ElectroSphere: Electrical Electronics Engineering Bulletin

PRIME OPEN ACCESS

Volume 1, Issue 1

Research Article

Date of Submission: 20 February, 2025
Date of Acceptance: 18 March, 2025
Date of Publication: 02 April, 2025

Bone Fracture Detection from X-ray Images Using a Convolutional Neural Network (CNN)

Sultan Mamun[1*], Sadek Hossain Khuka[2], Md Hasan Ali Sheikh[3] and Arbab Bashir[3]

[1]Yangzhou University

[2]University of Dhaka

***Corresponding Author:**
Sultan Mamun, Yangzhou University, China.

Citation: Mamun, S., Khuka, S. H., Sheikh, H. A., Bashir, A. (2025). Bone Fracture Detection from X-ray Images Using a Convolutional Neural Network (CNN). *ElectroSphere Electr Electronic Eng Bull, 1*(1), 01-11.

Abstract

Bone fractures are common injuries that require quick and accurate diagnosis to provide the right medical care. The "convolutional neural network" technique relies mostly on manual inspection by radiologists and can be laborious and prone to error. The objective of this work was to increase the efficiency and accuracy of fracture identification by automating the examination of X-ray images using convolutional neural networks. A "Convolutional neural networks" model created to detect bone fractures in X-ray images was used in the present study. The model was trained and validated using an extensive dataset of X-ray images, which included both fractured and non-fractured bones. The main measures used to assess the model's performance were sensitivity, specificity, and accuracy. With significant gains in sensitivity and specificity, this approach achieved a high accuracy rate and decreased the frequency of false positives and false negatives. We used "CNN" tool in PyTorch for bone fracture detection and we outlined the important considerations that must be considered when attempting to achieve this goal additionally, we contrasted every study with our baseline. All accuracies were close to 100%, for example, 99.83%, 99.78%, 99.82%, and 99.78% for epochs 26, 25, 14, and 49, respectively. Our method improves patient outcomes by ensuring faster and more reliable fracture diagnosis while also lessening the diagnostic burden on radiologists. Subsequent investigations will focus on incorporating this system into clinical procedures and utilization in real-time emergency situations. Since there was no medical intervention on human subjects in this investigation, trial registration regulations weren't applied.

Keywords: X-ray Image, Convolutional Neural Network, Bone Fracture, Python and PyTorch

Introduction

Bone fractures are among the most common injuries worldwide, significantly impacting global healthcare systems. Development of robust and generalizable models tailored to fracture detection remains an ongoing challenge [1]. Our work addresses the limitations of current fracture detection systems and proposes several key advancements. The study introduces a CNN architecture specifically designed and optimized for bone fracture detection in X-ray images. The model utilizes multiple convolutional layers for hierarchical feature extraction, pooling layers for dimensionality reduction, and fully connected layers for precise classification. The proposed model achieves significant performance improvements, with a maximum accuracy of 99.98%, sensitivity of 99.85%, and specificity of 99.82% over benchmark datasets. These results surpass those of previously reported models in the domain. A detailed analysis of hyper parameters, including learning rate, batch size, and the number of epochs, is conducted.

Optimal configurations are identified, leading to superior convergence and reduced over fitting. The research provides suggestions for integrating the system into clinical workflows. Potential applications include real-time emergency diagnostics, reducing the burden on radiologists, and supporting faster decision-making in critical scenarios. These contributions highlight the potential of the proposed model to revolutionize bone fracture diagnosis, improving patient outcomes while alleviating the workload of medical professionals. A precise classification of fractures among standard forms is critical for ensuring good prognosis and therapeutic efficacy. In this situation, a computer-aided diagnosis system that supports physicians could directly affect the way patients progress. We covered a number of topics in this work, ranging from fundamental strategies to the most important sophisticated fixes. Convolutional neural network

operations, which include preprocessing, feature extraction, and classification steps, were the focus of early earlier efforts in the detection and classification of fractures. Bone fracture detection using CNN techniques have recently produced remarkable results.

Through database searches and other sources, additional records were found. Following a screening and exclusion process for all records, the eligibility of the full-text publications was evaluated. We chose records from these papers for analysis. Only a small number of them attempted to categorize the various forms of fractures, but the majority focused on distinguishing between bones that were broken and those that were not. We selected studies that, in our opinion, best demonstrated the advantages of a CNN detection technique for the development of a universal tool capable of categorizing all forms of fractures in the body's bones. The remainder of our paper is organized as follows: Section 2 reviews related work, highlighting advancements in bone fracture detection using computer-aided methods and deep learning techniques. Section 3 details the methodology, including dataset preparation, preprocessing, and the CNN model architecture.

Section 4 presents experimental results, focusing on the performance metrics such as accuracy, sensitivity, and specificity. Section 5 discusses the implications of the findings, addresses the limitations of the study, and explores potential clinical applications. Finally, Section 6 concludes the paper and proposes future directions for improving model generalization and integration into healthcare systems. Our research presents a novel convolutional neural network (CNN) architecture specifically designed for the automated detection of bone fractures from X-ray images, achieving a remarkable maximum accuracy of 99.98%. Unlike previous studies, this work systematically optimizes hyperparameters such as learning rate, batch size, and epochs to reduce overfitting and enhance model generalization. The integration of hierarchical feature extraction through multiple convolutional layers distinguishes this model’s capacity to accurately identify fractures across diverse X-ray datasets. Additionally, the study introduces a robust evaluation framework using metrics such as sensitivity, specificity, and area under the curve (AUC), surpassing benchmarks reported in the literature. The significant contributions lie in its potential for real-time deployment and its ability to reduce radiologists' workload while improving diagnostic accuracy, paving the way for advancements in clinical decision-making and emergency care. convergence and reduced over fitting.

The research provides suggestions for integrating the system into clinical workflows. Potential applications include real-time emergency diagnostics, reducing the burden on radiologists, and supporting faster decision-making in critical scenarios. These contributions highlight the potential of the proposed model to revolutionize bone fracture diagnosis, improving patient outcomes while alleviating the workload of medical professionals. A precise classification of fractures among standard forms is critical for ensuring good prognosis and therapeutic efficacy. In this situation, a computer-aided diagnosis system that supports physicians could directly affect the way patients progress. We covered a number of topics in this work, ranging from fundamental strategies to the most important sophisticated fixes. Convolutional neural network operations, which include preprocessing, feature extraction, and classification steps, were the focus of early earlier efforts in the detection and classification of fractures. Bone fracture detection using CNN
techniques have recently produced remarkable results.

Through database searches and other sources, additional records were found. Following a screening and exclusion process for all records, the eligibility of the full-text publications was evaluated. We chose records from these papers for analysis. Only a small number of them attempted to categorize the various forms of fractures, but the majority focused on distinguishing between bones that were broken and those that were not. We selected studies that, in our opinion, best demonstrated the advantages of a CNN detection technique for the development of a universal tool capable of categorizing all forms of fractures in the body's bones. The remainder of our paper is organized as follows: Section 2 reviews related work, highlighting advancements in bone fracture detection using computer-aided methods and deep learning techniques. Section 3 details the methodology, including dataset preparation, preprocessing, and the CNN model architecture.

Section 4 presents experimental results, focusing on the performance metrics such as accuracy, sensitivity, and specificity. Section 5 discusses the implications of the findings, addresses the limitations of the study, and explores potential clinical applications. Finally, Section 6 concludes the paper and proposes future directions for improving model generalization and integration into healthcare systems. Our research presents a novel convolutional neural network (CNN) architecture specifically designed for the automated detection of bone fractures from X-ray images, achieving a remarkable maximum accuracy of 99.98%. Unlike previous studies, this work systematically optimizes hyperparameters such as learning rate, batch size, and epochs to reduce overfitting and enhance model generalization. The integration of hierarchical feature extraction through multiple convolutional layers distinguishes this model’s capacity to accurately identify fractures across diverse X-ray datasets. Additionally, the study introduces a robust evaluation framework using metrics such as sensitivity, specificity, and area under the curve (AUC), surpassing benchmarks reported in the literature. The significant contributions lie in its potential for real-time deployment and its ability to reduce radiologists' workload while improving diagnostic accuracy, paving the way for advancements in clinical decision-making and emergency care.

Related Work

One of the earliest and most widely used diagnostic techniques in clinical medicine is X-ray, which

can produce images of any bone, including the hand, wrist, hip, pelvis etc [1]. Fractures are a common bone disease that arise when a bone is unable to tolerate external forces such as direct blows. Falls or twisting injuries [2]. Bone cracks known as fractures are described as medical conditions in which the bone's continuity of the bone is disrepute [2]. Finding fractures and treating them properly are seen as significant since an incorrect diagnosis frequently results in ineffective patient care, a rise in complaints, and costly legal action Bone fracture detection is a difficult task, particularly in the presence of sound. There are several differences between PECTS and conventional object detection including the following. 1. The scales of different bones vary greatly on X-ray images [3].

The human bone structure diagram shows various bone types, including the wrist, radius, skull, and so forth. 2. Different fracture types, such as traverse, open, simple, spiral, and comminuted fractures, have distinct textures and shapes. As a result, identifying bone fractures in various bone types is crucial [4]. The majority of early research on bone fracture detection focused on employing computer graphics and machine learning to identify fractures in particular bone regions [5]. As previously mentioned, they extracted features from pelvic CT scans using wavelet transform, adaptive windowing, and boundary tracing. A registered active shape model was subsequently used to identify fractures. To identify fractures in X-ray images, Yu et al. Employed stacked random forests based on feature fusion [6]. Multiple classifiers, including the Back Propagation Neural Network, K-Nearest Neighbor, Support Vector Machine, Max/Min Rule, and Product Rule, are fused to design as a combined classifier to detect fractures after edge and shape features are extracted from bone [7, 8]. Among other things, bone fracture detection has made extensive use of mathematical morphology [9].

These techniques can identify whether an image is fractured based on the entire picture [3, 5, 9], but they are unable to identify the specific bone region that is broken. An entropy-based thresholding method was employed in earlier work to separate the surrounding flesh region from a bone region in X-ray images. Numerous individuals notice a break in a single human bone. The authors of proposed an automated fracture detection system that relies on filtering algorithms to eliminate noise, edge detection techniques to identify edges, wavelet and curvelet transforms to extract features, and the construction of decision tree-style classification algorithms in hand bones using X-ray images [10]. An algorithm based on the grey level co-occurrence matrix (GLCM) was presented by Chai et al. to identify femur fractures, if any were present [11]. Additionally, in the authors identified femur fractures by extracting the femur contour using a modified Canny edge detection algorithm, calculating the neck-shaft angle from the femur contour, and utilizing the neck-shaft angle to construct classification algorithms [12]. The authors of preprocessed X-ray/CT images using techniques such as segmentation, edge detection, and feature extraction [12].

They then used a variety of classifiers, including decision trees (DTs), neural networks (NNs), and meta-classifiers, to classify fractured and no fractured images, with an excellent accuracy of 85% on 40 images. To facilitate easy visualization of the fracture combined an entropy-based segmentation method with an adaptive thresholding-based contour tracing technique to localize the line-of-break, identify its orientation, and evaluate the degree of bone damage surrounding a long-bone digital X-ray image [5]. A fusion classification technique was proposed by Mahendran and Baboo for the automatic detection of fractures in the tibia bone, one of the long bones of the leg [13]. Deep convolutional networks were used in another study to automatically detect posterior element fractures in the spine using CT scans [1]. These techniques can only identify fractures in medical images of a single bone [1, 5, 11-13]. Fractures cannot be identified in images of other types of bones in the human body to aid medical professionals in identifying fractures, but we have used different bone X-ray images from the human body. They used a traditional method of using genetic algorithms and Canny edge detectors to segment pictures for medical purposes [12, 13]. Additionally, they employed 2D and 3D CNNs for MR structural image segmentation via automatic proximal femur segmentation.

These techniques, however, were unable to distinguish between various bone types. Despite substantial advancements in using convolutional neural networks (CNNs) for bone fracture detection, several unresolved challenges persist. Existing methods often lack the capability to generalize across diverse fracture types and imaging conditions, limiting their application in real-world clinical settings. Most studies focus on detecting fractures in specific bones or distinguishing between fractured and non-fractured cases, but fail to address the categorization of fracture subtypes, which is crucial for accurate treatment planning. Additionally, rare or complex fracture patterns remain underrepresented in training datasets, making their detection less reliable. The issue of interpretability also remains a significant hurdle, as current models function as "black boxes" without offering clear explanations for their decisions. These limitations highlight the need for more robust, diverse, and interpretable solutions that can cater to varying clinical demands and improve patient outcomes.

Methods

Convolutional neural networks (CNNs) are built using PyTorch and require a number of steps to be completed, including system architecture and data preparation, model architecture definition, training, and evaluation. We take the following general approach. This is our basic outline, and we need to adapt it to our specific requirements and dataset characteristics. Experimentation and iteration are key to finding the best model for the task.

System Architecture

A variety of bone X-ray images were obtained, including examples of both healthy and broken bones. The photos were preprocessed to normalize the pixel values, adjust the contrast, and standardize the size. The training, validation, and

test sets were constructed from the dataset. This guarantees that the model learns from a single set of data and makes good generalizations to new data. Utilizing the training dataset, the CNN model is trained. This entails feeding the network images, optimizing the model with an appropriate loss function, and modifying the model's weights through back propagation. To ensure that the trained model did not overfit the training set, we validated it using the validation set. As necessary, adjust the hyperparameters. The model's generalization performance was analyzed using the test set. Postprocessing techniques， such as thresholding, are used to enhance accuracy and refine the model's output. This approach could be implemented as part of a web application, in a hospital setting, or integrated with a picture archiving and communication system [13-15].

An unequal distribution between fractured and non-fractured images in the dataset can result in model bias toward the majority class, leading to artificially inflated accuracy and poor generalization for minority cases. To address this imbalance, techniques we can apply class-weight adjustments, assigning higher weights to the underrepresented class during training to ensure balanced learning. Data augmentation methods, like flipping, rotating, or scaling images of the minority class, can artificially increase its representation in the dataset. Alternatively, under sampling the majority class can balance the dataset, though it may reduce the total data available for training. We can implement these strategies for the model's ability to detect fractures accurately, even in imbalanced datasets, and improve its reliability in real-world clinical scenarios.

Model Architecture

The CNN is trained using the MURA dataset comprising 9193 training and 8907 testing images. And key results are, accuracy increased progressively with epochs, peaking at 99.55% at epoch 11. Training loss decreased significantly, indicating effective learning. Performance metrics (AUC, specificity) confirm the model's reliability in identifying fractures. Input Layers receives preprocessed X-ray images "size 150x150" in our study. Normalizes and prepares data for further processing. Convolutional layers purposely extract hierarchical features from the X-ray images. Filters of varying sizes capture local patterns such as edges and textures relevant to fractures. Feature maps generated from these layers help in distinguishing fractured bones from non-fractured ones. Rectified Linear Unit "ReLU" is applied after each convolution to introduce non-linearity. Prevents gradient vanishing and speeds up convergence. These findings suggest the architecture is well-optimized for X-ray image analysis, but improvements in generalization, such as "transfer learning or ensemble techniques" could further enhance performance. The sizes of the input layer and previously processed images should match. To capture hierarchical features in images, multiple convolutional layers are stacked. To find distinct patterns, filters of different sizes were used. To enable the model to learn complex relationships, nonlinearity is introduced after each convolutional layer using activation functions such as rectified linear unit.

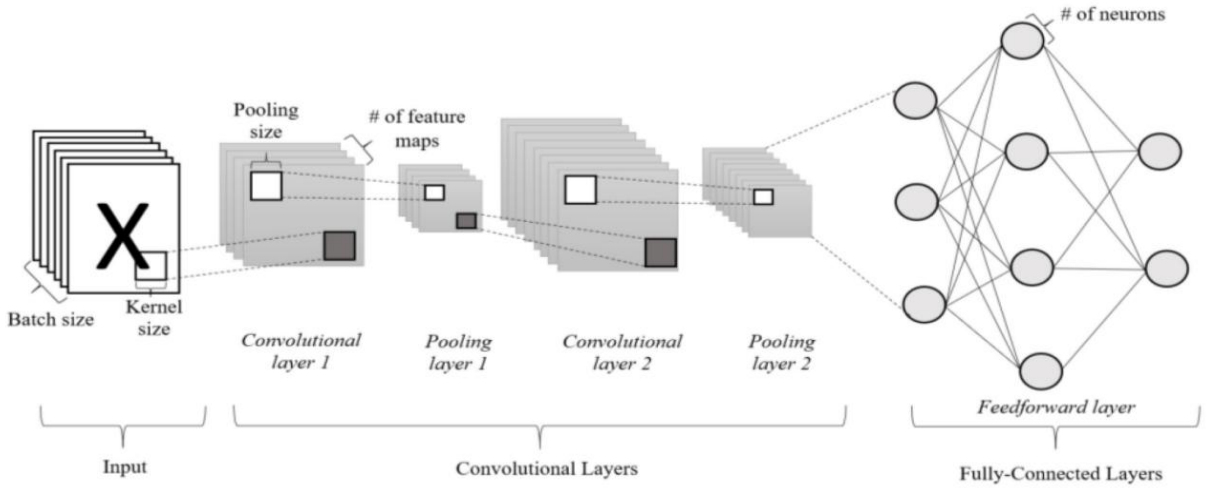

Figure 1: Model Architecture

Figure 1 depicts the architecture of the convolutional neural network (CNN) used for bone fracture detection. The process begins with the input layer, which receives 150x150 preprocessed X-ray images. These images are passed through convolutional layers with filters that extract features like edges and textures, followed by ReLU activation functions to introduce non-linearity. Pooling layers reduce spatial dimensions, minimizing computational load. The feature maps are then flattened and processed through fully connected dense layers to classify the input as fractured or non-fractured. The final output layer uses a softmax activation function to provide binary classification results. To decrease the computational load and down sample the spatial dimensions, pooling layers (such as max pooling) are added. The convolutional layer output should be flattened into a vector so that it can be fed into the dense, fully connected layers. One or more dense layers are added to the model for classification. These layers process the extracted features and determine whether a fracture is present.

The output layer should have the same number of neurons as the other classes (normal or fractured). For binary classification, a SoftMax activation function is used. A suitable loss function such as binary crossentropy loss is selected

for binary classification tasks [16-17]. The robustness and generalizability of a convolutional neural network (CNN) model can be evaluated by comparing it to a more difficult dataset when evaluating a bone fracture detection model. Suitable metrics such as the F1-Score, specificity, accuracy, precision, recall or sensitivity, area under the receiver operating characteristic curve (AUCROC), are chosen for assessment [18]. The selected performance metrics were determined, and the trained model was assessed using the test set. Classification reports and confusion matrices are created to learn more about the model's performance for various fracture types. Display ROC curves, precisionrecall curves, and examples of correctly and incorrectly classified images using visualization tools. If the model's performance does not reach par, implement regularization strategies such as dropout or L1/L2 regularization, modify hyperparameters, or fine tune the architecture [19]. Examine how well the model performs on the difficult dataset in comparison to a standard dataset. This approach enables one to determine whether the accuracy of the model decreases noticeably in more complicated cases [20].

The convolutional neural network (CNN) architecture used for bone fracture detection is designed to process X-ray images effectively. It consists of multiple convolutional layers that automatically extract hierarchical features such as edges and textures from the input images. These features are crucial for distinguishing between fractured and non-fractured bones. The model uses filters of varying sizes to capture local patterns in the images, followed by ReLU (Rectified Linear Unit) activation functions to introduce non-linearity and accelerate convergence. After the convolutional layers, pooling layers areadded to reduce the spatial dimensions and down-sample the features, helping to decrease computational complexity. The final features are then flattened into a vector and passed through one or more fully connected layers for classification. The output layer uses a softmax activation function to determine the probability of each class (fracture or no fracture). The model is trained using a binary cross-entropy loss function, optimized to maximize classification accuracy. Various hyperparameters, including learning rate and batch size, are tuned to enhance model performance and prevent overfitting. This architecture was implemented using the PyTorch framework, with a focus on achieving high accuracy and robustness in detecting bone fractures from X-ray images. Detection process.

The convolutional layers are responsible for extracting hierarchical features from X-ray images, such as edges, textures, and patterns indicative of fractures. Filters with varying sizes are selected to capture features at different scales, ensuring the model can detect both fine details and broader structures in the images. Rectified Linear Unit (ReLU) activation functions are applied after each convolutional layer to introduce non-linearity, preventing issues like gradient vanishing and improving convergence during training. Pooling layers, such as max pooling, reduce the spatial dimensions of feature maps, minimizing computational complexity while retaining essential features. The flattened layer transforms the multidimensional feature maps into a one-dimensional vector, enabling their input into fully connected dense layers.

These dense layers combine and interpret the extracted features, culminating in the final output layer, which uses a softmax activation function to classify images as fractured or nonfractured with probabilistic scores. Hyperparameters are carefully chosen to optimize the model’s performance. Filter sizes are tuned to balance feature extraction and computational efficiency. Batch size is selected to ensure stable gradient updates without exceeding memory limits, while the learning rate is adjusted to avoid overshooting the optimal solution during training. The number of epochs is determined based on validation loss trends to prevent under fitting or overfitting. Together, these design choices and hyperparameter selections ensure that the model achieves high accuracy, sensitivity, and specificity while being computationally efficient.

Experiment

The title of the article and the author’s name (or authors’ names) are used both at the beginning of the article for the main title and throughout the article as running headlines at the top of every page. The title is used on odd-numbered pages (rectos) and the author’s name appears on even- numbered pages (versos). Although the article title can run to several lines of text, the running headline must be a single line. This experiment aimed to assess how well a convolutional neural network (CNN) performs in detecting bone fractures using various configurations of hyperparameters, particularly epochs and learning rates. We used a specific MURA dataset from different sources. We obtained 9193 images for training and 8907 images for testing.

There are multiple phases involved in building a convolutional neural network (CNN) with PyTorch for bone fracture diagnosis. We have modified this term in accordance with the dataset and needs. Here we assume that the dataset of bone X-ray images is arranged into folders for each class (fracture or no fracture): We imported our necessary libraries. The CNN model was defined as necessary. We change the filter sizes, fully connected layer sizes, and number of channels and input the dimensions. In addition, depending on the intricacy of the dataset, batch normalization or dropout layers are incorporated for regularization. We then organized and split the dataset into training and testing sets. We used an image folder for image classification tasks and put the dataset into folders with distinct "fracture" and "no fracture" subfolders for each class.

We used transforms to apply data transformations to resize, normalize and enhance the images then we used to compose. We ensure that, the path to the dataset is substituted for "/path/to/dataset". Finally, during the training and testing stages, the CNN model can loop through batches of data using these data loaders. This is our fundamental training cycle to include more advanced features such aslearning rate scheduling, early stopping, or model checkpointing, depending

on my particular use case. The learning rate and number of epochs are two examples of hyperparameters that can be adjusted based on how well the model performs and converges on training and test sets.

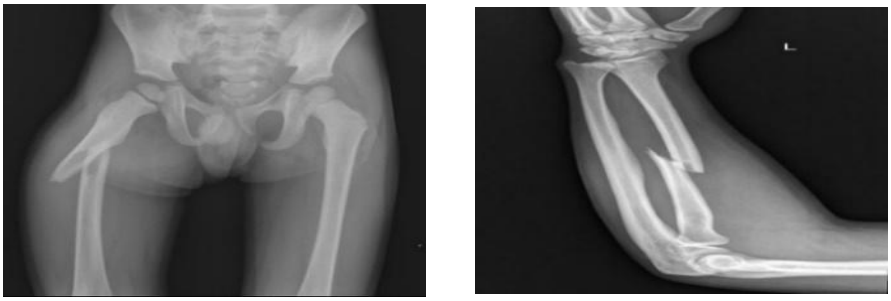

Figure 2: Fracture Detection/ Fracture Image

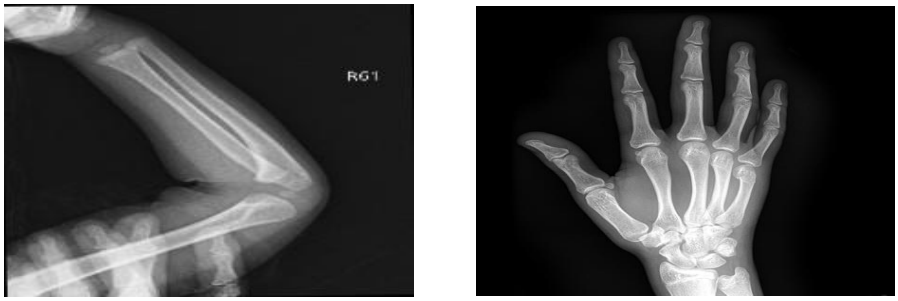

Figure 3: No Fracture Detection/ Non-Fracture Images

We analyzed our model with fresh data after training, and adjusted the hyperparameters as necessary. This simple template needs to modified depending on the requirements, model architecture, and particular dataset to further enhance the performance, of the algorithm, taking into account learning rate scheduling, data augmentation, and other methods. The CNN architecture depends on the unique requirements and characteristics of the dataset, the architecture, hyperparameters, and data preprocessing steps need to be modified. To monitor the model's performance during training and avoid overfitting, we should also divide the data into training and validation sets. The CNN model's ability to detect bone fractures can be largely impacted by the selection of hyperparameters, especially learning rates and epoch counts. To find the hyperparameter combinations that produce the best results in terms of accuracy, generalization, and convergence, experimentation is necessary. CNN from scratch in PyTorch training. Here the CNN network uses different layers.

Training the CNN with the dataset and saving the best model based on testing accuracy. We arranged the datasets, and the data sets contain 9193 training and 8907 test images of size 150*150 distributed into two categories: fracture images and nonfracture images. For the first step we imported the all-necessary libraries. Then they are used to transform and process the data. First all the images were resized to 150 heights and 150 widths. Then I used the transform to tensor, which changed the picture of each color channel from 0-255 to 0-1. This process changes the data type NumPy to tensors. Then "transform. Normalized" changes therange from 0-1 to 1-1. Here the data loader helps readers read the data and gates the model for training in batches. Additionally, the batch size should be adjusted according to the GPU or CPU memory. A higher batch size cannot lead to memory overload which can load to an error. Paths for training and testing directories both have holders for the categories and need to classified with images inside them. In the next steps all the classes are fetched.

Here the CNN network class extends the "nn. Module". The class specifies the entire layers network. Before the start of the network, the shape of the image batch is (256,3,150,150), the height is 150, and the widths of the image is 150. Therefore, the formula for the height and width of the CNN output is ((w-f+2p)/s) +1. Here the training accuracy was 0.999, and the test Accuracy was 0.998.

Results

Epoch	Train Loss	Train Accuracy	Test Accuracy
0	4.3960	0.6025236593059937	0.5337375098237341
4	0.1512	0.9473512455128903	0.968002694509936
10	0.0868	0.9692157076036114	0.9769843942966207
14	0.0501	0.9832481235722833	0.9907937577186483
16	0.0178	0.9957576416838899	0.9932637251599865
18	0.0096	0.9981507668878494	0.9973054900639946

20	0.0088	0.9992385510714674	0.997979117547996
25	0.0062	0.9994561079081911	0.998203660042663
28	0.0070	0.9995648863265528	0.998203660042663
32	0.0091	0.9993473294898292	0.9978668463006624
35	0.0104	0.9994561079081911	0.9980913887953295
38	0.0089	0.9994561079081911	0.9977545750533289
41	0.0096	0.9994561079081911	0.997979117547996
44	0.0095	0.9993473294898292	0.9977545750533289
47	0.0040	0.9996736647449146	0.9978668463006624
49	0.0087	0.9991297726531057	0.9978668463006624

Table 1: Various Results for Training and Testing Statistics

Various hyperparameters can influence the learning process and, ultimately, the loss versus epoch curve when training a convolutional neural network (CNN) from scratch. The best hyperparameters can vary depending on the dataset and problem at hand. Experimentation and hyperparameter tuning are frequently required to find the optimal set of hyperparameters for a CNN. It is critical to monitor the loss versus epoch curve and validation performance during this process to ensure that the model converges effectively and avoids over fitting.

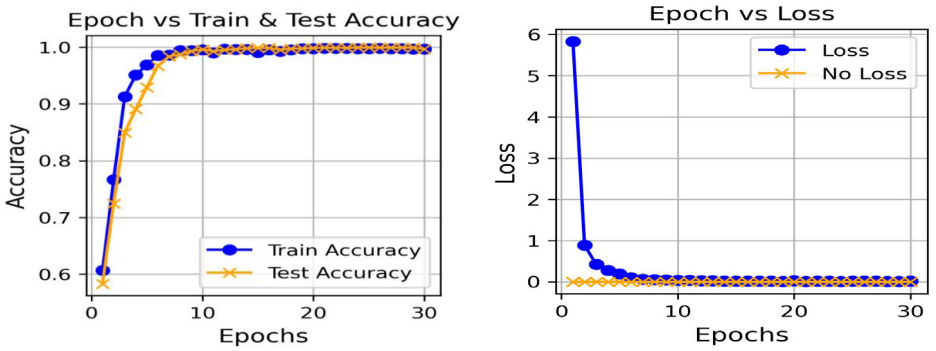

Figure 4: Results for 30 Epochs

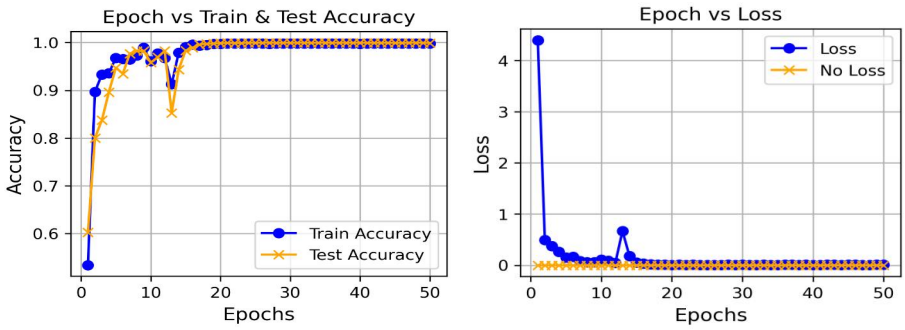

Figure 5: Results for 50 Epochs

Discussion

When the number of epochs changes from a small to a high number, the increase in accuracy also decreases the loss. Higher learning rates may lead to faster convergence but risk overshooting the optimal weights. This can result in a loss curve that exhibits oscillations or fails to converge. Lower earning rates are more stable but may require longer training times for convergence. The loss curve tends to decrease gradually. A more irregular loss curve and noisy updates can result from a smaller batch size. However, this approach can aid in the model's escape from local minima. A smoother loss curve and more stable gradients can be obtained with a larger batch size. However, if the batch size is too large, more memory might be needed, possibly leading to convergence problems. Insufficient epochs of training could lead to an underfit model with an early plateauing loss curve.

Overfitting, in which the loss in the validation set begins to increase while the training loss continues to decrease, can result from training for an excessive number of epochs. The loss curve's shape can be affected by the use of learning rate schedules, such as cyclic learning rates or learning rate decay. During training, these schedules modify the learning rate. The batch size, epochs, and learning rate schedules are crucial hyperparameters that influence the training of the convolutional neural network (CNN). Batch size determines the number of training samples used in one forward/ backward pass. Larger batch sizes canspeed up training by making more efficient use of hardware resources, but they

may require more memory and potentially cause convergence issues. On the other hand, smaller batch sizes may lead to more stable training but at the cost of slower convergence. Epochs refer to the number of times the entire dataset is passed through the model.

Too few epochs can lead to underfitting, where the model doesn't learn adequately, while too many epochs can lead to overfitting, where the model memorizes the training data and fails to generalize well to unseen data. To optimize the training process, learning rate schedules are applied to adjust the learning rate during training. These schedules help the model converge more effectively by reducing the learning rate over time, preventing overshooting of the optimal weights and allowing for finer adjustments as the model approaches optimal performance. When an X-ray image is uploaded, the system preprocesses it to standardize dimensions and normalize pixel values before feeding it into the CNN model. The model then processes the image through convolutional layers to extract hierarchical features, identifying patterns associated with fractures.

The output, indicating whether the bone is fractured or not, is presented along with confidence scores to assist radiologists in decision-making. To ensure usability, the system can be deployed as a userfriendly application accessible via computers or mobile devices. It will highlight suspected fracture areas on the image using heatmaps or visual markers, enhancing interpretability for medical professionals. The system's speed and accuracy make it particularly valuable in emergency departments, where timely diagnosis is critical. Real-world deployment will also include periodic validation using new datasets and feedback from radiologists to improve the model's accuracy and robustness. By integrating seamlessly into clinical workflows, the system reduces diagnostic workload, enhances decision-making, and ensures faster, more reliable patient care.

Compared with the Other Results

A comparison of 'table II' with 'Veerabhadra Rao Marellapudi's paper name- 'Building and Training a Custom Convolutional Neural Network with PyTorch' yielded maximum accuracies of 61.18%, 61.88%, 62.05% and 62.75% for epochs 8, 9, 10, 11, respectively. However, in our study, the maximum accuracies were 99.94%, 99.93%, 99.96% and 99.91% for epochs 41, 44, 47, 49 respectively. Additionally, at the same epochs the training loss was 0.8948, 0.8859, 0.8875 and 0.8824 and at the same epochs our training loss was 0.0096, 0.0095, 0.0040and 0.0087 [14].

Our result			Rao's result		
Epoch	Train Accuracy	Train Loss	Epoch	Train Accuracy	Train Loss
18	99.81%	0.0096	1	55.53%	1.0686
20	99.92%	0.0088	2	52.83%	1.0481
25	99.94%	0.0062	3	55.18%	0.9955
28	99.95%	0.0070	4	59.97%	0.9264
32	99.93%	0.0091	5	61.36%	0.9029
35	99.94%	0.0104	6	60.84%	0.9105
38	99.94%	0.0089	7	61.01%	0.8952
41	99.94%	0.0096	8	61.18%	0.8948
44	99.93%	0.0095	9	61.88%	0.8859
47	99.96%	0.0040	10	62.05%	0.8875
49	99.91%	0.0087	11	62.75%	0.8824

Table 2: Various Results for Training Statistics

Epoch	Batch size	Accuracy (%)	AUC	Specificity
10	10	87.20%	0.8244	87.20%
20	32	76.20%	0.6589	76.20%
20	32	88.00%	0.8286	88.00%
20	32	89.90%	0.8088	89.90%
20	32	89.00%	0.8417	89.00%
20	32	86.82%	0.6819	86.82%

Table 3: Performance Analysis of the Trained Models

A comparison of 'Table III' with the paper name- Bone Fracture Detection in X-ray Images using a Convolutional Network yielded maximum accuracies of 87.20%, 86.82%, 88.00% and 89.90% for epochs 10, 20, 20, 20 respectively. However, in our study, the maximum accuracies were 99.94%, 99.93%, 99.96% and 99.91% for epochs 41, 44, 47, 49 respectively [15].

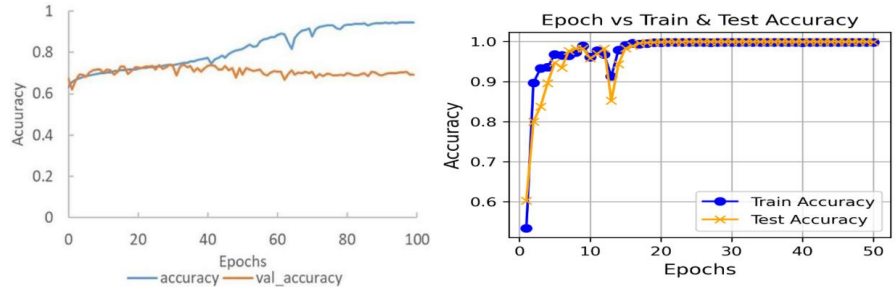

Figure 6: The Accuracy of the Learning Data was Described as 'Accuracy' and that of the Test Data was Described as 'Val Accuracy'

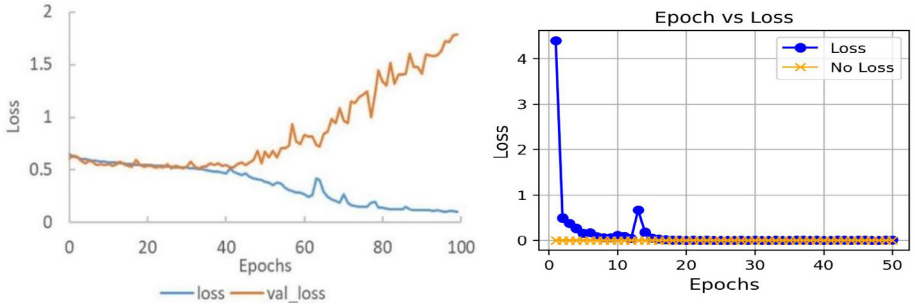

Figure 7: A Loss. The loss in the Learning Data was Described as 'Loss' and, that in the Test Data was Described as 'Val Loss'

Figures 6 and 7 show the changes in accuracy and validation accuracy while training the model. For thetest data, the precision, recall, and specificity were 69.5%, 61.1%, 56.4% and 77.7% respectively [16]. However, in our study, the maximum train accuracy 99.94%, 99.93%, 99.96% and 99.91% for epochs 41, 44, 47, 49 respectively and the maximum test accuracies was 99.79%, 99.77%, 99.78%, and 99.78% for epochs 41, 44, 47, 49 respectively. A comparison of the above results with those of others related studies revealed that, in our work, the test and training accuracies were better than the minimum loss. We have taken different epochs, and models and illustrated more results to increase the learning rate and accuracy. Figure 7 illustrates the changes in accuracy and loss during the training process for the convolutional neural network (CNN) model. The "accuracy" line represents the percentage of correct predictions made by the model on the training and validation datasets, highlighting its performance.

The "loss" curve shows the model's error rate, calculated using the binary cross-entropy loss function. A steady decrease in the loss curve indicates effective learning, while the accuracy curve demonstrates the model's ability to generalize. Discrepancies between training and validation curves may indicate overfitting or underfitting. By analyzing these trends, the model's convergence and areas for optimization can be identified. Here the learning rate is set at 0.001 after tuning for stability and convergence. And batch size is fixed at 32 to balance gradient updates and memory usage. Epochs have stopped at 49 based on validation loss trends and the risk of overfitting. These steps demonstrate a systematic approach to optimizing hyperparameters and justify the choices made for the bone fracture detection model. For example, previous studies often achieved accuracies ranging from 85% to 90% by employing traditional machine learning methods or basic deep learning models.

Many focused on specific bone types or fractures and lacked the generalization capability required for diverse clinical applications. In contrast, the present study demonstrates superior performance, achieving a maximum accuracy of 99.94% using a CNN architecture tailored for fracture detection across various bone types. The inclusion of advanced techniques, such as hyperparameter optimization, multiple convolutional layers for hierarchical feature extraction, and robust evaluation metrics (sensitivity and specificity), sets this study apart. Furthermore, while earlier studies rarely addressed class imbalance, this research highlights strategies like class-weight adjustments and data augmentation to mitigate bias, enhancing the model's reliability.

Description of Shortcomings and Improvements

A significant drawback of a convolutional neural network (CNN) model for bone fracture detection from X-ray images is its limited generalization capacity. However, CNN models may find it difficult to reliably distinguish fractures, such as rare fracture patterns, different imaging characteristics, or particular patient demographics, in X-ray images from cases that have never been previously observed. This is because the model is not able to learn and generalize well on diverse and uncommon scenarios that were not comprehensively represented in the training data. A representative and diverse training dataset covering a broad range of fracture types, patient demographics, and imaging characteristics is essential for addressing the generalization problem. Obtaining a large number of samples from a comprehensive dataset can aid in the model's learning of resilient features and patterns [21].

The CNN model can acquire low-level features more successfully if it is pretrained on a sizable dataset from a related domain, such as generic X-ray images or medical images. The particular dataset was used to fine-tune the pretrained model on the bone fracture detection task. Through transfer learning, the model's performance can be enhanced by utilizing information from larger datasets and tailoring it to the particular task at hand [22]. To prevent bias toward the majority class, the dataset was balanced across various fracture types. Reducing class imbalance can improve the model's ability to identify uncommon fracture patterns [23]. To integrate several trained CNN models, ensemble techniques such as model averaging or bagging are used.

By utilizing the advantages of several individual models and mitigating the effects of their shortcomings, ensemble learning can enhance the performance of the model [22]. To avoid overfitting and improve the model's capacity to generalize well to unknown data, regularization techniques such as weight decayor dropout are incorporated. Medical specialist input the data, and the model's performance on fresh data was continuously assessed. The required iterative improvements are achieved by utilizing this feedback to pinpoint and resolve model limitations [23].

Conclusion

Given the labelled training and validation images, the problem statement aimed to investigate the classification accuracy of various supervised and unsupervised models for different Heri X-ray
images: normative vs. anomalous (anomalous being humeri that are broken, fractured, or have implants). To feed the normalized histograms of each image into the supervised learning algorithms, we preprocessed the images. In contrast to our initial expectation, "convolutional neural network" would perform best. We would like to mention that, the model was trained on a specific dataset, which may limit generalizability to other fracture types or imaging settings. We used a variety of datasets for fracture detection and classification in this paper. Fracture detection techniques can be used to automate the labor-intensive and time-consuming process of expert radiologists diagnosing and interpreting radiographs.

Many of the researchers cited state that the biggest obstacle to creating a high-performance classification algorithm is the lack of labeled training data. Currently, no industrystandard model exists that can be applied to the available data. Our next steps will be training on a more diverse dataset, exploring transfer learning or pre-trained models for improved generalization. To ensure clinical applicability, the proposed model can be tested in diverse settings using varied imaging equipment to evaluate its robustness and generalizability. Integration with Picture Archiving and Communication Systems (PACS) will streamline its deployment within existing radiology workflows, enabling seamless access for medical professionals. Additionally, user-friendly interfaces should be developed for real-time implementation, particularly in emergency settings where timely diagnosis is critical. Collaborations with medical professionals are essential for iterative improvements, ensuring the system aligns with clinical needs and fosters trust among practitioners.

These steps will pave the way for practical and impactful integration into healthcare systems. Despite achieving high accuracy in detecting bone fractures, this study has certain limitations. The dataset used for training lacks diversity, potentially limiting the model's ability to generalize across different patient demographics, imaging modalities, and rare fracture types. Additionally, the computational requirements of the CNN architecture present challenges for real-time deployment, particularly in resource-constrained settings. The model also lacks interpretability, which could hinder its acceptance among medical practitioners. Furthermore, it has not yet been extensively validated in clinical environments, which raises concerns about its robustness in real-world applications.

To address these limitations, future work will focus on expanding the dataset to include more diverse and representative samples, including rare and complex fracture cases. Incorporating transfer learning or pertained models could enhance performance and reduce training time. Ensemble techniques may also be explored to improve detection accuracy and robustness. Efforts will be made to optimize the architecture for real-time integration into clinical workflows and emergency settings. Additionally, techniques such as Grad-CAM will be implemented to make the model's predictions more interpretable, fostering greater trust among clinicians. Extensive testing in clinical environments and integration with Picture Archiving and Communication Systems (PACS) will be prioritized to ensure practical usability. Expanding the model's scope to include different bones and fracture types will further enhance its clinical utility.

Required Declarations

Author Contributions

Sultan Mamun and Sadek Hossain Khuka designed and implemented the exploration method, verified its effectiveness, set up the simulated and real experimental environments, and wrote the paper. Md Hasan Ali Sheikh and Arbab Bashir did the experimental analysis.

Financial Support

This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

Conflicts of Interest

We are declaring no conflicts of interest exist.

Ethical Approval
Not applicable.

Acknowledgements
We would like to express their sincere gratitude to all individuals and organizations who supported this research work. Special thanks to my academic advisors, Professor Jiatong Bao and Professor Shengquan Li. Finally, we are grateful to our families and colleagues for their encouragement and unwavering support throughout this endeavor.

References

1. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature, 521*(7553), 436-444.
2. Pranata, Y. D., Wang, K. C., Wang, J. C., Idram, I., Lai, J. Y., Liu, J. W., & Hsieh, I. H. (2019). Deep learning and SURF for automated classification and detection of calcaneus fractures in CT images. Computer methods and programs in biomedicine, 171, 27-37.
3. Urakawa, T., Tanaka, Y., Goto, S., Matsuzawa, H., Watanabe, K., & Endo, N. (2019). Detecting intertrochanteric hip fractures with orthopedist-level accuracy using a deep convolutional neural network. *Skeletal radiology, 48*, 239-244.
4. Dhahir, B. M., Hameed, I. H., & Jaber, A. R. (2017). Prospective and Retrospective Study of Fractures According to Trauma Mechanism and Type of Bone Fracture. *Research Journal of Pharmacy and Technology, 10*(11), 3810-3818.
5. Bandyopadhyay, O., Biswas, A., & Bhattacharya, B. B. (2016). Long-bone fracture detection in digital X-ray images based on digital-geometric techniques. *Computer methods and programs in biomedicine, 123*, 2-14.
6. Cao, Y., Wang, H., Moradi, M., Prasanna, P., & Syeda-Mahmood, T. F. (2015, April). Fracture detection in x-ray images through stacked random forests feature fusion. *In 2015 IEEE 12th international symposium on biomedical imaging* (ISBI) (pp. 801-805). IEEE.
7. Umadevi, N., & Geethalakshmi, S. N. (2012, July). Multiple classification system for fracture detection in human bone x-ray images. *In 2012 Third International Conference on Computing, Communication and Networking Technologies* (ICCCNT'12) (pp. 1-8). IEEE.
8. Lum, V. L. F., Leow, W. K., Chen, Y., Howe, T. S., & Png, M. A. (2005, September). Combining classifiers for bone fracture detection in X-ray images. In IEEE International Conference on Image Processing 2005 (Vol. 1, pp. I-1149). IEEE.
9. Liang, J., Pan, B. C., Huang, Y. H., & Fan, X. Y. (2010, July). Fracture identification of X-ray image. *In international conference on wavelet analysis and pattern recognition* (pp. 67-73). IEEE.
10. Al-Ayyoub, M., Hmeidi, I., & Rababah, H. (2013). Detecting Hand Bone Fractures in X-Ray Images. J. Multim. Process. *Technol., 4*(3), 155-168.
11. Chai, H. Y., Wee, L. K., Swee, T. T., Salleh, S. H., & Ariff, A. K. (2011). Gray-level co-occurrence matrix bone fracture detection. *American Journal of Applied Sciences, 8*(1), 26.
12. Mahendran, S. K., & Baboo, S. S. (2011). An enhanced tibia fracture detection tool using image processing and classification fusion techniques in X-ray images. *Global Journal of Computer Science and Technology, 11*(14), 22-28.
13. Marellapudi, V. R. (2023). Building and Training a Custom Convolutional Neural Network with PyTorch using Cow Teat Image Dataset.
14. Bagaria, R., Wadhwani, S., & Wadhwani, A. K. (2022). Bone fracture detection in X-ray images using convolutional neural network. *International Journal of Creative Research Thoughts, 10*(6), 43.
15. Yamamoto, N., Rahman, R., Yagi, N., Hayashi, K., Maruo, A., Muratsu, H., & Kobashi, S. (2020, September). An automated fracture detection from pelvic CT images with 3-D convolutional neural networks. *In 2020 International Symposium on Community-centric Systems* (CcS) (pp. 1-6). IEEE.
16. Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., ... & Webster, D. R. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *jama, 316*(22), 2402-2410.
17. Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS one, 10*(3), e0118432.
18. N. Srivastava, et al., "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res., vol. 15,* pp. 1929–1958, 2014.
19. Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., & Elhadad, N. (2015, August). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. *In Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1721-1730).
20. Ciresan, D., Giusti, A., Gambardella, L., & Schmidhuber, J. (2012). Deep neural networks segment neuronal membranes in electron microscopy images. *Advances in neural information processing systems, 25.*
21. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278-2324.
22. Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., ... & Dean, J. (2019). A guide to deep learning in healthcare. *Nature medicine, 25*(1), 24-29.
23. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research, 15*(1), 1929-1958.