# AI-Driven Medical Image Analysis Application

## Abstract

This project introduces a lightweight, AI-powered desktop application tailored for medical image analysis, with a primary focus on chest X-ray interpretation. Leveraging Google's Gemini API, the application performs advanced multimodal analysis remotely, eliminating the need for local ML infrastructure. The solution is designed for healthcare professionals and researchers seeking quick, AI-assisted insights from radiological images without technical overhead.

## Overview

The application offers a user-friendly interface built with Python’s native Tkinter framework. By offloading complex analysis to Gemini’s cloud-based model ('gemini-2.0-flash-thinking-exp-01-21'), it provides efficient image processing and diagnostic support while maintaining local system simplicity. It acts as a bridge between high-performance AI and non-specialized computing environments.

## Technical Implementation

The system employs a modular client-server design. The local client handles GUI operations, image loading, and result display. Images are encoded in base64 and transmitted via secure requests to Gemini’s API. The server responds with structured, AI-generated insights. Code modularity ensures easy maintenance and future expansion.

## User Interface

The application features a dual-tab layout: one tab for image analysis, another for configuration. Users can preview X-rays, initiate analysis, and view structured output with explanations. Images are resized dynamically for optimal visualization. Async operations maintain responsiveness, accompanied by visual progress indicators during remote analysis.

## Medical Functionality

The system supports clinically relevant tasks including:

1. Binary classification (normal vs. abnormal)

2. Detection of potential pathologies

3. Radiological feature highlighting

4. AI confidence reporting

Structured prompts ensure consistent, focused outputs from the Gemini model.

## Accessibility Features

A built-in image directory system allows users to test with sample data. This addresses common bottlenecks in accessing quality test images for medical AI applications, streamlining validation and experimentation workflows.

## Technical Requirements

Minimal dependencies include Pillow (image processing) and Requests (API communication). The GUI is handled by built-in Tkinter. This ensures maximum compatibility, no complex installations, and efficient performance on standard hardware—making it ideal for resource-constrained settings.

## Limitations and Considerations

Designed for educational and research use, not clinical diagnostics. The application emphasizes responsible AI by clearly indicating confidence levels and avoiding clinical decision-making. It complements professional analysis without replacing human judgment.

## Future Directions

The modular architecture supports extensions for:

- Additional imaging modalities

- Integration with EHR systems

- Support for new AI models

This ensures adaptability and long-term utility in evolving medical AI landscapes.