SkeletaX: An AI-Powered System for Bone Fracture Analysis and Medical Information Retrieval

Caslow Chien, Sindhuja Kumar, Serena Theobald

May 2025

1 Abstract

This paper presents SkeletaX, a novel interactive system designed to assist in the analysis of elbow, hand, and shoulder X-ray images and to provide evidence-based information regarding bone fractures. The system integrates deep learning-based image classification techniques with a retrieval-augmented generation (RAG) architecture to deliver accurate medical information from curated sources. SkeletaX features a dual-interface system, allowing users to either query a knowledge base about fracture-related topics or upload X-ray images for automated analysis accompanied by contextually relevant medical information. Our system demonstrates the potential of AI-assisted tools in orthopedic healthcare settings while maintaining appropriate medical disclaimers and human oversight requirements.

The complete project, including models and training details, is available on our GitHub repository: https://github.com/CaslowChien/bone-fracture-detection-chatbot/. The bone fracture model is revised from the work of Alkoby et al.

Keywords: Medical imaging analysis, bone fractures, deep learning, retrieval-augmented generation, healthcare AI

2 Introduction

Bone fractures are among the most common injuries in orthopedic practice, with significant variation in presentation, diagnosis, and treatment protocols. Accurate interpretation of X-ray images requires specialized training, and access to expert radiologists or orthopedic specialists may be limited in many healthcare settings. Additionally, both healthcare professionals and patients often need quick access to reliable information about fracture management, rehabilitation protocols, and expected outcomes.

SkeletaX addresses these challenges through an integrated approach that combines:

- Automated analysis of bone fracture X-ray images using deep learning models
- Contextual information retrieval from medical literature using vector embeddings
- A user-friendly interface to facilitate interaction with the system

This paper describes the system architecture, technical implementation, and potential applications of SkeletaX in clinical and educational settings.

3 Motivation

Bone fractures are a significant global health concern, requiring accurate and timely diagnosis to enable effective treatment. However, fracture detection remains highly dependent on expert radiologists, leading to diagnostic variability, delayed care, and limited accessibility, particularly in resource-constrained settings. Misdiagnosed or undiagnosed fractures can result in severe complications, prolonged recovery, and increased healthcare costs. The rising burden on radiology departments and the need for scalable solutions have fueled interest in AI-driven computer vision systems to assist in fracture detection.

In this study, we present an AI-powered fracture detection system that classifies X-ray images as either "Fractured" or "Normal", accompanied by a confidence score. Unlike traditional approaches that

rely on bounding boxes, our method emphasizes global classification and segmentation-friendly strategies, making outputs easier to interpret and integrate into real-world clinical settings without the need for manual region annotation.

Our pipeline is further integrated with a fine-tuned Large Language Model (LLM) chatbot, trained to provide more in-depth medical suggestions and risk assessments based on the model's predictions. This component delivers contextualized medical suggestions and risk assessments based on the model's outputs. Together, this multimodal system offers scalable, explainable diagnostics that reduce human error and streamline clinical decision-making.

While segmentation-based methods can provide fine-grained localization and more precise accuracy, our focus on classification enables rapid and efficient fracture identification across diverse radiology image types. This approach supports earlier intervention, broadens access to care, and helps democratize diagnostic support across both-well resourced hospitals and underserved regions.

4 Related Work

4.1 LLM for ChatBot

The LLM + RAG (Retrieval-Augmented Generation) architecture, which augments large language models (LLMs) with retrieval systems, is pertinent to the proposed project's aim of developing an automated fracture detection and diagnostic reasoning system.

This architecture enhances the accuracy and relevance of model-generated outputs by incorporating external knowledge sources, such as a vector database like Milvus. The retrieval process enables the model to access information that is not contained within its training data, thereby improving its capacity to answer complex queries and provide precise, context-specific insights. Relevant research demonstrates the efficacy of this approach.

For instance, Anthropic's work on building effective agents highlights the importance of leveraging external data to refine decision-making, a strategy that is directly applicable to enhancing the diagnostic accuracy of AI systems in medical contexts [1]. Similarly, AWS defines RAG as the combination of LLMs and external data retrieval, facilitating the generation of more detailed and context-aware responses [2].

This approach aligns with the project's objective of improving diagnostic precision in fracture detection. Furthermore, a tutorial on integrating Milvus with LangChain illustrates how transforming data into vector embeddings and performing similarity searches can enhance the model's ability to retrieve relevant information [3].

In summary, the LLM + RAG architecture, by integrating retrieval techniques with language generation, offers a robust framework for producing more accurate and informed outputs, making it a highly suitable method for improving diagnostic reasoning and accessibility in the context of automated fracture detection.

4.2 Computer Vision for Fracture Detection

Several studies in the field of computer vision have contributed valuable insights and methodologies that are directly applicable to our project on automated fracture detection. One such study, WCAY Object Detection of Fractures for X-ray Images of Multiple Sites [4], explores the use of object detection techniques for identifying fractures across various anatomical sites in X-ray images. This work underscores the potential for applying deep learning-based object detection to accurately locate fractures in radiographic images, which is a key component of our approach.

In FracAtlas: A Dataset for Fracture Classification, Localization, and Segmentation of Musculoskeletal Radiographs, the authors introduce a comprehensive dataset of 4,083 X-ray images with 922 instances of fractures [5]. These images are manually annotated for bone fracture classification, localization, and segmentation, making the dataset a valuable resource for training and evaluating machine learning algorithms for bone fracture diagnosis. The dataset's availability and the high-quality annotations it provides make it an essential tool for advancing research in the field, especially in areas of fracture detection and segmentation.

In FractuVision: Enhancing Bone Fracture Diagnostics, the authors fine-tune YOLOv8, achieving localization precision of 77%, recall of 55%, and mAP50 of 0.53. This study demonstrates the effectiveness of the YOLO architecture in fracture localization, providing a benchmark for model performance in medical image analysis, and informing the choice of segmentation-based methods in our project [6].

Another relevant work, Window Loss for Bone Fracture Detection and Localization in X-ray Images with Point-based Annotation [7], introduces a novel loss function, Window Loss, which addresses the ambiguity in fracture scales and boundaries in X-ray images by utilizing point-based annotations. This methodology is significant for improving fracture localization accuracy, particularly in cases where traditional boundary definitions are unclear, thus offering insights for refining our segmentation techniques.

Together, these studies provide a solid foundation for advancing fracture detection systems through deep learning, offering both technical methodologies and performance metrics that are crucial for our project's development of a robust, accurate, and interpretable fracture detection system.

5 Approach (Methodology)

5.1 Model & Training approach

5.1.1 Bone Fracture Detection

Our approach involves modifying existing open-source models and repo from Alkoby et al., which implemented ResNet50-based classification models for four anatomical regions: elbow, hand, shoulder, and other body parts.

The original model achieved approximately 80% status accuracy. To extend and improve upon this foundation, we explore alternative deep learning architectures in PyTorch, including transformer-based models, and apply transfer learning techniques to enhance model generalizability across datasets.

Rather than relying on traditional bounding box detection, our approach focuses on whole-image classification and segmentation-aware strategies, aiming to improve diagnostic precision while simplifying deployment in clinical environments. We evaluate model performance using metrics such as accuracy, AUC, and confusion matrices.

For the chatbot component, we intend to utilize an open-source large language model (LLM), such as deepseek-r1:1.5b, which offers a balance between computational efficiency and performance. This model will be fine-tuned to our available resources to optimize its capability to assist with fracture detection and related inquiries.

We will implement knowledge distillation after training if the model exhibits long inference times or high computational costs. This technique will allow us to transfer knowledge from a larger, more complex model to a smaller, more efficient one, reducing the number of parameters while preserving performance.

For X-ray analysis, the system enhances user queries with classification results to provide more targeted information:

```
def process_xray_and_query(image, user_query):
    """
    Function to process uploaded X-ray image and user query.
    """
    # Process the image with the classifier
    classification_result = inference(temp_img_path, user_query)

    # Enhance the query with the classification results
    enhanced_query = f"Information about {classification_result['body_part']} fracture:
    text_response = query_vector_store(vector_db, enhanced_query)

    # Format the detailed report
    confidence_percentage = f"{classification_result['confidence'] * 100:.1f}%"

    report = f"""# X-Ray Analysis Report

## Classification Results:
- **Body Part**: {classification_result['body_part']}
- **Diagnosis**: {classification_result['fracture_status'].capitalize()}
- **Confidence**: {confidence_percentage}

## Expert Analysis:
{text_response}
```

Note: This is an AI-assisted analysis and should not replace professional medical advice. Please consult with a healthcare provider for proper diagnosis and treatment.

5.1.2 Retrieval-Augmented Generation (RAG)

SkeletaX implements a robust RAG architecture to ensure that responses are factually grounded in medical literature. The query vector store function demonstrates this approach:

```
def query_vector_store(vector_db: Optional[Chroma], query: str) -> str:
    """
    Queries the vector store with a given query and retrieves relevant documents.
    Implements a two-stage prompt.
    """
    # Step 1: Staging Prompt to reformulate the query
    staged_query_prompt = staging_prompt.format(user_query=query)
    staged_response = llm(staged_query_prompt).strip()

    # Use the staged response for document retrieval
    effective_query = staged_response

    # Step 2: Document Retrieval
    retrieved_docs = vector_db.similarity_search(effective_query, k=TOP_K)

    # Step 3: Context Compilation
    context_text = ""
    for i, doc in enumerate(retrieved_docs):
        source = doc.metadata.get('source', 'Unknown')
        heading = doc.metadata.get('heading', 'General Information')
        context_text += f"Document {i+1}: {source} - {heading}\n{doc.page_content}\n\n"

    # Step 4: Generate Response
    final_prompt = main_prompt.format(context=context_text, question=query)
    response = llm(final_prompt).strip()

    return clean_response
```

The system incorporates multiple retrieval strategies (similarity search, maximum marginal relevance) to enhance the relevance and diversity of retrieved information.

5.2 Implementation Details

5.2.1 Configuration

SkeletaX employs a centralized configuration approach to manage system parameters:

```
# Define project-level constants
VECTOR_DB_DIR = os.path.join(os.getcwd(), "SkeletaX_chroma_db")
TEMP_PDF_DIR = "../temp_pdfs"
DATA_DIR = "../data/bone_fractures_RAG_data.zip"
DEFAULT_LLM = "ollama"
OLLAMA_DEFAULT_MODEL = "deepseek-r1:1.5b"
CHUNK_SIZE = 500
CHUNK_OVERLAP = 50
TOP_K = 5
SEARCH_TYPE = "similarity"
```

This configuration enables easy tuning of system parameters without modifying core code.

5.2.2 Error Handling and Logging

The system implements comprehensive error handling and logging to ensure robustness:

```
# Set up logging
logging.basicConfig(
    level=logging.INFO,
    format='%(asctime)s - %(levelname)s - %(message)s',
    handlers=[
        logging.FileHandler("skeletax_app.log"),
        logging.StreamHandler(sys.stdout)
    ]
)

logger = logging.getLogger(__name__)
```

Each major function includes try-except blocks with appropriate fallback mechanisms:

```
def ask_skeletaX(user_query):
    try:
        # Process the query
        # ...
    except Exception as e:
        error_message = f"Error processing query: {e}"
        logger.error(error_message)
        return "I encountered a problem while processing your question. Please try again
```

5.3 Datasets Source

For the X-ray images, several datasets will be utilized. The "Bone Fracture v2" dataset available on Roboflow (https://universe.roboflow.com/capjamesg/bone-fracture-v2/dataset/3) will serve as one of the primary sources for training the model. Additionally, the GRAZPEDWRI-D dataset (https://www.nature.com/articles/s4 022-01328-z) will also be integrated into the training pipeline, supported by an open-source GitHub model for processing.

The MONAI (Medical Open Network for AI) platform (https://monai.io/) will provide valuable tools and resources for medical image processing. Finally, the Figshare dataset (https://figshare.com/articles/dataset/) will be considered for its variety and quality, enhancing the model's ability to generalize across different fracture types and imaging conditions. These datasets will form the foundation of our AI-powered fracture detection system, ensuring robust and diverse data coverage for accurate training and evaluation.

5.4 Exploratory Data Analysis on Dataset

The training dataset for image labeling provides several key insights. Each image is categorized by bone type—Elbow, Shoulder, or Hand—based on its folder location. Labels also include fracture status (e.g., study1-positive or study3-negative), patient ID, image path, and laterality, which indicates whether the image corresponds to the left or right side (denoted by L and R). Additional metadata markers appear in the images, such as the initials of the radiologic technologist, machine or room codes, location identifiers, and device IDs. These are likely not institution codes, as such information is typically removed to comply with HIPAA and anonymization standards. Some images also contain a circular symbol (O), which serves as a calibration marker to help with positioning, scaling, measurement accuracy, and implant sizing or surgical planning.

Frequent metadata markers such as PORT, INT, OR, and ROT reflect the imaging setup, with PORT indicating a portable X-ray, ROT referring to rotation instructions, OR denoting the operating room, and INT suggesting internal, interventional, or intensive settings. These markers imply that the dataset includes X-rays taken in diverse clinical environments. Other markers like R and L represent laterality, helping to infer whether the image is from the right or left side of the body (e.g., right/left hand, shoulder, or elbow). Additionally, markers such as AEF, ASC, and ZO are most likely technologist IDs or room/machine codes, suggesting that multiple technicians across different imaging setups contributed to the dataset.

6 Evaluation

6.1 Evaluation Matrix

6.1.1 Data Quality and Preprocessing

To evaluate a medical assistance agent for describing X-ray images of fractures using LLM + RAG technology, we should focus on the following key aspects:

6.1.2 Data Quality and Preprocessing

To ensure model effectiveness, we will curate a comprehensive, diverse, and accurately labeled dataset of X-ray images and fracture descriptions, which will serve as the foundation for training and validation. We will also evaluate preprocessing methods, such as image segmentation and object detection, to extract relevant features from the X-rays, enhancing the model's ability to identify and classify fractures accurately.

6.1.3 Model Performance

We will evaluate the model's accuracy in identifying fracture severity and type using metrics such as IoU, precision, recall, and F1 score. Additionally, we will assess the recall and retrieval efficiency of the system in extracting relevant data from external knowledge bases to aid in generating accurate descriptions. The model's ability to combine visual data from X-rays with textual information from medical literature will also be evaluated to ensure effective multimodal understanding.

6.1.4 Clinical Validity and Reliability

We will compare the agent's predictions to expert assessments to ensure clinically useful results. Cross-validation will be employed to evaluate the model's ability to generalize to new images. Additionally, we will ensure clinical safety by assessing the system's reliability and minimizing false positives and false negatives.

6.1.5 Explainability and Transparency

We will assess the agent's interpretability by evaluating whether it explains its reasoning in a manner understandable to clinicians. Additionally, error analysis will be conducted to identify areas where the model fails, helping to refine its predictions and retrieval systems.

6.1.6 User Experience

We will evaluate the system's ease of use by assessing its intuitiveness for clinicians. Additionally, we will measure its speed and scalability, focusing on how quickly it processes and provides results, particularly in real-time environments.

6.1.7 External Feedback

A feedback loop will be implemented, allowing clinicians to provide continuous input to help improve the model's performance.

6.1.8 Regulatory Compliance

We will ensure the system complies with healthcare regulations, such as HIPAA, and safeguards patient data.

6.1.9 Key Metrics

- Accuracy of diagnosis.
- AUC curve/ confusion matrix for classification.
- Retrieval quality (relevance of information).
- Processing speed.

Target	Accuracy (validation)	Improvement
Shoulder 	84.58%	+ 1.59%
Hand 	84.66%	+ 0.49%
Elbow 	86.41%	+ 2.95%

Figure 1: Results of three of the models with accuracy validation.

- Clinician satisfaction.
- Error rate (false positives/negatives).

This comprehensive evaluation ensures the agent provides accurate, clinically valid insights and meets regulatory and user needs.

6.2 Evaluation Results

6.2.1 Data Quality and Preprocessing

Our model performance depends heavily on the quality and structure of the data. In this project, we implement targeted preparation workflows for both X-ray image data and unstructured medical documents to support our vision and language components.

For image preparation and segmentation, we preprocess all X-ray images by standardizing resolution and pixel intensity distributions across datasets. Images are converted to grayscale if needed, normalized, and subjected to various augmentations (e.g., small rotations, flips) to enhance generalization. Rather than relying on bounding boxes, we prioritize segmentation-based labeling, outlining the specific region of interest such as a fractured area of the hand, elbow, or shoulder. This approach improves model focus, especially in cases where fractures are subtle, fragmented, or span irregular boundaries.

Images without detailed segmentation are labeled at the image level (e.g., 'Fractured' vs. 'Normal'), while others include precise region annotations. This hybrid labeling strategy supports both coarse classification and fine-grained localization, allowing the model to learn from diverse annotation types.

To power the chatbot's retrieval-augmented capabilities, we parsed unstructured medical content, including PDFs and web articles, into clean, structured text. The text is chunked into short, semantically coherent units (e.g., sentences or paragraphs), which are then embedded into vector representations using pretrained transformer models. These vectors are indexed in Milvus or Chromadb to enable efficient semantic search at inference time.

This two-tiered data preparation pipeline—one for vision, one for retrieval—ensures that the system is both diagnostically robust and contextually informed when answering user medical queries or classifying fractures.

6.2.2 Model Performance - Bone Fracture Detection

To ensure the robustness and generalizability of our bone fracture detection model, we carefully monitored key performance metrics across the training, validation, and testing stages. Specifically, we aimed to maintain loss values across all three sets—training, validation, and test—below our predefined baseline to avoid issues related to overfitting or underfitting, where all the training data results are agent's predictions. Additionally, we benchmarked the model's accuracy against the original implementation, confirming that it consistently achieved higher performance.

Beyond basic metrics, we conducted further evaluation using the Area Under the ROC Curve (AUC) and the confusion matrix. These analyses helped verify that the model performs consistently across different classes, ensuring fair and unbiased predictions—even though our training dataset was already balanced.

Given that the baseline model already exhibited high accuracy, achieving further improvements posed a considerable challenge. Nevertheless, through iterative experimentation and fine-tuning, we success-

fully enhanced the model's performance without introducing any significant bias, as confirmed by our evaluation metrics.

7 Limitations and Future Work

Despite promising results, SkeletaX has several limitations that should be addressed in future work:

1. Limited anatomical coverage: Currently focused only on elbow, hand, and shoulder fractures
2. Binary classification: Only distinguishes between fractured and normal states, without fracture subtyping
3. Single-modal analysis: Limited to X-ray images without integration of other clinical data

Future work will focus on:

1. Expanding anatomical coverage to include lower extremities and spine
2. Incorporating fracture subtyping and severity assessment
3. Integrating multimodal inputs (clinical history, physical examination findings)
4. Enhancing the explainability of the classification models Implementing differential diagnosis capabilities

8 Conclusion

SkeletaX demonstrates the potential of integrated AI systems in orthopedic care and education. By combining deep learning-based image analysis with retrieval-augmented generation, the system provides valuable assistance for fracture assessment and information retrieval. The modular architecture allows for continual improvement and expansion of capabilities.

While not intended to replace clinical judgment, SkeletaX represents a step toward AI-augmented orthopedic care, with applications in education, remote healthcare settings, and clinical decision support.

References

[1] Anthropic, "Building effective agents," 2024. Accessed: 2025-03-02.

[2] A. W. Services, "What is retrieval-augmented generation (rag)?," 2024. Accessed: 2025-03-02.

[3] GettingStarted.ai, "How to integrate milvus vector database into your rag llm langchain app," 2024. Accessed: 2025-03-02.

[4] P. Chen, S. Liu, W. Lu, F. Lu, and B. Ding, "WCAY object detection of fractures for x-ray images of multiple sites," *Scientific Reports*, vol. 14, p. 26702, Nov 2024.

[5] I. Abedeen, M. Rahman, F. Prottyasha, *et al.*, "Fracatlas: A dataset for fracture classification, localization and segmentation of musculoskeletal radiographs," *Scientific Data*, vol. 10, p. 521, 2023.

[6] A. Guermazi, C. Tannoury, A. J. Kompel, A. M. Murakami, A. Ducarouge, A. Gillibert, X. Li, A. Tournier, Y. Lahoud, M. Jarraya, E. Lacave, H. Rahimi, A. Pourchot, R. L. Parisien, A. C. Merritt, D. Comeau, N. E. Regnard, and D. Hayashi, "Improving radiographic fracture recognition performance and efficiency using artificial intelligence," *Radiology*, vol. 302, pp. 627–636, Mar 2022. Epub 2021 Dec 21.

[7] X. Zhang, Y. Wang, C. Cheng, L. Lu, A. P. Harrison, J. Xiao, C. Liao, and S. Miao, "A new window loss function for bone fracture detection and localization in x-ray images with point-based annotation," *CoRR*, vol. abs/2012.04066, 2020.

Appendix - Midterm Checkpoint

Model

- Reproduced X-ray fracture detection model from GitHub repository as baseline models. For different body parts, there are different models. Here are the baseline models results: model of hand with validation accuracy 0.842; model of shoulder with validation accuracy 0.832; model of elbow with validation accuracy 0.839. Our next plan is to improve all different models. The model is stored on a local laptop for now, instead of GitHub due to the file size. Please let us know if you would like to have a look.
- Conducted initial dataset inspection; detailed findings available in Section 5.4
- Explore the possibility of implementing bounding box
- Modified training code to integrate with Weights and Biases (WandB) and generated baseline model using original implementation.

Chatbot

The methodology seen in the bone-fracture-detection-chatbot GitHub repository employs a Retrieval-Augmented Generation (RAG) architecture, integrating large language models (LLMs) with document retrieval mechanisms to answer medical queries about bone fractures. At the core of the system is a vector database that stores vector embeddings of medical documents. These embeddings capture the semantic meaning of the content, enabling efficient retrieval of the most relevant information based on the meaning of the user's query, rather than relying on exact keyword matches. This retrieval mechanism ensures that even if the user's query is phrased differently from the document content, the system can still retrieve contextually accurate results.

The document processing pipeline plays a critical role in preparing the data for retrieval. Medical documents, typically in unstructured formats like PDFs, are processed to extract the text, which is then cleaned and preprocessed. After extraction, the text is chunked into smaller, more manageable segments, such as sentences or paragraphs, to improve the searchability within the vector database. This chunking allows the system to focus on the most relevant portions of the text, enhancing the accuracy and efficiency of the retrieval process. These smaller chunks of text are then stored in the vector database, where they can be quickly searched when responding to user queries.

When a user asks a question, the system performs a retrieval process by searching the vector database for the most relevant document chunks. These retrieved segments are then passed to an LLM, which generates a human-like response grounded in the retrieved information. The LLM interprets the context of the query and the retrieved documents, providing an accurate and relevant answer. By combining document retrieval with generative language modeling, the RAG architecture enables the chatbot to answer complex medical inquiries with contextually rich, fact-based responses. This methodology allows the system to deliver highly accurate and informative answers, grounded in a broad repository of medical knowledge, ensuring that users receive reliable assistance regarding bone fractures.