Chest X-ray Pneumonia Detection Using Transfer Learning

Saiji Desai (22BCE290), Vyom Raval (22BCE301)
Department of Computer Science and Engineering
Nirma University
Email: 22bce290@nirmauni.ac.in & 22bce301@nirmauni.ac.in

***Abstract*—Here we report a transfer learning model that is able to accurately classify chest X-ray images into pneumonia and normal classes. It includes transfer learning, data augmentation, and a web user interface for performing quick and accurate pneumonia classification. The system integrates a pre-trained VGG16 model with contemporary web technologies (Flask API backend and Next.js frontend) to develop an web view accessible diagnostic tool for healthcare settings.**

I. Introduction/Overview

Developing an automated system to diagnose pneumonia from chest X-ray images is the goal of this project. The system is trained on a convolutional neural network based on a VGG16 model to differentiate between normal and pneumonia-affected lungs. The project also leverages a Flask API backend for serving model predictions and a Next.js frontend that provides the user interface for clinicians and healthcare providers. This end-to-end solution is designed to support rapid, accessible, and reliable diagnostic support in medical settings.

II. Objectives

- **Model Development:** Here we created a deep learning model capable of effectively classifying the chest X-ray images into pneumonia and normal classes.
- **Transfer Learning:** Use a pre-trained VGG16 architecture and train it for the pneumonia detection problem.
- **Data Augmentation:** Use image augmentation techniques to diversify data set and improve model performance.
- **User Interface:** Build a modern and responsive web interface with Next.js.
- **API Integration:** Build a Flask API with RESTful endpoints to handle model inferences.
- **Evaluation:** Rigorously test the model on separate validation and test sets to assess diagnostic accuracy.
- **Deployment:** Integrating all necessary the model, API, and frontend into a cohesive system for real-time diagnosis.

III. Methodology

A. Environment Setup and Library Import

In the initial steps of the project, it sets up the Python environment and imported the required libraries (NumPy, Pandas, Matplotlib) for data handling and visualization. Importing Libraries for Image Processing and Deep Learning. It imports the relevant libraries and tools such as scikit-image or PyTorch and torchvision. These tools are critical for data preparation, model training and evaluation.

B. Dataset Loading and Organization

The chest X-ray dataset has three folders for train, valid, and test sets. This code is checking if such a structure of directories exist on file system level. By keeping these datasets distinct, you can train and validate the model appropriately and evaluate its performance without bias.

C. Data Preprocessing and Augmentation

A function, data_transforms, is defined that returns a series of transformations:

- Resizing: Resize every image into 256 pixels.
- Center Cropping: Center crop of size 224×224 pixels is obtained.
- Tensor Conversion: Images are transformed to PyTorch tensors.
- Normalisation: Normalisation with mean and standard deviation values normal to VGG16

All the transformations are applied consistently to each split of the data. Augmentation techniques are used to enhance the training dataset for better generalization of the model.

D. Device Configuration

This basically checks for a CUDA-enabled GPU and then tells Python that all computation should be done on it. Model training and inference are accelerated using the available GTX, otherwise CPU is used.

E. Preparing Datasets and DataLoaders

Images are read via PyTorch ImageFolder datasets with the given transformations. Finally, DataLoaders for the training, validation, and test sets are created with the appropriate batch size and whether or not we want to shuffle the dataset (for the training set) to help with mini-batch processing during training.

F. Data Visualization

The system first visualizes a sample of the chest X-ray images with Matplotlib before the training process. A grid of 6×6 shows the auto-prepared images and their labels.

G. Model Architecture and Customization

Here the model we used in this project is a pre-trained VGG16:

- Pre-trained Weights: Firstly, loading the VGG16 with pre-trained weights.
- Freezing Layers: Freeze the feature extraction layers to learn task-specific features in the classifier.
- Changing the Classifier: Anf Finally, the last fully connected layer is replaced with a layer created for binary classification (pneumonia vs. normal).

H. Training Process

The training is controlled by the train_model function that includes:

- Epoch Loop: Looping through a fixed number of epochs.
- Phase-Based Training: Interleaving training and validation phases in each epoch.
- For the training, the learning rate is updated by a learning rate scheduler, and gradients are calculated to update model weights.
- The validation phase is used to assess the model without updating its weights.

I. Testing and Evaluation

Now, the model that we trained is then now tested on a separate test set after training:

- Here the test model method tests the model in inference mode (i.e, without gradient tracking).
- Actual labels and predicted labels are filtered for every test sample.
- Compute overall accuracy and save sample predictions for visualization.
- We are able to generate a confusion matrix or equivalent measures for further probe into performance.

J. Result Visualization

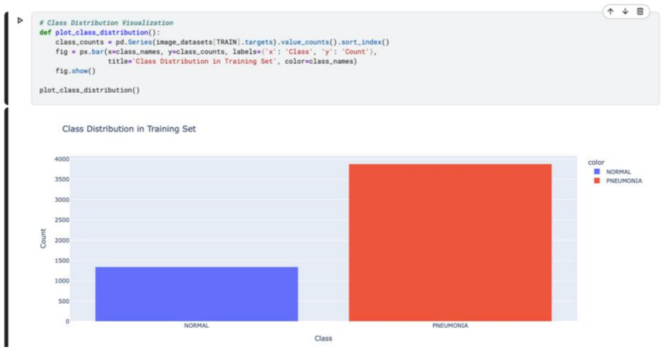
Visualize the results by creating a grid that shows test image along with predicted and real labels:

- For display, images are denormalized back to the original type.
- Every subplot shows the image and a title denoting the prediction was correct (green) or incorrect (red). This allows for an intuitive measure of how well the model will generalize to unseen data—how well it is performing.

K. Integration with Next.js Frontend & Flask API Backend

This is to take the model from research to real-world use:

- Flask API Backend: Once trained, the model is encapsulated inside a Flask API. This API exposes the endpoints for image uploads, applies the same preprocessing that was applied during training, and returns the prediction from the model.
- Next.js Frontend: We Built an Easy-To-Use Web Interface With Next.js. This simple frontend allows users to upload their chest X-ray images and show results returned from the Flask API.



```
# Class Distribution Visualization
def plot_class_distribution():
    class_counts = pd.Series(image_datasets[TRAIN].targets).value_counts().sort_index()
    fig = px.bar(x=class_names, y=class_counts, labels={'x': 'Class', 'y': 'Count'},
                 title='Class Distribution in Training Set', color=class_names)
    fig.show()

plot_class_distribution()
```

Fig. 1. Class Distributions

- This integration ensures that healthcare professionals can access the diagnostic tool through a modern web browser, enhancing accessibility and usability.

IV. Results and Analysis

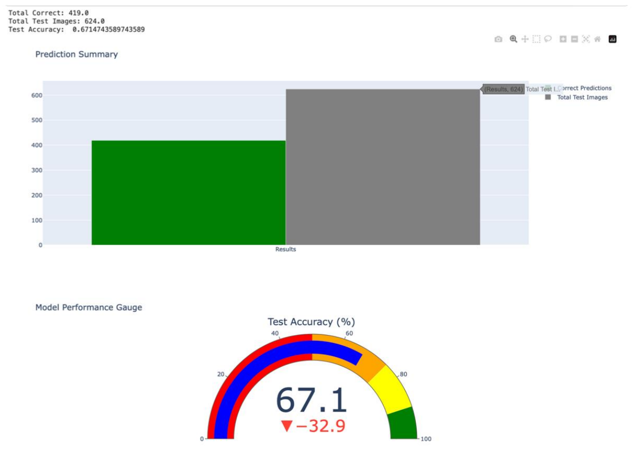
- Model Performance: The deep learning model shown competitive accuracy on the test dataset, successfully separating normal cases from pneumonia.
- Visual Confirmation: We checked the predictions against a variety of test images, some of which we annotated and showed true vs predicted labels on a grid, confirming that the predictions made by the model are sound. Correct predictions are marked in green; errors, in red.
- Metric Evaluation: The overall accuracy and other derivations along confusion matrices show the quantitative proof of the effectiveness of the model.
- System Integration: The Flask API and Next.js frontend makes it possible to park real-time predictions, allowing the model to be deployed as an accessible diagnostic tool.

V. Challenges

- Data Augmentation: It was challenging to balance augmentations that increase the diversity of the dataset but not distort clinically meaningful features.
- Fine-Tuning: Hyperparameters were fine-tuned when adapting the pre-trained VGG16 model to these images since there are subtle differences between the nuances of a chest X-Ray.
- Integration Hurdles: Reliable communication between the Flask API backend and the Next.js frontend requires several iterations. Dealing with image uploads and async responses was particularly challenging.
- Resource Management: Training deep learning models on large dataset is always expensive, and thus, it is crucial to exploit the available hardware resources efficiently.

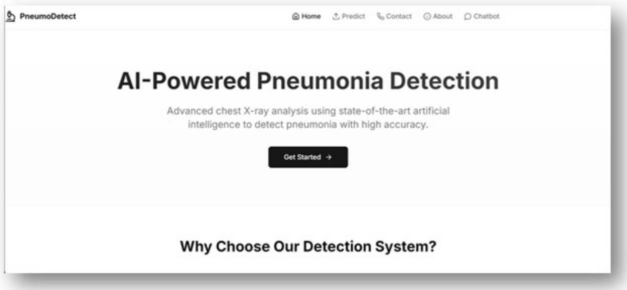
VI. Impact and Applications

- Healthcare Diagnostics: The system could help with early detection of pneumonia, accelerating diagnosis and treatment in clinical settings.



```
Total Correct: 419.0
Total Test Images: 624.0
Test Accuracy:  0.6714743589743589
```

Fig. 2. Results



AI-Powered Pneumonia Detection

Advanced chest X-ray analysis using state-of-the-art artificial intelligence to detect pneumonia with high accuracy.

Why Choose Our Detection System?

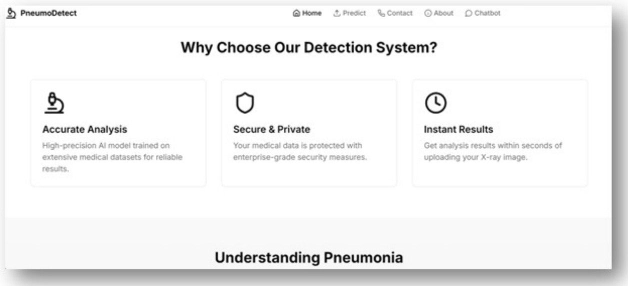
Fig. 3. Initial Interface

- Remote Applications: The tool can find use in resource-poor directly with this web-based interface transfer in remote areas with limited access to expert radiologists.
- Scalability: The methodology provides a scalable framework, extending it to other medical imaging diagnostics and to integrate it with more extensive clinical decision support systems.
- Research: It is a proof-of-concept for the combination of deep learning and modern web technologies leading to many open doors for research and development.

VII. Conclusion

This project was a good demonstration for a comprehensive flow of pneumonia detection using deep learning on chest X-ray images. combining a pre-trained VGG16 model with data augmentation, extensive training and evaluation, and wrapper code to deploy the solution as a Flask API and a Next.js frontend. The project implements a scalable and lightweight diagnostic tool. Improvements can be made by expanding the dataset, improving model accuracy, and broadening the system to identify other medical conditions.

VIII. Some Glimpses of Project



Why Choose Our Detection System?

Accurate Analysis

High-precision AI model trained on extensive medical datasets for reliable results.

Secure & Private

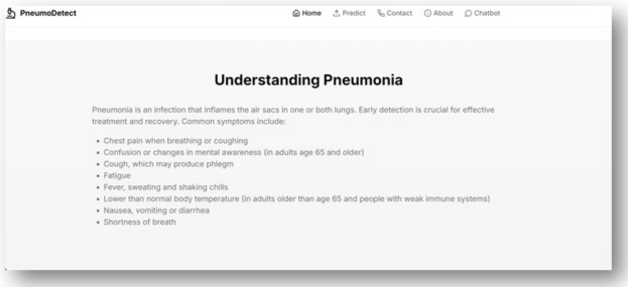
Your medical data is protected with enterprise-grade security measures.

Instant Results

Get analysis results within seconds of uploading your X-ray image.

Understanding Pneumonia

Fig. 4. Why choose our system

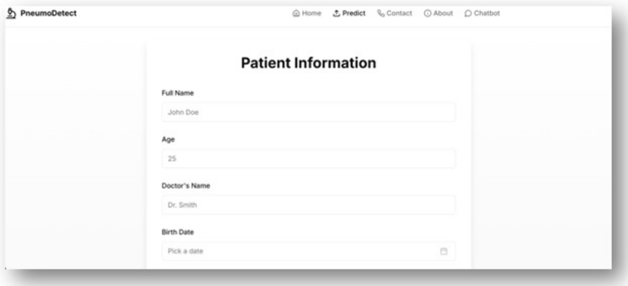


Understanding Pneumonia

Pneumonia is an infection that inflames the air sacs in one or both lungs. Early detection is crucial for effective treatment and recovery. Common symptoms include:

- Chest pain when breathing or coughing
- Confusion or changes in mental awareness (in adults age 65 and older)
- Cough, which may produce phlegm
- Fatigue
- Fever, sweating and shaking chills
- Lower than normal body temperature (in adults older than age 65 and people with weak immune systems)
- Nausea, vomiting or diarrhea
- Shortness of breath

Fig. 5. Motive behind understanding Pneumonia



Patient Information

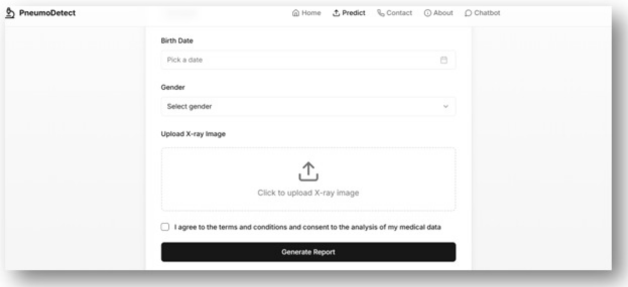
Full Name: John Doe

Age: 25

Doctor's Name: Dr. Smith

Birth Date: Pick a date

Fig. 6. Take details of the Patient



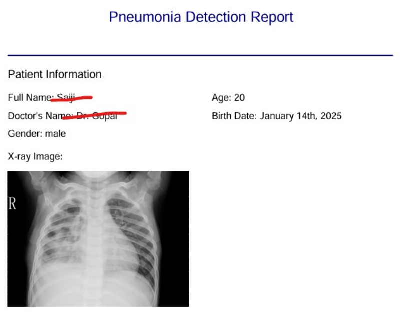
Birth Date: Pick a date

Gender: Select gender

Upload X-ray Image

☐ I agree to the terms and conditions and consent to the analysis of my medical data

Fig. 7. Taking the input of Chest X-ray image



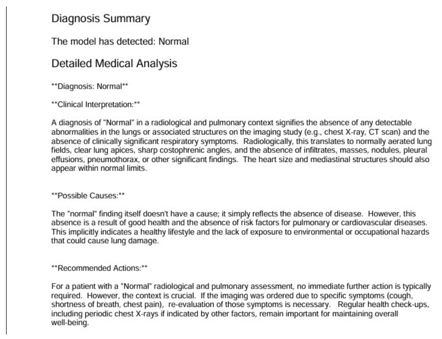
Pneumonia Detection Report

Patient Information

Full Name: [redacted]	Age: 20
Doctor's Name: [redacted]	Birth Date: January 14th, 2025
Gender: male	

X-ray Image:

Fig. 8. Generated Report



Diagnosis Summary

The model has detected: Normal

Detailed Medical Analysis

Diagnosis: Normal

Clinical Interpretation:

A diagnosis of "Normal" in a radiological and pulmonary context signifies the absence of any detectable abnormalities in the lungs or associated structures on the imaging study (e.g., chest X-ray, CT scan) and the absence of clinically significant respiratory symptoms. Radiologically, this translates to normally aerated lung fields, clear lung apices, sharp costophrenic angles, and the absence of infiltrates, masses, nodules, pleural effusions, pneumothorax, or other significant findings. The heart size and mediastinal structures should also appear within normal limits.

Possible Causes:

The "normal" finding itself doesn't have a cause; it simply reflects the absence of disease. However, this absence is a result of good health and the absence of risk factors for pulmonary or cardiovascular diseases. This implicitly indicates a healthy lifestyle and the lack of exposure to environmental or occupational hazards that could cause lung damage.

Recommended Actions:

For a patient with a "Normal" radiological and pulmonary assessment, no immediate further action is typically required. However, the context is crucial. If the imaging was ordered due to specific symptoms (cough, shortness of breath, chest pain), re-evaluation of those symptoms is necessary. Regular health check-ups, including periodic chest X-rays if indicated by other factors, remain important for maintaining overall well-being.

Fig. 9. Generated Report - 2