Capstone 3 Project Documentation: Medical Image Analysis for Pneumonia Detection

---

Sona Guliyeva

1. Project Overview and Problem Identification

1.1 Problem Statement

The core problem is the need for an efficient and accurate method to assist in the early and precise detection of pneumonia from chest X-ray images, thereby reducing diagnostic delays and potential human error in medical settings.

1.2 Context

Pneumonia requires prompt diagnosis, but manual X-ray interpretation can be slow and variable. This project aims to demonstrate how an automated image processing system can efficiently assist in pneumonia detection, improving patient outcomes and healthcare workflow, especially in high-volume or resource-limited settings.

1.3 Criteria for Success

- **High Accuracy & Balance:** Achieve high classification accuracy (e.g., >90%) with a good balance of sensitivity and specificity to minimize false positives and negatives.
- **Robustness:** The model should perform consistently across different subsets of the data.
- **Explainability (Desirable):** Ability to highlight image regions influencing the model's prediction, offering insights to medical professionals.

1.4 Scope of Solution Space

The solution will involve developing and evaluating a Convolutional Neural Network (CNN) for binary image classification (Pneumonia vs. Normal). Key aspects include image pre-processing, model training, and performance evaluation using standard metrics.

1.5 Constraints

- **Data Limitations:** Real-world medical data is more complex than publicly available datasets.

- **Computational Needs:** Training deep learning models is resource-intensive, requiring GPU access.
- **Ethical & Regulatory:** Model is an assistive tool, not diagnostic; regulatory approvals are beyond scope.
- **Time:** Project scope is limited by capstone timeline, focusing on a proof-of-concept.

1.6 Stakeholders

- **Primary Users:** Radiologists, medical imaging technicians, and general practitioners who interpret chest X-rays.
- **Indirect Beneficiaries:** Patients (through faster and potentially more accurate diagnoses).
- **Client (Conceptual):** A hospital or healthcare system interested in leveraging AI for diagnostic support and improving operational efficiency.
- **Data Scientists:** Those involved in developing, evaluating, and deploying the AI model.

1.7 Key Data Sources

- **Dataset:** Chest X-Ray Images (Pneumonia)
- **Source:** Kaggle
- **Link:** https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

2. Methodology

2.1 Data Wrangling

The initial step in data wrangling involved collecting and organizing metadata for all images in the dataset. Since the images are stored in subfolders by label (`NORMAL`, `PNEUMONIA`) and dataset split (`train`, `val`, `test`), the process involved traversing all image file paths, extracting key information (path, label, and dataset split), and building a unified DataFrame for analysis.

The dataset initially contained 5856 images. A check of the class distribution across the train, validation, and test splits was performed to understand the balance between NORMAL and PNEUMONIA images.

Data Wrangling Summary:

- Image file paths were successfully gathered into a DataFrame.
- Each image is labeled as `NORMAL` or `PNEUMONIA` and assigned to `train`, `val`, or `test` splits.
- The data is now ready for further exploratory analysis and preprocessing.

2.2 Exploratory Data Analysis (EDA)

Exploratory Data Analysis was performed to understand the data structure, class balance, and image characteristics before training a deep learning model.

2.2.1 Class Distribution by Dataset Split Visualization of the class distribution showed how many images are in each category (`NORMAL` vs `PNEUMONIA`) across train, validation, and test sets.

2.2.2 Image Dimension Analysis The width and height of all images were extracted to understand size variability. This step is crucial for deciding on an input resizing strategy for the model, as CNNs typically require fixed-size inputs. Any unreadable or tiny images (e.g., those with width or height less than or equal to 100 pixels) were filtered out.

2.2.3 Image Size Distribution Histograms were used to visualize the distribution of image widths and heights, providing insights into the typical dimensions and variability within the dataset.

2.2.4 Image Dimensions by Class Box plots were used to compare image sizes (widths and heights) between the `NORMAL` and `PNEUMONIA` classes, checking for any systematic differences in dimensions that might be correlated with the diagnosis.

2.2.5 Sample Chest X-Ray Images A visual preview of a few random sample images from both `NORMAL` and `PNEUMONIA` classes across different splits was conducted to gain qualitative understanding of the data quality and visual characteristics of each class.

2.3 Pre-processing and Training Data Development

This critical phase prepared the image data for consumption by the deep learning model, involving resizing, normalization, and data augmentation to enhance model generalization and prevent overfitting.

Data Loading and Initial Preparation: The dataset follows a standard directory structure (`main_dir/train/{NORMAL, PNEUMONIA}`, etc.). Image paths and labels were loaded into separate DataFrames for the training, validation, and test sets.

- Training samples: 5216
- Validation samples: 16
- Test samples: 624

2.3.1 Image Dimensions and Batch Size: Consistent image dimensions of 224x224 pixels were set for CNN input, with a batch size of 32. Two classes (`NORMAL`, `PNEUMONIA`) were defined.

2.3.2 Data Augmentation Strategy: An `ImageDataGenerator` was configured for the training set to perform various transformations (rescaling, shear, zoom, horizontal flip, rotation, width/height shifts, brightness adjustments) to artificially expand the dataset and improve model robustness. Validation and test sets were only rescaled.

2.3.3 Data Generators Creation: `ImageDataGenerator.flow_from_directory` was used to efficiently load images in batches directly from their respective directories, applying the defined pre-processing and augmentation steps.

- Class indices: `{'NORMAL': 0, 'PNEUMONIA': 1}`

2.3.4 Class Imbalance Handling: Class weights were calculated based on the inverse frequency of each class in the training set to prevent the model from becoming biased towards the majority class (Pneumonia).

- Calculated class weights: `{'NORMAL': 1.94, 'PNEUMONIA': 0.67}` (approximate values)

2.4 Modeling

This section details the construction, compilation, and training of the deep learning model, utilizing transfer learning with a pre-trained CNN.

2.4.1 Model Architecture Selection: ResNet50, a pre-trained Convolutional Neural Network (CNN) on the ImageNet dataset, was chosen as the base model. Its layers were frozen to leverage learned features, and a custom classification head (GlobalAveragePooling2D, Dense layers, Dropout) was added for binary classification.

2.4.2 Model Compilation: The model was compiled using the Adam optimizer with a learning rate of 0.001. `binary_crossentropy` was selected as the loss function, and `accuracy`, `Precision`, `Recall`, and `AUC` were chosen as metrics to monitor during training.

2.4.3 Callbacks Definition: Several callbacks were defined to control the training process:

- `EarlyStopping`: To stop training if validation loss does not improve for 10 epochs, restoring the best weights.
- `ReduceLROnPlateau`: To reduce the learning rate by half if validation loss does not improve for 5 epochs.
- `ModelCheckpoint`: To save the best performing model based on validation accuracy.

2.4.4 Model Training: The model was trained for 50 epochs using the prepared data generators, with `steps_per_epoch` and `validation_steps` calculated based on batch size. The calculated class weights were applied during training to mitigate class imbalance.

3. Model Evaluation and Interpretation

After training, the model's performance was rigorously evaluated on the unseen test set using various metrics and visualizations.

3.1 Load the Best Model

The `best_pneumonia_model.h5` file, saved by the `ModelCheckpoint` callback, was loaded for final evaluation.

3.2 Evaluate on Test Data

The model was evaluated on the test set to determine its performance on unseen data.

- Test Loss: 0.5258
- Test Accuracy: 0.7549
- Test Precision: 0.9213
- Test Recall: 0.6578
- Test AUC: 0.8752

3.3 Generate Predictions

True labels from the test generator were obtained, and the model generated predicted probabilities. These probabilities were converted to binary predictions (0 or 1) using a 0.5 threshold, and then mapped back to label names (`NORMAL`, `PNEUMONIA`).

3.4 Classification Report

The classification report provided detailed metrics for each class:

Class	Precision	Recall	F1-score	Support
NORMAL	0.61	0.91	0.73	234
PNEUMONIA	0.92	0.66	0.77	390
accuracy	**0.75**			**624**
macro avg	0.77	0.78	0.75	624
weighted avg	0.81	0.75	0.75	624

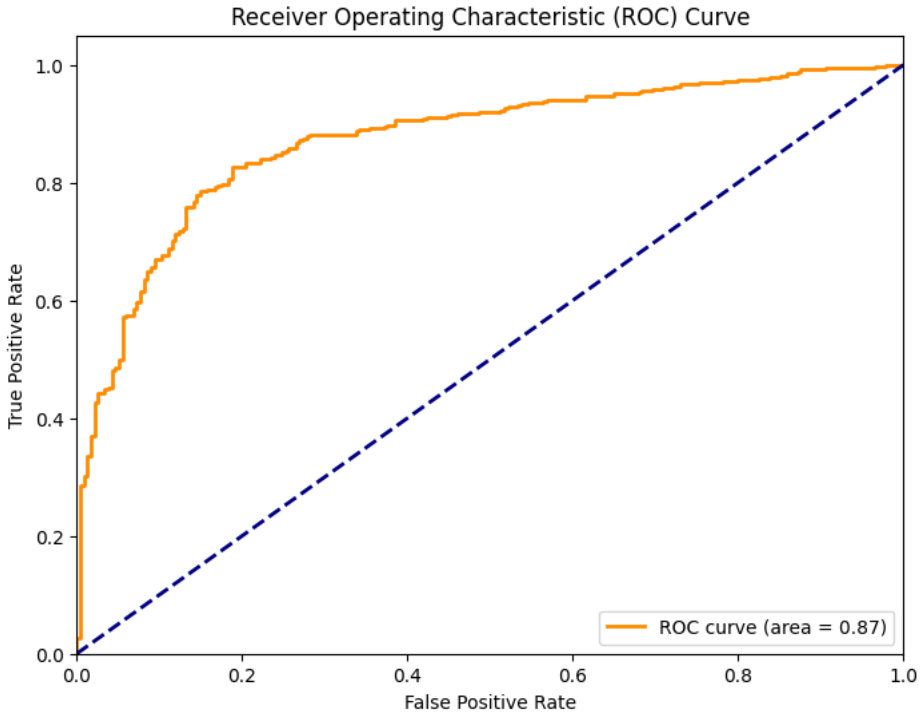
3.5 Confusion Matrix

A confusion matrix was generated to visualize the counts of true positives, true negatives, false positives, and false negatives.

- True Negatives (Normal correctly predicted Normal): 213
- False Positives (Normal predicted Pneumonia): 21
- False Negatives (Pneumonia predicted Normal): 133
- True Positives (Pneumonia correctly predicted Pneumonia): 257

3.6 ROC Curve and AUC

The Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC) were plotted. An AUC of **0.88** was achieved, indicating a good ability to distinguish between the two classes.



3.7 Assessment of Model Predictions vs. Actual Outcomes

The model achieved an overall **Test Accuracy of 75.49%**. While this indicates a reasonable ability to classify X-ray images, a deeper look at the precision, recall, and confusion matrix provides more nuanced insights:

- **Pneumonia (Class 1) Prediction:** The model shows high precision (0.92), meaning when it predicts "Pneumonia," it is correct 92% of the time, resulting in few false positives for pneumonia. However, its recall (0.66) indicates that it correctly identifies only 66% of actual pneumonia cases, missing 34% (false negatives).
- **Normal (Class 0) Prediction:** The model has high recall (0.91) for the "Normal" class, meaning it rarely misclassifies a normal case as pneumonia. Its precision (0.61) is lower, indicating a higher rate of false positives for "Normal" (i.e., predicting Normal when it's actually Pneumonia).

The confusion matrix confirms that the model is very good at identifying normal cases. However, the relatively high number of **False Negatives (133)** for pneumonia is critical. In a medical context, missing a pneumonia diagnosis can have serious consequences.

Overall Assessment: The model demonstrates promising capabilities, particularly in its high precision for pneumonia detection and its ability to correctly identify healthy cases. However, the relatively high number of false negatives for pneumonia (missed diagnoses) is a significant area for improvement in a real-world clinical application. This could be due to the class

imbalance (despite using class weights), the complexity of pneumonia patterns, or the need for further hyperparameter tuning and more advanced augmentation strategies. Future work should prioritize reducing false negatives to make the model a more reliable assistive tool for medical professionals.