X-Care: A Suite of Rehabilitation Tools To Assist X-Ray Patients

Dev Shah
University of Waterloo
d73shah@uwaterloo.ca

Hargun Mujral
University of Waterloo
hmujral@uwaterloo.ca

Ali Al-Hamadani
University of Waterloo
a2alhama@uwaterloo.ca

Helen Li
University of Waterloo
h273li@uwaterloo.ca

Guruprasanna Suresh
University of Waterloo
grajukan@uwaterloo.ca

Mukund Chettiar
University of Waterloo
msayeega@uwaterloo.ca

Mica Shatil
University of Waterloo
mshatil@uwaterloo.ca

Abstract—**This paper introduces a novel diagnostic tool that combines Supervised Learning in image classification, and Retrieval-Augmented Generation (RAG) using LLMs, to support the detection and diagnosis of injuries based on X-ray images for patients. By applying transfer learning with specialized datasets on DenseNet and EfficientNet convolutional neural networks (CNNs), which were pretrained on the ImageNet dataset, the tool approaches expert-level accuracy in detecting and classifying various skeletal fractures. Then, the RAG-powered rehabilitation agent allows for informative plans and recommendations, using clinically verified documents from a comprehensive medical database (PubMed). The fusion of these technologies offers significant potential to improve patient care by providing rapid and sophisticated analyses, in an effort to bridge the gap between diagnostic imaging and consultation with a professional to enable a quicker recovery and reduce patient uncertainty.**

I. Introduction

There is an urgent necessity to advance the quality and speed of X-ray diagnosis. As it currently stands, there is an average delay of 14-days [Vancouver Coastal Health, 2024] between performing an X-ray scan and an appointment with a specialist to discuss results. This gap not only hinders the diagnostic process but also leaves patients in a state of uncertainty and anxiety while awaiting further consultation with healthcare specialists. And considering that up to 25% of X-rays are rejected or must be repeated [Little et al., 2017], this issue is exacerbated. Thus we hope to tackle the problem of the absence of immediate rehabilitative guidance for patients throughout prevalent X-ray procedures.

A. Motivation

Conventional methods of interpreting X-ray images are often hampered by speed and variability of human analysis, which can yield inconsistent diagnostic outcomes [Kuo et al., 2022]. By employing Supervised Learning and Retrieval-Augmented Generation (RAG), we strive for diagnostic precision and consistency surpassing traditional human-led evaluations.

Our aspirations, however, extend beyond diagnostics. Leveraging AI to scrutinize X-ray data, we aim to offer patients personalized rehabilitation plans and provide immediate, informed responses to their queries. This approach is designed to kickstart recovery ahead of specialist consultations, providing vital, timely advice when it is most needed [Sharma, 2023].

The fundamental objective is to elevate patient outcomes through rapid, precise diagnostics and early intervention strategies. Through this project, we underscore our commitment to the application of state-of-the-art AI technologies within the healthcare field, contributing to diagnosis and facilitating a seamless recovery journey.

II. Related Work

Multiple studies have highlighted the efficacy of artificial intelligence in the realm of radiography-based fracture detection. A comprehensive review by a team of researchers in England, encompassing 42 studies, revealed comparable diagnostic performance between AI and clinicians, with both exhibiting a sensitivity of 91-92% [Kuo et al., 2022]. Notably, the BoneView model released by startup Gleamer in France, demonstrated proficiency to a degree higher than musculoskeletal radiologists in detecting fractures across diverse anatomical regions [Oppenheimer et al., 2023]. Their approach reduced missed fractures by 29% and enhanced sensitivity by 16%, illustrating the promise of AI in the field of orthopaedic radiology [Guermazi et al., 2022].

Presently, the most common methodology involves training AI models using deep learning networks. A study demonstrated the effectiveness of re-training a pre-existing CNN model, specifically an Inception v3 model, which achieved a binary classification model with 90% sensitivity and 88% specificity in fracture detection [Kim and MacKinnon, 2018]. Additionally, a group of doctors and researchers from Sweden utilized openly available deep learning networks to create AI fracture detectors that performed similarly to senior orthopaedic surgeons when analyzing medical images [Olc, 2017].

Although numerous works have shown the effectiveness of employing pre-trained deep-learning networks for the task of classifying fractures, integrating AI into real-world systems to benefit patients beyond mere fracture detection persists to be an issue. Novel techniques using Large Language Models (LLMs), such as RAG, have proven to reduce hallucination and provide accurate information when summarizing radiology papers, extracting relevant information and returning deterministic data [Lewis et al., 2021]. This approach can provide patients with recent and precise information extending from the training data in the LLM, and when used effectively toward creating rehabilitation plans, can allow for the initiation of treatment without delays [Sharma, 2023].

III. Methodology

A. X-Ray Classification & Recognition: Approach

Our methodology for X-ray image recognition and interpretation would be divided into two phases: binary classification to determine the exstence of a fracture, and the subsequent identification of the fracture's precise location within the X-ray scan.

a) Initial Research and Planning: We commenced the project with an in-depth investigation into the prevalent challenges within X-ray dagnostics, followed by strategic planning. For the binary classification task, we selected to experiment between DenseNet and EfficientNet architectures, targeting the detection of fractures with the FracAtlas Dataset. For localization of fractures, we planned to incorporate an object detection model, such as Faster R-CNN.

b) Revised Approach: During the exection phase, we encountered hurdles with the quality of our R-CNN model and the practicality of integrating an object-detection approach with our planned workflow. Theses challenges necessitated a revised approach, leading to the decision of building two CNN-based classification models to independently handle each phase: A binary classifier for fractures, and a multiary classificatier for localization. For the latter, this also required searching for a new dataset more comprehensive in body part classification.

B. Step 1: Presence of a Fracture

1) Model Information: To train our fracture detection model, we utilized the FracAtlas dataset [Abedeen et al., 2023], comprising 4,083 annotated X-ray images. Using the DenseNet121 and EfficientNet architectures as backbones, and weights from pre-trained artifacts using the ImageNet dataset, we trained the models to classify X-ray images as either fractured (1) or non-fractured (0). We used a 70:15:15 split for training, testing and validation to ensure the model's generalization.

2) Testing Results: We iterated on testing different hyperparameters and configurations to optimize the model's performance. By using Optuna, a hyperparameter optimization framework, we employed the Tree-structured Parzen Estimation algorithm to search for the best hyperparameters. This approach of Bayesian optimization allowed us to fine-tune the model's hyperparameters, including the number of epochs, batch size, optimizer, and learning rate.

The initial reference models' hyperparameters, configurations, and accuracy are located in Table I and Table II.

TABLE I
COMPARISON OF MODEL PARAMETERS AND PERFORMANCE BEFORE AND AFTER TUNING (EFFICIENTNET)

Parameter	Initial	Tuned
#(Epochs)	5	3
Batch Size	64	64
Loss Function	BCE	BCE
Optimizer	SGD	Adam
Learning Rate	0.001	0.001
Validation Accuracy	74.56%	88.40%
Test Accuracy	71.84%	86.60%

TABLE II
COMPARISON OF MODEL PARAMETERS AND PERFORMANCE BEFORE AND AFTER TUNING (DENSENET)

Parameter	Initial	Tuned
#(Epochs)	5	2
Batch Size	64	32
Loss Function	BCE	BCE
Optimizer	SGD	Adam
Learning Rate	0.001	0.00019
Validation Accuracy	73.69%	86.30%
Test Accuracy	70.65%	85.15%

BCE: Binary Cross Entropy Loss.

We had inferred from manual inspection of the various trials with different hyperparameter configurations that 3 epochs were sufficient for the model to converge, and more epochs did not significantly impact the test accuracy. After fine-tuning the hyperparameters, we achieved a validation accuracy of 86.6% with the EfficientNet model, a significant improvement over the initial 71.84% accuracy.

Upon experimentation, we saw that, almost universally with the same hyperparameter configurations, EfficientNet outperformed DenseNet in terms of accuracy. Thus, we opted to allocate more resources to testing and tuning the EfficientNet model.

C. Step 2: Fracture Location (Multiary Classification)

In this step, we focus on the body part classification of detected fractures based on their location within X-ray images. Identifying a suitable dataset for this purpose required a comprehensive review of available resources, as many lacked specificity and comprehensive labelling. Below, we detail our evaluation of several datasets considered for this task.

a) UNIFESP X-Ray Body Part Classification Dataset: This dataset consists of 2,481 X-ray images across 22 body parts, annotated in a multilabel format, making it ideal for detailed classification tasks in our second step.

b) VinDr-BodyPartXR: Derived from DICOM scans, it includes general labels such as ”abdominal,” ”pediatric,” and ”adult.” Its broad categorizations, however, lacked the specificity needed for precise classification in our study.

c) MURA (Musculoskeletal Radiographs): A large collection focusing on upper body parts, including 7 specific areas. While extensive, its scope was too narrow for the broad classification goals of our project.

We thus selected the UNIFESP dataset for its comprehensive coverage of body parts, which aligned with our classification objectives.

1) Model Information: We built off of a DenseNet architecture, and tailored it's output layer to support 22 output classes. The training and testing were conducted on an 80:20 split.

2) Data Preprocessing: Upon examination, it was discovered that only the published training set of the UNIFESP dataset had labels. Thus, the number of usable images dropped to 1738 images. Additionally, some images in the dataset were labelled with multiple body parts, which were removed to simplify the classification task and remove unnecessary complexity. This resulted in a final dataset of 1606 images.

The data labels, represented as numerical values ranging from 0 to 21 for each body part, and a 80:20 ratio was employed ratio for training and testing respectively. Due to the small size of the dataset, we wanted to include as many images as possible in the training set, and thus did not use a test set. We intend on experimenting with data augmentation techniques to artificially increase the size of the dataset in the future.

3) Testing Results: In the initial trial, the model was trained with configurations summarized in Table III.

TABLE III
MODEL CONFIGURATION AND PERFORMANCE

Parameter	**Initial**	**Tuned**
#(Epochs)	9	5
Batch Size	64	32
Optimizer	SGD	Adam
Learning Rate	0.001	0.0001
Validation Accuracy	96.18%	99.38%
Test Accuracy	90.06%	91.30%

As shown, the validation accuracy was exceptionally high, reaching up to 99.38%. This hints at a high likelihood of overfitting, and further investigation is required to ensure the model's generalizability. Notably, the final test accuracy was 91.30%, which is higher than expected and a potential cause for concern, considering the task of classifying between 22 labels.

In an attempt to tackle this overfitting, we programmatically checked for image duplicates. Although all images were unique, many shared similarities in terms of backgrounds and bone placement. This lead to data invariance, and potentially contributed to the overfitting. Consequently, we concluded that training the model with a reduced number of epochs (3-5) would be more appropriate, however this needs to be revisited in the future.

Next, ten different model configurations with 3-5 epochs were tested, and the highest scoring model was selected as the final ”tuned” result. We employed manual hyperparameter tuning for this task, by running multiple trials with different configurations in parallel, and iteratively adjusting the hyperparameters based on the results of the trials.

Among the ten models, three exhibited high accuracy (> 70%), while the remaining models yielded accuracy below 50%. This hinted at very high variance, potentially related to the earlier issues of overfitting. To ensure consistent weight initialization across all models, the top-performing models were assessed with a fixed seed.

In general, regarding our specific task, the batch size seemed the least relevant, as successful models were achieved with varying batch sizes. Optimizers SGD and Adam performed well, alongside effective learning rates of 0.001 and 0.0001, as well as weight decays of 0.001 and 0.1.

Further investigation showed that a learning rate of 0.01 would adversely affect the highest accuracy model, and Adam remains the ideal optimizer for this task.

4) Exploratory Advances in Fracture Detection - Employing Faster R-CNN: This research employs a Faster R-CNN model with a ResNet50 backbone, renowned for its object detection prowess, to pinpoint fractures in medical imaging data from the FracAtlas dataset. The methodology integrates data pre-processing, such as image resizing and normalization, and leverages transfer learning to utilize pre-trained weights, enhancing the model's capability to detect fractures accurately [Ren et al., 2015]. Hyperparameter optimization is performed using Optuna, focusing on learning rate, weight decay, and the number of epochs to refine model performance. The training process involves batch processing, forward and backward passes, and loss computation, with the Stochastic Gradient Descent (SGD) optimizer chosen for its effectiveness in deep learning model training. The model is periodically evaluated on a validation set to monitor performance and adjust hyperparameters accordingly [Tor,].

TABLE IV
PERFORMANCE METRICS OF MODEL TRAINING ACROSS DIFFERENT HYPERPARAMETERS

S.No	**Mean IOU**	**Lr**	**Weight_decay**	**No of epochs**
1	0.2032	0.00082	0.00011	6
2	0.2270	0.00095	0.00011	5
3	0.2177	0.00091	0.00010	5
4	0.2033	0.00044	0.00042	5
5	0.2470	0.00098	0.00022	5
6	0.2175	0.00032	0.00037	6

The table IV lists the outcomes of multiple runs with varying hyper parameters. The ‘Mean_IOU‘ column represents the mean Intersection over Union, a common metric for evaluating the accuracy of an object detector on a particular dataset. The ‘Lr‘ column indicates the learning rate, ‘Weight_decay‘ is a regularization term to prevent over fitting, and ‘No of epochs‘ shows how many complete passes the algorithm has made over the entire training dataset.

Results: The dataset records indicate that the model performed best with a mean IOU of approximately 0.2471 when the learning rate was set to roughly 0.000988, paired with a weight decay of about 0.000225 over the course of 5 epochs. Notably, this configuration achieved the highest recorded mean IOU without the need for the most extended training period, indicating a more efficient learning process compared to other configurations. In contrast, the lowest mean IOU observed was approximately 0.2033 with a significantly lower learning rate of 0.000442 and a higher weight decay of 0.000420 across 5 epochs. This outcome could suggest that either the learning rate was too low for the model to make significant updates to its weights or the high weight decay overly penalized the model parameters, hindering its ability to fit the data effectively. In general, for our specific task, the batch size was least relevant, as successful models were achieved with varying batch sizes. Optimizers SGD and Adam both performed well, alongside effective learning rates of 0.001 and 0.0001 and weight decays of 0.001 and 0.1.

D. Patient Diagnosis: RAG

To generate an accurate diagnosis for the provided X-ray, a Retrieval Augmented Generation (RAG) pipeline was developed. This pipeline integrates two embedding models: the open-source HuggingFace model and OpenAI's text-embedding-002-ada model. These models were selected to balance cost and performance, with the HuggingFace model serving as a cost-effective alternative to the more expensive OpenAI model. Additionally, the pipeline utilizes two Large Language Models (LLMs): Cohere's prompting LLM and OpenAI's GPT-4, providing alternatives for cost efficiency and accuracy.

The RAG approach was preferred over fine-tuning models like GPT-4 due to its demonstrated accuracy with smaller datasets and the impracticality of annotating each prompt for GPT-4 fine-tuning. In the pre-processing stage, medical diagnosis documents from PubMed and Springer were converted into vector embeddings using Cohere's Embedding model.

Upon classification of the user's X-Ray image and embedding of the fracture location and other patient information, this data is used to query a vector database. Relevant documents are then retrieved to provide context for the GPT-4 LLM, which generates the diagnosis response.

The first stage in building the RAG system was to identify high value medical diagnosis documents that can be considered factual information. After reviewing over ten sources of medical research papers, Pubmed and Springer were decided as the top two sources due to the quality, scope, and size of their databases. Furthermore, Pubmed's API enables easy data ingestion. In Springer's case, beautiful soup was used to scrape the articles.

The second step of implementing the RAG pipeline is to chunk these documents. We explored chunking at a paragraph and sentence level before finally settling on a 100 character chunk size. Paragraphs vary too much in size making it difficult to properly embed each paragraph, and sentences often did not contain enough context. using a 100 character chunk size we found that the chunks have adequate context, and being a consistent size allows for better embedding of the chunks, and thus improved search for relevant documents.

Once the documents are chunked they are embedded using HuggingFace or OpenAI embeddings, both utilized from the Langchain Python interface. HuggingFace embeddings are free, but OpenAI's embeddings are of a much higher dimension by default, enabling more refined search. Each chunk of the documents is then embedded and stored with some metadata in a Chroma database.

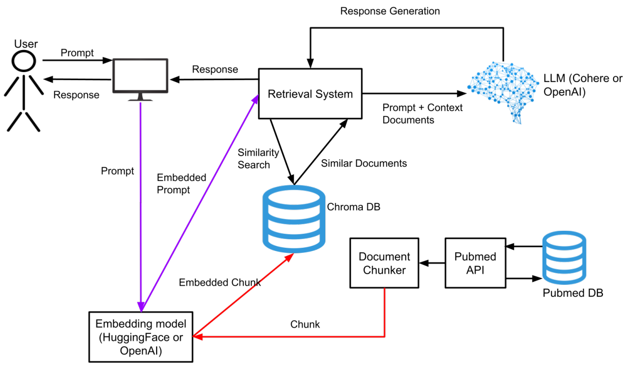

Fig. 1. RAG Pipeline

The chunks are stored in a Chroma DB JEEVAN WRITE THIS PART

Now that all of the data has been ingested, we can utilize these documents in our RAG flow. When a user sends a query, their query is embedded into the same vector space the Chroma-db is in. A similarity search is run in Chroma between the document chunks and the query to return the top 5 most relevant document chunks to the query. We take these 5 most similar chunks and send them as RAG documents to the llm model when we submit the user query. Both Cohere's prompting LLM and OpenAI's GPT-4 llm take the similar chunks as document parameters and use them to form a response to the query using not only their general knowledge, but also the specific fact based information in the document chunks. Cohere's prompting LLM is used from Cohere's python interface, and OpenAI's GPT-4 is used from the Langchain interface. The llm's process the query and RAG documents, creating a response that is informed on related and factual medical information from the RAG documents. The response is sent back to the user along with document metadata to provide a legitimate citation for the sources of medical information.

For a visual representation of this flow see Fig. 1. RAG Pipeline.

E. Full Stack

The interface for user interaction with the system was developed using a React frontend and a Flask backend. The X-Ray diagnosis architecture facilitates user input through three distinct steps:

1) Step 1: Fracture Identification: The user uploads an X-Ray image, which is analyzed by the CV model to detect the presence of a fracture. Depending on the outcome, the system either stops with a notification of no issues or proceeds after identifying the fracture type, allowing user corrections if necessary.

2) Step 2: Location Identification: This step involves the CV model pinpointing the fracture's location, visually indicated by a labeled box. Users can correct the identified location if they believe there's an error.

3) Step 3: Diagnosis and Treatment Suggestions: Finally, users receive a diagnosis summary and a suggested treatment plan, generated by the RAG pipeline. They have the option to request a diagnosis regeneration for additional considerations.

IV. Results

Our results demonstrate that our initial goal of producing a pipeline capable of diagnosing a fracture given an X-ray was successful. For Step 1 & 2 (identifying the presence of a fracture and classifying a fracture's location), the DenseNet and EfficientNet models demonstrated high-quality results: following training and hyperparameter tuning, the models were able to achieve an accuracy of 86.60% and 91.30% respectively on an unseen 'test' dataset.

V. Conclusion

Having introduced a diagnostic tool that integrates Computer Vision and RAG AI, this product significantly improves the accuracy and efficiency of X-ray image diagnosis in the medical field. Leveraging DenseNet and EfficientNet CNNs trained on extensive datasets, the Computer Vision models achieved high functional accuracies. The RAG element ensures that users receive the most accurate and scientifically valid information by retrieving relevant data from the PubMed papers database. Delivering quick, accurate, and up-to-date information through an electronic matter generates impactful outcomes for patient care, including the elimination of extended waiting times for doctor consultations regarding results and follow-up procedures.

VI. Future Work

Moving forward, we intend to begin refining this pipeline for real usage in the medical field. Primarily, this involves interviewing various clinicians and medical researchers to gain further insight into deployment for our product. Additionally, we intend to improve the accuracy of our product by refining our testing and iterative processes, as well as expanding our available suite of tools.

References

[Tor,] Torchvision models documentation. https://pytorch.org/vision/stable/models.html. Accessed: [Insert Access Date Here].

[Olc, 2017] (2017). Artificial intelligence for analyzing orthopedic trauma radiographs. *Acta Orthopaedica*, 88(6):581–586. PMID: 28681679.

[Abedeen et al., 2023] Abedeen, I., Rahman, M., Prottyasha, F., Ahmed, T., Chowdhury, T., and Shatabda, S. (2023). Fracatlas: A dataset for fracture classification, localization and segmentation of musculoskeletal radiographs. *Scientific Data*, 10.

[Guermazi et al., 2022] Guermazi, A., Tannoury, C., Kompel, A. J., Murakami, A. M., Ducarouge, A., Gillibert, A., Li, X., Tournier, A., Lahoud, Y., Jarraya, M., Lacave, E., Rahimi, H., Pourchot, A., Parisien, R. L., Merritt, A. C., Comeau, D., Regnard, N.-E., and Hayashi, D. (2022). Improving radiographic fracture recognition performance and efficiency using artificial intelligence. *Radiology*, 302(3):627–636. PMID: 34931859.

[Kim and MacKinnon, 2018] Kim, D. and MacKinnon, T. (2018). Artificial intelligence in fracture detection: transfer learning from deep convolutional neural networks. *Clinical Radiology*, 73(5):439–445.

[Kuo et al., 2022] Kuo, R. Y., Harrison, C., Curran, T.-A., Jones, B., Freethy, A., Cussons, D., Stewart, M., Collins, G. S., and Furniss, D. (2022). Artificial intelligence in fracture detection: A systematic review and meta-analysis. *Radiology*, 304(1):50–62. PMID: 35348381.

[Lewis et al., 2021] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., tau Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. (2021). Retrieval-augmented generation for knowledge-intensive nlp tasks.

[Little et al., 2017] Little, K., Reiser, I., Liu, L., Kinsey, T., Sánchez, A., Haas, K., Mallory, F., Froman, C., and Lu, Z. (2017). Unified database for rejected image analysis across multiple vendors in radiography. *J Am Coll Radiol*, 14(2):208–216. Epub 2016 Sep 20. PMID: 27663061.

[Oppenheimer et al., 2023] Oppenheimer, J., Lüken, S., Hamm, B., and Niehues, S. (2023). A prospective approach to integration of ai fracture detection software in radiographs into clinical workflow. *Life (Basel)*, 13(1):223. PMID: 36676172; PMCID: PMC9864518.

[Ren et al., 2015] Ren, S., He, K., Girshick, R. B., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *ArXiv*, abs/1506.01497.

[Sharma, 2023] Sharma, S. (2023). Artificial intelligence for fracture diagnosis in orthopedic x-rays: current developments and future potential. *SICOT J*, 9:21. Epub 2023 Jul 6. PMID: 37409882; PMCID: PMC10324466.

[Vancouver Coastal Health, 2024] Vancouver Coastal Health (2024). X-Ray. https://www.vch.ca/en/service/x-ray#short-description--6916. Online; accessed 29 January 2024.