**Attention-Based Multi-Label Classification of Thoracic Diseases from Chest X-rays**

**Problem Domain and Project Description**

Chest X-ray images are critical tools in medical diagnostics, enabling early detection and treatment of conditions such as pneumonia, effusion, and atelectasis. However, interpreting these images requires skilled radiologists, and resource constraints in healthcare systems often delay diagnosis. This project aims to address this gap by developing a deep learning-based system for automated multi-label classification of chest X-rays. This project focuses on building a deep learning-based model that classifies chest X-ray images into specific disease categories. Leveraging transfer learning and attention mechanisms, the model identifies patterns and anomalies associated with thoracic diseases such as 'No Finding,' 'Infiltration,' 'Effusion,' and 'Atelectasis'.

Following the input, output of the project:

Input: Preprocessed and augmented chest X-ray images.

Output:

1. **Model Predictions**:
   * Probabilities for each label (No Finding, Infiltration, Effusion, Atelectasis) using sigmoid activation.
2. **Evaluation Metrics**:
   * Classification performance (e.g., ROC AUC, F1-score, Precision, Recall).
   * Classification reports for individual labels.
3. **Trained Model**:
   * A saved Keras model (best\_model.keras) for inference on unseen data

**Approach**

**High-Level Methodology**

**Dataset**

The NIH Chest X-Ray dataset from Kaggle is the primary data source, which contains 112,120 chest X-ray images with disease labels from 30,805 unique patients. The disease labels are divided into 14 categories: Atelectasis, Cardiomegaly, Consolidation, Edema, Effusion, Emphysema, Fibrosis, Hernia, Infiltration, Mass, Nodule, Pleural Thickening, Pneumonia, Pneumothorax, No Findings.

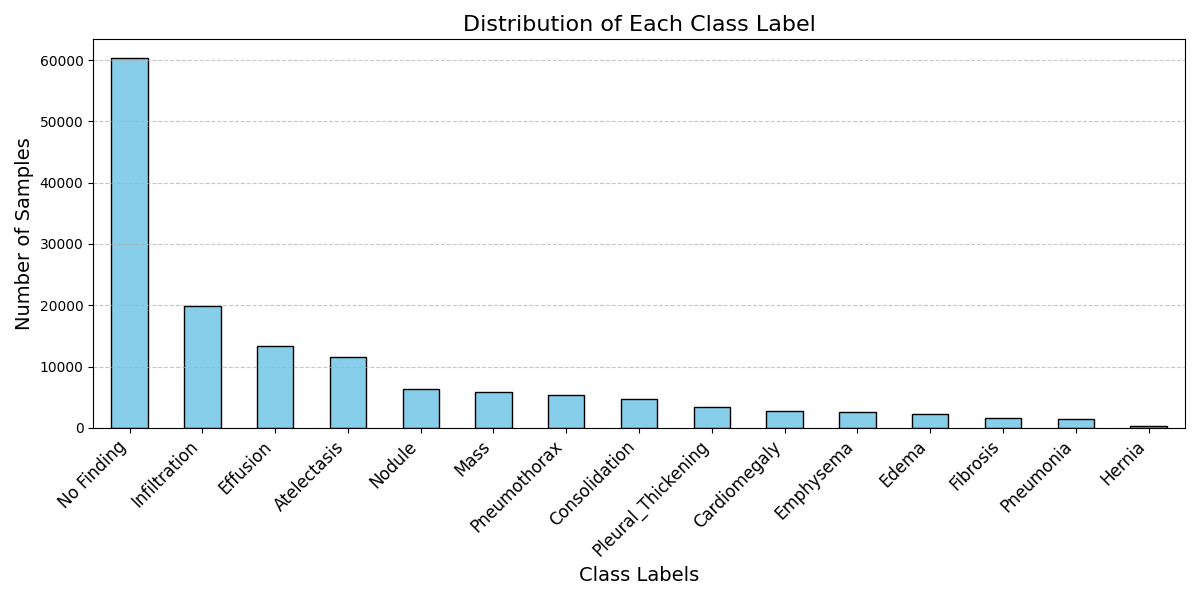
The following are the key features of the dataset:

Image Index: The name of the image

Finding Labels: Diseases associated with that particular patient.

Labels were processed to handle multi-label classification. For example, a sample might have labels like "Effusion|Atelectasis". These were split and one-hot encoded for all unique labels in the dataset. Each image's path was constructed using the metadata and directory structure. This ensured the efficient loading of images during training.

Then, we plotted a bar chart to check the distribution of the labels. We found that the dataset had a high class imbalance as shown in the plot below.



From the above plot, we can see that the ‘No Finding’ label has the highest number of samples (approximately 60,000). Classes such as Infiltration, Effusion, and Atelectasis also have many samples but far fewer than No Finding. Less frequent classes, such as Hernia*,* Pneumonia, and Fibrosis, have very few samples, indicating they are underrepresented.

To improve the performance of the model, we decided to consider the top 4 labels having the highest number of samples – Non-Finding, Infiltration, Effusion, and Atelectasis. For training the model, we decided to consider 50,000 images.

Further to handle class imbalance, data augmentation was applied to artificially expand the dataset and improve model generalization:

* **Geometric Transformations**: Rotations, zooms, and shifts were used to simulate variations in image capture.
* **Photometric Adjustments**: Brightness scaling helped the model become robust to variations in exposure.
* **Preprocessing**: Images were resized to a fixed size of 512x512 pixels and normalized using the VGG16 preprocessing pipeline.

**Model Design**

The methodology employs a **Combined Attention Model**, leveraging a pre trained VGG16 backbone for feature extraction and an attention mechanism to enhance interpretability and performance.

1. **Feature Extraction with VGG16:**
   * The VGG16 model, pretrained on ImageNet, is used to extract high-level image features.
   * The model's convolutional layers were frozen to retain pretrained knowledge, while additional layers were added for task-specific learning.
2. **Attention Mechanism**:

An attention mechanism was added to help the model focus on relevant regions of the image:

* **Attention Layers**: A sequence of convolutional layers generates an attention map, which highlights the most critical areas in the image for each condition.
* **Feature Masking**: The attention map modulates the extracted features, effectively weighting important regions more heavily.
* **Rescaled Global Average Pooling (GAP)**: Combined information from the attention map and features is passed through pooling layers to reduce dimensionality.

1. **Fully Connected Layers**:
   * Dense layers were added after feature extraction and attention pooling to map the processed features to multi-label outputs.
   * Dropout layers were used to prevent overfitting.
2. **Output Layer:**
   * The final output layer uses a sigmoid activation function, providing probabilities for each of the selected labels ("No Finding", "Infiltration", "Effusion", and "Atelectasis").

**Weighted Loss Function**

To handle class imbalance, a weighted binary cross-entropy loss function was implemented: Class weights were computed based on the prevalence of each label in the dataset. The loss function dynamically adjusts the penalty for misclassifications of underrepresented classes, ensuring the model pays more attention to rare conditions.

**Training Pipeline**

1. **Data Splitting:**
   * The dataset was split into training (80%) and validation (20%) sets.
2. **Training Configuration:**
   * Batch size: 32 for training, 40 for validation.
   * Optimizer: Adam optimizer with a learning rate of 5e-5.
   * Loss function: Weighted binary cross-entropy.
   * Augmented image batches were fed to the model during training to improve robustness.
3. **Callbacks for Optimization:**
   * **Model Checkpoint:** Saves the best model based on validation loss.
   * **Early Stopping:** Halts training if validation loss does not improve for five consecutive epochs.
   * **Learning Rate Scheduler:** Reduces the learning rate when validation loss plateaus.
4. **Hardware Acceleration:**
   * The training process utilized GPU acceleration with TensorFlow’s CUDA backend for faster computation.

**Interpretation of Evaluation Metrics and Model Performance**

* **ROC-AUC (Macro Average: 0.7343):**
  + Indicates the model's ability to distinguish between the presence and absence of diseases across all labels.
  + A value of 0.7343 shows moderate discrimination performance, suggesting the model can differentiate well but may need improvement for higher accuracy.
* **F1-Score (Macro Average: 0.4432):**
  + Balances precision and recall, offering a single metric for overall performance.
  + A relatively low score indicates that the model struggles to maintain a good trade-off between detecting true positives (recall) and avoiding false positives (precision).
* **Precision (Macro Average: 0.3755):**
  + Measures the proportion of correct positive predictions out of all positive predictions.
  + A low precision suggests the model produces a significant number of false positives, especially for less frequent classes.
* **Recall (Macro Average: 0.6721):**
  + Measures the proportion of true positives correctly identified out of all actual positives.
  + A higher recall indicates the model is sensitive to detecting diseases.

**2. Classification Report Interpretation (Per Class):**

* **No Finding:**
  + Precision: 0.81, Recall: 0.72, F1-Score: 0.76
  + Strong performance reflects the abundance of "No Finding" samples, enabling the model to learn this class effectively.
* **Infiltration:**
  + Precision: 0.27, Recall: 0.60, F1-Score: 0.38
  + High recall but low precision shows the model is overpredicting infiltration, leading to false positives.
* **Effusion:**
  + Precision: 0.24, Recall: 0.74, F1-Score: 0.36
  + Similar to infiltration, the model is sensitive to detecting effusion cases but at the expense of precision.
* **Atelectasis:**
  + Precision: 0.17, Recall: 0.64, F1-Score: 0.27
  + The model has a high false positive rate for this rare class, showing difficulty in correctly predicting it.