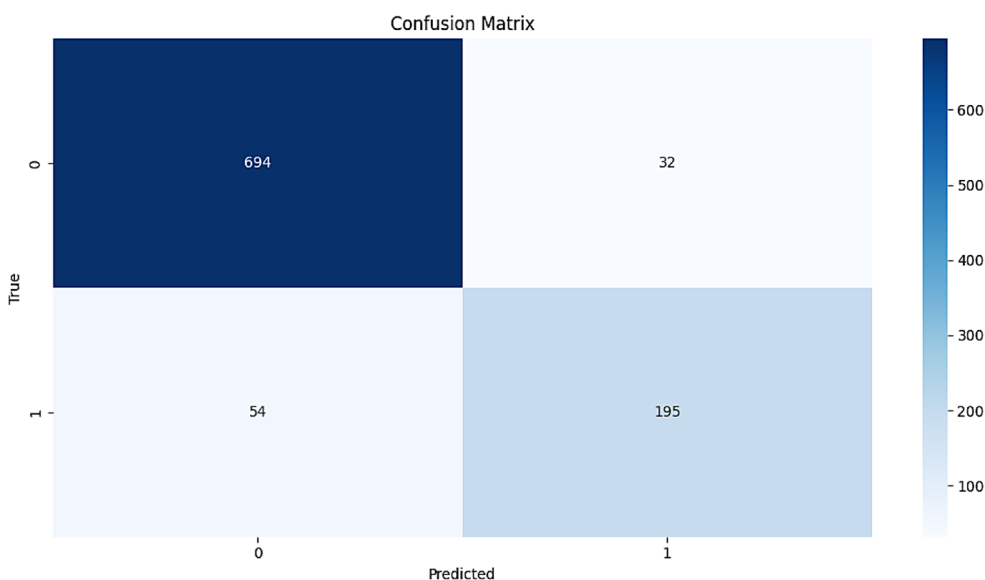


Figure 6. The CM results

actual fractures present in the data. For F1-Score (0.94): combining the above, the model performs well in both aspects for the “bone fractured” class. On the other hand, concerning bone not-fractured, for precision (0.86): The model is moderately good at identifying non-fractures when it predicts them. There’s an 86% chance that a negative prediction (not-fractured) by the model is a true non-fracture.

For recall (0.78): The model misses a higher proportion of actual non-fractures compared to fractures. It only correctly identifies 78% of all actual non-fractures present in the data. For F1-score (0.82): Due to the lower recall, the overall performance for the “bone not-fractured” class is slightly lower compared to the “bone fractured” class and the overall F1-score.

Table 1. The values of precision, recall, and F1-Score for each category

Category	Precision	Recall	F1-Score	MCC
Bone fractured	0.93	0.96	0.94	0.762
Bone not-fractured	0.86	0.78	0.82	

**Table 2.** Performance comparison of the proposed model with a group of previous studies

Reference	Technique	Performance
[6]	DCFPN	AP = 82.1%
[7]	Improved R-CNN	AP = 62.04%
[8]	ParallelNet	AP = 87.8%
[9]	Faster R-CNN & CrackNet	Precision = 89.09%, Recall = 87.5%, Accuracy = 88.39%
[11]	Deep Learning	AP = 68.8%, Accuracy = 71.5%
Proposed model	CNN	AP = 89.5%, Recall = 87%, Accuracy = 91%, MCC=0.762

Overall, the model shows promising performance in classifying bone fractures, particularly in terms of identifying actual fractures. However, there's room for improvement in correctly classifying non-fractures. Based on the results shown in Table 1, we can reduce the overall performance of our model by calculating the arithmetic average of precision, recall and F1-score and then calculating the overall accuracy of the system. The following is the model's overall performance. Precision (0.895): this metric, at 0.895, indicates that out of all the cases the model predicted as positive (fractures), 89.5% were true fractures. In other words, for every 100 positive predictions, the model was right about 89.5 of them. Recall (0.87): this metric, at 0.87, signifies that the model captured 87% of the actual fractures present in the data. Conversely, it missed 13% of the actual fractures. F1-score (0.88): This metric, at 0.88, is the harmonic mean of precision and recall, providing a balanced view of the model's performance. It considers both the ability to correctly predict positive cases (fractures) and avoid false positives (predicting fractures when they're absent). And accuracy (0.91): this metric, at 0.91, represents the proportion of all predictions (both positive and negative) that the model got right. In this case, the model correctly classified 91% of the cases. A Matthews correlation coefficient of 0.762 suggests that our model has a strong ability to correctly classify instances into their respective categories (fracture vs. non-fracture). This indicates that our model's predictions are in strong agreement with the actual labels.

The model demonstrates moderately good overall performance in classifying bone fractures. It achieves a good balance between

precision and recall, as reflected by the F1-score close to 0.9. Additionally, the high accuracy of 0.91 suggests the model makes relatively few errors in both positive and negative predictions. By comparing the performance of the proposed model with a group of previous studies on the same topic, the results are shown in Table 2.

## CONCLUSIONS

In the realm of medical imaging, the accurate and swift identification of bone fractures plays a pivotal role in facilitating timely and effective patient care. This research addresses this critical need by harnessing the power of deep learning, specifically employing a convolutional neural network (CNN) model as the cornerstone of our methodology. In this work, we proposed a new deep-learning method for bone fracture detection in X-rays. The experiment results show that the proposed method could achieve average precision of 89.5%, average recall of 87%, average F1\_score of 88% and accuracy of 91% on bone fracture detection and it remarkably outperforms other state-of-the-art deep learning methods. If minimizing false positives is crucial (to avoid unnecessary interventions), exploring techniques to improve the model's ability to correctly classify non-fractures might be beneficial. Future research could delve into comparative analysis with the referenced techniques, ensuring consistent metrics and experimental setups for a fair evaluation. This would provide a clearer picture of the proposed model's relative strengths and weaknesses compared to existing approaches.



## REFERENCES

1. Edison G. Transforming medical decision-making: A comprehensive review of AI's impact on diagnostics and treatment. *BULLET: Jurnal Multidisiplin Ilmu*. 2023; 2(4): 1121–1133.
2. Joshi D., Singh T.P. A survey of fracture detection techniques in bone X-ray images. *Artificial Intelligence Review*. 2020; 53(6): 4475–4517. <https://doi.org/10.1007/s10462-019-09799-0>
3. Malone J. X-rays for medical imaging: Radiation protection, governance and ethics over 125 years. *Physica Medica: European Journal of Medical Physics*. 2020; 79: 47–64. <https://doi.org/10.1016/j.ejmp.2020.09.012>
4. Sarvamangala D.R., Kulkarni R.V. Convolutional neural networks in medical image understanding: A survey. *Evolutionary Intelligence*. 2022; 15(1): 1–22. <https://doi.org/10.1007/s12065-020-00540-3>
5. Zhang X., Xu J., Yang J., Chen L., Zhou H., Liu X., et al. Understanding the learning mechanism of convolutional neural networks in spectral analysis. *Analytica Chimica Acta*. 2020; 1119: 41–51. <https://doi.org/10.1016/j.aca.2020.03.055>
6. Guan B., Yao J., Zhang G., Wang X. Thigh fracture detection using deep learning method based on new dilated convolutional feature pyramid network. *Pattern Recognition Letters*. 2019; 125: 521–526. <https://doi.org/10.1016/j.patrec.2019.06.015>
7. Guan B., Zhang G., Yao J., Wang X., Wang M. Arm fracture detection in X-rays based on improved deep convolutional neural network. *Computers & Electrical Engineering*. 2020; 81: 106530. <https://doi.org/10.1016/j.compeleceng.2019.106530>
8. Wang M., Yao J., Zhang G., Guan B., Wang X., Yue-ming Z. ParallelNet: Multiple backbone network for detection tasks on thigh bone fracture. *Multimedia Systems*. 2021; 27: <https://doi.org/10.1007/s00530-021-00783-9>
9. Ma Y., Luo Y. Bone fracture detection through the two-stage system of Crack-Sensitive Convolutional Neural Network. *Informatics in Medicine Unlocked*. 2021; 22: 100452. <https://doi.org/10.1016/j.imu.2020.100452>
10. Wu H.-Z., Yan L.-F., Liu X.-Q., Yu Y.-Z., Geng Z.-J., Wu W.-J., et al. The feature ambiguity mitigate operator model helps improve bone fracture detection on X-ray radiograph. *Scientific Reports*. 2021; 11(1): 1589. <https://doi.org/10.1038/s41598-021-81236-1>
11. Qi Y., Zhao J., Shi Y., Zuo G., Zhang H., Long Y., et al. Ground truth annotated femoral X-ray image dataset and object detection based method for fracture types classification. *IEEE Access*. 2020, 8, 189436–189444. <https://doi.org/10.1109/ACCESS.2020.3029039>
12. Thian Y.L., Li Y., Jagmohan P., Sia D., Chan V.E.Y., Tan R.T. Convolutional neural networks for automated fracture detection and localization on wrist radiographs. *Radiology: Artificial Intelligence*. 2019; 1(1): e180001. <https://doi.org/10.1148/ryai.2019180001>
13. Plaksyvyi A., Skublewska-Paszowska M., Powroznik P. 2023. A comparative analysis of image segmentation using classical and deep learning approach. *Advances in Science and Technology. Research Journal*. 17(6); 127–139. <https://doi.org/10.12913/22998624/172771>
14. Zamir M., Ali N., Naseem A., Frasteen A.A., Zafar B., Assam M., Othman M., Attia E. Face detection & recognition from images & videos based on CNN & Raspberry Pi. *Computation*. 2022; 10(9): 148. <https://doi.org/10.3390/computation10090148>
15. Yang F., Choi W., Lin Y. Exploit All the Layers: Fast and Accurate CNN Object Detector with Scale Dependent Pooling and Cascaded Rejection Classifiers. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June, 2016, 2129–2137. <https://doi.org/10.1109/CVPR.2016.234>
16. Albawi S., Mohammed T.A., Al-Zawi S. Understanding of a convolutional neural network. In: 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 August, 2017, 1–6. <https://doi.org/10.1109/ICEngTechnol.2017.8308186>
17. Jie H., Wengang Z., Houqiang L., Weiping L. Sign language recognition using 3D convolutional neural networks. In: 2015 IEEE International Conference on Multimedia and Expo (ICME), Turin, 29 June-3 July, 2015, 1–6, <https://doi.org/10.1109/ICME.2015.7177428>
18. Basha S.H.S., Dubey S.R., Pulabaigari V., Mukherjee S. Impact of fully connected layers on performance of convolutional neural networks for image classification. *Neurocomputing*. 2020; 378: 112–119. <https://doi.org/10.1016/j.neucom.2019.10.008>
19. Shorten C., Khoshgoftaar T.M. A survey on image data augmentation for deep learning. *Journal of Big Data*. 2019; 6(1): 60. <https://doi.org/10.1186/s40537-019-0197-0>
20. Wong T.T., Yeh P.Y. Reliable accuracy estimates from k-fold cross validation. *IEEE Transactions on Knowledge and Data Engineering*. 2020; 32(8): 1586–1594. <https://doi.org/10.1109/TKDE.2019.2912815>
21. Haghighi S., Jasemi M., Hessabi S., Zolanvari A., 729. PyCM: Multiclass confusion matrix library in Python. *Journal of Open Source Software*. 2018; 3(25): 729. <https://doi.org/10.21105/joss.00729>
22. Markoulidakis I., Kopsiaftis G., Rallis I., Georgoulas I., editors. Multi-Class Confusion Matrix Reduction method and its application on Net Promoter Score classification problem. In: Proceedings of the 14th Pervasive Technologies Related to Assistive Environments Conference, Corfu, Greece, June, 2021,

- 412–419, <https://doi.org/10.1145/3453892.3461323>
23. Roth K., Pemula L., Zepeda J., Schölkopf B., Brox T., Gehler P. Towards total recall in industrial anomaly detection. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June, 2022, 14298–14308. <https://doi.org/10.1109/CVPR52688.2022.01392>
24. MacEachern S.J., Forkert N.D. Machine learning for precision medicine. *Genome*. 2021; 64(4): 416–425. <https://doi.org/10.1139/gen-2020-0131>
25. Fu G., Sun P., Zhu W., Yang J., Cao Y., Yang M.Y., et al. A deep-learning-based approach for fast and robust steel surface defects classification. *Optics and Lasers in Engineering*. 2019; 121: 397–405. <https://doi.org/10.1016/j.optlaseng.2019.05.005>
26. Karpiński R., Krakowski P., Jonak J., Machrowska A., Maciejewski M. 2023. Comparison of Selected Classification Methods Based on Machine Learning as a Diagnostic Tool for Knee Joint Cartilage Damage Based on Generated Vibroacoustic Processes. *Applied Computer Science*, 19(4), 136–150. <https://doi.org/10.35784/acs-2023-40>