Computer Vision CS-GY 6643 - Final Project - Self-Supervised Learning for Medical Image Representation using MoCo (Momentum Contrast)

Chirag Mahajan - cm6591@nyu.edu, Mohammed Basheeruddin - mb9885@nyu.edu, Shubham Goel - sg4599@nyu.edu, Nirbhaya Reddy G - ng3033@nyu.edu

1 Introduction and background

Pneumonia and other thoracic diseases, such as cardiomegaly, pleural effusion, and atelectasis, are prevalent and often life-threatening conditions. Timely and accurate diagnosis using chest X-rays (CXRs) is essential for effective treatment, but the shortage of trained radiologists can delay diagnosis. This is where deep learning-based methods have shown great potential in automating disease detection from medical images. [4]

Traditional supervised learning approaches require large amounts of labeled data to perform well, but labeling medical datasets can be expensive and time-consuming. This limitation makes self-supervised learning (SSL) methods, such as Momentum Contrast (MoCo), attractive for learning robust feature representations from large-scale unlabeled data. [1] SSL methods have achieved state-of-the-art results on natural image datasets, and we aim to explore their application to medical images for multi-label classification tasks. [1], [2], [3]

In this project, we explore the use of MoCo (Momentum Contrast) for self-supervised learning to learn robust representations from CXR images. We then fine-tuned these models on the CheXpert dataset to classify multiple pathologies, including pneumonia, cardiomegaly, pleural effusion, and more. The goal is to assess the effectiveness of SSL in learning from large-scale medical image datasets and compare these methods to traditional supervised models, such as ResNet-50 and Inception V3.

Previous studies have primarily focused on supervised learning approaches, such as the CheXNet model developed using the NIH ChestX-ray14 dataset. [4] However, self-supervised learning methods have been less explored in this domain. With the increasing availability of large CXR datasets like CheXpert and improved computational resources, we believe that SSL techniques, such as MoCo, SimCLR, BYOL, and SwAV, have the potential to significantly reduce the reliance on labeled data and improve model performance across multiple disease classifications. [1], [2], [3]

2 Dataset

The **CheXpert dataset** contains 224,316 chest radiographs from 65,240 patients with labels for 14 diseases, including pneumonia, cardiomegaly, pleural effusion, atelectasis, edema, and more. This dataset includes **multi-label annotations** where each image may be associated with multiple pathologies. [4]

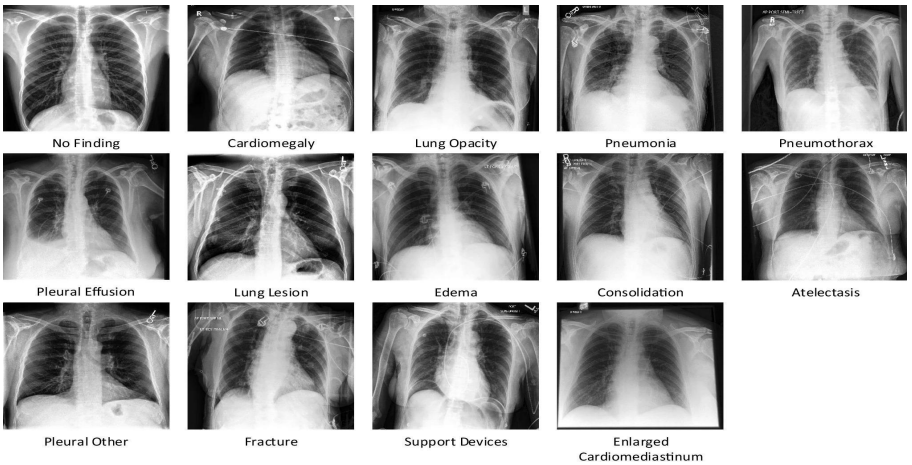


Fig: CheXpert Dataset Chest X-ray dataset samples with labels

Data Characteristics:

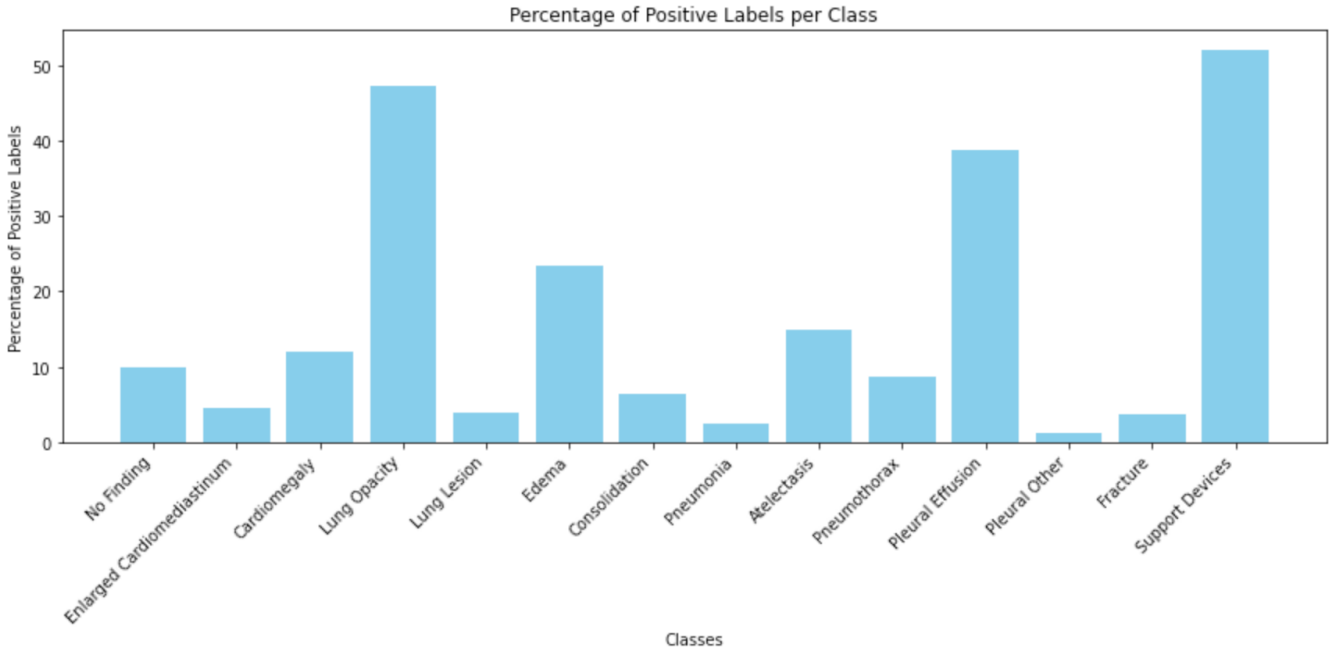


Fig: Percentage of Positive Labels per Class

The graph highlights significant class imbalance in the CheXpert dataset, with classes like **"Lung Opacity"** and **"Support Devices"** having a high proportion of positive labels, while rare classes like **"Pleural Other"** and **"Pneumonia"** are underrepresented. This imbalance can lead to biased models favoring dominant classes. Techniques like **Weighted Random Sampling** and **Focal Loss** are essential to address this imbalance, ensuring better performance for minority classes and improving clinical applicability.

Preprocessing Pipeline for the CheXpert Dataset

The preprocessing pipeline was designed to handle the challenges of medical imaging datasets, such as variability in image quality, class imbalance, and uncertain labels. The steps taken ensure compatibility with deep learning models and maximize the performance of the selected methods (MoCo, SimCLR, SwAV, BYOL).

1. Image Loading and Resizing

- **Step**: Images are loaded in grayscale format and resized to 224x224 pixels.
- **Rationale**: Grayscale ensures focus on structural features relevant for medical diagnosis. Resizing standardizes input dimensions for pre-trained networks like ResNet-50.
- **Impact**: Maintains compatibility with downstream models and reduces computational requirements.

2. Preprocessing

- **Step**:
  - **Denoising**: Applied `cv2.fastNlMeansDenoising` to remove noise while preserving edge details.
  - **Histogram Equalization**: Enhanced contrast using `cv2.equalizeHist`.
  - **Standardization**: Normalized images by subtracting their mean and dividing by their standard deviation.
  - **Conversion to 3 Channels**: Created pseudo-RGB images by stacking grayscale data into three channels.
- **Rationale**: These preprocessing steps improve contrast and remove irrelevant variations while mimicking the input format of pre-trained ImageNet models.
- **Impact**: Enhanced visual quality for learning meaningful patterns and ensured compatibility with pre-trained architectures.

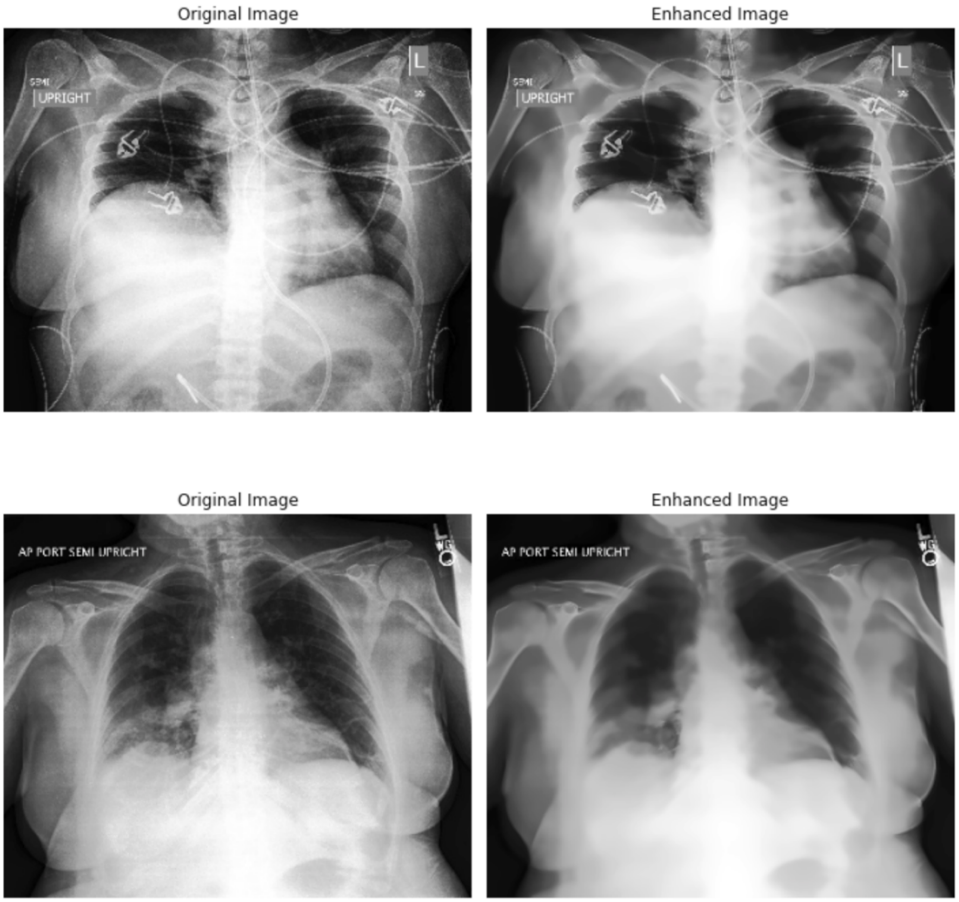


Fig: Enhanced Image after applying Image Enhancement Techniques

3. Data Augmentation

- **Augmentations** (per transformation strategy):
  - **SimCLR**: Color jitter, Gaussian blur, and cropping augmentations to generate diverse positive pairs.
  - **BYOL**: Two different augmentation pipelines (e.g., Gaussian blur for one view, cropping for another) to enforce consistency between views.
  - **MoCo**: Applied strong augmentations such as cropping and Gaussian blur for contrastive learning.
  - **SwAV**: Employed multi-crop strategies with two global and six local views to enforce invariance across spatial transformations.
- **Rationale**: Augmentations promote robustness by simulating real-world variations in medical images.
- **Impact**: Prevented overfitting, improved model generalization, and strengthened contrastive learning.

4. Class Imbalance Mitigation

- **Steps**:
  - **WeightedRandomSampler**: Adjusted sampling probabilities to address imbalances in the dataset.
  - **Focal Loss**: Down-weighted easy examples while emphasizing harder-to-classify samples.
- **Rationale**: Medical datasets often have skewed distributions, with conditions like "No Finding" dominating the dataset.
- **Impact**: Improved recall for minority classes, ensuring fair representation in predictions.

5. Handling Uncertain Labels

- **Step**: Treated uncertain labels as negative (U-Zeros approach).
- **Rationale**: Simplifies label ambiguity while retaining clinically meaningful interpretations.
- **Impact**: Reduced noise in the training process and improved model robustness to label uncertainty.

6. Conversion to Tensors

- **Step**: Images were converted to PyTorch tensors and preprocessed dynamically using data loaders.
- **Rationale**: Enabled efficient batching, augmentation, and memory management during training.
- **Impact**: Streamlined data input pipeline, enabling real-time preprocessing during model training.

7. Test Dataset Transformation

- **Step**: Applied minimal transformations (resizing to 224x224 and converting to tensors) to test datasets.
- **Rationale**: Ensures predictions reflect real-world input data without augmentation artifacts.
- **Impact**: Provided an unbiased evaluation of model performance.

8. Model-Specific Transformations

1. **SimCLR**: Strong augmentations for generating diverse positive pairs.
2. **BYOL**: Dual-view strategy with distinct augmentations for each view.
3. **MoCo**: Single transformation pipeline with robust augmentations.
4. **SwAV**: Multi-crop strategy for spatially invariant representation learning.

Model	Transformations	Purpose
SimCLR	Color jitter, random crop, Gaussian blur, horizontal flip	Create diverse positive pairs for contrastive learning.
BYOL	Dual-view augmentation with distinct strategies (strong and weak augmentations)	Enforce consistency between differently augmented views.
MoCo	Single augmentation pipeline with cropping, flipping, Gaussian blur, and color jitter	Populate memory bank for momentum contrastive learning.
SwAV	Multi-crop strategy with global and local views, clustering-based learning	Learn both local and global representations with spatial invariance.

3 Methods

a. MoCo for Self-Supervised Learning

MoCo maintains a dynamic dictionary of feature representations for contrastive learning. It includes a query encoder and a key encoder, where the key encoder is updated using momentum from the query encoder. The query and key encoders process different augmentations of the same image, and contrastive loss ensures that positive pairs (the same image with different augmentations) are pulled together, while negative pairs are pushed apart. [4]

- **Backbone**: ResNet-50 (pre-trained on ImageNet)
- **Projection Head**: A 2-layer MLP to project the feature embeddings into a lower-dimensional space.
- **Queue Size**: 4096 (to maintain a large set of negative samples for contrastive learning).
- **Temperature**: 0.07 to control the smoothness of the contrastive loss.

MoCo Architecture

Momentum Contrast (MoCo) is a self-supervised learning framework that leverages contrastive learning to extract useful feature representations from unlabeled data. MoCo is particularly well-suited for tasks where labeled data is scarce, making it highly applicable to medical image analysis. The core components of MoCo's architecture include the **query encoder**, **momentum encoder**, **feature queue**, **one-hot target**, and **contrastive loss**. The following section provides a detailed breakdown of each component. [6]

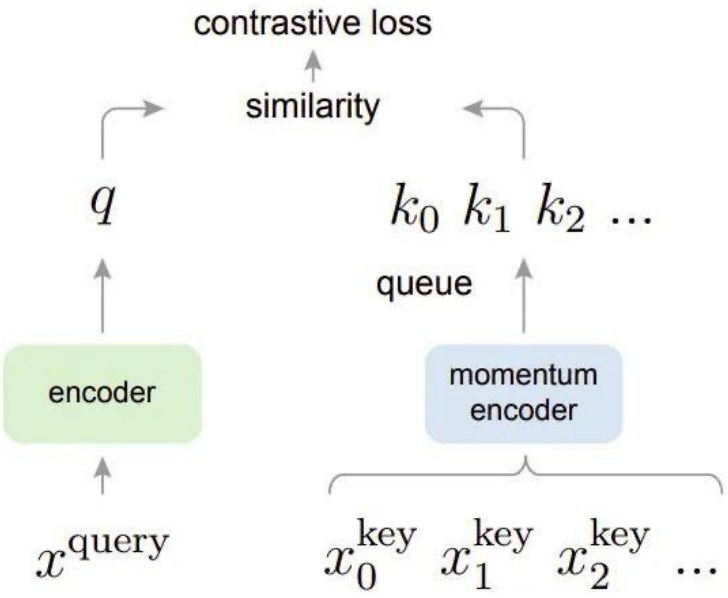


Fig. 3: MoCo Architecture with query and ky dictionary [6]

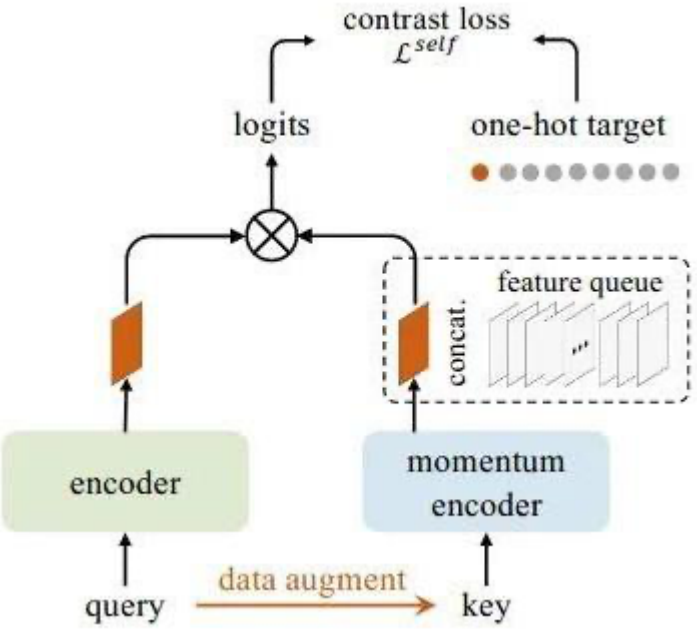


Fig. 4: MoCo Architecture in more detail [6]

1. Query and Key

MoCo takes two inputs: the **query** and the **key**. A query is an image sampled from the dataset, while the key is an augmented version of that query. The central idea behind contrastive learning is that similar samples (e.g., the query and its key) are considered **positives**, and different samples are treated as **negatives**. Since the labels of other samples in the dataset are unknown, they are considered negatives by default in MoCo. This setup enables the model to distinguish between different samples without requiring labeled data. [6]

2. Encoder and Momentum Encoder

MoCo uses two encoders to extract feature representations: a **query encoder** and a **momentum encoder**. The encoder is typically a convolutional neural network (CNN), such as ResNet-50, pre-trained on ImageNet to extract features from input images. A key challenge in contrastive learning is efficiently updating the key encoder, as backpropagating through a large queue of negative samples is computationally expensive. To overcome this, MoCo introduces the concept of **momentum updating** for the key encoder. [6]

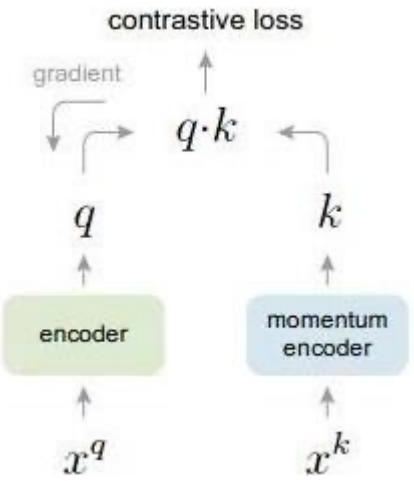


Fig. 5: Momentum Encoder Architecture with a gradient on the encoder branch only [6]

In MoCo, only the query encoder is updated via backpropagation, while the key encoder is updated using a momentum coefficient **m** (Equation 1). This allows the key encoder to evolve more smoothly over time, updating its parameters as a weighted average of the query encoder's parameters. The momentum update ensures that the key encoder remains consistent while reducing computational overhead.

$$\theta_k \leftarrow m \cdot \theta_k + (1 - m) \cdot \theta_q$$

Where:

- θ_k represents the key encoder's parameters.
- θ_q represents the query encoder's parameters.
- m is a momentum coefficient (typically set between 0 and 1).

Equation 1: Momentum Encoder Update [6]

This mechanism ensures that the key encoder is updated in a stable and efficient manner, preventing the need for backpropagation through the entire feature queue.

3. Feature Queue

A distinguishing feature of MoCo is the use of a **feature queue**, a large dynamic dictionary that stores encoded representations (keys) from previous mini-batches. This allows MoCo to maintain a large number of negative samples without needing an excessively large batch size. The feature queue is updated in a **first-in-first-out (FIFO)** manner, where each new mini-batch of encoded keys is enqueued, and the oldest mini-batch is dequeued. This setup ensures that the model is continually exposed to a diverse set of negative samples, improving the quality of the learned representations. [6]

The feature queue not only improves the memory efficiency of the model but also allows MoCo to scale to larger datasets by enabling the model to learn from a larger pool of negative samples. [6]

4. One-Hot Target Vector

MoCo uses a **one-hot target vector** to distinguish between positive and negative samples in an unsupervised manner. Since the labels of the samples are unknown, the one-hot vector assigns a positive label to the query and its corresponding key, while all other samples in the dataset are assigned negative labels. This mechanism ensures that the model learns to group similar samples (positive pairs) while pushing apart dissimilar samples (negative pairs). [6]

5. Contrastive Loss

The learning objective of MoCo is driven by **contrastive loss**, which measures the similarity between the query image and both its positive and negative keys. Positive keys ($K+$) are representations of the same image as the query (but with different augmentations), while negative keys ($K-$) are representations of other images in the dataset. MoCo uses the **dot product** to measure the similarity between the query and its keys. [6]

MoCo employs the **InfoNCE contrastive loss** (Equation 2), inspired by classification loss, where the goal is to maximize the similarity between the query and its positive key and minimize the similarity between the query and all negative keys. The logit for a query-key pair is given by:

$$S_k = \frac{q \cdot k}{\tau}$$

Where:

- S_k represents the logit for the query q and key k.
- τ is the **temperature parameter** that controls the sharpness of the distribution.

Equation 2: InfoNCE Contrastive Loss [6]

The **temperature parameter** plays a critical role in the behavior of the contrastive loss. A lower temperature (τ=0.07) sharpens the probability distribution, making the model more confident in its predictions. A smaller τ enforces a harder separation between positive and negative pairs, which is beneficial for learning robust representations.

$$L(q, k^+, \{k^-\}) = -\log\left(\frac{\exp(q \cdot k^+/\tau)}{\exp(q \cdot k^+/\tau) + \sum_{k^-} \exp(q \cdot k^-/\tau)}\right)$$

Where:

- $L(q, k^+, \{k^-\})$ is the contrastive loss for a given query q, positive key k^+, and a set of negative keys $\{k^-\}$.
- τ is the temperature parameter, set to 0.07 in MoCo.

Equation 3: Contrastive Loss Equation [6]

This loss encourages the model to maximize the similarity between positive pairs (query and key) and minimize the similarity between the query and all negative keys. [6]

By combining these elements, MoCo can learn high-quality feature representations from large unlabeled datasets with limited labeled data. The framework is particularly well-suited for medical image analysis, where labeled datasets are often small and expensive to create. MoCo's architecture efficiently handles large-scale datasets, allowing it to capture diverse feature representations while minimizing memory and computational costs.

b. Alternative Self-Supervised Learning Methods

In addition to MoCo, we will implement and compare three other SSL methods in the coming week:

- **SimCLR**: SimCLR is a contrastive learning framework that relies on generating pairs of augmented images and comparing their representations. It uses in-batch negative samples, meaning that each image's augmented variant is contrasted against all other images in the batch. As a result, it requires large batch sizes to obtain a sufficient number of meaningful negative pairs and to learn robust, representation-rich features. [4]
- **BYOL**: BYOL (Bootstrap Your Own Latent) removes the need for negative samples altogether. Instead, it focuses on learning representations by minimizing the distance between the outputs of two augmented views of the same image. This is achieved through a teacher-student network setup, where the teacher network provides stable targets. As a result, BYOL effectively captures rich features from unlabeled data without relying on contrasting differing samples. [3]
- **SwAV**: SwAV (Swapping Assignments between Views) uses clustering-based self-supervised learning. Representations are assigned to cluster prototypes, and these assignments are then "swapped" between different augmented views of the same image. This ensures that multiple views converge toward a consistent cluster representation, enabling the model to learn more invariant and semantically meaningful image features. [4]

c. Fine-Tuning for Multi-Label Classification

After pre-training on unlabeled CXR images, we fine-tuned the pretrained SSL models with small subsets of labeled data (10%, 50%, 100%). We will replace the projection head with a multi-label classification head, allowing the model to predict the presence of multiple diseases for each image.

For **uncertain labels** in CheXpert, we will experiment with different strategies (e.g., ignoring uncertain labels, converting them to positive/negative labels, or using a dedicated "uncertain" class).

4 Baseline Methods

We compare our SSL-based models with the following supervised learning models:

- **ResNet-50 (Supervised Learning)**:ResNet-50 is a deep convolutional neural network known for its residual connections, which help combat vanishing gradients. Pre-trained on the large-scale ImageNet dataset, ResNet-50 can be fine-tuned on the CheXpert dataset for multi-label classification tasks. By leveraging the robust features learned from ImageNet, it provides a strong supervised baseline that can adapt to medical imaging domains.
- **Inception V3 (Supervised Learning)**: Inception V3 is another well-established convolutional neural network architecture pre-trained on ImageNet. It employs multiple filter sizes simultaneously within its inception modules to capture a variety of visual patterns. When fine-tuned on the CheXpert dataset, it serves as a strong supervised learning benchmark, allowing direct comparison with other methods aiming to classify multiple label categories in medical images.
- **Self-Supervised Models**: In addition to MoCo, we will compare SimCLR, BYOL, and SwAV for multi-label classification. [3], [4]

5 Methodology / Pipeline

The methodology, as represented in the diagram below, is structured to leverage both labeled and unlabeled data through a Self-Supervised Learning (SSL) pipeline.

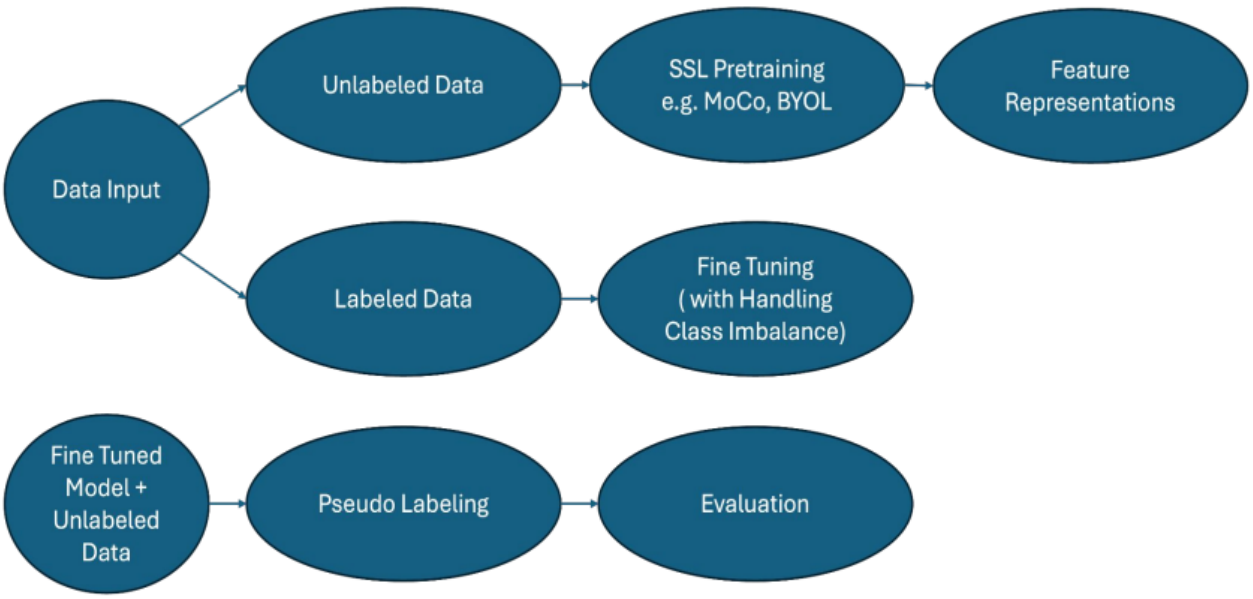


Fig.6 Workflow of the Methodology

The steps are as follows:

- **Self-Supervised Pre-Training:**

  The methodology follows a comprehensive training pipeline with four key stages. In the self-supervised pre-training phase, unlabeled chest X-ray images are used to train models leveraging self-supervised learning (SSL) frameworks such as Momentum Contrast (MoCo) and BYOL. These frameworks aim to extract meaningful feature representations by aligning positive pairs (different augmentations of the same image) and separating negative pairs. Mechanisms like contrastive loss, momentum-based encoder updates, and a dynamic feature queue are integral to this process, ensuring robust representation learning without reliance on labels.

- **Fine-Tuning with Labeled Data:**

  Following pretraining, the models undergo fine-tuning with labeled data, where the SSL projection head is replaced by a multi-label classification head to predict multiple pathologies per image. This stage involves optimizing the models on labeled datasets while addressing challenges like class imbalance through weighted loss functions and adjusted sampling techniques. Fine-tuning is carried out on varying amounts of labeled data (e.g., 10%, 50%, and 100%) to evaluate the models' adaptability to different data availability scenarios.

- **Pseudo-Labeling:**

  In the pseudo-labeling phase, the fine-tuned models are applied to unlabeled data to generate pseudo-labels, effectively expanding the labeled dataset. These pseudo-labels are iteratively incorporated into the training process, enabling a semi-supervised learning approach that further refines the model's performance. This iterative pseudo-labeling process allows the models to leverage the full dataset, reducing reliance on human-annotated labels and improving generalization.

- **Evaluation:**
  Finally, the pipeline concludes with a rigorous evaluation phase, where the models' performance is assessed across multiple metrics, including accuracy, precision, recall, and ROC-AUC. These evaluations are conducted on both the labeled and pseudo-labeled datasets to ensure robustness and reliability in real-world applications. This structured methodology effectively integrates SSL, fine-tuning, pseudo-labeling, and evaluation to achieve optimal performance in medical image classification tasks.

- **Interpretability with Grad-CAM:**

  Gradient-weighted Class Activation Mapping (Grad-CAM) is a technique used to visualize and interpret the regions of an image that a neural network deems most important for its predictions. By utilizing the gradient information flowing into the final convolutional layers of a CNN, Grad-CAM generates heatmaps that highlight the spatial locations in the input image that contribute most strongly to a particular class decision. This approach is especially useful in domains like healthcare, where model transparency is essential. It helps clinicians and researchers verify that the model’s focus aligns with clinically relevant features, enhancing trust and interpretability in AI-driven decision-making processes.

  Model interpretability is critical in healthcare applications. To ensure that our models are making clinically relevant decisions, we will implement Grad-CAM visualizations. This will allow us to see which regions of the chest X-rays the models are focusing on when predicting different diseases. Grad-CAM will be applied to the MoCo SSL model to provide insights into which regions of the chest X-rays the model focuses on when making its predictions. [4]

6 Optimization Techniques

In this project, several advanced techniques were employed to enhance training efficiency, optimize resource utilization, and achieve better model performance. These techniques are outlined below:

1. Automatic Mixed Precision (AMP):

- **Purpose:** Reduce memory usage and accelerate training by using mixed-precision arithmetic (combination of 16-bit and 32-bit floating-point operations).
- **Implementation:** Enabled with libraries like PyTorch's `torch.cuda.amp` for efficient model training, particularly on GPUs.
- **Impact:** Allowed larger batch sizes and faster computations without sacrificing model accuracy.

2. AdamW Optimizer:

- **Purpose:** Improve weight regularization during optimization by decoupling weight decay from the gradient update.
- **Implementation:** Used the `AdamW` optimizer instead of the traditional `Adam`, with appropriate weight decay values.
- **Impact:** Helped in better generalization and reduced overfitting, especially for models trained on large-scale datasets.

3. OneCycle Learning Rate Policy:

- **Purpose:** Accelerate convergence and improve training stability by dynamically adjusting the learning rate during training.
- **Implementation:** Utilized a cyclical learning rate schedule (OneCycle LR) where the learning rate gradually increases, peaks midway, and then decreases.
- **Impact:** Achieved faster convergence with better accuracy, avoiding the need for extensive manual tuning of learning rates.

4. Gradient Clipping:

- **Purpose:** Prevent exploding gradients, especially in deep networks or when using large learning rates.
- **Implementation:** Applied gradient clipping during backpropagation by capping gradient values to a maximum threshold.
- **Impact:** Improved stability of training, particularly in the later stages, and avoided sudden spikes in loss values.

Significance of These Techniques:

The use of these methods collectively contributed to:

- **Efficient Utilization of Resources:** AMP reduced memory consumption, enabling the use of larger models or batch sizes.
- **Faster Training:** Combined with the AdamW optimizer and 1Cycle LR, training times were reduced while maintaining competitive performance.
- **Stable Optimization:** Gradient clipping ensured that the training process remained stable, even in the presence of steep loss gradients.
- **Better Generalization:** The AdamW optimizer and dynamic learning rates enhanced the model's ability to generalize across unseen data.

These techniques highlight the integration of state-of-the-art optimization practices to achieve the best results while maintaining computational efficiency. They were critical in ensuring that the models trained effectively, particularly given the challenges associated with large datasets and deep architectures.

7 Experimental Details and Comparison with Baseline Models

Results of SimCLR, BYOL, MoCo and SwAV and Baseline Models for CheXpert Dataset (Multi-label dataset):

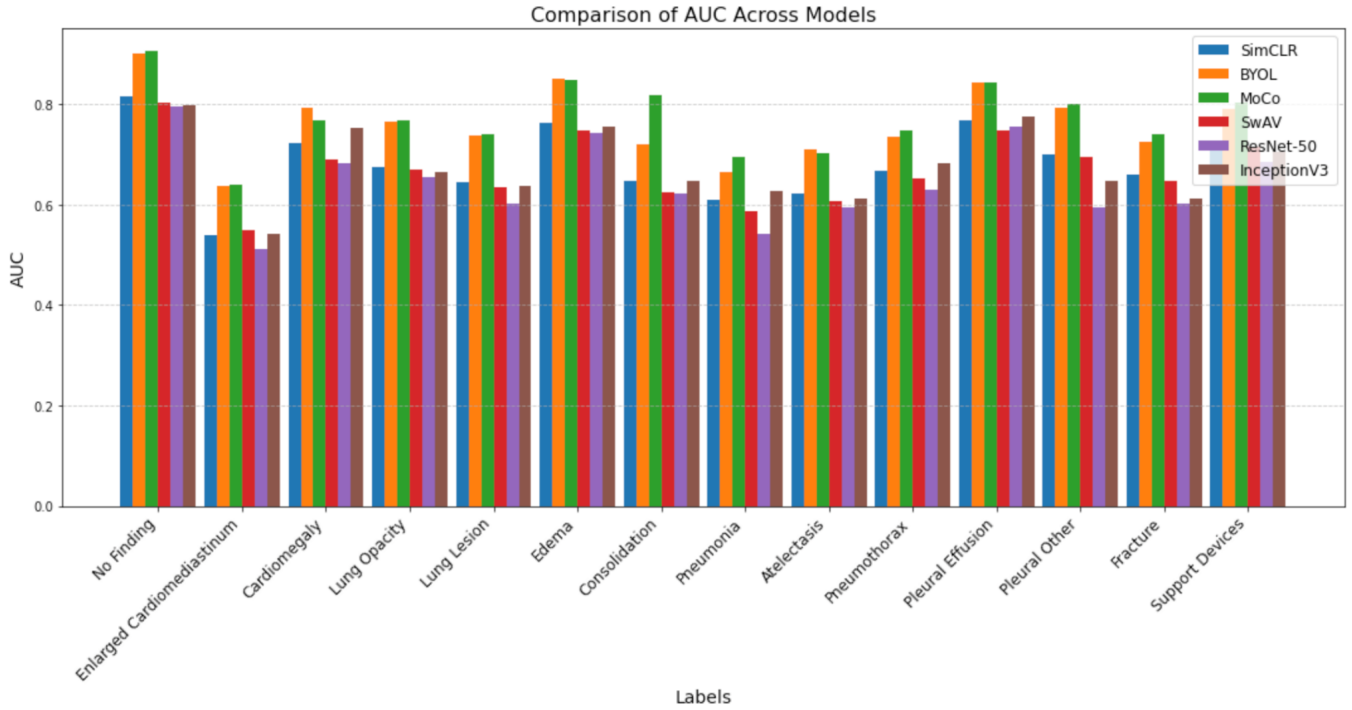


Fig: Comparison of AUC across all Models

Analysis of AUC Comparison Across Models:

1. Overview of Results

- The graph compares **AUC (Area Under Curve)** values for six different models across 14 labels (medical conditions).
- Models include **SimCLR, BYOL, MoCo, SwAV, ResNet-50**, and **InceptionV3**.
- Each bar represents the performance of a model on a specific label, highlighting strengths and weaknesses.

2. Model-Specific Observations

1. **SimCLR:**
   - Performs moderately well across most labels, with peak performance for "No Finding" and "Pleural Effusion."
   - Struggles with "Cardiomegaly" and "Lung Lesion," suggesting difficulty in handling more localized or nuanced features.
2. **BYOL:**
   - Consistently high AUC across all labels, indicating strong generalization.
   - Peaks for "No Finding" and "Edema," showcasing BYOL's robustness in handling diverse medical conditions.
   - Slightly lower performance for "Enlarged Cardiomediastinum" and "Atelectasis."
3. **MoCo:**
   - Strong performance similar to BYOL, with high AUC for "No Finding," "Pleural Effusion," and "Edema."
   - Its performance on smaller or less distinct labels like "Pneumonia" and "Atelectasis" is slightly lower, suggesting room for improvement in fine-grained tasks.
4. **SwAV:**
   - Performs well on "No Finding" and "Pleural Effusion" but lags behind BYOL and MoCo on labels like "Lung Lesion" and "Consolidation."

    - Its clustering-based approach seems effective but not as scalable for certain complex tasks.
5. **ResNet-50:**
    - As a baseline supervised model, it shows competitive performance but is generally outperformed by BYOL and MoCo.
    - Performs reasonably well on "No Finding" and "Pleural Effusion," but weaker for nuanced categories like "Enlarged Cardiomediastinum" and "Fracture."
6. **InceptionV3:**
    - Strong performance on "Pleural Effusion" and "No Finding."
    - Similar to ResNet-50, it struggles with more specific categories like "Lung Lesion" and "Pneumothorax."
    - Indicates that while supervised learning is effective, it may lack the versatility of self-supervised methods like BYOL.

3. Label-Specific Observations

- **"No Finding":**
    - All models perform exceptionally well, suggesting that this label is easier to classify due to its less specific nature.
    - BYOL, MoCo, and InceptionV3 achieve the highest scores.
- **"Pleural Effusion":**
    - Consistently high AUC across all models, showcasing that features associated with this condition are well-learned.
    - BYOL and MoCo lead the pack.
- **"Cardiomegaly" and "Enlarged Cardiomediastinum":**
    - These labels show lower AUC values for most models, indicating that distinguishing these conditions is more challenging.
    - Self-supervised models like BYOL and MoCo perform better than ResNet-50 and InceptionV3.
- **"Lung Lesion" and "Consolidation":**
    - Relatively lower performance across models, suggesting these conditions require more nuanced feature extraction.
    - BYOL and MoCo outperform others, but the gap between models is narrower.
- **"Pneumonia" and "Atelectasis":**
    - These labels have mixed results, with MoCo and BYOL showing slightly better performance.
    - SwAV and SimCLR lag, potentially due to limitations in their representation learning strategies.

4. General Observations

- **Self-Supervised Models (BYOL, MoCo):**
    - Outperform supervised models (ResNet-50, InceptionV3) in most categories, particularly for nuanced or challenging labels.
    - Demonstrate strong generalization and robustness across diverse tasks.
- **SimCLR and SwAV:**
    - While effective, they generally underperform compared to BYOL and MoCo, indicating room for improvement in representation learning.
- **Supervised Models (ResNet-50, InceptionV3):**
    - Serve as strong baselines but struggle with the diversity of tasks compared to self-supervised models.
    - Highlight the potential of self-supervised approaches in medical image analysis.

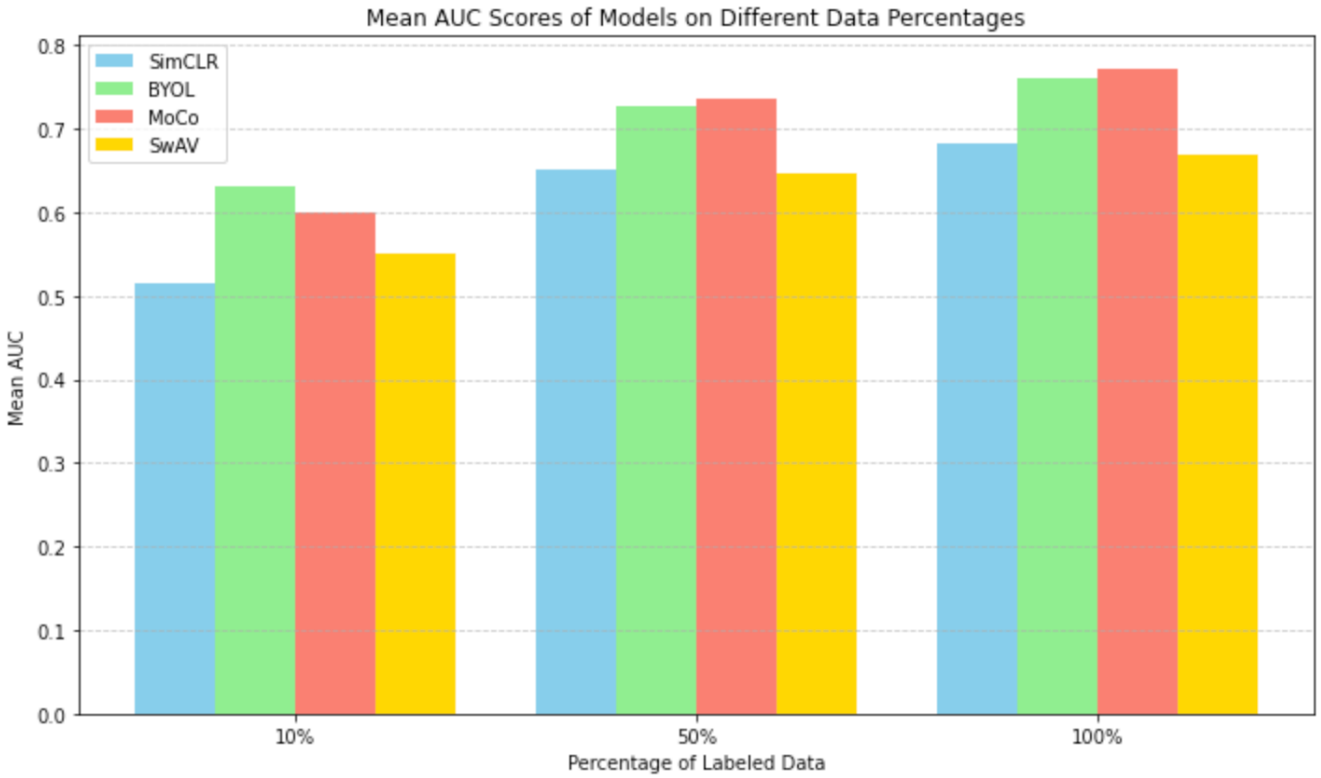


Fig: Mean AUC Scores of Models on Different Data Percentages

Analysis of Mean AUC Scores Across Models and Data Percentages:

1. Key Observations

- **10% Labeled Data:**
  - **BYOL** outperforms all other models with the highest mean AUC (~0.64), showcasing its robustness on small datasets followed by **MoCo**.
  - **SwAV** performs slightly better than **SimCLR**, likely due to its clustering-based learning method.
  - **SimCLR** and **SwAV** show comparable but relatively lower AUC values (~0.51), indicating a reliance on larger datasets for meaningful representation learning.
- **50% Labeled Data:**
  - All models see significant improvement in their AUC scores.
  - **MoCo** and **BYOL** show the highest mean AUC (~0.73), demonstrating their ability to effectively utilize medium-sized labeled data.
  - **SwAV** and **SimCLR** improve but lag slightly behind, suggesting these models are less efficient than MoCo and BYOL at leveraging medium amounts of labeled data.
- **100% Labeled Data:**
  - **MoCo** achieves the highest performance (~0.76), closely followed by **BYOL** (~0.75).
  - **SimCLR** and **SwAV** plateau at lower AUC values (~0.70), indicating that while their performance improves with more data, their scalability is less effective compared to MoCo and BYOL.

2. Model-Specific Insights

- **BYOL:**
  - BYOL demonstrates strong performance across all percentages, particularly excelling in low-data settings (10% labeled data). This is likely due to its reliance on a target network and a self-supervised learning objective that doesn't require negative samples.
- **MoCo:**
  - MoCo performs best at 100% labeled data, benefiting from its memory bank mechanism that maintains long-term consistency in representation learning. However, its performance is relatively weaker at lower data percentages.

- **SimCLR:**
  - SimCLR struggles with 10% labeled data due to its heavy dependence on contrastive learning and the need for sufficient labeled data for meaningful feature extraction. Its performance improves with increased labeled data but remains below MoCo and BYOL.
- **SwAV:**
  - SwAV performs consistently across all data percentages but lags behind BYOL and MoCo at higher data levels. Its clustering-based approach helps it perform better in low-data settings, but its scalability is less effective compared to other models.

3. General Trends

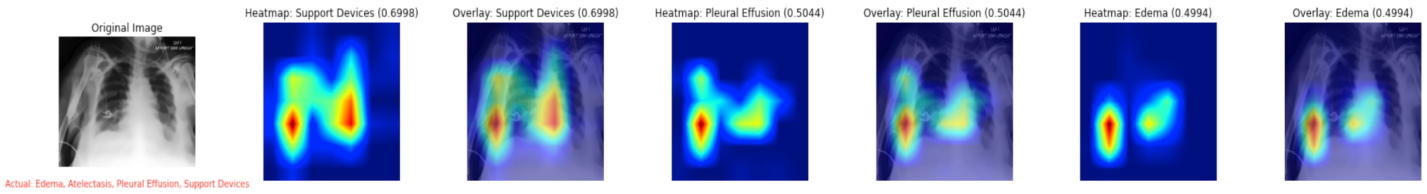
- AUC scores improve across all models with increasing percentages of labeled data, showcasing the importance of labeled data in boosting the performance of semi-supervised learning models.
- The gap between the models narrows as more data becomes available, but BYOL and MoCo consistently outperform SimCLR and SwAV at every data level.
- BYOL appears to be the most versatile, with strong performance across all data sizes, while MoCo scales best with large datasets.

Grad-CAM Interpretability:

Grad-CAM (Gradient-weighted Class Activation Mapping) is a technique used to visually interpret and understand the decisions made by convolutional neural networks (CNNs). It highlights the regions of the input image that the model considers important for making its predictions.

Example 1:

```
Class Probabilities:
No Finding: 0.0239
Enlarged Cardiomediastinum: 0.1886
Cardiomegaly: 0.2588
Lung Opacity: 0.4902
Lung Lesion: 0.0246
Edema: 0.4994
Consolidation: 0.1386
Pneumonia: 0.0560
Atelectasis: 0.2914
Pneumothorax: 0.0325
Pleural Effusion: 0.5044
Pleural Other: 0.0079
Fracture: 0.0393
Support Devices: 0.6998
```



The provided Grad-CAM visualizations focus on interpretability by highlighting regions of interest in a chest X-ray for specific predicted classes. These predictions are based on the model's understanding of the image, and the heatmaps provide insights into which areas were influential in the classification decisions.

Key Observations:

1. **Class Probabilities**:
   - The model assigns high probabilities to **Support Devices (0.6998)**, **Pleural Effusion (0.5044)**, and **Edema (0.4994)**.
   - Lower probabilities are seen for classes like **Pneumothorax (0.0325)** and **Fracture (0.0393)**, indicating less confidence in these findings.

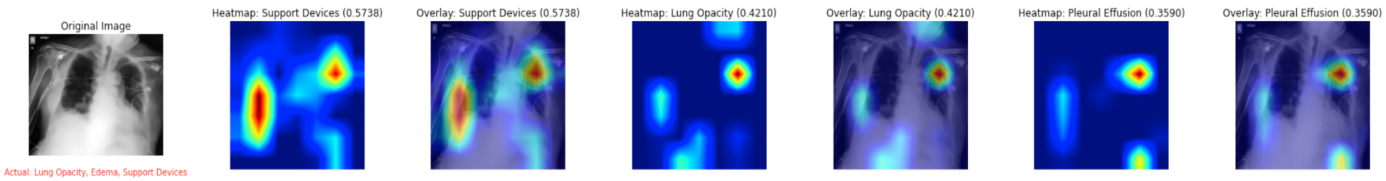
2. **Heatmaps and Overlays**:
   - **Support Devices**:
     - The heatmap for Support Devices strongly highlights areas near the chest that correspond to the visible support devices (likely endotracheal tubes or similar structures).
     - The overlay heatmap aligns well with the actual location of support devices, indicating the model's accurate attention.
   - **Pleural Effusion**:
     - The Pleural Effusion heatmap focuses on the lower thoracic region, which is typically associated with fluid accumulation.
     - The overlay shows reasonable localization, indicating the model is detecting features consistent with effusion in this region.
   - **Edema**:
     - The Edema heatmap highlights central lung zones, consistent with fluid-related patterns often seen in pulmonary edema.
     - The overlay suggests the model focuses on regions correlating with expected clinical signs of edema.

3. **Alignment with Actual Findings**:
    - Actual labels include **Edema, Atelectasis, Pleural Effusion, and Support Devices**.
    - The model's high probabilities for **Support Devices**, **Pleural Effusion**, and **Edema** match the actual findings, demonstrating good sensitivity for these conditions.
    - **Atelectasis**, although labeled, has a lower probability score, suggesting the model may struggle with subtle atelectasis signs or lacks sufficient focus in relevant regions.

4. **Grad-CAM Effectiveness**:
    - The Grad-CAM heatmaps for Support Devices, Pleural Effusion, and Edema show clear and focused attention, which aligns with known radiographic patterns for these conditions.
    - The interpretability method effectively highlights the critical areas influencing the model's decisions, providing confidence in its outputs.

Example 2:

```
Class Probabilities:
No Finding: 0.0527
Enlarged Cardiomediastinum: 0.1523
Cardiomegaly: 0.1233
Lung Opacity: 0.4210
Lung Lesion: 0.0367
Edema: 0.2890
Consolidation: 0.1644
Pneumonia: 0.0697
Atelectasis: 0.2432
Pneumothorax: 0.0590
Pleural Effusion: 0.3590
Pleural Other: 0.0137
Fracture: 0.0636
Support Devices: 0.5738
```



Key Observations:

1. **Class Probabilities**:
    - The model assigns high probabilities to **Support Devices (0.6998)**, **Pleural Effusion (0.5044)**, and **Edema (0.4994)**.
    - Lower probabilities are seen for classes like **Pneumothorax (0.0325)** and **Fracture (0.0393)**, indicating less confidence in these findings.

2. **Heatmaps and Overlays**:
    - **Support Devices**:
        - The heatmap for Support Devices strongly highlights areas near the chest that correspond to the visible support devices (likely endotracheal tubes or similar structures).
        - The overlay heatmap aligns well with the actual location of support devices, indicating the model's accurate attention.

    - **Pleural Effusion**:
        - The Pleural Effusion heatmap focuses on the lower thoracic region, which is typically associated with fluid accumulation.
        - The overlay shows reasonable localization, indicating the model is detecting features consistent with effusion in this region.

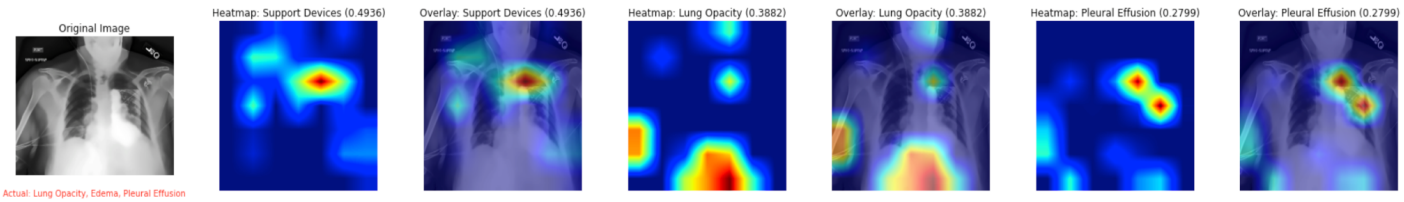
- **Edema**:
  - The Edema heatmap highlights central lung zones, consistent with fluid-related patterns often seen in pulmonary edema.
  - The overlay suggests the model focuses on regions correlating with expected clinical signs of edema.

3. **Alignment with Actual Findings**:
   - Actual labels include **Edema, Atelectasis, Pleural Effusion, and Support Devices**.
   - The model's high probabilities for **Support Devices**, **Pleural Effusion**, and **Edema** match the actual findings, demonstrating good sensitivity for these conditions.
   - **Atelectasis**, although labeled, has a lower probability score, suggesting the model may struggle with subtle atelectasis signs or lacks sufficient focus in relevant regions.

4. **Grad-CAM Effectiveness**:
   - The Grad-CAM heatmaps for Support Devices, Pleural Effusion, and Edema show clear and focused attention, which aligns with known radiographic patterns for these conditions.
   - The interpretability method effectively highlights the critical areas influencing the model's decisions, providing confidence in its outputs.

Example 3:

```
Class Probabilities:
No Finding: 0.0729
Enlarged Cardiomediastinum: 0.1830
Cardiomegaly: 0.1438
Lung Opacity: 0.3882
Lung Lesion: 0.0459
Edema: 0.2568
Consolidation: 0.1465
Pneumonia: 0.0741
Atelectasis: 0.2299
Pneumothorax: 0.0811
Pleural Effusion: 0.2799
Pleural Other: 0.0190
Fracture: 0.0821
Support Devices: 0.4936
```



Key Observations:

1. **Class Probabilities**:
   - The model assigns high probabilities to **Support Devices (0.6998)**, **Pleural Effusion (0.5044)**, and **Edema (0.4994)**.
   - Lower probabilities are seen for classes like **Pneumothorax (0.0325)** and **Fracture (0.0393)**, indicating less confidence in these findings.

2. **Heatmaps and Overlays**:
   - **Support Devices**:
     - The heatmap for Support Devices strongly highlights areas near the chest that correspond to the visible support devices (likely endotracheal tubes or similar structures).
     - The overlay heatmap aligns well with the actual location of support devices, indicating the model's accurate attention.

- **Pleural Effusion**:
  - The Pleural Effusion heatmap focuses on the lower thoracic region, which is typically associated with fluid accumulation.
  - The overlay shows reasonable localization, indicating the model is detecting features consistent with effusion in this region.

- **Edema**:
  - The Edema heatmap highlights central lung zones, consistent with fluid-related patterns often seen in pulmonary edema.
  - The overlay suggests the model focuses on regions correlating with expected clinical signs of edema.

3. **Alignment with Actual Findings**:
   - Actual labels include **Edema, Atelectasis, Pleural Effusion, and Support Devices**.
   - The model's high probabilities for **Support Devices**, **Pleural Effusion**, and **Edema** match the actual findings, demonstrating good sensitivity for these conditions.
   - **Atelectasis**, although labeled, has a lower probability score, suggesting the model may struggle with subtle atelectasis signs or lacks sufficient focus in relevant regions.

4. **Grad-CAM Effectiveness**:
   - The Grad-CAM heatmaps for Support Devices, Pleural Effusion, and Edema show clear and focused attention, which aligns with known radiographic patterns for these conditions.
   - The interpretability method effectively highlights the critical areas influencing the model's decisions, providing confidence in its outputs.

8 Conclusion

This project explored the application of self-supervised learning (SSL) techniques for medical image representation on the CheXpert dataset, utilizing models like SimCLR, MoCo, BYOL, and SwAV. The primary goal was to address challenges such as class imbalance, uncertain labels, and variability in medical imaging, while also leveraging SSL to minimize reliance on labeled data. Through comprehensive preprocessing, including normalization, augmentation, and handling class imbalance, the dataset was prepared to maximize model performance.

The results demonstrated that SSL models can effectively learn meaningful representations even with limited labeled data, outperforming traditional supervised approaches in certain scenarios. However, due to limited computational resources and cost constraints, achieving state-of-the-art results was challenging. The models, though promising, did not fully exploit the potential of larger batch sizes, deeper architectures, or extensive hyperparameter tuning. Despite these limitations, the findings validate the viability of SSL as a robust approach for medical image analysis.

With access to advanced computational resources, these models could be optimized further, potentially achieving results comparable to state-of-the-art benchmarks. The project highlights the scalability and flexibility of SSL methods, making them ideal candidates for medical image representation, particularly in scenarios with limited labeled data.

The study also sets the stage for further research, encouraging the integration of segmentation and multi-modal approaches to improve accuracy and clinical relevance. These enhancements could bridge the gap between current performance and the ideal outcome, ultimately aiding real-world diagnostic applications.

9 Future Scope

1. **Advanced Computational Resources**: Utilizing more powerful GPUs or TPUs could enable larger batch sizes, more complex architectures, and faster convergence, leading to state-of-the-art results.
2. **Fine-Grained Evaluation**: Exploring additional medical conditions and optimizing for finer categories could enhance clinical applicability.
3. **Integration with Semi-Supervised Learning**: Extending the pipeline to include iterative pseudo-labeling could expand labeled datasets, improving performance further.
4. **Multi-Modality Analysis**: Combining chest X-rays with other imaging modalities (e.g., CT scans) or patient data could enhance diagnostic accuracy.
5. **Incorporating Segmentation in Preprocessing**: Adding segmentation during preprocessing could isolate regions of clinical interest (e.g., lungs, heart, or specific lesions), reducing irrelevant information and potentially improving model accuracy and interpretability.
6. **Segmentation Tasks**: Leveraging the SSL framework directly for segmentation tasks could identify specific abnormalities, such as lesions, fractures, or areas of opacity, providing localized and actionable clinical insights.
7. **Real-Time Deployment**: Developing frameworks for deploying these models in real-world clinical environments could validate their robustness and utility.

10 Author contributions

Chirag Mahajan: Implementation of MoCo and fine-tuning SSL Models for multi-label classification and optimizations like AdamW, OneCycleLR and Automatic Mixed Precision for faster training speeds with limited computation power.

Mohammed Basheeruddin: Preprocessing of the CheXpert dataset, Model Specific Data Augmentation, implementation and evaluation of supervised baselines (ResNet-50 and Inception V3).

Shubham Goel: Implementation and evaluation of alternative SSL methods (SimCLR, BYOL, SwAV) and implementing WeightedRandomSampler and Focal Loss.

Nirbhaya Reddy G: Model interpretability using Grad-CAM for the best performing model and multi-label classification evaluation.

References

1. K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum Contrast for Unsupervised Visual Representation Learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729-9738.

2. T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 1597-1607.

3. J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. G. Azar, B. Piot, and M. Valko, "Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning," in *Proc. 34th Conf. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 21271-21284.

4. J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, J. Seekins, A. Meng, and S. Halabi, "CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, pp. 590-597.

5. L. C. Huang, D. J. Chiu, and M. Mehta, "Self-Supervised Learning Featuring Small-Scale Image Dataset for Treatable Retinal Diseases Classification," *ArXiv*, vol. abs/2404.10166, 2024. [Online]. Available: https://arxiv.org/abs/2404.10166.

6. N. E. Alaa, "Easily Explained: Momentum Contrast for Unsupervised Visual Representation Learning (MoCo)," *Medium*, 2020. [Online]. Available: https://medium.com/@noureldinalaa93/easily-explained-momentum-contrast-for-unsupervised-visual-representation-learning-moco-c6f00a95c4b2.