BRAUDE

College of Engineering, karmiel

Software Engineering Department
Braude College

Capstone project Phase B

Automated Medical Diagnosis: Enhancing Chest X-ray Analysis with Convolutional Neural Networks

Project Code: 24-1-D-8

Supervisor: Dr. Natali Levi

Noam Ofir

Oded Morag

GitHub Repo:

https://github.com/GoodMoodMan/xray_classifier

1. General Description

The project "Automated Medical Diagnosis: Enhancing Chest X-ray Analysis with Convolutional Neural Networks" aimed to develop an AI-powered system for automated chest X-ray analysis to improve efficiency and diagnostic accuracy in detecting pulmonary conditions (PC). This ambitious undertaking sought to address the growing need for efficient and accurate diagnostic tools in radiology, particularly in resource-constrained healthcare settings.

The system leverages deep learning techniques, specifically Convolutional Neural Networks (CNNs), to analyze chest X-ray images and identify various PCs. The architecture explored multiple CNN models, including ResNet50, DenseNet121, EfficientNet B0, and VGG16, each chosen for their proven performance in image classification tasks and their potential adaptability to medical imaging challenges.

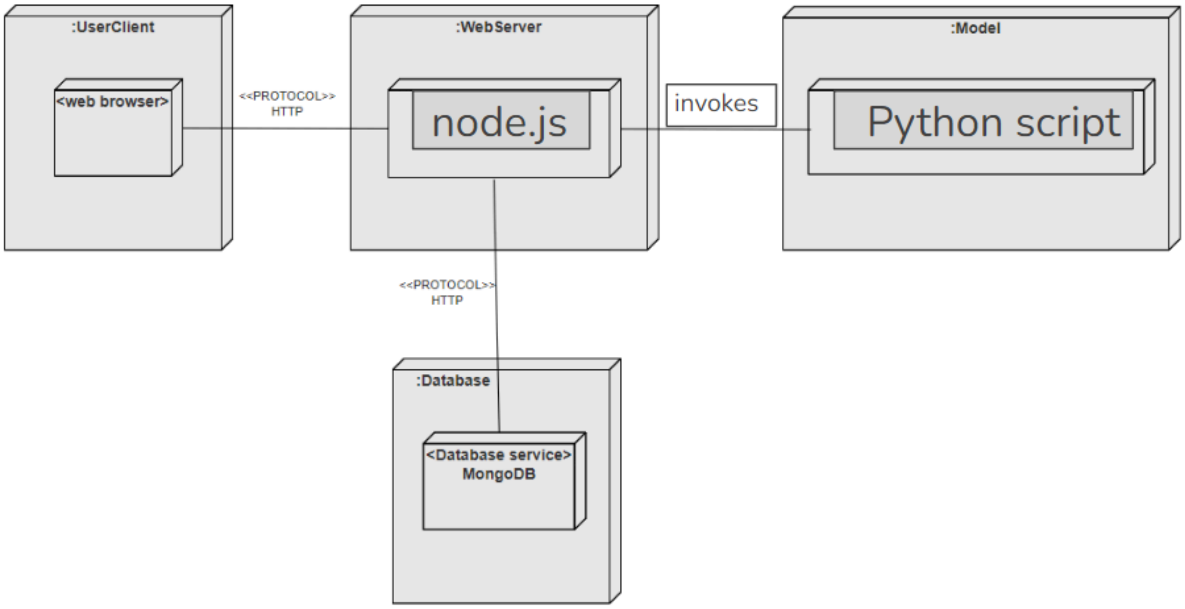
A key feature of the system is its user-friendly web interface, designed to seamlessly integrate into existing clinical workflows. This interface allows healthcare professionals to upload chest X-ray images for real-time analysis, potentially reducing the time and effort required for manual interpretation. The target audience includes radiologists, physicians, and other healthcare professionals involved in diagnosing pulmonary conditions, with a particular focus on supporting healthcare providers in areas with limited access to specialist radiologists.

The project also aimed to contribute to the broader field of AI in healthcare by exploring the limitations and potential of current deep learning models in medical image analysis. By tackling the complex task of distinguishing between visually similar clinical findings, the project sought to push the boundaries of what's possible with current CNN architectures in medical diagnostics.

1.1 Deployment Diagram

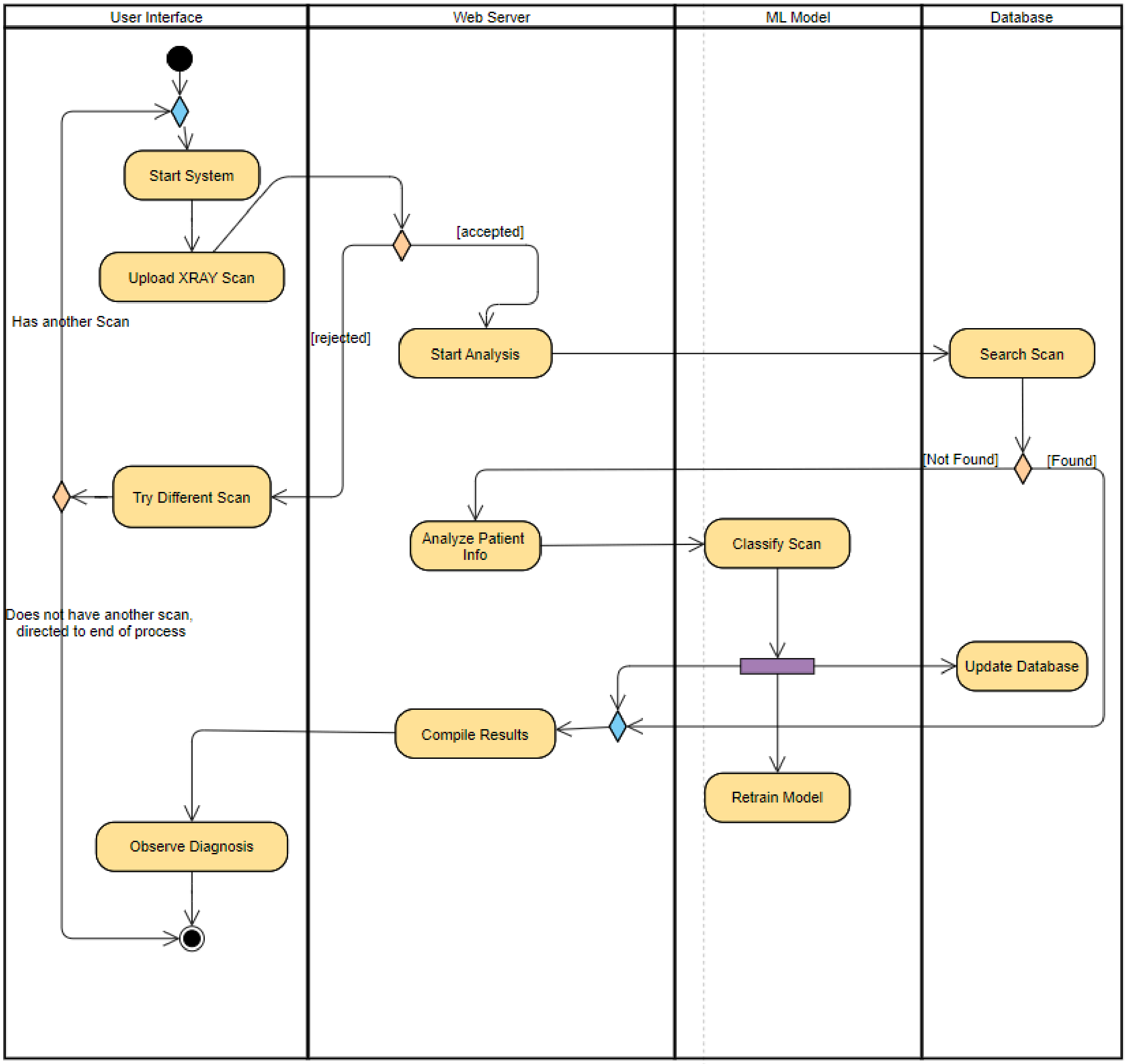
We deployed our application using a MERN (MongoDB, Express.js, React, Node.js) stack, which provides a robust and scalable architecture for web applications. Here's a brief overview of our deployment structure:

1. Frontend: Developed using React, providing a responsive and interactive user interface.
2. Backend Server: Implemented with Node.js and Express.js, handling HTTP requests, business logic, and database interactions.
3. Database: MongoDB used for storing necessary data.
4. Machine Learning Integration:
   - We integrated a Python runtime environment within the Node.js server.
   - This allows us to execute our Python-based machine learning script directly from the Node.js environment.
   - The Python script processes the uploaded X-ray images and performs the classification.
   - Results are then passed back to the Node.js server for further handling and client response.
5. API Layer: RESTful APIs facilitate communication between the frontend and backend, including image upload and retrieval of classification results.



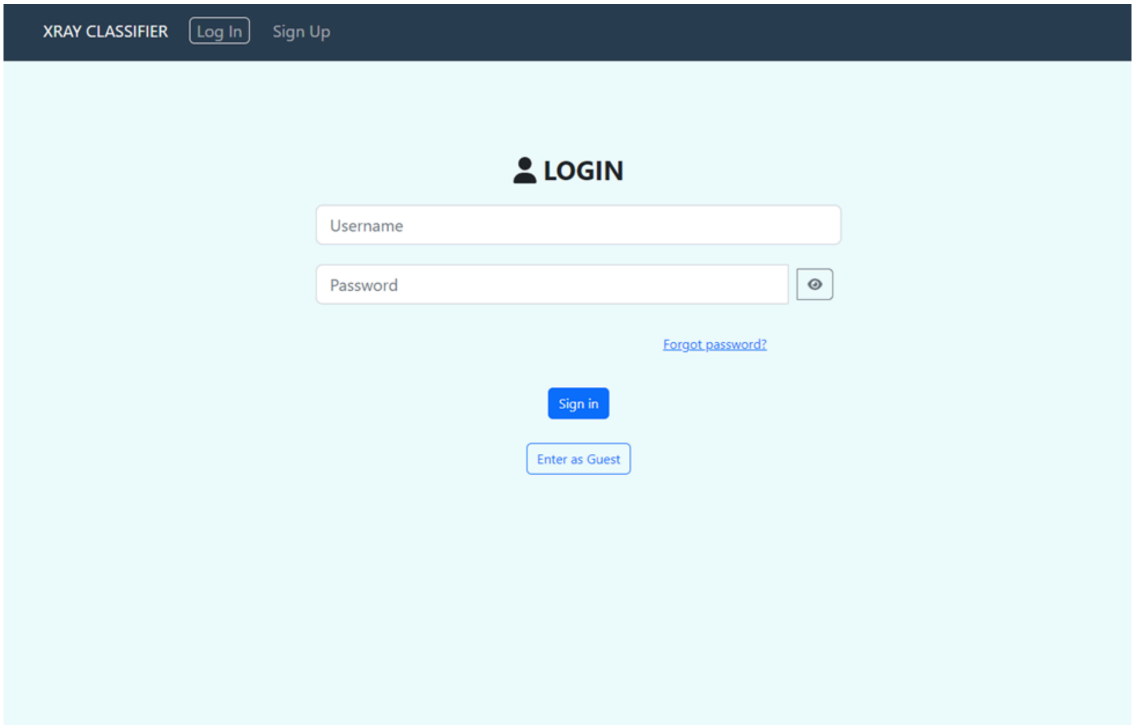
1.2 Activity Diagram

The activity diagram provides a clear overview of the automated chest X-ray analysis system's workflow. It illustrates the sequential steps from uploading a scan to observing the diagnosis, highlighting the interaction between the user interface, web server, machine learning model, and database components. The diagram effectively communicates the key activities and decision points involved in the process.



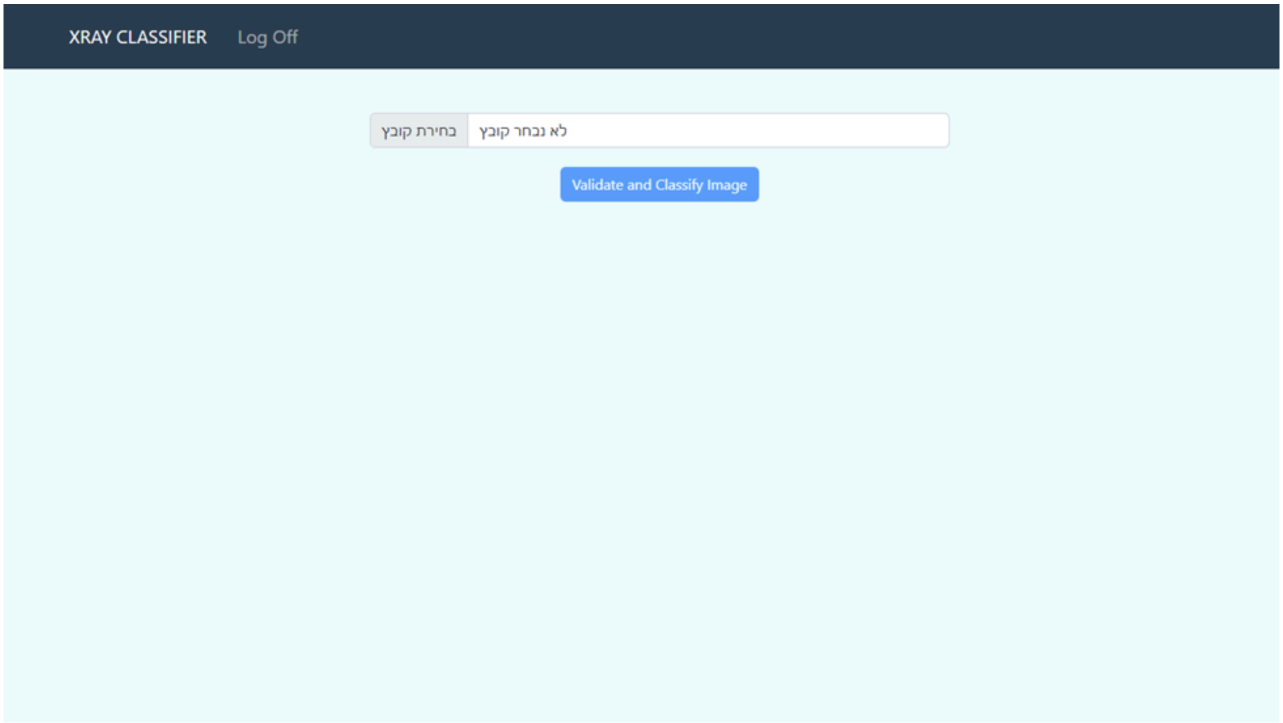
1.3 User Interface

Log in Dashboard



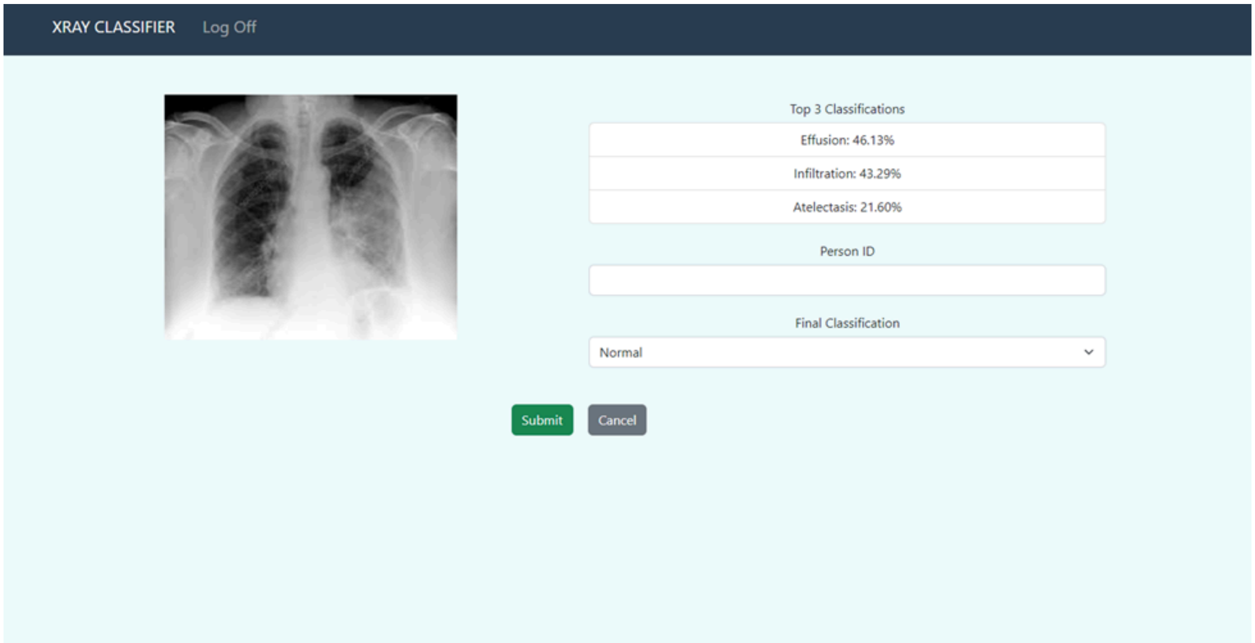
This is the login page for the XRAY CLASSIFIER system. It features a clean, minimalist design with a dark blue header containing the system name and login/sign up options. The main content area has a centered login form with fields for username and password. There's a "Forgot password?" link below the password field. The page also includes a blue "Sign in" button and an option to "Enter as Guest" below it.

Scan Upload Page



This page is the interface for uploading X-ray scans. The header now shows a "Log Off" option, indicating the user is logged in. The main content area features a file selection or drop zone for uploading images. There are two options for selecting the type of scan. A blue button labeled "Validate and Classify Image" is prominently displayed for initiating the analysis process.

Scan Results Page



Top 3 Classifications

Effusion: 46.13%

Infiltration: 43.29%

Atelectasis: 21.60%

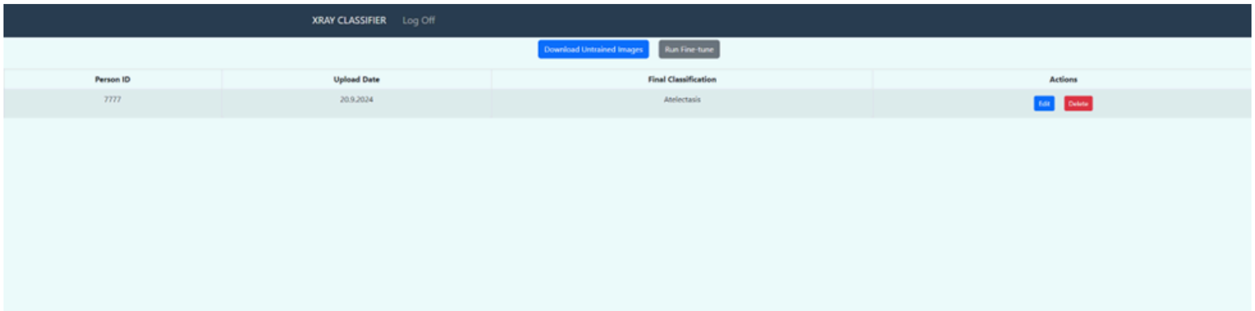
Person ID

Final Classification

Normal

This page displays the results of an X-ray scan analysis. On the left side, there's an X-ray image of a chest. On the right, the system shows the top 3 classifications with their corresponding probabilities. Below this, there's a field for entering a Person ID and a dropdown menu for selecting the final classification. The page includes "Submit" and "Cancel" buttons at the bottom.

Admin Functionality Page



Person ID	Upload Date	Final Classification	Actions
7777	20.9.2024	Atelectasis	Edit Delete

This is an administrative interface for managing scanned images. It displays a table with columns for Person ID, Upload Date, and Final Classification. Each row has "View" and "Delete" action buttons. At the top of the page, there are options to "Download Unclassified Images" and "Run Filter Script", additional administrative functions for managing the database of scans.

2. Development Process

The development process was comprehensive and multifaceted, involving several key steps:

2.1 Literature Review:

An extensive review of existing techniques in AI-based medical imaging analysis was conducted. This involved studying recent advancements in CNN architectures, transfer learning techniques, and specific applications in chest X-ray analysis. The team also investigated best practices in medical AI development, including data handling, model validation, and ethical considerations.

2.2 Data Acquisition and Preprocessing:

A large and diverse dataset of chest X-rays was obtained, likely from public repositories such as the NIH Chest X-ray dataset or the CheXpert dataset. The preprocessing pipeline included:

- Image normalization and resizing to ensure consistency across the dataset
- Data augmentation techniques such as random rotations, flips, and contrast adjustments to artificially expand the dataset and improve model generalization
- Handling of class imbalances, which are common in medical datasets where some conditions are rarer than others

2.3 Model Development:

Various CNN architectures (ResNet50, DenseNet121, EfficientNet B0, VGG16) were implemented and compared. This involved:

1. Setting up each model architecture with appropriate modifications for the chest X-ray classification task
2. Implementing transfer learning by using pre-trained weights from ImageNet and adapting the models to the specific medical imaging task
3. Experimenting with different hyperparameters and training strategies for each model

2.4 Training and Optimization:

Models were trained on the preprocessed data, with techniques like data augmentation and layer freezing employed to optimize performance. The training process likely involved:

- Splitting the data into training, validation, and test sets
- Implementing early stopping to prevent overfitting
- Using learning rate scheduling to improve convergence
- Experimenting with different optimizers (e.g., Adam, SGD with momentum) to find the best performance

2.5 Web Interface Development:

A user-friendly web interface was created for easy interaction with the system. This likely involved:

- Designing an intuitive UI/UX tailored for healthcare professionals
- Implementing secure file upload and handling for patient data protection
- Developing a backend system to queue and process uploaded images
- Creating clear and interpretable result displays, possibly including visualization of model attention or decision-making process

2.6 Integration and Testing:

The ML model was integrated with the web interface, followed by comprehensive testing. This phase probably included:

- End-to-end system testing to ensure smooth data flow from upload to result display
- Performance testing to assess system responsiveness under various load conditions
- User acceptance testing with healthcare professionals to gather feedback on usability and clinical relevance

2.7 Tools And Algorithms:

Tools used included Python for the core development, PyTorch for implementing and training the deep learning models, Pandas and NumPy for data manipulation and preprocessing, and Google Colab as the primary development environment to leverage its GPU resources. For the web interface, technologies like Node.js and Express for the backend and React + Bootstrap for the frontend have been employed.

3. Challenges Faced and Solutions:

The development process encountered several significant challenges:

3.1 Resource Limitations:

As an undergraduate project without funding, accessing sufficient computational resources for training large models on a diverse dataset was a major hurdle. This challenge was particularly acute given the size and complexity of modern CNN architectures and the large datasets required for medical imaging tasks.

Solutions:

- Implemented efficient training techniques like layer freezing in pretrained models to reduce the number of trainable parameters
- Utilized Google Colab's free GPU resources, carefully managing training sessions to maximize the use of available compute time
- Explored model compression techniques like pruning and quantization to reduce model size and computational requirements

3.2 Complex Clinical Findings:

Many of the clinical findings being tested had very similar visual cues, making it difficult for the models to differentiate between them accurately. This challenge is inherent in medical imaging tasks and reflects the complexity of diagnosis even for human experts.

Solutions:

- Experimented with attention mechanisms to help models focus on relevant areas of the X-ray images
- Implemented multi-task learning approaches to leverage potential correlations between different clinical findings
- Explored ensemble methods to combine predictions from different models, potentially capturing a wider range of visual cues

3.3 Model Performance:

Achieving the targeted accuracy levels proved challenging, especially for complex clinical findings. This reflected both the inherent difficulty of the task and the limitations of current AI technologies in medical diagnosis.

Solutions:

- Implemented advanced data augmentation techniques, including mixup and cutout, to improve model generalization
- Experimented with curriculum learning, starting with easier examples and gradually introducing more difficult cases during training
- Explored few-shot learning techniques to improve performance on rare conditions with limited training examples

4 Results and Conclusions:

The project's results provided valuable insights into the challenges and current limitations of AI in medical imaging:

- Performance varied significantly across different clinical findings:
    - Some findings achieved decent accuracy ratings suggesting potential clinical utility with further refinement
    - More complex clinical findings had very low accuracy ratings across validation sets, highlighting the difficulty of the task and the need for more advanced techniques
- The best-performing model architecture varied depending on the specific clinical finding, suggesting that ensemble methods or condition-specific models might be more effective than a one-size-fits-all approach
- The project demonstrated the feasibility of developing an end-to-end system for automated chest X-ray analysis, from image upload to result display, even with limited resources
- The challenges faced in achieving high accuracy across all conditions underscored the complexity of medical diagnosis and the current limitations of AI in this domain

These results highlight the challenging nature of the project, especially given the resource constraints and the complexity of distinguishing between similar clinical findings in chest X-rays. They also provide valuable data points for future research, indicating areas where current CNN architectures struggle and where more innovative approaches might be needed.

5 Lessons Learned:

The project provided a wealth of insights and learning opportunities:

- The importance of realistic goal-setting in medical AI projects, especially considering available resources and expertise. Future projects might benefit from a more focused approach, targeting a smaller set of clinical findings or a specific use case.

- The challenge of differentiating between visually similar clinical findings in medical imaging, emphasizing the need for more advanced techniques or additional data modalities. This suggests that future work might explore combining imaging data with clinical history or laboratory results.

- The value of efficient training techniques like data augmentation and transfer learning when working with limited computational resources. These approaches proved crucial in maximizing the performance achievable with constrained resources.

- The complexity of medical AI projects and the high level of expertise and resources required to achieve clinical-grade performance. This underscores the importance of interdisciplinary collaboration in medical AI, combining expertise in machine learning, medicine, and healthcare systems.

- The critical role of data quality and diversity in training effective medical AI models. Future projects should prioritize not just the quantity of data, but also its quality, representativeness, and careful annotation.

- The importance of interpretability and explainability in medical AI systems. While not fully implemented in this project, the experience highlighted the need for AI systems that can provide clear reasoning for their predictions to support, rather than replace, clinical decision-making.

- The potential of AI in augmenting medical diagnosis, even if full automation remains a distant goal. The project demonstrated that AI could potentially assist in triaging or flagging cases for further review, even if it can't yet match human-level performance across all conditions.

6 Achievements:

While the project did not fully meet its initial accuracy targets, it achieved several important objectives and laid groundwork for future research:

- Successfully implemented and compared multiple state-of-the-art CNN architectures for chest X-ray analysis, providing valuable data on their relative strengths and weaknesses in this domain.

- Developed a functional automated chest X-ray analysis system with a user-friendly web interface, demonstrating the feasibility of integrating AI into clinical workflows.

- Gained valuable insights into the challenges of medical AI development, particularly in resource-constrained environments. This experience will be invaluable for team members in future academic or professional pursuits in the field.

- Demonstrated the potential and limitations of current CNN models in distinguishing between complex clinical findings in chest X-rays, contributing to the broader understanding of AI capabilities in medical imaging.

- Successfully navigated the ethical and regulatory considerations involved in medical AI development, gaining practical experience in handling sensitive medical data and designing systems with potential clinical impact.

- Developed a robust data preprocessing and augmentation pipeline, which could serve as a foundation for future medical imaging projects.

- Gained hands-on experience with the full lifecycle of an AI project, from initial research and data collection to model development, deployment, and evaluation.