Convolutional Neural Network on X-ray Images

Diya Vinod

I. Problem Statement

Bone fractures are among the most common injuries seen in emergency and orthopedic departments, accounting for millions of hospital visits each year [1]. Due to population growth and aging, the number of people who endure fractures has been increasing [2]. Osteoporotic fractures impose a substantial economic burden, with the Bone Health and Osteoporosis Foundation estimating 3 million fractures and $25.3 billion in annual healthcare costs by 2025. Costs are expected to rise further, with estimates exceeding $95 billion by 2040, largely due to undertreatment and disease mismanagement [3]. Bone fractures require accurate and timely diagnosis to prevent complications such as improper healing, chronic pain, or functional impairment [2].

However, interpreting X-ray images can be challenging due to subtle fracture appearances, overlapping anatomical structures, and variability in imaging quality, particularly in high-volume emergency settings where clinicians must make rapid decisions. The consequences associated with inaccurate diagnosis of bone fractures for patients, doctors, and healthcare systems motivate the development of automated tools that can assist in fracture detection, improving both the speed and consistency of diagnosis. This project looks into the use of a Convolutional Neural Network (CNN), a type of deep learning model, to determine the presence of a fracture in X-ray images.

II. Solution

To aid in better fracture detection, I built a deep learning-based computer vision solution to identify bone fractures using a dataset consisting of X-ray images. Given that the input into the model is an X-ray image, a machine learning model suited for image classification will work best. CNNs are a type of machine learning algorithm highly effective for image classification tasks due to their ability to automatically learn hierarchical features such as edges, textures, and complex shapes. CNNs are well-suited for fracture detection because they can automatically learn complex visual patterns and generalize across patients and imaging conditions, critical for clinical applications. In this project, a CNN is applied to a binary classification task. The input to the CNN is an X-ray image, and the output indicates whether a fracture is present. This CNN model can support doctors by offering an additional perspective during X-ray analysis, helping to improve diagnostic accuracy and support more effective treatment decisions.

III. Assumptions, Constraints & Implications

For this project, we made several assumptions regarding the dataset. First, we assumed that the labeled data is accurate, meaning that each X-ray image is correctly classified as fractured or non-fractured. Any errors or inconsistencies in the labels would directly affect the model's performance. Second, we assumed the training data were representative of real-world clinical cases. The X-ray images used for training and testing should reflect the types of images encountered in real clinical settings, including variation in patient anatomy, fracture types, and imaging conditions. When implementing this model in clinical settings, it is critical that a representative dataset is used for training.

One of the main constraints in this project was the size and quality of the dataset, including some class imbalance, which could bias the model's predictions. This was addressed through data augmentation and careful validation. Another constraint was that training CNNs can be computationally expensive; running the model on a CPU required a significant amount of time. Hardware limitations restricted the model's size, number of layers, batch size, and training epochs. These considerations are important when implementing this model in real-world settings.

Since this is a medical imaging application, the implications must be carefully considered. Incorrect predictions could have serious consequences and put patients at risk. Therefore, the model should only be used as a decision-support tool and not as a standalone diagnostic system. A human clinician should always have the final authority in making the diagnosis.

IV. Method

The dataset used for this project is the "Bone Fracture Multi-Region X-ray Data" and is sourced from Kaggle [4]. The data contains X-ray images of fractured and non-fractured bones covering all anatomical body regions. It consists of 9246 training images, 828 validation images, and 506 test images. First, we preprocessed the data and applied data augmentation techniques. For the training, validation, and test images, I normalized the pixel values to the range of 0 to 1. Normalization is necessary to help the optimizer converge faster, and neural networks train better. I set every image to 224 pixels high by 224 pixels wide to ensure all input images have the same shape, and the batch size to 32 to allow the model to process a manageable number of images at a time, improving training efficiency and stability. Additionally, I applied random transformations such as rotations, shifts, zooms, and flips to images during training only, to help the model generalize better and avoid overfitting. *Figure 1* shows a snippet of the training data and its correct labels.

Figure 1

Next, I built the CNN architecture. The CNN consists of four convolutional blocks with increasing filter sizes (32-256), each using ReLU activation and followed by batch normalization and max pooling. The extracted features are flattened and passed through a fully connected layer with ReLU activation and dropout regularization before a sigmoid-activated output layer performs binary classification.

Then we train the model with a learning rate of 0.006. This value was determined using random search, which is an efficient hyperparameter optimization technique for CNNs, allowing us to determine the best value for the learning rate. The model was trained for up to 15 epochs using training data, with validation performance monitored at each epoch. Early stopping and model checkpointing were employed to prevent overfitting and retain the best-performing model. If the validation loss did not improve for three epochs, training stops, and the model restores the weights from the best epoch. During one specific run of the model, the final training accuracy was **0.963**, the training loss was **0.099**, the validation accuracy was **0.932**, and the validation loss was **0.165**. *Figures 2* and *3* show training and validation accuracy and training and validation loss over the 15 epochs.

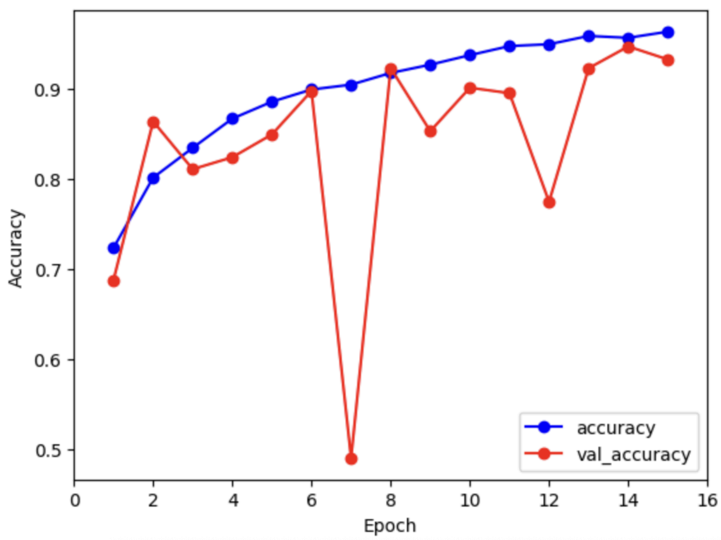

Figure 2

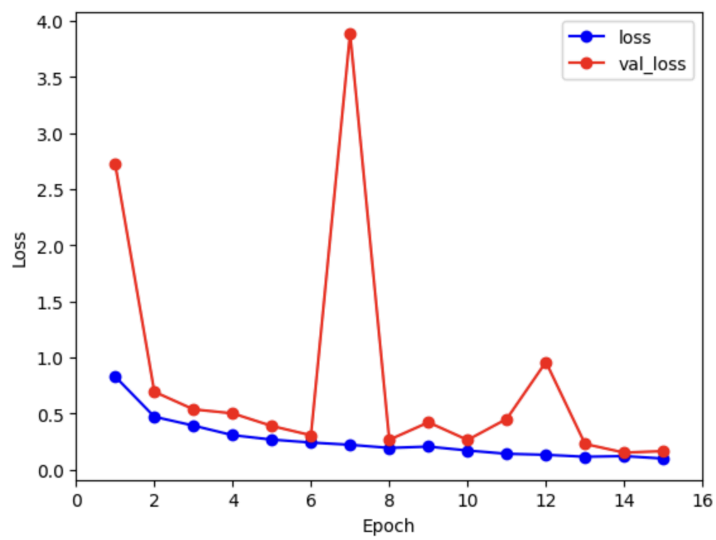

Figure 3

The training and validation accuracy for this model are both strong. When evaluated on the test dataset, the model achieves a test loss of **0.203** and a test accuracy of **0.928.** *Figure 4* presents the confusion matrix, and *Figure 5* shows the precision, recall, and F1-score values. From the confusion matrix, it is evident that the model correctly classifies the majority of both fracture and non-fracture cases. The model produces a few false positives but a slightly larger number of false negatives, which is particularly important in a medical context since missed fractures can have serious clinical consequences.

For class 0 (non-fracture), the precision is **0.89**, indicating that when the model predicts a non-fracture, it is correct 89% of the time. The recall for this class is **0.97**, meaning the model correctly identifies 97% of all non-fracture cases. For class 1 (fracture), the precision is **0.97**, showing that fracture predictions are highly reliable. The recall for this class is **0.89**, indicating that the model correctly identifies 89% of fracture cases.

Overall, the model demonstrates strong performance, with high test accuracy and balanced precision and recall across both classes. However, the presence of false negatives highlights the importance of using this model as a clinical decision-support tool rather than as a standalone diagnostic system.

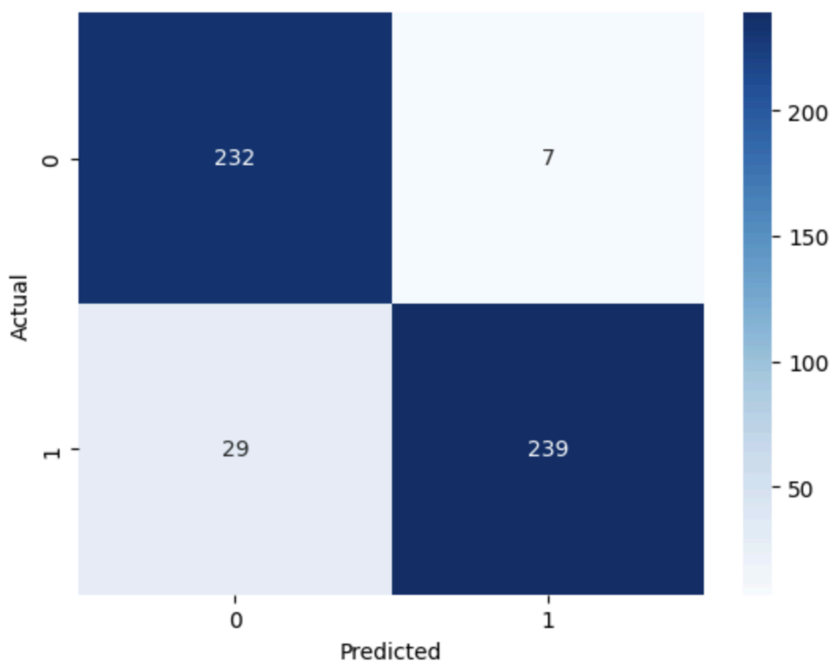

Figure 4

	precision	recall	f1-score	support
0	0.89	0.97	0.93	239
1	0.97	0.89	0.93	268
accuracy			0.93	507
macro avg	0.93	0.93	0.93	507
weighted avg	0.93	0.93	0.93	507

Figure 5

V. Summary

Accurately detecting bone fractures from X-ray images is challenging due to subtle fracture patterns, overlapping anatomical structures, and variability in image quality. In high-pressure clinical settings, these challenges increase the risk of misdiagnosis, which can lead to delayed or inappropriate treatment and negative patient outcomes. To address this problem, a CNN was developed to classify X-ray images as either fractured or non-fractured. The model learns visual features directly from the images and serves as a decision-support tool to assist clinicians by providing an additional perspective during X-ray interpretation. The CNN demonstrated strong performance on test data, achieving high accuracy and balanced precision and recall across both classes. Despite its strong performance, the model has several limitations. It relies on the quality and representativeness of the training data and may be biased by class imbalance or limited dataset size. Additionally, the presence of false negatives (missed fracture cases) is a concern in a medical context. The model is also computationally expensive to train and lacks inherent interpretability, reinforcing the need for human oversight. As a result, the model should be used as a decision-support tool rather than a standalone diagnostic system.

VI. References

[1] Office of the Surgeon General. (2019). *The Burden of Bone Disease*. Nih.gov; Office of the Surgeon General (US). https://www.ncbi.nlm.nih.gov/books/NBK45502/

[2] World. (2024, September 25). *Fragility fractures*. Who.int; World Health Organization: WHO. https://www.who.int/news-room/fact-sheets/detail/fragility-fractures?utm_source=chatgpt.com

[3] Singer, A., McClung, M. R., Tran, O., Morrow, C. D., Goldstein, S., Kagan, R., McDermott, M., & Alon Yehoshua. (2023). Treatment rates and healthcare costs of patients with fragility fracture by site of care: a real-world data analysis. *Archives of Osteoporosis*, 18(1). https://doi.org/10.1007/s11657-023-01229-7

[4] Rodrigo, M. (2024). *Bone Fracture Multi-Region X-ray Data*. Kaggle.com. https://www.kaggle.com/datasets/bmadushanirodrigo/fracture-multi-region-x-ray-data/data