

Artificial Intelligence (AI) in Medical Image-Pneumonia and Bone Fracture Detection

Shaikh Tojibul Islam¹, Bivas Dewan², Mohammad Miraj³, Md. Al-Mamun⁴, Swarnaly Hakim⁵

^{1,2,4,5} Biomedical Engineering, Bangladesh University of Health Sciences (BUHS), Dhaka, Bangladesh

³Biomedical Engineering, University of Malaya, Kuala Lumpur, Malaysia.

DOI: <https://doi.org/10.51583/IJLTEMAS.2026.150900050>

Received: 13 September 2026; Accepted: 18 September 2026; Published: 07 October 2026

ABSTRACT

Medical X-ray image interpretation requires considerable clinical expertise and may become challenging when the workload is high or access to experienced medical professionals is limited. This study presents a web-based medical image analysis application developed using the Django framework for the automated classification of pneumonia from chest X-ray images and bone fractures from skeletal X-ray images. The proposed system integrates deep learning-based image classification with a web application interface, allowing users to upload X-ray images and obtain model-generated predictions through a single platform. The application also incorporates image validation, prediction-result storage, and an AI-assisted explanation module using the Google Gemini API to present the prediction in a more understandable form. The system was developed to demonstrate the practical integration of deep learning models into an accessible medical imaging workflow rather than to replace professional clinical diagnosis. The application provides a structured workflow from image submission and preprocessing to model inference and presentation of the predicted result. The study also discusses the limitations associated with image quality, dataset diversity, classification coverage, and the absence of explicit lesion or fracture localization. The proposed system demonstrates how a web-based deep learning application can be used as a supportive tool for preliminary medical image assessment and as a platform for further research in AI-assisted medical imaging.

Keywords: Artificial Intelligence predictions, Medical Image, Convolutional Neural Network, Django, PostgreSQL, HTML5, CSS3, Bootstrap, Javascript, JSON API

INTRODUCTION

Medical imaging is an essential component of modern healthcare because it provides non-invasive information about the internal structures of the human body and supports the diagnosis and management of a wide range of diseases and injuries. Among the available imaging modalities, X-ray remains one of the most commonly used because of its relatively low cost, wide availability, and rapid image acquisition. Chest X-ray imaging is frequently used in the assessment of respiratory conditions such as pneumonia, while skeletal X-rays are commonly used to identify abnormalities including bone fractures. Despite the widespread use of X-ray imaging, accurate interpretation generally requires appropriate clinical knowledge and experience, and the increasing volume of medical images can place considerable demands on healthcare professionals.

The application of Artificial Intelligence (AI), particularly Machine Learning (ML) and Deep Learning (DL), has created new opportunities for computer-assisted medical image analysis. Deep learning models, including Convolutional Neural Networks (CNNs), can learn relevant visual patterns directly from image data and have therefore been extensively investigated for medical image classification. In the context of X-ray analysis, such approaches can assist in identifying patterns associated with specific diseases or abnormalities. However, the output of an AI model should be considered within the limitations of the training data, image quality, model design, and validation methodology. Therefore, AI-based image analysis is more appropriately considered a supportive technology rather than a replacement for professional clinical assessment.

In addition to the prediction capability of an AI model, the practical usability of a medical image analysis system is an important consideration. A standalone trained model may demonstrate classification capability, but its practical use requires an appropriate mechanism for image submission, preprocessing, inference, result presentation, and data management. A web-based application can provide an integrated environment through which users can interact with an AI model without directly handling the underlying implementation. Such an approach can also provide a structured workflow for image validation, prediction, result storage, and presentation. This study presents a Django-based web application for AI-assisted analysis of medical X-ray images. The system focuses on two application areas: pneumonia classification from chest X-ray images and bone-fracture classification from skeletal X-ray images. Users can upload X-ray images through a web interface, after which the images are processed and passed to the corresponding trained deep learning model for prediction. The application also incorporates image validation and prediction-result management to provide a structured user workflow.

To improve the interpretability of the model output for non-specialist users, the application integrates the Google Gemini API to generate human-readable explanations based on the prediction results. The explanation component is intended to improve the presentation and understanding of the system output; it is not used as an independent diagnostic engine and does not replace the prediction generated by the trained image-classification model. This distinction is important because generative AI systems may produce plausible explanations that should not be interpreted as clinical evidence without appropriate validation. The primary contribution of this work is the integration of deep learning-based X-ray classification with a web-based application workflow. Rather than focusing only on model development, the study demonstrates how trained image-classification models can be incorporated into an application that manages image submission, processing, prediction, explanation, and result presentation. The system is designed as a research and decision-support platform and is not intended to provide autonomous clinical diagnosis.

The study also recognizes several limitations of the current implementation. The system is limited to the supported X-ray categories and does not currently cover other imaging modalities such as Computed Tomography (CT), Magnetic Resonance Imaging (MRI), or ultrasound. Furthermore, the present classification approach does not provide explicit localization of abnormal regions through bounding boxes, segmentation masks, or heatmaps. Model performance may also be affected by image quality and by differences between the training data and images encountered in practical settings. These limitations provide directions for future development and further validation of AI-assisted medical imaging systems.

Problem Statement

X-ray imaging is widely used for the assessment of various medical conditions, including respiratory diseases and skeletal injuries. Although X-ray examinations are generally accessible and relatively cost-effective, the interpretation of medical images requires appropriate clinical expertise. The increasing volume of imaging examinations can place additional workload on healthcare professionals and may become challenging in settings where experienced specialists are not readily available. The interpretation of X-ray images can also be affected by variations in image quality, acquisition conditions, and the experience of the individual interpreting the image. These factors highlight the potential value of computer-assisted methods that can provide an initial classification of selected abnormalities. Deep learning-based image classification offers a possible approach for analyzing X-ray images by learning patterns from previously labeled image datasets.

However, developing an image-classification model alone does not provide a complete user-oriented solution. A practical system requires an interface for image submission, appropriate image validation and preprocessing, model inference, result presentation, and management of prediction information. In addition, users may require understandable information about the generated prediction rather than a classification label alone.

Therefore, this project addresses the need for an integrated web-based platform that combines deep learning-based X-ray classification with a practical application workflow. The proposed system focuses on pneumonia classification from chest X-ray images and bone-fracture classification from skeletal X-ray images. It also

provides AI-assisted textual explanations of prediction results through integration with the Google Gemini API. The system is intended to function as a supportive research and decision-support tool and is not designed to replace professional clinical diagnosis.

Objectives of the Project

The Major Objectives Of This Project Are As Follows:

Development of a User-Friendly Web Interface

To develop a web-based interface using the Django framework that allows users to upload medical X-ray images and interact with the image analysis system through a structured workflow.

Automated X-ray Image Classification

To integrate deep learning-based image classification models for the detection of pneumonia from chest X-ray images and bone fractures from skeletal X-ray images.

Image Validation and Preprocessing

To implement appropriate image validation and preprocessing procedures so that uploaded images can be prepared for model inference in a consistent manner.

Prediction Result Management

To provide a mechanism for presenting and managing the prediction results generated by the trained deep learning models.

AI-Assisted Explanation of Results

To integrate the Google Gemini API for generating human-readable explanations of model prediction results, while keeping the generated explanation separate from the underlying image-classification prediction.

Integration of AI Models into a Practical Application

To demonstrate the integration of deep learning-based medical image classification models into a complete web application rather than evaluating the models only as standalone machine learning components.

Support for Research and Decision-Making

To develop the system as a supportive platform for preliminary assessment, educational use, and further research in AI-assisted medical image analysis, without replacing professional clinical diagnosis.

Dataset and Data Preparation

The development of the proposed medical image analysis system required labeled X-ray image datasets for the two targeted classification tasks: pneumonia detection from chest X-ray images and bone-fracture detection from skeletal X-ray images. Publicly available datasets were used for the development and evaluation of the deep learning-based classification models.

For pneumonia classification, the Chest X-Ray Images (Pneumonia) dataset available through Kaggle was used. The dataset contains chest X-ray images categorized into pneumonia and normal classes. The images provide examples of different radiographic appearances and were used to develop the model for binary classification of chest X-ray images.

For bone-fracture classification, the MURA (Musculoskeletal Radiographs) dataset was considered as the source of skeletal radiographic images. MURA is a publicly available musculoskeletal radiograph dataset containing studies from different anatomical regions. The dataset was used as the basis for developing the fracture-classification component of the proposed system. Before model inference, the uploaded images undergo preprocessing so that their format and dimensions are compatible with the corresponding deep learning model. Image processing operations include image loading, resizing, normalization, and, where required by the model input configuration, conversion of the image representation. These operations help maintain a consistent input format between the images used during model development and the images submitted through the web application.

The datasets used in this study were obtained from publicly available sources and were not collected directly from patients through the developed application. Consequently, the system was designed around the characteristics and limitations of the selected datasets. Differences in image acquisition conditions, patient populations, image quality, and data distribution may affect the generalizability of the trained models to images obtained from other clinical environments. The exact number of images used for training, validation, and testing depends on the dataset subset and model-development configuration used in the implementation. Therefore, dataset counts and data-splitting details should be reported based on the actual experimental configuration rather than estimated values.

Image Preprocessing

Image preprocessing is an important stage of the proposed system because the X-ray images obtained from publicly available datasets or submitted through the web application may differ in dimensions, file formats, and image characteristics. The preprocessing workflow was designed to prepare the images in a consistent format before they were provided to the corresponding deep learning model. Initially, the uploaded image is checked to ensure that it is a valid image file and can be successfully opened by the application. Pillow and OpenCV are used for image loading and image-processing operations within the application. Invalid or unsupported files are rejected before model inference. The valid images are then resized according to the input requirements of the corresponding trained model. Resizing ensures that the image dimensions are compatible with the model input layer. Pixel values are also normalized according to the preprocessing requirements used by the model so that the input data remain within the expected numerical range.

For grayscale X-ray images, the image representation is handled according to the input configuration of the corresponding model. Where required, the processed image is converted or reshaped to match the expected number of input channels. The resulting image array is then converted into the appropriate numerical representation and supplied to the deep learning model for prediction. The same general preprocessing principles are applied to images submitted through the web application so that the inference pipeline remains consistent with the model-development workflow. The preprocessing stage therefore acts as an interface between the uploaded medical image and the trained classification model. It should be noted that preprocessing does not guarantee correct classification. Variations in image quality, acquisition equipment, anatomical positioning, contrast, and other characteristics may still affect model predictions. These factors are considered important limitations when applying models trained on public datasets to images from different sources.

Deep Learning Model and Classification

The proposed system uses deep learning-based image classification models to analyze X-ray images for the two targeted medical conditions: pneumonia in chest X-ray images and bone fractures in skeletal X-ray images. The models are integrated into the Django application so that an uploaded image can be processed and classified through the web interface. The overall classification workflow consists of image preprocessing, model input preparation, model inference, and prediction generation. After an uploaded image passes the validation and preprocessing stages, it is converted into the numerical representation required by the corresponding trained model. The processed image is then provided to the deep learning model, which analyzes the visual patterns present in the image and produces a classification output. For pneumonia

detection, the model performs binary classification of chest X-ray images into the supported classes, such as pneumonia and normal. Similarly, the fracture-detection model analyzes skeletal X-ray images and generates a classification indicating whether the image belongs to the supported fracture or non-fracture category. The corresponding model is selected according to the type of X-ray image being analyzed.

The models were developed using the Keras deep learning framework and were based on publicly available labeled X-ray datasets. The use of trained models allows the application to perform inference on new images without retraining the model for every prediction request. During application execution, the trained model is loaded and used to generate predictions from the preprocessed input image. The prediction generated by the classification model represents the output of the image-analysis component of the system. Where confidence information is available from the model output, it can also be presented to the user together with the predicted class. The prediction is subsequently stored and displayed through the web application.

It is important to distinguish the deep learning classification model from the Gemini-based explanation component. The classification model is responsible for analyzing the X-ray image and generating the prediction, whereas the Gemini API is used separately to generate a human-readable explanation based on the prediction and relevant application context. Gemini is therefore not treated as the primary X-ray image classification model in the proposed system. The performance and generalizability of the classification models depend on factors such as the quality and diversity of the training data, preprocessing procedures, model configuration, and differences between public datasets and real-world clinical images. Therefore, the predictions generated by the system should be considered supportive information and should not be interpreted as a substitute for professional clinical diagnosis.

Tools and Technologies

Front		End	Development
Technology	Purpose	Description	
HTML	Page Structure	HTML is used to create the basic structure of the web pages. It defines forms, buttons, input fields, image upload sections, and result display areas. It works as the skeleton of the front end.	
CSS	Styling and Design	CSS is used to improve the visual appearance of the application. It controls colors, fonts, layout spacing, button design, and overall look to make the interface clean and user-friendly.	
Bootstrap	Responsive Layout	Bootstrap is used to create responsive and mobile-friendly layouts. It provides pre-built components like forms, buttons, cards, and grid systems, ensuring consistent design across different devices.	
JavaScript	Interactivity	JavaScript is used to add dynamic behavior to the frontend. It handles form validation, user interactions, loading indicators, and dynamically displays results received from the backend.	

Backend Development

Technology	Purpose	Description
Python	Backend Programming	Python is the main programming language used to implement backend logic. It handles image processing workflows, API communication, data handling, and overall application logic.
Django	Web Framework	Django is used to manage backend operations such as handling HTTP

		requests, controlling application flow through views, managing URLs, and integrating external APIs like Gemini. It also ensures security and scalability.
PostgreSQL	Database System	PostgreSQL is used to store user information, uploaded image data, analysis results, and system logs. It provides reliable, secure, and efficient data storage for the application.

Image Processing and AI Libraries

Category	Libraries	Purpose
Image Handling	Pillow (PIL)	Handles image file operations in the Django application, including image loading, format conversion, and saving uploaded files.
Deep Learning Framework	Keras	Used for building, training, and loading deep learning models for pneumonia and bone fracture detection from X-ray images [1].

Explanation of the Result with AI

AI Tool	Purpose
Gemini API	The application uses the Gemini API to generate clear and human-readable explanations for the diagnostic results produced by the deep learning models. After analyzing an X-ray image, the model's prediction is sent to the Gemini API along with relevant context. The API then generates a textual explanation that helps users understand the predicted result in a simple and meaningful way. This approach improves the interpretability and transparency of the system. It also enhances user trust and supports educational and clinical understanding of AI-based medical diagnosis.

Database for Storing Data

In this application, PostgreSQL is used as the primary database management system for storing and managing all application data. PostgreSQL is an open-source, object-relational database system known for its reliability, data integrity, and high performance. The database stores various types of data including user information, uploaded medical images (X-ray, MRI, CT scan), prediction results, timestamps, and system logs. PostgreSQL efficiently handles large datasets and supports complex queries, which is essential for medical image-based applications. The database is integrated with the Django framework using Django's built-in ORM (Object Relational Mapping) system. This allows the application to interact with the database using Python code instead of writing raw SQL queries, making data operations more secure and easier to maintain. PostgreSQL also provides strong support for ACID properties, ensuring data consistency and accuracy. Due to its scalability, security features, and compatibility with Django, PostgreSQL is a suitable choice for this medical image analysis system.

Features of the Application

The application is mobile-responsive, ensuring that users can access all functionalities seamlessly on smartphones, tablets, and desktops. The responsive UI enhances usability and accessibility, allowing healthcare professionals and users to use the system conveniently from any device.

X-ray Image Upload

The system allows users to upload medical images such as chest and bone X-ray images through a web-based

interface. Uploaded images are validated for correct file format and size before being stored on the server. This ensures data security and prevents unauthorized or invalid files from entering the system.

AI Based Pneumonia and Bone Fracture Detection

The application uses trained deep learning models to analyze uploaded X-ray images for pneumonia and bone fracture classification. For chest X-ray images, the corresponding model classifies the image into the supported pneumonia or normal categories. Similarly, the fracture-detection model analyzes skeletal X-ray images and provides the corresponding classification result[5]. The trained models are integrated into the Django application so that predictions can be generated through the web interface. The use of trained models allows the application to perform image classification without requiring model training during each prediction request.

Diagnostic Result Visualization

After image analysis, the system displays diagnostic results in a clear and user-friendly format. The output includes the predicted condition (such as pneumonia detected or fracture detected) along with confidence information, helping users understand the diagnostic outcome easily.

Explanation of Results Using Gemini API

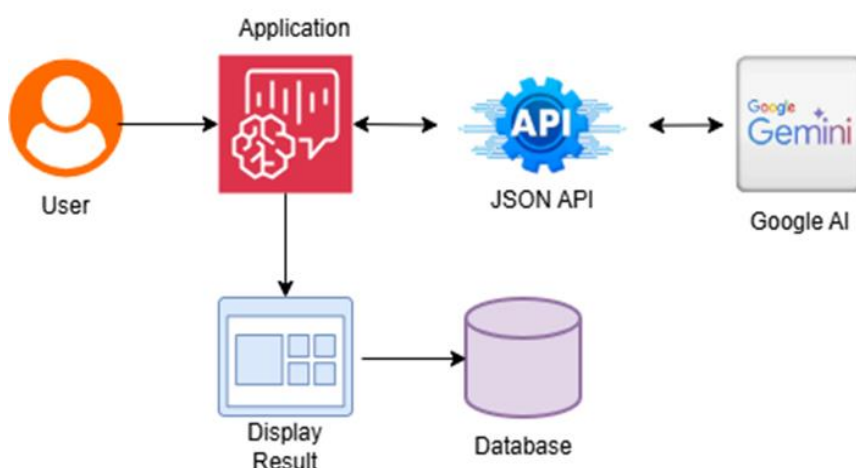
The application uses the Gemini API to generate clear and human-readable explanations of the prediction results produced by the deep learning models. After the X-ray image is processed by the corresponding deep learning model, the prediction result and relevant context are sent to the Gemini API. Gemini then generates a textual explanation to help users better understand the predicted result[3]. The Gemini API is therefore used as an explanation and interpretation component and does not perform the primary X-ray image classification. This separation allows the deep learning model to remain responsible for image classification while Gemini provides a more understandable description of the generated result.

Display Archive Data from Previous History

The system maintains a comprehensive history of previously uploaded images and their diagnostic results. Users can access this archived data at any time, enabling easy review of past records, comparison of results over time, and better tracking of patient progress.

System Architecture User

High Level Design of the application



The entry point of the system. This represents the medical professional or technician. The user interacts with the application frontend to upload an X-ray image and input basic patient data.

Application (Django Framework)

This is the "brain" of application. Django handles the logic and routing.

- It receives the uploaded image, validates it, and prepares a request for the AI.
- It processes the uploaded image and displays the result

JSON API

The JSON API is used for structured communication between the Django application and the Google Gemini API. The application sends the prediction result and relevant contextual information in a structured request, and the Gemini API returns a textual explanation of the result. The JSON format allows the information to be exchanged in a structured and machine-readable form.

Google AI (Gemini)

Google Gemini acts as an AI-assisted explanation component of the system. It receives the prediction result and relevant context from the application and generates a human-readable explanation. Gemini does not perform the primary X-ray image classification; the classification is performed by the trained deep learning models integrated into the application.

Display Result

Once Django receives the response from Gemini, it renders a new view for the user. The user sees the original X-ray alongside the AI's findings (e.g., "Positive for Fracture").

Database

This is the data storage layer of the application. Every request and result is stored here. This is crucial for medical applications for patient history allowing professionals to look back at previous X-ray

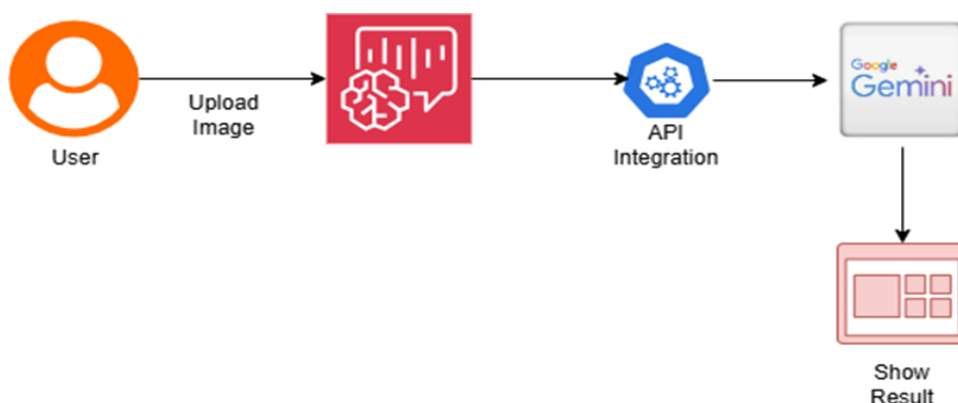
Development Process (Coding)

Code Repository: The python codes for this application can be found in below link of github:

<https://github.com/tojib/Pneumonia-Bone-Fracture-Detection-Using-Deep-Learning>

Gemini API Integration

High Level Diagram of API Integration

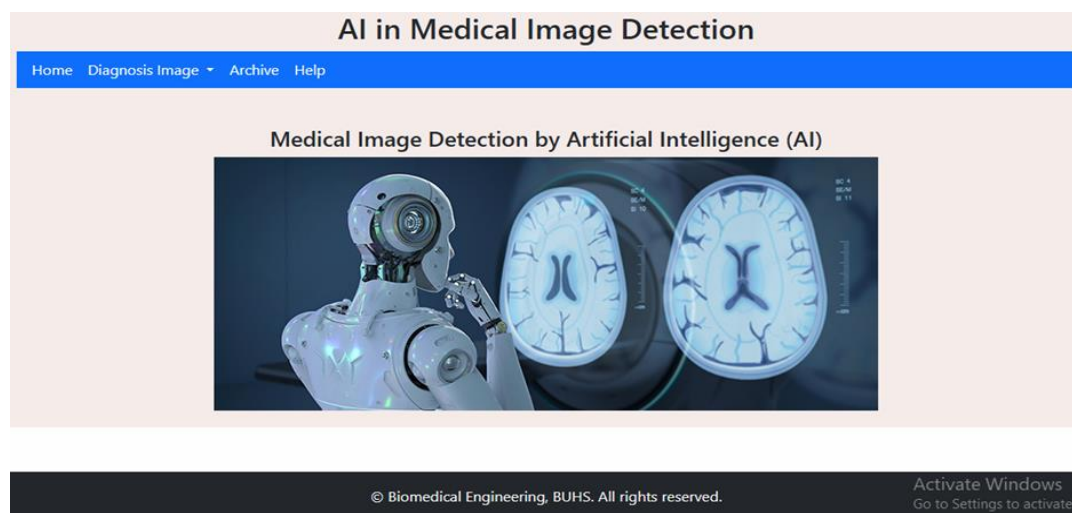


Required Python Libraries and Purpose

Library	Purpose
opencv-python	Preprocessing images (resize, grayscale)
pillow	Django image handling
tensorflow	Loading and using pretrained deep learning models
numpy	Array and matrix manipulation for model input
django	Backend logic and template management
psycopg2	PostgreSQL driver for Django

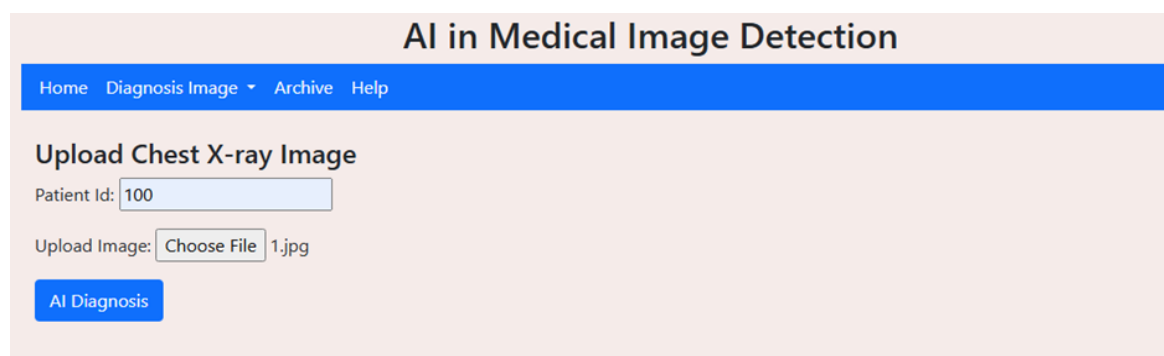
User Interface of the Application

Home Page



Pneumonia Detection

Uploading Chest X-ray Image in Application



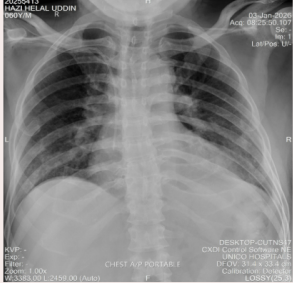
Showing Result:

AI in Medical Image Detection

Home Diagnosis Image ▾ Archive Team Members

Patient Id: 100

Uploaded Image



AI Result:

Pneumonia Detected!

Explanation:

In simple terms, "Pneumonia Detected!" means that **the image analysis (like an X-ray or CT scan) found signs consistent with an infection in your lungs.** Here's what that generally means: **What it is:** Pneumonia is an infection that causes inflammation in the air sacs of one or both lungs. These air sacs can then fill with fluid or pus. **What it can cause:** This can make it difficult to breathe, lead to coughing (often with phlegm), fever, chills, and chest pain. **What was seen:** The image analysis likely picked up on these areas of inflammation or fluid in your lungs. **Next Steps:** This result typically means a doctor will need to evaluate you further and discuss treatment options, which often involve antibiotics (if bacterial) or other supportive care.

Upload Another Image

Bone Fracture Detection

Uploading Bone Xray Image:

AI in Medical Image Detection

Home Diagnosis Image ▾ Archive Help

Upload Bone X-ray Image

Patient Id:

Upload Image: images.jfif

Showing Result:

AI in Medical Image Detection

Home Diagnosis Image ▾ Archive Team Members

Patient Id: 200

Uploaded Image



AI Result:

Fractured

Explanation:

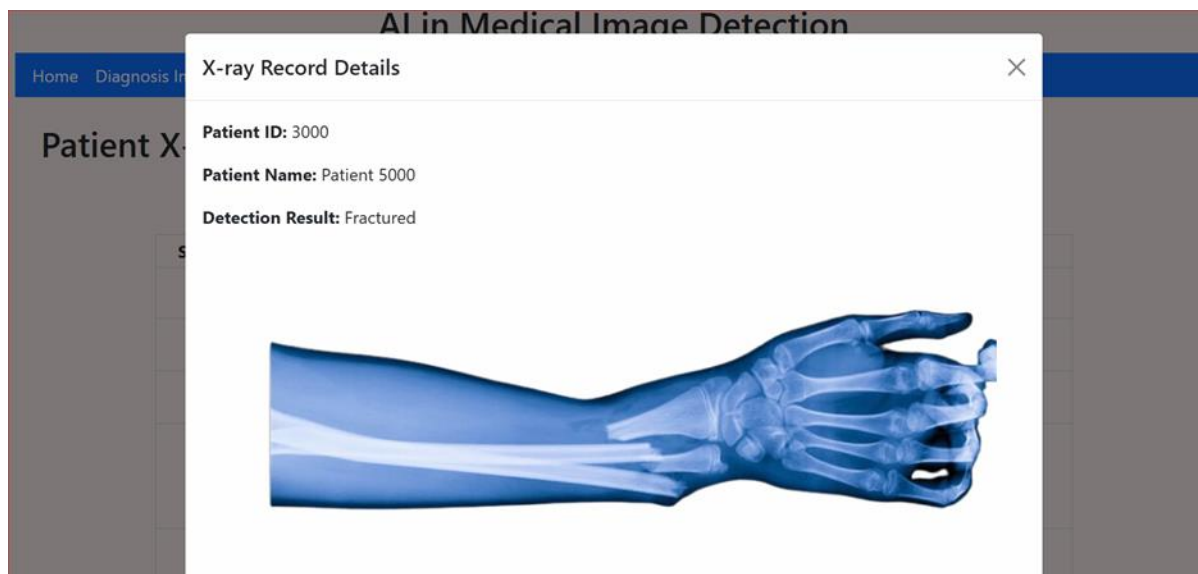
When an image analysis result says "Fractured," it means there's a **broken bone**. It's not just a crack; it's an actual break in the bone. Here's what should be done in simple terms: 1. **Don't Move It:** The absolute most important thing is to keep the injured part as still as possible. Moving it can cause more pain and potentially more damage. 2. **Don't Try to Fix It Yourself:** Do not try to push the bone back into place or adjust it. This is a job for medical professionals. 3. **Get Medical Help Immediately:** **Go to an Emergency Room (ER):** This is the best course of action for a suspected broken bone. **Call an ambulance:** If the person is in severe pain, can't move, or if the injury looks very serious (like a bone sticking out), call emergency services. 4. **Support the Injury (Gently, if needed):** If you have to move the person or wait for help, gently support the injured area to prevent movement. You can use a pillow, rolled-up towel, or even your hands to keep it stable, but don't apply pressure or try to straighten it. 5. **What Doctors Will Do:** At the hospital, doctors will likely take X-rays to see exactly where and how the bone is broken. They will then **"set"** the bone (put it back in the correct position if needed) and usually put it in a cast, splint, or sometimes surgery is required to help it heal properly. **In simple terms:** If an image shows something is "fractured," it means there's a broken bone. The most important thing to do is **get medical help right away** and **don't try to move or fix it yourself.**

Upload Another Image

Show Archive Report

AI in Medical Image Detection					
Home Diagnosis Image ▾ Archive Team Members					
Patient X-ray Records					
Search by Patient ID				Search	
SL No	Patient ID	Patient Name	Report	X-ray Image	Action
1	3000	Patient 5000	Bone		View Edit Delete
2	500	Patient 500	Chest		View Edit Delete
3	500	Patient 500	Chest		View Edit Delete
4	100	Patient 0	Bone		View Edit Delete

Detailed view of a report



Edit a report for correction

AI in Medical Image Detection	
Home Diagnosis Image ▾ Archive Team Members	
Edit X-ray Record	
Patient ID: 3000	Current Image Preview: 
Patient Name: Patient 5000	
Upload Image: Choose File No file chosen	
Update Cancel	

Challenges Faced

During the development and deployment of our Django-based medical image analysis application, we encountered several challenges that required careful consideration and iterative problem-solving. These challenges are outlined below.

Ensuring Image Quality for Better Prediction

Medical image analysis is highly sensitive to image quality. Low-resolution or blurred images can significantly reduce the accuracy of AI-based predictions for conditions such as pneumonia or bone fractures.

- **Problem:** Images collected from various sources (X-ray machines, hospital databases, or user uploads) often had inconsistent resolution, contrast, and noise levels.
- **Impact:** Poor-quality images led to misclassification or uncertain predictions by the model, affecting the reliability of diagnostic results.
- **Approach:** We implemented preprocessing techniques such as resizing, normalization, and contrast enhancement to standardize images before feeding them into the AI model. Additionally, we emphasized the importance of high-quality image capture in user guidelines.

Handling Large Model Loading in Production

The AI models used for image analysis, especially deep learning models, were large in size and required significant memory and computational resources.

- **Problem:** Loading these models on a production server caused slow startup times and occasionally led to memory errors during concurrent requests.
- **Impact:** Users experienced delays in prediction, and server stability was sometimes affected.
- **Approach:** To mitigate this, we explored model optimization techniques such as model quantization and caching the model in memory to avoid repeated loading. We also ensured that the server had sufficient RAM and GPU resources when available.

Text Generation Consistency from Gemini API

The application integrates with the Gemini API to generate descriptive explanations for AI-predicted results. Ensuring consistent and accurate text output was challenging.

- **Problem:** The API sometimes produced responses that were incomplete, redundant, or inconsistent in phrasing, which could confuse end users.
- **Impact:** Inaccurate or unclear explanations could reduce user confidence in the system, especially in a medical context.
- **Approach:** We implemented post-processing of API responses to refine and standardize the text, including removing duplicates, correcting terminology, and ensuring clarity. Furthermore, we maintained structured templates for the explanation to guide the API output.

Making UI Responsive for All Devices

Ensuring that the application's user interface worked seamlessly across various devices, screen sizes, and resolutions was another critical challenge.

- **Problem:** Users access the application from desktops, laptops, tablets, and smartphones. Without a responsive design, some interface elements could overlap, break layout, or become unreadable.
- **Impact:** Poor UI responsiveness could lead to a frustrating user experience, reducing the usability of the platform.

- **Approach:** We adopted modern front-end frameworks (HTML5, CSS3, Bootstrap) and tested the interface extensively on different screen sizes. Media queries and flexible layout designs were applied to ensure the application remained user-friendly and visually consistent across devices

Security

Consideration

Security is a critical aspect of any medical image analysis application, as it deals with sensitive patient data and AI model predictions. During development, we implemented several measures to ensure the integrity, confidentiality, and reliability of the system. The main security considerations are described below:

Image Upload Validation

Allowing users to upload medical images introduces the risk of malicious files or invalid formats being submitted.

- **Problem:** Users could inadvertently or intentionally upload unsupported file types, corrupted images, or files containing harmful code.
- **Impact:** Invalid or malicious files could cause application crashes, compromise server security, or lead to incorrect AI predictions.

Approach:

- **File type restriction:** Only allowed image formats (JPEG, PNG, DICOM where applicable) were accepted.
- **File size limitation:** Maximum upload size was enforced to prevent denial-of-service (DoS) attacks.
- **Content validation:** Uploaded files were programmatically checked to confirm they were valid image files and could be processed by the model.
- **Sanitization:** File names were sanitized to avoid directory traversal or script injection attacks.

Error Handling for Model Prediction Failures

AI model predictions can sometimes fail due to unexpected input, server overload, or corrupted data. Proper error handling is essential to prevent application crashes and maintain user trust.

- **Problem:** Without robust error handling, prediction failures could expose sensitive information, crash the projection, or display incomplete results to users.
- **Impact:** Security and reliability of the platform would be compromised, reducing confidence in the application's use for medical purposes.

Approach:

- **Try-except blocks:** All prediction calls were wrapped in error handling routines to catch exceptions gracefully.
- **Logging:** Prediction failures were logged securely for troubleshooting without exposing sensitive patient data.
- **User feedback:** Users were shown clear, non-technical error messages and instructions for re-uploading images if necessary.
- **Fallback mechanisms:** In case of failure, the system prevents corrupted or incomplete results from being saved to the database

These security measures collectively ensured that the application could safely handle medical images and AI predictions while minimizing potential threats and maintaining data integrity

LIMITATIONS

Despite the effectiveness of the proposed AI-based medical image analysis system, the project has several limitations that should be acknowledged.

Limited Image Diagnosis Coverage

The current version of the application is designed to analyze only chest X-ray and bone X-ray images. It can detect pneumonia from chest X-rays and identify the presence of bone fractures from skeletal X-rays. The system does not support other medical images such as MRI, CT scan, or ultrasound. Additionally, it is limited to these specific conditions and cannot be used to detect other diseases or abnormalities.

Dependency on Image Quality

The performance of the AI models heavily depends on the quality of the input images. High-resolution and properly scanned X-ray images are required to obtain reliable predictions. In rural or resource-limited areas, access to advanced digital X-ray machines or high-quality scanning devices may be limited. As a result, low quality, blurred, or noisy images may lead to incorrect or unreliable diagnostic results.

Lack of Localization Capabilities

The system can classify whether pneumonia or a bone fracture is present in an uploaded X-ray image; however, it does not identify or highlight the exact location or affected region of the disease. The application provides only a classification result and does not include advanced features such as bounding boxes, heatmaps, or region-based visualization to indicate the precise area of infection or fracture.

These limitations indicate potential areas for future improvement and provide direction for enhancing the system's accuracy, usability, and clinical relevance.

Future Plan and Enhancement

As the Django-based medical image analysis application continues to evolve, several enhancements and expansions are planned to improve functionality, scalability, and user experience. The key future plans are outlined below:

Add More Disease Classifications

Currently, the application focuses on detecting pneumonia and bone fractures from X-ray images.

- Extend the AI model to classify a broader range of medical conditions using additional imaging modalities such as MRI, Ultrasonogram, and CT Scan.
- This expansion will make the platform more comprehensive for diagnostic purposes, allowing healthcare professionals to analyze multiple types of medical images in one system.
- New datasets will be collected and preprocessed for these modalities, and separate or multi-task deep learning models will be trained to ensure accurate classification across different diseases.

Create a Patient Login Portal and Personal Dashboard

Currently, the system does not have personalized access for patients or doctors.

- Introduce user authentication with secure login for patients and medical professionals, along with a personal dashboard to view previous scans, AI predictions, and reports.
- Personalized access enhances data privacy, enables tracking of patient history, and improves overall usability of the application.
- Implement Django's authentication framework with role-based access control and encrypted storage for sensitive patient information.

Collect Feedback for AI Accuracy Improvement

AI model performance may vary depending on the quality and diversity of input data.

- Implement mechanisms for users (doctors or radiologists) to provide feedback on AI predictions, including corrections or confirmations.
- Feedback will enable continuous learning and fine-tuning of models, improving accuracy and reliability over time.
- Develop a feedback module integrated with the dashboard, store feedback securely, and use it to periodically retrain and update AI models.

Integration with existing system like EHR

The application is designed to integrate with existing healthcare information systems, such as Electronic Health Records (EHR). This integration allows automatic transfer of diagnostic results, uploaded images, and AI-generated explanations directly into patients' digital medical records. By connecting with EHR systems, clinicians can access a centralized repository of patient data, ensuring continuity of care and reducing manual data entry errors. The integration also enables secure data sharing across departments and facilities while maintaining compliance with healthcare data privacy standards. Overall, this feature enhances workflow efficiency, supports informed clinical decision-making, and provides a more comprehensive view of a patient's x-ray reports.

Ethical Considerations and Disclaimer

The development and use of artificial intelligence (AI) in medical image analysis raise important ethical concerns related to patient safety, data privacy, and responsible decision-making. This project has been designed with careful consideration of these ethical aspects to ensure that the system is used in a safe and supportive manner.

Firstly, The application is intended solely as a decision support and educational tool and is not designed to replace professional medical diagnosis[1]. The AI generated results are not meant to replace professional medical judgment, diagnosis, or treatment. Final clinical decisions should always be made by qualified medical professionals based on comprehensive clinical evaluation. The system provides preliminary predictions to assist users, not to act as an independent diagnostic authority.

Secondly, patient data privacy and confidentiality are important ethical responsibilities. The application stores medical images and diagnostic results in a secure database and restricts access to authorized users only. No patient-identifiable information is intentionally exposed, and all stored data is handled in a controlled manner to minimize the risk of misuse or unauthorized access.

Thirdly, the system acknowledges the limitations of AI-based predictions. Since the accuracy of results depends on image quality, dataset diversity, and model performance, there is a possibility of false positives or false negatives. Users are advised not to rely solely on the system's output and to use the results as supportive information rather than final conclusions.

Additionally, the use of the Google Gemini API for generating explanatory text is limited to improving interpretability and user understanding. The generative AI component does not perform medical diagnosis; instead, it provides simplified explanations based on the model's prediction. Efforts have been made to ensure that the generated content remains informative, non misleading, and easy to understand.

Finally, this project follows ethical principles of transparency, responsibility, and user safety. By clearly defining its intended purpose and limitations, the system aims to support healthcare professionals while promoting responsible and ethical use of AI in medical imaging.

CONCLUSION

This Django-based medical image analysis application demonstrates the practical integration of artificial intelligence (AI), machine learning, and deep learning techniques into a web-based medical imaging workflow. By analyzing supported X-ray images, the system provides a supportive platform for preliminary image assessment through trained deep learning models.

The integration of model-generated predictions with AI-assisted textual explanations provides users with a more understandable presentation of the analysis results. Features such as image validation, result management, and responsive web design further support the practical use of the application.

The current system is limited to the supported chest X-ray and skeletal X-ray classification tasks and does not replace professional clinical diagnosis. Future development may include additional imaging modalities, expanded disease and abnormality classification, patient dashboards, electronic health record integration, and further model validation using diverse clinical datasets. Overall, the project demonstrates the potential of web-based AI-assisted medical image analysis as a supportive platform for healthcare-related research and future development.

Resources

- TensorFlow Documentation- <https://www.tensorflow.org>
- Django Documentation- <https://docs.djangoproject.com>
- Gemini API- <https://ai.google.dev>
- OpenCV Docs- <https://docs.opencv.org>
- Kaggle Pneumonia Dataset- <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>
- MURA Dataset- <https://stanfordmlgroup.github.io/competitions/mura/>

REFERENCES

1. Abdel-Basset M, Chang V, Mohamed R. "A novel equilibrium optimization algorithm for multi-thresholding image segmentation problems." 2021.
2. "Analysis of Biomedical Images Through Deep Learning Model for the Identification of Diseases." 2025.
3. "Automated Analysis of Biomedical Images Using Convolutional Neural Networks and Deep Learning." 2025.
4. J. Brown et al., "Automated segmentation of cardiac structures in CT angiography: Leveraging deep learning for precise diagnosis," 2023.
5. G. Calzada-Jasso et al., "Generative Fabrication of Medical Images for Machine Learning Training," 2025.
6. G. Carneiro, J. Nascimento, and A. Bradley, "Automated analysis of unregistered multi-view mammograms with deep learning," 2017.
7. Chen A, Zhu L, Zang H, Ding Z, Zhan S. "Computer-aided diagnosis and decision-making system for medical data analysis: a case study on prostate MR images." 2019.
8. P. Corcoran et al., "Chest X-Ray Visual Saliency Modeling: Eye-Tracking Dataset and Saliency Prediction Model," 2025.
9. Das A, Rad P. "Opportunities and challenges in explainable artificial intelligence (XAI): A survey." 2020.
10. Gadgil SU, Endo M, Wen E, Ng AY, Rajpurkar P. "Proceedings of the Fourth Conference on Medical Imaging with Deep Learning." 2021.
11. G. Garcia et al., "Deep learning-based segmentation of liver lesions in ultrasound images," 2023.
12. Ghnemat R, Alodibat S, Abu Al-Haija Q. "Explainable Artificial Intelligence (XAI) for Deep Learning Based Medical Imaging Classification." 2023.
13. Guo K, Ren S, Bhuiyan MZA, Li T, Liu D, Liang Z, Chen X. "MDMaaS: Medical-Assisted Diagnosis Model as a Service with artificial intelligence and trust." 2020.
14. Jafari M, Auer D, Karimi D. "DRU-Net: An efficient deep convolutional neural network for medical image segmentation." 2020.

15. Janik A., Dodd J., Ifrim G., Sankaran K., Curran K. "Interpretability of a deep learning model in the application of cardiac mri segmentation with an acdc challenge dataset." 2021.
16. Kazi A, Farghadani S, Aganj I, Navab N. "IA-GCN: Interpretable attention based graph convolutional network for disease prediction." 2024.
17. Karimi D, Dou H, Warfield SK, Gholipour A. "Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis." 2020.
18. Kayalibay B, Jensen G, van der Smagt P. "CNN-based segmentation of medical imaging data." 2017.
19. Kebaili et al., "Deep learning approaches for data augmentation in medical imaging," 2023.
20. Lekadir K, Osuala R, Gallin C, Lazrak N, Kushibar K, Tsakou G, et al. "FUTURE-AI: Guiding principles and consensus recommendations for trustworthy artificial intelligence in medical imaging." 2023.
21. P. Li, Y. Pei, J. Li, and H. Xie, "Medical Image Recognition Using a Novel Neural Network Construction Controlled by a Kernel Method Encoder," 2024.
22. Liu Q et al., "Imaging-Process-Informed Generative Strategy for Microwave Medical Image Processing," 2025.
23. Liu X, Song L, Liu S, Zhang Y. "A review of deep-learning-based medical image segmentation methods." 2021.
24. G. Litjens et al., "A survey on deep learning in medical image analysis," 2017.
25. M. A. Mazurowski et al., "Deep learning in radiology: State of the art," 2019.
26. Meng Y, Zhang H, Wang Z, Chen M, Hu P, Du Y, et al. "BI-GCN: Boundary-aware input-dependent graph convolution network for biomedical image segmentation." 2021.
27. Miao J, Chen C, Liu F, Wei H, Heng PA. "CauSSL: Causality-inspired semi-supervised learning for medical image segmentation." 2023.
28. E. Najdenovska et al., "Automated Identification of Eye Motion in Raw MRI Data Using Machine Learning," 2025.
29. R. Rajendran and S. G. Shiva Shankar, "An overview of X-ray imaging in the medical field," 2012.
30. Ruan J, Xiang S. "VM-UNet: Vision Mamba UNet for Medical Image Segmentation." 2024.
31. Singh D, Somani A, Horsch A, Prasad DK. "Counterfactual explainable gastrointestinal and colonoscopy image segmentation." 2022.
32. Smith and B. Johnson, "Revolutionizing X-ray imaging: CNN-based lung tissue segmentation in chest X-rays," 2020.
33. Stoyanov D, Taylor Z, Kia SM, Oguz I, Reyes M, Martel A, Maier-Hein L, Marquand AF, Duchesnay E, Löfstedt T, Landman B, Cardoso MJ, Silva CA, Pereira S, Meier R. "Understanding and interpreting machine learning in medical image computing and applications." 2018.
34. Z. Teng, L. Li, Z. Xin, D. Xiang, J. Huang, H. Zhou, F. Shi, W. Zhu, J. Cai, T. Peng, and X. Chen, "A literature review of artificial intelligence (AI) for medical image segmentation: From AI and explainable AI to trustworthy AI," 2024.
35. Tragakis A, Kaul C, Murray-Smith R, Husmeier D. "The fully convolutional transformer for medical image segmentation." 2023.
36. Williams D, Brown L. "Multimodal integration of clinical data, demographics, and imaging for enhanced disease classification in X-ray images." 2021.
37. Yadav, N. L., Singh, S., Kumar, R., & Singh, S. "Medical image analysis for detection, treatment and planning of disease using artificial intelligence approaches." 2024.
38. W. Zhang et al., "Annular Prior Prompt Learning for Medical Images Segmentation," 2026.
39. M. A. Mazurowski et al., "Deep learning in radiology: State of the art," 2019.