Bone Fracture Detection Using Convolutional Neural Networks

Chiranjeevi Inamati[1], Rajashekhar G C[2]

[1]Student, Department of MCA, GM University, Davangere

[2]Associate Professor & Director , FCIT , GM University, Davanagere

E-mail: chiranjeeviinamati@gmail.com

Abstract

Bone fractures represent a significant and common clinical challenge, with diagnostic accuracy being paramount for effective patient treatment and recovery. While radiographic imaging is the standard diagnostic modality, the manual interpretation process is susceptible to human error, particularly in identifying subtle or hairline fractures. This can lead to missed diagnoses and adverse patient outcomes. In response, deep learning, particularly Convolutional Neural Networks (CNNs), has emerged as a powerful tool for automating medical image analysis. This study presents a rigorous comparative analysis of three distinct CNN-based architectural paradigms for the automated detection of bone fractures from X-ray images. The objective was to evaluate the efficacy of a foundational CNN model, a deep transfer learning model (VGG16), and a region-based object detection model (R-CNN) on a curated dataset of 221 radiographic images. The models were trained and evaluated on their ability to classify images as either "fractured" or "normal." The experimental results demonstrate a clear performance hierarchy, with the R-CNN model achieving a superior accuracy of 86.32%, significantly outperforming both the VGG16 model (65.8%) and the baseline CNN (55.69%). This principal finding underscores the critical importance of architectural design in medical imaging tasks. The superior performance of the R-CNN framework suggests that for pathologies like bone fractures, which are often localized, two-stage object detection architectures that explicitly identify regions of interest before classification are more effective than global image classification approaches. This research contributes empirical evidence to guide the development of more accurate and reliable automated diagnostic systems, highlighting the potential of region-based models to serve as powerful assistive tools for radiologists.

1. Introduction

1.1. The Clinical Imperative for Automated Fracture Detection

A bone fracture is a break in the continuity of a bone and stands as one of the most common traumatic injuries seen in clinics across the globe. Fractures can be caused by high-impact events like falls and accidents or by underlying health conditions that weaken the bone structure. A prompt and correct diagnosis is fundamental in emergency and orthopedic medicine because it determines the treatment plan and is vital for avoiding long-term issues such as improper healing, chronic pain, or permanent disability.

The main tool for diagnosing fractures is radiography, commonly known as X-ray imaging. However, even with its widespread use, manually interpreting these images has its difficulties. Diagnostic precision can be affected by factors like the subtlety of the fracture (e.g., hairline fractures), the intricacy of the bone structure, or poor image quality. Moreover, in busy environments like emergency rooms, radiologist fatigue can heighten the risk of errors and missed diagnoses. Even skilled doctors can miss minor fractures, which, if not treated, could develop into more serious problems. This potential for error in manual diagnosis highlights a clear clinical need for reliable automated systems that can support medical professionals, boost diagnostic accuracy, and make patient care more efficient

1.2. The Emergence of Deep Learning in Radiographic Analysis

Over the last ten years, the field of medical image analysis has been transformed by significant progress in artificial intelligence (AI), especially deep learning. Within this field, Convolutional Neural Networks (CNNs) have been particularly successful at tasks involving visual information, changing the way complex medical images are analyzed. Traditional machine learning methods required human experts to manually identify and select features from an image, a subjective and time-consuming process. In contrast, CNNs can learn hierarchical feature representations directly from the image data itself.

A CNN processes an image by passing it through a series of layers, where each layer is trained to detect patterns of increasing complexity. The initial layers might identify basic features like edges or colors, while subsequent layers learn to recognize more abstract concepts, such as anatomical structures or signs of disease. This ability to automate feature extraction has allowed CNNs to match or even surpass human performance on many medical diagnostic tasks. The proven success of CNNs provides a solid technological basis for tackling the challenges associated with manual bone fracture detection.

1.3. Research Objectives and Contributions

The main goal of this research is to systematically compare the performance of three different CNN-based architectures for classifying bone fractures in X-rays. Our investigation seeks to identify which design is most suitable for this particular clinical application. The three models we compared are:

- **A foundational, custom-built CNN (Conv-Net):** This model was created to set a baseline for performance.
- **A VGG16 model:** This represents a deep CNN using transfer learning, which allows us to test the effectiveness of using a pre-trained feature extractor.
- **A Region-Based CNN (R-CNN):** This model represents a two-stage object detection framework, used to assess an architecture that first localizes potential areas of interest before classifying them.

The main contribution of this work is the empirical evidence showing that the network's architectural design is a critical factor in the performance of automated fracture detection. By demonstrating that the R-CNN model significantly outperformed the others, our study offers important insights into which deep learning strategies are best suited for medical imaging challenges. Specifically, our findings suggest that for identifying localized conditions like bone fractures, an object detection approach may be inherently more effective than a standard image-wide classification method.

2. Background and Related Work

2.1. The Architectural Principles of Convolutional Neural Networks

To understand the models we evaluated, it's helpful to know the basic components of a CNN. CNNs are a special type of neural network built to handle data with a grid-like structure, such as an image. A standard CNN is made up of several key layers that work together to process an input image and produce an output, like a classification.

- **Convolutional Layer:** This is the core component of a CNN. It uses a set of learnable filters, or kernels, that slide across the input image. Each filter is designed to find a specific feature, like an edge or a texture. As the filter moves, it creates a feature map that shows where that specific feature is located in the image. By stacking these layers, the network can learn a hierarchy of features, from simple to complex.
- **Activation Function:** After each convolutional layer, an activation function like the Rectified Linear Unit (ReLU) is applied. This function introduces non-linearity, which is vital for allowing the model to learn the complex relationships in the data. ReLU is computationally efficient and helps prevent certain training problems.
- **Pooling Layer:** These layers are used to reduce the spatial size of the feature maps. The most common method, Max Pooling, reduces the size of the feature map while retaining the most important information. This helps decrease computational load and makes the model more robust to small shifts in the image.
- **Fully Connected Layer:** After passing through several convolutional and pooling layers, the resulting feature maps are flattened into a single vector. This vector is then fed into one or more fully connected layers, which analyze the features to perform the final classification.

2.2. Seminal Architectures in Image Analysis

The evolution of CNNs has been marked by the development of several landmark architectures that have significantly advanced the state-of-the-art in computer vision. The models used in this study, VGG16 and R-CNN, represent two distinct and influential architectural philosophies.

2.2.1. VGG16: Deep Architectures for Feature Extraction

The VGG16 model, developed by the Visual Geometry Group at the University of Oxford, is a deep CNN architecture that achieved outstanding performance in the 2014 ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The defining characteristic of the VGG architecture is its simplicity and uniformity. It demonstrates that significant performance gains can be achieved by increasing the network's depth using a consistent and stacked arrangement of very small (3×3) convolutional filters. The VGG16 variant consists of 16 layers with learnable weights—13 convolutional layers and 3 fully connected layers.

The profound depth of VGG16 allows it to learn a rich hierarchy of features, making it a powerful feature extractor. A key application of VGG16, particularly relevant in the medical domain where labeled data is often scarce, is *transfer learning*. This technique involves using a model pre-trained on a large, general-purpose dataset like ImageNet and adapting it to a new, more specialized task. The underlying hypothesis is that the features learned from natural images (e.g., edges, shapes, textures) are transferable to other domains, such as medical imaging. By leveraging the pre-trained weights of VGG16, researchers can build highly effective models for medical image analysis with significantly less data than would be required to train a deep network from scratch.

2.2.2. R-CNN: A Two-Stage Framework for Object Localization

The Region-Based Convolutional Neural Network (R-CNN) represents a pioneering effort that fundamentally shifted the approach to object detection, moving beyond simple image classification. Unlike architectures like VGG16 that process an entire image to produce a single class label, R-CNN introduced a multi-stage pipeline designed to first identify potential object locations and then classify the content within those locations. This two-stage framework is particularly well-suited for tasks where the object of interest may occupy only a small portion of the image. The R-CNN workflow consists of three key steps :

1. **Region Proposal:** The process begins by generating a set of candidate object locations, known as "regions of interest" (ROIs). The original R-CNN implementation uses an external algorithm called Selective Search, which employs hierarchical grouping of image segments based on color, texture, and size to propose approximately 2,000 ROIs per image. This step effectively narrows down the search space from the entire image to a manageable number of relevant patches.

2. **Feature Extraction:** Each proposed region is then warped to a fixed size (e.g., 224×224 pixels) and passed through a pre-trained CNN (such as AlexNet or VGG) to extract a fixed-length feature vector. This vector serves as a rich, high-level representation of the content within the ROI.

3. **Classification and Refinement:** Finally, a set of class-specific linear Support Vector Machines (SVMs) are trained to classify the object within each region based on its extracted feature vector. A separate bounding box regression model is also trained to refine the coordinates of the proposed ROIs, improving the localization accuracy of the final detected objects.

This architectural distinction is critical. While a standard CNN asks, "Is there a fracture in this image?", the R-CNN framework effectively asks, "Where in this image might a fracture be, and what is in those specific locations?". This inherent localization capability makes it a compelling candidate for detecting discrete pathological findings like bone fractures.

2.3. A Review of Automated Systems for Bone Fracture Diagnosis

The application of deep learning to automate bone fracture detection has become an active and promising area of research. Numerous studies have demonstrated the potential of CNN-based models to assist radiologists and improve diagnostic accuracy across various anatomical regions and imaging modalities, including radiographs and computed tomography (CT) scans.

The literature reveals the use of a wide range of CNN architectures for this task. Many studies have successfully employed well-established classification networks like VGG, ResNet, and DenseNet, often using transfer learning to leverage features learned from large-scale natural image datasets. These models have shown high accuracy in classifying images containing fractures of the wrist, hip, and other areas. Other research has focused on segmentation-based approaches using architectures like U-Net, which are designed to produce pixel-level masks that delineate the exact boundaries of the fracture, providing more detailed information than a simple classification label.

Despite these successes, the field continues to face challenges. The detection of subtle, non-displaced, or hairline fractures remains difficult, as these features can be easily missed by models that analyze images globally. Furthermore, ensuring that models generalize well across different patient populations, imaging equipment, and clinical settings is a significant hurdle that requires validation on large, diverse datasets. Recent trends in the field have moved towards incorporating mechanisms that mimic the diagnostic process of human experts, such as attention mechanisms that allow the model to focus on the most salient regions of an image, and object detection frameworks that explicitly localize potential abnormalities. This study contributes to this body of work by directly comparing a global classification approach with a region-based object detection framework, aiming to provide clear evidence on the importance of architectural choice for this specific clinical problem.

3. Materials and Methods

3.1. Dataset Curation and Preprocessing

The data for this study was gathered from publicly available bone fracture image collections, including sources like Kaggle and GitHub[73]. Our final dataset contained 221 radiographic X-ray images from various parts of the body, such as the hand, leg, and wrist[74]. Every image was manually labeled as either "fractured" or "normal"[75]. The dataset was divided into a training set for teaching the models and a testing set for final evaluation[76]. The data was slightly imbalanced, with 57.47% of images labeled as "fractured" and 42.53% as "normal"[77].

To prepare the images for the neural networks, we performed several preprocessing steps[78]. These included:

- **Image Resizing:** All images were resized to a standard dimension of 224×224 pixels to meet the input requirements of the network architectures[79].
- **Normalization:** Pixel values were rescaled from the original 0-255 range to a smaller range, such as 0 to 1, to help stabilize and speed up the training process[80].
- **Data Conversion:** The processed images were converted into numerical tensors, which is the standard format used by deep learning frameworks[81].
- **Data Augmentation:** To expand our small dataset and reduce the risk of overfitting, we used data augmentation[82]. During training, we applied on-the-fly transformations to the images, such as random rotations, shifts, and zooms, using the ImageDataGenerator class in Keras[83]. This created more training examples and helped the models learn more generalizable features[84].

3.2. Design of Experimental Models

- **Model A: Baseline CNN (Conv-Net):** We built a standard, relatively simple CNN to serve as a performance baseline[86]. Its architecture consisted of several blocks, each with a convolutional layer (using 3×3 kernels), a ReLU activation function, and a Max Pooling layer (with a 2×2 window) for down-sampling[87]. The final feature maps were flattened into a vector and passed through a dense layer, leading to a single output neuron with a Sigmoid activation function to predict the probability of a fracture[88].
- **Model B: Transfer Learning with VGG16:** Our second model utilized the VGG16 architecture via transfer learning[89]. We used a VGG16 model pre-trained on the ImageNet dataset as our convolutional base for feature extraction[90]. We removed the original top classification layers and added our own custom classifier, which included a flatten layer, a dense layer, and a final Sigmoid output layer[91]. During the initial training, the weights of the pre-trained VGG16 base were frozen to preserve the powerful features learned from ImageNet, and only the new classifier head was trained on our fracture dataset[92929292].
- **Model C: Region-Based Detection with R-CNN:** The third model adopted the two-stage R-CNN approach, shifting from image classification to object detection[93]. The process followed the classic R-CNN workflow [94]: first, the Selective Search algorithm was used to generate around 2,000 region proposals (ROIs) for each image[95]. Each ROI was then warped to a fixed size and processed by a pre-trained CNN (VGG16) to extract a feature vector[96]. Finally, a pre-trained linear SVM classifier determined whether each ROI contained a fracture, and a bounding box regressor refined the location of positive detections.

3.2.1. Model A: A Baseline Convolutional Neural Network (Conv-Net)

To establish a performance baseline, a standard, relatively shallow CNN was constructed. This model was designed with a sequential architecture comprising fundamental CNN components. The architecture consisted of several blocks, each containing a convolutional layer with a set of learnable filters (e.g., 32 or 64 filters with a 3×3 kernel size), followed by a ReLU activation function to introduce non-linearity. After each convolutional layer, a Max Pooling layer with a 2×2 window was used to down-sample the feature maps, reducing dimensionality and building spatial invariance. The final block of convolutional and pooling layers was followed by a Flatten layer, which converted the 2D feature maps into a 1D vector. This vector was then passed through a Dense (fully connected) layer for high-level feature integration, and finally to a single-neuron output layer with a Sigmoid activation function to produce a probability score between 0 and 1, indicating the likelihood of the image containing a fracture.

3.2.2. Model B: Transfer Learning with VGG16

The second model leveraged the power of transfer learning using the VGG16 architecture. A pre-trained VGG16 model, with weights learned from the extensive ImageNet dataset, was instantiated as the convolutional base. This base serves as a sophisticated feature extractor. The top classification layers of the original VGG16 model (the fully connected layers) were removed, and a new custom classifier was added on top. This new head consisted of a Flatten layer, followed by one or more Dense layers with ReLU activation, and a final Sigmoid output layer for the binary fracture classification task. During the initial training phase, the weights of the pre-trained convolutional base were "frozen," meaning they were not updated. This ensures that the powerful, generalized features learned from ImageNet are preserved. Only the weights of the newly added classifier head were trained on the fracture dataset, allowing the model to adapt the pre-trained features to the specific task at hand.

3.2.3. Model C: Region-Based Detection with R-CNN

The third model implemented the two-stage R-CNN pipeline, fundamentally shifting the approach from image classification to object detection. The implementation followed the classic R-CNN workflow:

1. For each input image, the Selective Search algorithm was first applied to generate approximately 2,000 candidate region proposals (ROIs).

2. Each of these ROIs was then extracted from the image and warped to the fixed input size required by the feature extractor.

3. A pre-trained CNN (consistent with the VGG16 base used in Model B for fair comparison) was used to perform a forward pass on each warped ROI, extracting a high-dimensional feature vector.

4. Finally, a pre-trained linear Support Vector Machine (SVM) classifier, specifically trained for the binary task of distinguishing "fracture" from "non-fracture" regions, was used to classify each ROI based on its feature vector. A bounding box regressor was also employed to refine the coordinates of the positively classified regions for more precise localization.

3.3. Training Protocol and Hyperparameter Configuration

All models were implemented and trained in a consistent environment using Python with the TensorFlow and Keras libraries. We used Scikit-learn for the SVM classifier in the R-CNN model. Experiments were run on a system with at least 8 GB of RAM.

For training, we used the Binary Cross-Entropy loss function, which is suitable for binary classification problems. After experimenting with different optimizers, we found that RMSprop delivered the most stable and accurate results for the R-CNN model, so it was used for the final experiments. To prevent overfitting, we incorporated Dropout regularization in the fully connected layers of our models. Hyperparameters such as the learning rate, batch size, and number of epochs were carefully tuned for each model to achieve the best possible performance

4. Conclusion

This study conducted a rigorous comparative evaluation of a baseline CNN, a VGG16 transfer learning model, and a region-based R-CNN for the task of automated bone fracture detection in radiographic images. The results conclusively demonstrated the superior efficacy of the R-CNN architecture, which achieved an accuracy of 86.32%, significantly surpassing the performance of the global image classification models.

The primary takeaway from this research is the critical importance of aligning the architectural design of a deep learning model with the specific characteristics of the clinical problem. For the detection of localized pathologies like bone fractures, an object detection framework that first proposes and then analyzes specific regions of interest proves to be a more effective strategy than a standard classification approach that processes the image globally. This finding provides a strong rationale for prioritizing the development and refinement of region-based and attention-aware models for similar diagnostic tasks in medical imaging.

While acknowledging the limitations of the study, particularly the small dataset size, the clear performance gap observed offers a valuable contribution to the field. It underscores a fundamental principle that can guide future research in computer-aided diagnosis. By continuing to explore more advanced object detection architectures and integrating explainability methods, the potential to develop powerful, reliable, and trustworthy AI-driven tools to assist radiologists is immense. Such tools promise to enhance diagnostic accuracy, improve workflow efficiency, and ultimately lead to better patient care and outcomes.

References

Goyal, A. R., Mehta, A., & Goel, H. (2024). BONE FRACTURE DETECTION USING CONVOLUTIONAL NEURAL NETWORKS. *International Journal of Novel Research and Development*, *9*(4).

SCITEPRESS. (2024). *Convolutional Neural Networks for Medical Image Processing*.

MDPI. (2025). *Deep Convolutional Neural Networks in Medical Image Analysis: Evolution, Architectures, and Applications*.

MDPI. (2024). *A Review of Convolutional Neural Network-Based Methods for Medical Image Understanding*.

RSNA Publications. (2018). *Deep Learning: A Primer for Radiologists*.

PMC. (2020). *A survey on applications of convolutional neural networks for medical image analysis*.

PMC. (2024). *Application of convolutional neural networks in medical imaging: a bibliometric analysis*.

Frontiers in Medicine. (2024). *Attention-based deep learning model with dilated convolutions and skip connections for bone fracture detection in X-ray images*.

Kaggle. (n.d.). *Bone Fracture Detection*.

PMC. (2020). *Deep learning for fracture detection on radiographs and CT*.

PMC. (2023). *Deep learning in bone fracture diagnosis from X-ray images: a scoping review*.

MDPI. (2024). *Artificial Intelligence in Bone Fracture Detection: A Systematic Review*.

PMC. (2022). *A Systematic Review of the Use of Deep Learning in Bone Imaging*.

GeeksforGeeks. (n.d.). *VGG-16 | CNN Model*.

GitHub. (n.d.). *VGG16*.

Built In. (2024). *Beginner's Guide to VGG16 Implementation in Keras*.

Medium. (n.d.). *Everything you need to know about VGG16*.

PyTorch. (n.d.). *torchvision.models.vgg16*.

Wikipedia. (n.d.). *Region Based Convolutional Neural Networks*.

MathWorks. (n.d.). *Getting Started with R-CNN, Fast R-CNN, and Faster R-CNN*.

Ultralytics. (n.d.). *What is R-CNN? A Quick Overview*.

GeeksforGeeks. (n.d.). *R-CNN (Region-based CNNs) – Machine Learning*.