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 will then fine-tune 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 Datasets

We will use two datasets for this project:

a. UC Mendeley Chest X-ray Dataset

Used for preliminary results in binary pneumonia classification to demonstrate the feasibility of the approach. This smaller dataset consists of CXRs labeled as either "Normal" or "Pneumonia."

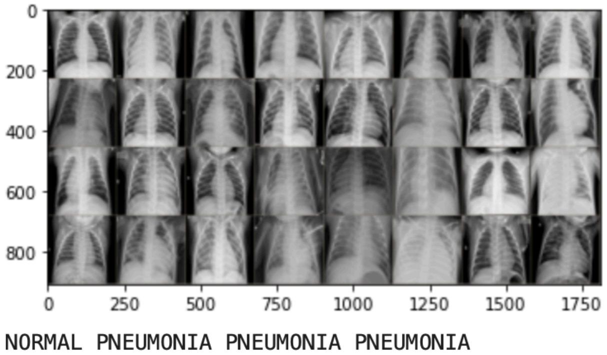


Fig. 1: UC Mendeley Chest X-ray dataset samples

b. CheXpert 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]

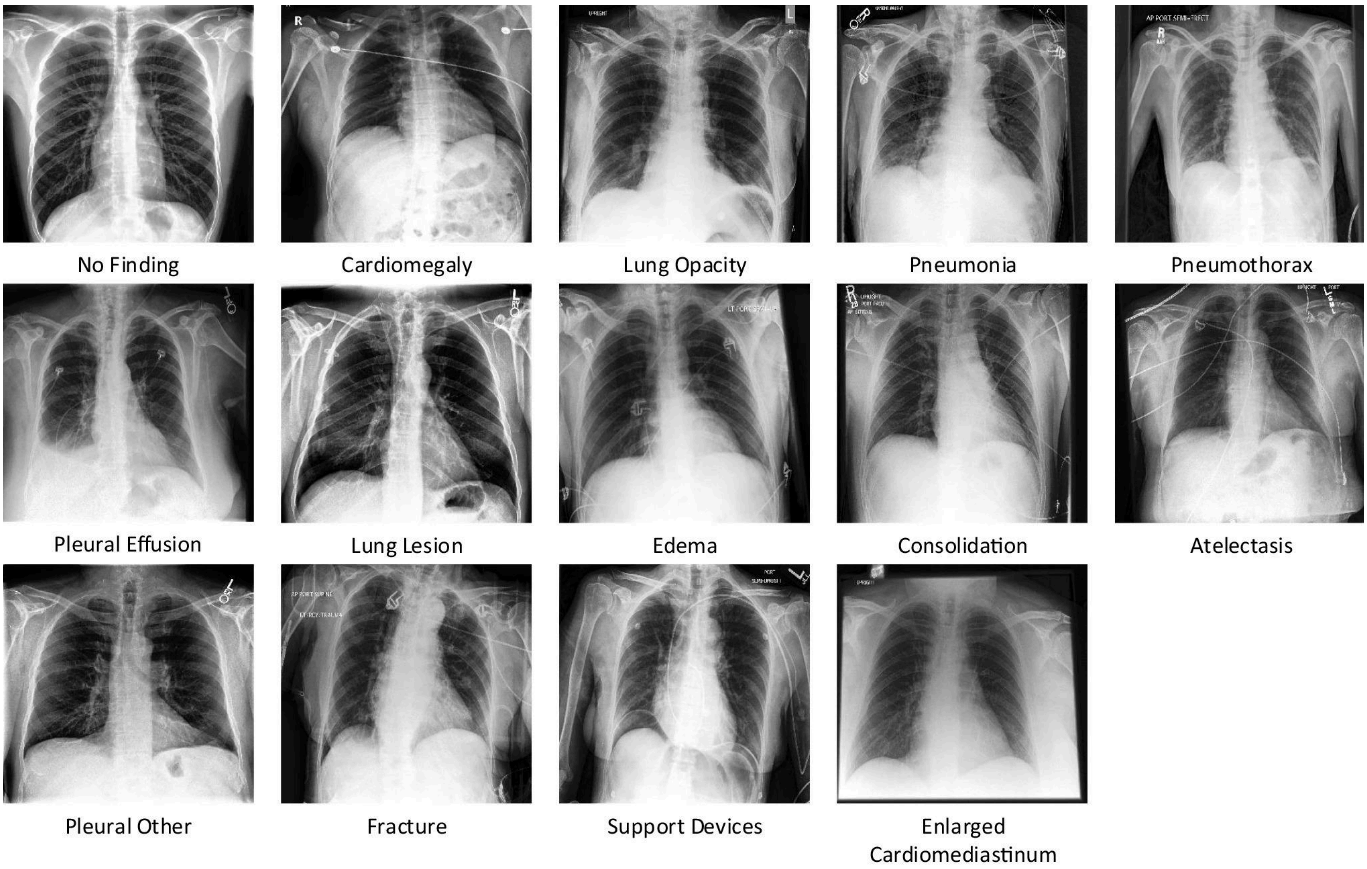


Fig. 2: CheXpert Dataset Chest X-ray dataset samples with labels

Data Characteristics

- **Uncertainty labels**: CheXpert includes three classes for each label - positive, negative, and uncertain. We will explore various techniques to handle these uncertainty labels, such as U-Ones, U-Zeros, and U-MultiClass (methods previously explored in the CheXpert paper). [4]
- **Class Imbalance**: Some pathologies (e.g., pneumonia) are less frequent than others (e.g., pleural effusion). We will use class balancing techniques or weighted loss functions to address this imbalance.
- **Preprocessing**: The images will be resized to **224x224** and normalized using ImageNet statistics. Data augmentation, such as random flipping, rotation, and brightness adjustments, will be used during training.

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]

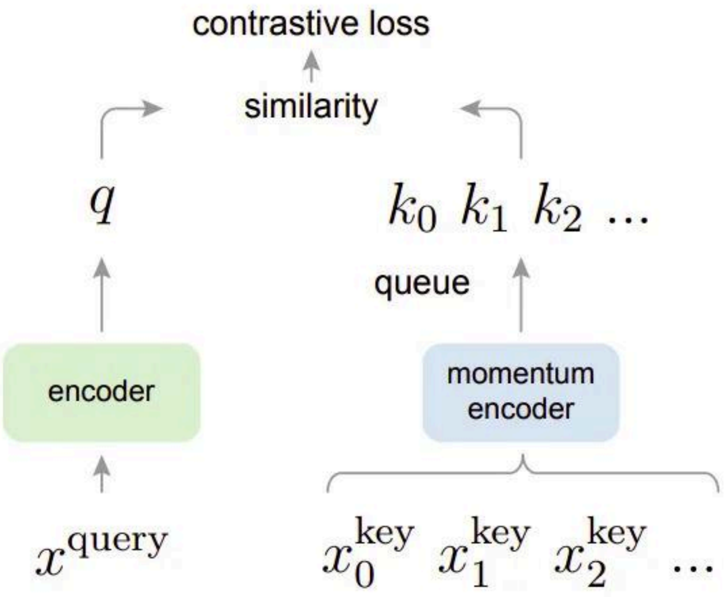


Fig. 3: MoCo Architecture with query and ky dictionary [6]

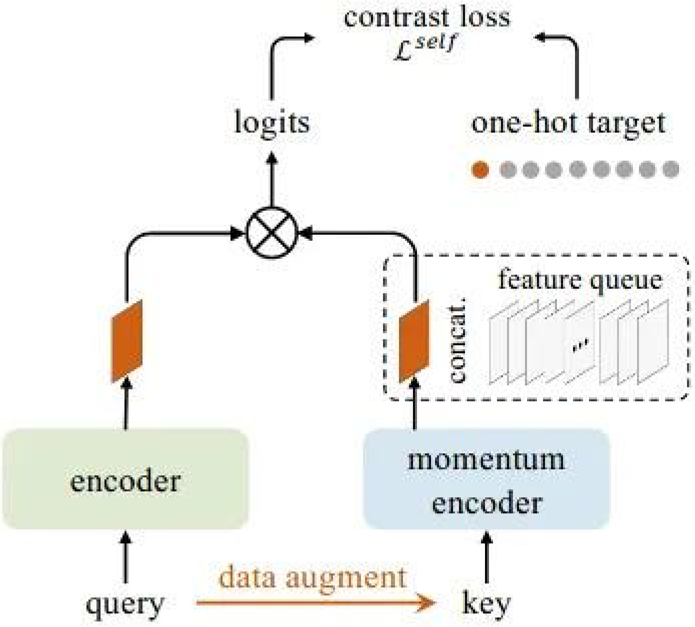


Fig. 4: MoCo Architecture in more detail [6]

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]

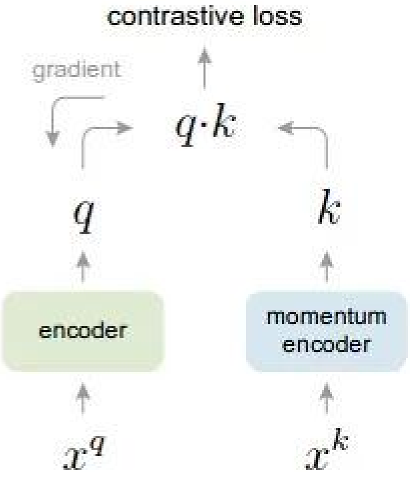


Fig. 5: Momentum Encoder Architecture with a gradient on the encoder branch only [6]

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]

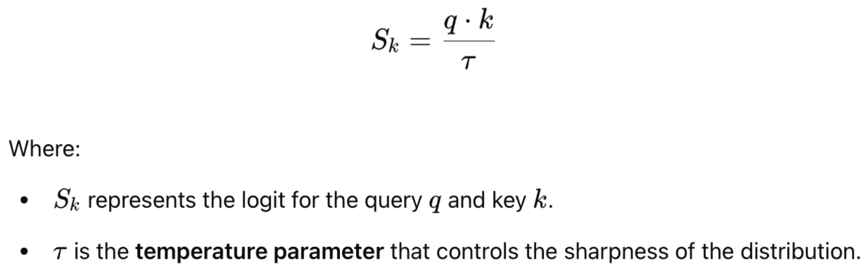
4. One-Hot Target Vector

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:



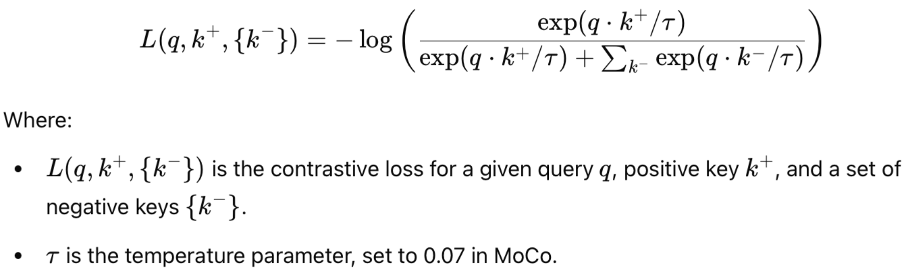
$$S_k = \frac{q \cdot k}{\tau}$$

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:

- **SimCLR**: Uses in-batch negative samples and contrastive loss, requiring large batch sizes to generate meaningful negative pairs. [4]

- **BYOL**: Avoids negative samples altogether, learning by minimizing the distance between the outputs of two differently augmented versions of the same image. [3]

- **SwAV**: A clustering-based SSL method that assigns representations to clusters and swaps the assignments between views of the same image to ensure consistency. [4]

c. Fine-Tuning for Multi-Label Classification

After pre-training on unlabeled CXR images, we will fine-tune the SSL models on the labeled CheXpert dataset for multi-label classification. 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 will compare our SSL-based models with the following supervised learning models:

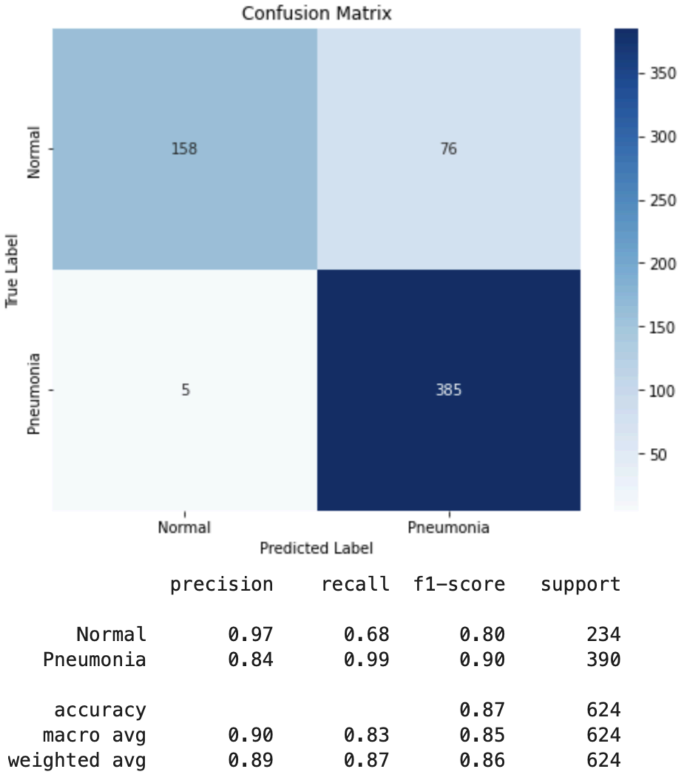
- **ResNet-50 (Supervised Learning)**: Pre-trained on ImageNet and fine-tuned on the CheXpert dataset for multi-label classification.
- **Inception V3 (Supervised Learning)**: Another supervised learning baseline, pre-trained on ImageNet and fine-tuned for multi-label classification.
- **Self-Supervised Models**: In addition to MoCo, we will compare SimCLR, BYOL, and SwAV for multi-label classification. [3], [4]

5 Preliminary Results

In our initial experiments, we evaluated three models for the task of classifying **Normal** and **Pneumonia** chest X-rays using the smaller **UC Mendeley Dataset**. The models included **MoCo (Momentum Contrast)**, **ResNet-50**, and **Inception V3**.

The evaluation metrics (Accuracy, Precision, Recall, F1-Score, and ROC-AUC) were used to assess each model's performance.

MoCo (Momentum Contrast):



```
              precision    recall  f1-score   support

      Normal       0.97      0.68      0.80       234
   Pneumonia       0.84      0.99      0.90       390

    accuracy                           0.87       624
   macro avg       0.90      0.83      0.85       624
weighted avg       0.89      0.87      0.86       624
```

Fig. 6: Confusion Matrix for MoCo

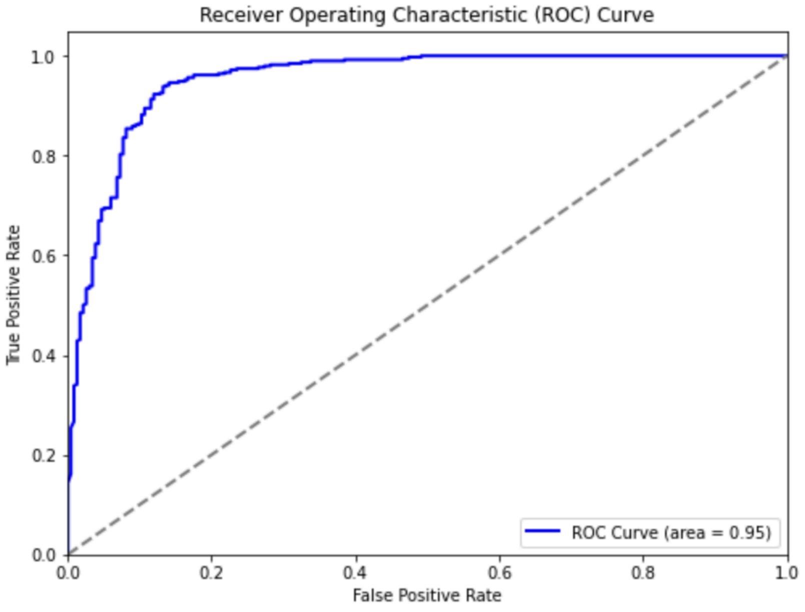
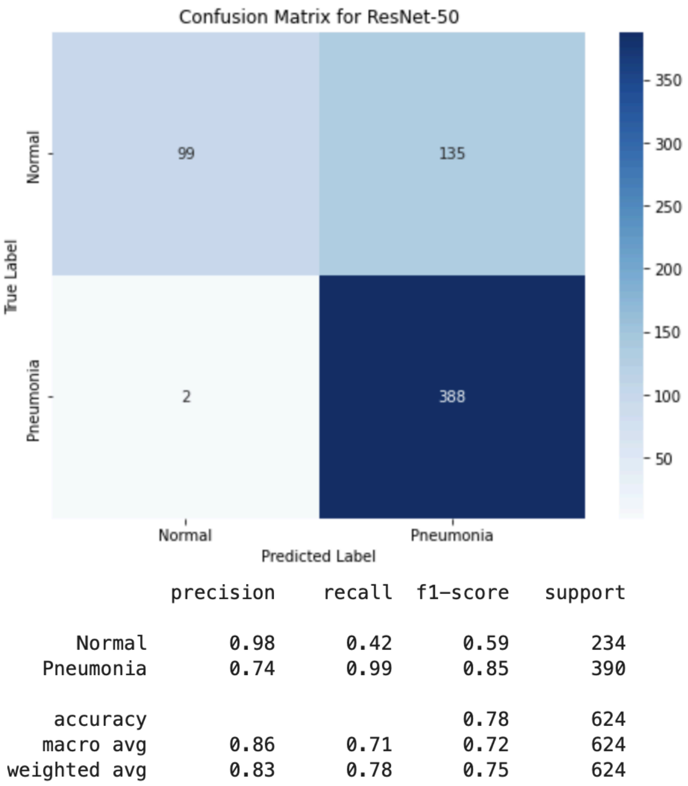


Fig. 7: AUC-ROC Curve for MoCo

- **Test Accuracy**: 87.02%
- **ROC-AUC Score**: 0.9605
- **Confusion Matrix**:
  - **True Positives for Normal**: 161
  - **False Negatives for Normal**: 73
  - **True Positives for Pneumonia**: 382
  - **False Negatives for Pneumonia**: 8
- **Normal**:
  - Precision: 0.95
  - Recall: 0.69
  - F1-Score: 0.80
- **Pneumonia**:
  - Precision: 0.84
  - Recall: 0.98
  - F1-Score: 0.90
- **Conclusion**: MoCo demonstrated high performance in detecting both **Normal** and **Pneumonia** cases, with the highest accuracy and a well-balanced precision, recall, and F1-score for both classes.

ResNet-50:



```
              precision    recall  f1-score   support

      Normal       0.98      0.42      0.59       234
   Pneumonia       0.74      0.99      0.85       390

    accuracy                           0.78       624
   macro avg       0.86      0.71      0.72       624
weighted avg       0.83      0.78      0.75       624
```

Fig. 8: Confusion Matrix for ResNet-50

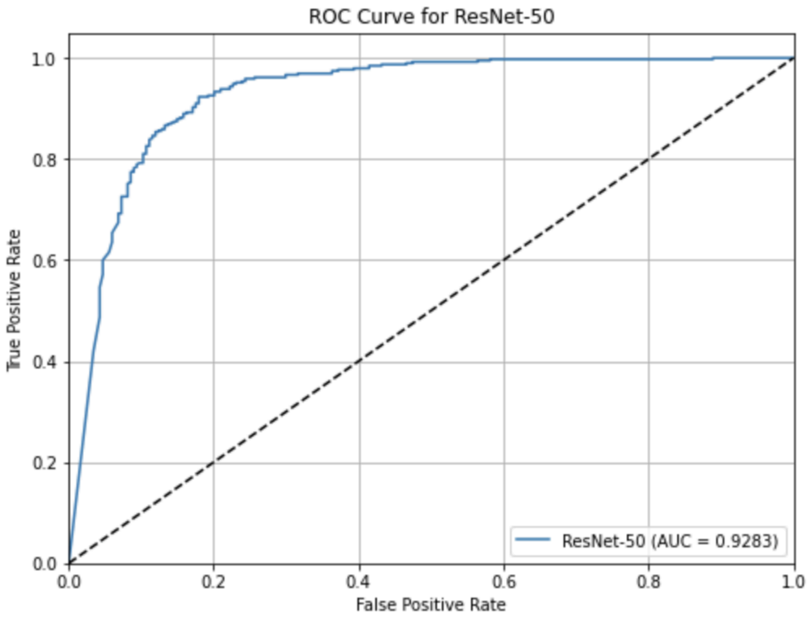
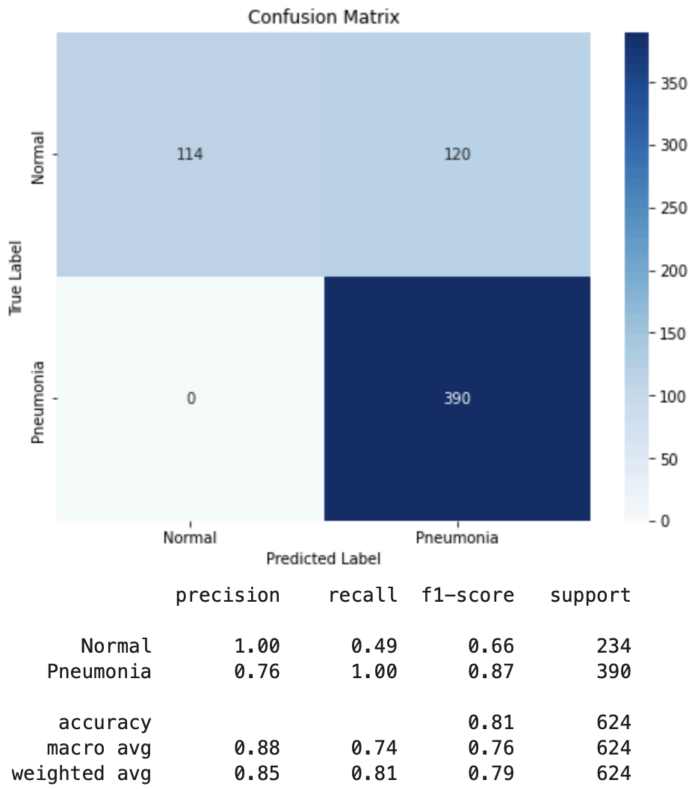


Fig. : AUC-ROC Curve for ResNet-50

- **Test Accuracy**: 78.04%
- **ROC-AUC Score**: 0.9283
- **Confusion Matrix**:
  - **True Positives for Normal**: 99
  - **False Negatives for Normal**: 135
  - **True Positives for Pneumonia**: 388
  - **False Negatives for Pneumonia**: 2
- **Normal**:
  - Precision: 0.98
  - Recall: 0.42
  - F1-Score: 0.59
- **Pneumonia**:
  - Precision: 0.74
  - Recall: 0.99
  - F1-Score: 0.85
- **Conclusion**: ResNet-50 struggled to detect **Normal** cases, reflected in a low recall of 0.42, while it performed strongly on **Pneumonia** detection with a recall of 0.99. The overall accuracy of 78.04% was lower compared to MoCo.

Inception V3:



```
              precision    recall  f1-score   support

      Normal       1.00      0.49      0.66       234
   Pneumonia       0.76      1.00      0.87       390

    accuracy                           0.81       624
   macro avg       0.88      0.74      0.76       624
weighted avg       0.85      0.81      0.79       624
```

Fig. 10: Confusion Matrix for Inception V3

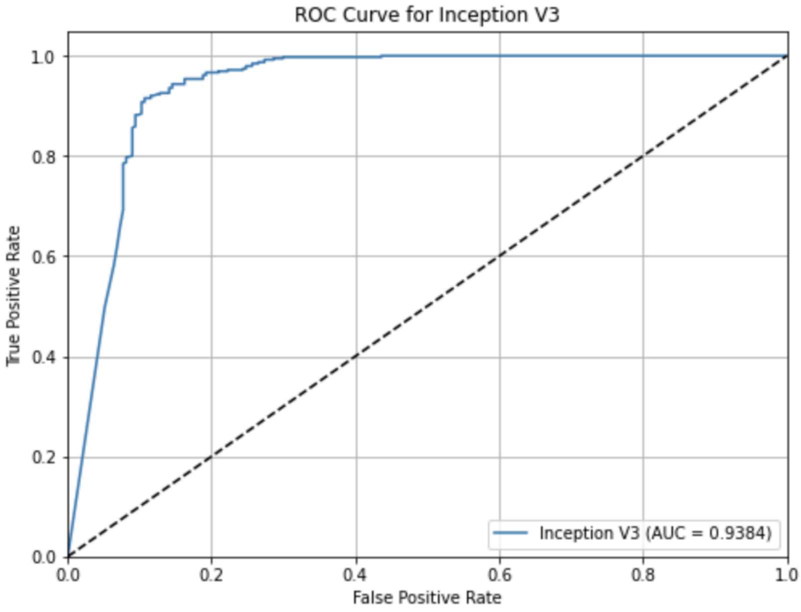


Fig. 11: AUC-ROC Curve for Inception V3

- **Test Accuracy**: 80.77%
- **ROC-AUC Score**: 0.9384
- **Confusion Matrix**:
  - **True Positives for Normal**: 114
  - **False Negatives for Normal**: 120
  - **True Positives for Pneumonia**: 390
  - **False Negatives for Pneumonia**: 0
- **Normal**:
  - Precision: 1.00
  - Recall: 0.49
  - F1-Score: 0.66

- **Pneumonia**:
  - Precision: 0.76
  - Recall: 1.00
  - F1-Score: 0.87
- **Conclusion**: Inception V3 had perfect recall for **Pneumonia** cases but struggled to detect **Normal** cases accurately, with a recall of 0.49. This caused a high number of normal cases being misclassified as pneumonia, as shown in the confusion matrix.

Model	Accuracy	ROC-AUC	Normal Recall	Pneumonia Recall	Normal F1-Score	Pneumonia F1-Score
ResNet-50	78.04%	0.9283	0.42	0.99	0.59	0.85
MoCo	87.02%	0.9605	0.68	0.99	0.80	0.90
Inception V3	80.77%	0.9384	0.49	1.00	0.66	0.87

Table 1: Comparison of ResNet-50, MoCo and Inception V3

MoCo outperformed both ResNet-50 and Inception V3 in terms of overall accuracy and balanced performance across both classes. **ResNet-50** struggled significantly in detecting normal cases, while **Inception V3** showed high precision but lower recall for **Normal** cases. MoCo provided the best balance between precision and recall for both classes.

6 Final project plan

The preliminary results indicate that **MoCo** provided the most balanced performance in classifying both **Normal** and **Pneumonia** cases. It had the highest accuracy and ROC-AUC, outperforming both **ResNet-50** and **Inception V3**. Based on these findings, we plan to proceed with MoCo as the primary model for the final project.

The insights gained from the experiments will shape our final project as follows:

1. **Dataset Transition**:
   - The preliminary experiments were conducted on the **UC Mendeley dataset** due to its smaller size, but for the final project, we will transition to the larger **CheXpert dataset**, which includes more diverse labels (such as Edema, Cardiomegaly, Atelectasis, etc.). We will extend the model's evaluation to these additional labels, making the classification task more complex. [1]
2. **Self-Supervised Learning Models**:
   - Apart from **MoCo**, we will also evaluate other **Self-Supervised Learning (SSL)** techniques such as **SimCLR**, **SwAV**, and **BYOL** for comparison. These methods are known for their strengths in

representation learning without requiring labeled data. The objective is to determine which SSL technique performs best in the medical domain, particularly in chest X-ray classification. [4]

3. **Evaluation of All Labels**:
    - While our preliminary results focused solely on **Pneumonia** vs. **Normal**, the final project will extend to evaluating multiple medical conditions in the **CheXpert dataset**. This will allow for a broader analysis of how well SSL techniques generalize across various conditions.

4. **Hyperparameter Tuning**:
    - We will continue to fine-tune hyperparameters (learning rate, batch size, momentum, and temperature) for MoCo and the other SSL models (SimCLR, SwAV, BYOL) to further optimize their performance on the **CheXpert dataset**.

5. **Comparative Study**:
    - The final project will include a comprehensive comparison between **Supervised Learning** techniques (ResNet-50, Inception V3) and **Self-Supervised Learning** techniques (MoCo, SimCLR, SwAV, BYOL). We will compare the models using evaluation metrics such as **Accuracy**, **ROC-AUC**, **Precision**, **Recall**, and **F1-Score**.

6. **Visualization and Interpretability**:
    - We will incorporate **Grad-CAM** visualizations to improve interpretability of the model's predictions, helping us understand which regions of the chest X-rays are most important for the model's decision-making process. [4]

7. **Model Generalization**:
    - We will assess the models' performance in a **low-data regime** to evaluate how well they generalize with limited labeled data. This is particularly relevant in the medical domain, where obtaining labeled data can be challenging.

8. **Timeline**:
    - The project will be completed in the next two months, allowing time for training and evaluating multiple SSL models, integrating interpretability methods, and preparing a comparative analysis across labels in the CheXpert dataset.

By leveraging the findings from our preliminary results and extending them to more complex tasks with the **CheXpert dataset**, we aim to provide valuable insights into the applicability of **Self-Supervised Learning** techniques in medical imaging.

Approach for the Final Project

Given the insights from the preliminary experiments, the final project will focus on:

1. **Multi-Label Classification on CheXpert Dataset**:
   In the final phase, we will transition from binary classification to multi-label classification using the **CheXpert dataset**. The dataset includes labels for 14 different pathologies, and we will adapt the MoCo architecture (and alternative SSL methods) to handle multiple labels per image. This transition will require:

    - **Handling uncertainty labels**: CheXpert includes "uncertain" labels in addition to positive and negative ones. We will experiment with methods like **U-Ones, U-Zeros**, and **U-MultiClass** to handle these uncertainty labels effectively.

    - **Adapting loss functions**: Instead of using binary cross-entropy, we will implement a **multi-label loss function** (e.g., binary cross-entropy with logits) that supports multi-pathology classification.

2. **Further Optimization of MoCo**:
   To improve MoCo's performance, we plan to:

   ○ **Tune the temperature parameter (ττ)**: In the preliminary experiments, we used a default value of 0.07. In the final phase, we will experiment with smaller and larger values to find the optimal temperature for separating positive and negative samples.

   ○ **Adjust the queue size**: The size of the **feature queue** plays a crucial role in the performance of contrastive learning. We plan to experiment with different queue sizes to ensure that MoCo has enough negative samples to learn from, while balancing memory and computational efficiency.

   ○ **Increase training epochs**: We will scale up training to more epochs as the CheXpert dataset is larger and richer in terms of pathology diversity. We hypothesize that a longer training time will allow MoCo to refine its learned representations further.

3. **Comparison with Other SSL Methods (SimCLR, BYOL, SwAV)**:
   Alongside MoCo, we will implement and evaluate **SimCLR, BYOL, and SwAV** as alternative self-supervised learning techniques. Each of these methods approaches contrastive learning differently (e.g., SimCLR requires large batches for negative sampling, BYOL avoids negative samples altogether), and we will compare their performance on the CheXpert dataset. This comparison will help us understand which SSL method is best suited for medical image classification, specifically in the context of multi-label classification. [4]

4. **Fine-Tuning and Transfer Learning**:
   After pre-training the models with SSL methods, we will fine-tune them on the labeled portion of the CheXpert dataset. Fine-tuning allows us to leverage the representations learned during self-supervised pre-training and adapt them to the specific downstream task of multi-label classification. We will use techniques like **early stopping** and **learning rate schedules** to ensure that fine-tuning is efficient and avoids overfitting.

5. **Interpretability with Grad-CAM**:
   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 both the SSL models (e.g., MoCo, SimCLR) and the supervised baselines (ResNet-50, Inception V3), allowing for a comparison of interpretability between SSL and traditional models. [4]

6. **Multi-Label Evaluation Metrics**:
   We will evaluate the performance of our models using metrics that are suitable for multi-label classification tasks:

   ○ **Accuracy**: To measure overall prediction accuracy across all labels.

   ○ **ROC-AUC (per class)**: To evaluate the model's ability to distinguish between positive and negative samples for each disease class.

   ○ **Precision, Recall, and F1-Score**: To assess how well the models balance false positives and false negatives.

   ○ **Micro and Macro Averages**: Since CheXpert is a highly imbalanced dataset, we will calculate micro- and macro-averaged versions of the evaluation metrics to understand both overall performance and performance on individual disease classes.

7 Author contributions

Chirag Mahajan: Implementation of MoCo and fine-tuning for multi-label classification, hyperparameter tuning.

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

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

Nirbhaya Reddy G: Model interpretability using Grad-CAM and multi-label classification evaluation.

8 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.