FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Self-Supervised Learning for Medical Image Classification: A Study on MoCo-CXR

Hugo Miguel Monteiro Guimarães

Master in Informatics and Computing Engineering

Supervisor: Luís Filipe Teixeira

Co-Supervisor: Isabel Rio-Torto

July 27, 2023

Self-Supervised Learning for Medical Image Classification: A Study on MoCo-CXR

Hugo Miguel Monteiro Guimarães

Master in Informatics and Computing Engineering

July 27, 2023

Abstract

The success of Artificial Intelligence (AI) and Deep Learning (DL) is challenging the traditional medical image analysis paradigm. Most works in this area encompass Supervised Learning (SL) approaches, which require large amounts of labeled datasets to teach models how to find the function that best maps the inputs into the desired outputs. The current trend is to use massive networks with increasing layers to achieve better performance, which requires large datasets to avoid overfitting.

Even though there are high amounts of available medical imaging exams, they lack annotations. Furthermore, labeling all images can be cumbersome and expensive, especially in medical scenarios, since it would require an expert.

Self-Supervised Learning (SSL) is a promising approach to efficiently learn visual representations without needing labeled data. Unlike SL, it mainly benefits from image characteristics, such as texture, position, and color, automatically generating a label from the data. Although already thoroughly explored in generic computer vision scenarios, SSL is still largely unexplored in medical imaging and computer-aided diagnosis.

To address these issues, we studied MoCo-CXR to identify and mitigate any shortcomings associated with the algorithm. MoCo-CXR, an adaptation of the momentum contrast (MoCo) method, was explicitly designed for learning representations from chest radiographs in the CheXpert dataset. By employing contrastive learning, MoCo-CXR extracts meaningful representations from a vast amount of unlabelled data. Our research involved adapting the pretraining phase of MoCo-CXR and integrating it with downstream image classification using the linear and finetuning approaches proposed by *Anton et al.*

Seven experiments were performed, initially focusing on the "Pleural Effusion" pathology. Subsequent experiments extended the study to the remaining observations. We evaluated the impact of MoCo-CXR pretraining augmentations, batch sizes, and learning steps on downstream image classification. Notably, we found that increasing the number of steps to 100k in finetuning significantly improved accuracy by 4.41%. The influence of multiple labels was also investigated, with experiments assessing individual pathology classification, multilabel classification, and evaluation on a test set annotated by radiologists.

Resumo

O sucesso da inteligência artificial e do DL está a desafiar o paradigma tradicional de análise de imagens médicas. A maioria dos trabalhos nesta área abrange abordagens de aprendizagem supervisionada, que exigem grandes quantidades de conjuntos de dados anotados para ensinar modelos como encontrar a função que melhor mapeia as entradas nos resultados desejados. A tendência atual é usar redes massivas com camadas cada vez maiores para alcançar melhor desempenho, o que exige grandes conjuntos de dados para evitar *overfitting*.

Apesar de existirem grandes quantidades de exames de imagens médicas disponíveis, não possuem anotações. Além disso, anotar todas as imagens pode ser uma tarefa morosa e dispendiosa, especialmente em cenários médicos, considerando a necessidade de um especialista.

A Self-Supervised Learning (SSL) é uma abordagem promissora para aprender representações visuais eficientemente sem a necessidade de dados com anotações. Ao contrário da aprendizagem supervisionada, a SSL beneficia principalmente das características das imagem, como textura, posição e cor, gerando automaticamente uma anotação a partir dos dados. Apesar de já ser amplamente explorada em cenários genéricos de visão por computador, a SSL ainda está pouco explorada em imagens médicas e diagnóstico assistido por computador.

Para resolver essas questões, realizámos um estudo sobre MoCo-CXR, com o objetivo de identificar e mitigar quaisquer desvantagens associadas ao algoritmo. MoCo-CXR, uma adaptação do método MoCo, foi desenhado especificamente para aprender representações de raio-x ao tórax no dataset CheXpert. Aplicando aprendizagem contrastiva, MoCo-CXR extrai o significado de representações de uma vasta quantidade de dados não anotados. Esta pesquisa envolveu a adaptação da fase de pré-treino do MoCo-CXR, integrando a abordagem de classificação de imagem proposta por *Anton et al.* através das metodologas *linear* e *finetuning*.

Este estudo envolve a realização de 7 experiências, com o foco inicial na patologia "Pleural Effusion", embora experiências subsequentes estendam o estudo para as restantes patologias. De seguida, avaliámos de diferentes pré-treinos, *batch sizes*, e número de passos na classificação através de *finetuning*. A descoberta mais relevante está relacionada com o aumento do número de passos de *finetuning* para 100k, resultando numa melhoria de 4.41% na *accuracy*. Também foi estudada a influência de diferentes anotações, através de experiências que avaliaram a classificação de cada patologia individualmente, assim como a classificação de multiplas anotações em simultâneo, terminando por avaliar os dados num conjunto de teste manualmente anotado por radiologistas, simulando o desempenho do modelo num cenário real.

Acknowledgments

I would like to thank the following people who have provided their invaluable assistance in carrying out this research: my supervisors Luís Filipe Teixeira and Isabel Rio-Torto, who guided me through every step of this project, always being there to help.

Hugo Miguel Monteiro Guimarães

"Some of the worst mistakes in my life were haircuts"

Jim Morrison

Contents

List of Figures

List of Tables

Abbreviations

AI Artificial Intelligence. i

AUC Area Under the ROC Curve. 6, 11, 12

CNN Convolutional Neural Network. 1, 9

CV Computer Vision. 1, 7, 37

DL Deep Learning. i, iii, 1, 3, 37

EMA Exponential Moving Average. 12

FN False Negative. 6

FP False Positive. 6

GAN Generative Adversarial Networks. 9, 10

MICLe Multi-Instance Contrastive Learning. xi, 11, 12, 15, 37

MLP Multi-Layer Perceptron. 9

NCE Noise Contrastive Estimation. 8

NLP Natural Language Processing. 7, 9

SGD Stochastic Gradient Descent. 19

SKD Self-Knowledge Distillation. 12

SL Supervised Learning. i, 1, 3, 10, 37

SSL Self-Supervised Learning. i, iii, xi, 1–7, 9–15, 17, 20, 24, 37

TN True Negative. 6

TP True Positive. 6

Chapter 1

Introduction

1.1 Context and Motivation

DL has become one of the main components in many intelligent systems worldwide. The medical scenario is no different. The ability to learn patterns from the vast amount of data available has made Convolutional Neural Networks (CNNs) a compelling approach to Computer Vision (CV) tasks such as image classification. However, most works in this area encompass SL approaches, whose labeling requirement has made it reach its bottleneck [27] due to the intense labor and cost required for the manual annotation of millions of data samples [21; 36]. In the medical image scenario, high amounts of unlabeled data are constantly being added to the medical workflow, generated from exams performed on millions of patients; thus, data is being produced faster than it is humanly possible to provide annotations, pushing researchers to find alternative approaches to leverage the existing unlabelled data.

This is where self-supervised methods emerged, becoming a viable alternative due to their capabilities to learn features from the data and provide supervision tasks without needing labeled datasets, producing results comparable with the state-of-the-art SL approaches [21].

1.2 Objectives

This study aims to perform an in-depth analysis of MoCo-CXR, [35] an adaptation of the Momentum Contrast [17] (MoCo) algorithm designed to create models able to learn meaningful representations from CheXpert [20], a large dataset of unlabelled chest radiographs. The objective is to identify and, if possible, suggest solutions for the limitations of the experiments performed on this SSL algorithm. To achieve this, we will investigate how various characteristics of the algorithm affect the performance of the model on an image classification downstream task, in the two scenarios proposed by [2]: linear probing and finetuning.

1.3 Structure

Besides the Introduction, this dissertation contains 4 chapters:

Chapter 2 describes the literature review, where several essential concepts associated with the topic are detailed, introducing the reader to technical terms, as well as the typical steps involved in an SSL approach.

Chapter 3 describes the architecture and methodology used in the project, with a description of the selected dataset as well as the experiments performed.

Chapter 4 presents and discusses the obtained results.

Chapter 5 summarizes the key findings and insights derived from the conducted study, providing a comprehensive understanding of the importance of the investigation.

Chapter 2

Literature Review

Despite the extensive research on SSL techniques in computer vision, their application in the medical image classification domain remains largely uncharted. Hence, this section describes a literature review of existing SSL approaches for image classification, specifically exploring their use in medical scenarios and determining the most effective methodologies for these applications.

2.1 Self-Supervised Learning

SL is by far the most widely known and used DL paradigm. It can be split into regression or classification, both methods learning to find a computational model that can predict labels of data not previously seen in a training phase [1]. To achieve better performance and avoid overfitting, these approaches use massive networks with several deep layers, requiring large datasets with labeled data, whose collection can be expensive, non-trivial, and time-consuming [21; 36].

SSL has recently gained attention as a way to effectively learn visual representations without needing labeled data. In this approach, annotations are automatically generated from the unlabelled data and used as pseudo-labels in a supervised way, presenting itself as a promising solution to mitigate issues associated with labeled datasets in a supervised scenario. Besides, this approach is particularly relevant in the medical image scenario as it streamlines the workflow process, from data generation following a medical exam to the point where the data is prepared for use by machine learning models, eliminating the costly step of manual labeling.

2.2 Pipeline

SSL algorithms are comprised of two main stages, as described in Figure 2.1; first, we pretrain the model to learn relevant features in a pretext task. This task aims to automatically generate labels from the data without human assistance so traditional supervised strategies can be employed. Subsequently, a downstream task is employed with the knowledge transferred from the pretext

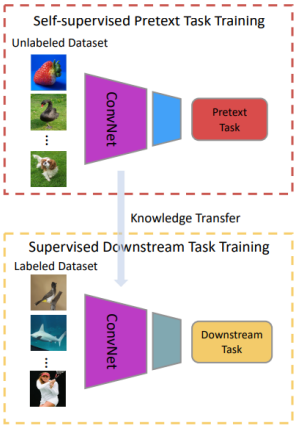


Figure 2.1: Typical SSL pipeline [22].

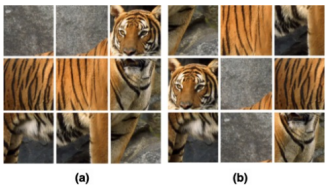


Figure 2.2: Solving a jigsaw puzzle being used as a pretext task to learn representation [21]. (a) Original Image. (b) Reshuffled image.

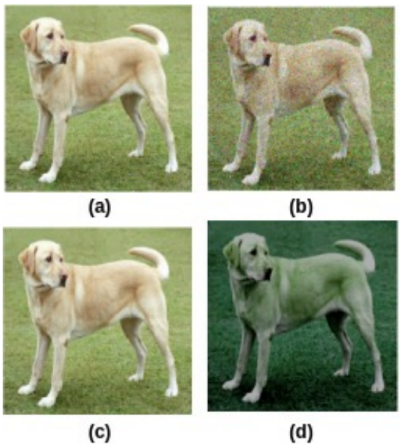


Figure 2.3: Color Transformation as pretext task [7]. (a) Original. (b) Gaussian noise. (c) Gaussian blur. (d) Color distortion [21].

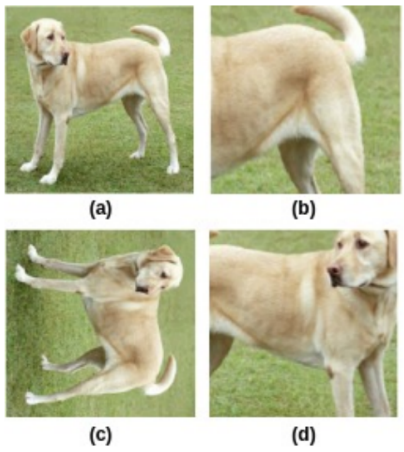


Figure 2.4: Geometric transformation as pretext task [7]. (a) Original. (b) Crop and resize. (c) Rotation. (d) Crop, resize and flip [21].

task to perform the final training and finetuning. In the words of *Li et al.* [24], "The art of self-supervised learning primarily lies in defining proper objectives for unlabelled data.", highlighting the importance of using the pretext tasks related to the downstream task.

Pretext tasks enable a model to learn visual representations and patterns in data, often using networks with deeper and more complex layers, requiring larger datasets to learn the features. Pretext tasks are not the final objective of self-supervision, positioning themselves as a pretraining task to be later passed down to a downstream task.

Figure 2.2, 2.3, and 2.4 exemplify different pretext tasks, displaying how other visual representations can be taught to the model. Figure 2.3 describes the color transformation task by which the model learns to identify what color the dog should be. Figure 2.2 contains a puzzle image, making the model grab the idea of what the form of a tiger should be, as well as the relationship between the different body parts of the tiger. Finally, a rotation task is applied in Figure 2.4 to make the model learn the orientation of the dog image.

Downstream Tasks are the final objective that leverages the visual features learned in the pretext tasks, operating similarly to supervised methods but requiring smaller labeled datasets [22], being used to evaluate and finetune the features learned in the previous phase. This dissertation focuses on the specific downstream task of image classification.

2.3 Backbones

In SSL, "backbones" refers to the fundamental architecture models employed in neural networks for extracting feature representations from input data. This architecture consists of multiple convolutional layers responsible for processing the input data and transforming it into a more meaningful representation able to capture relevant information for a given task, being initially pretrained on a large dataset without labels to learn generic and high-quality representations, such as relevant patterns, textures, and shapes that can be transferred and finetuned for the image classification downstream task. Several SSL backbones have been proposed and successfully applied to computer vision tasks, namely image classification in the medical image scenario, such as:

- ResNet (Residual Neural Network) - It introduces skip connections, which allow the training of very deep networks. Architectures like ResNet18, ResNet50, or even deeper networks are commonly used for frameworks like SimCLR, [7] MoCo [17], and BYOL, [16] including the medical image scenario.

- Densenet (Densely Connected Convolutional Network) - Contains a dense block where each layer receives the feature map from all preceding layers, improving the flow of information, encouraging feature propagation, and reducing the number of parameters compared to traditional convolutional networks. DenseNet121 is a commonly used backbone with a deep network.

These backbone architectures are combined with SSL methods that learn representations from unlabeled data. The most commonly used frameworks for effective feature learning without explicit labels are:

- SimCLR [7] (Simple Contrastive Learning) - Aims to learn meaningful representations by maximizing the similarity between different views of the same image and minimizing the similarity between views of different images.

- BYOL [16] (Bootstrap Your Own Latent) - Utilizes a target network that provides a moving average of the model's weights and a prediction network that attempts to predict the representation produced by the target network, thus learning meaningful representations.

- MoCo [17] (Momentum Contrast) - Contrastive learning frameworks create many negative pairs for each positive pair, which can be computationally expensive and impractical for large-scale datasets. MoCo [17] addresses this limitation by introducing a memory bank

serving as a large queue of negative samples while enabling a large and consistent dictionary for learning visual representations.

- SwAV [5] (Swapped and Shared Representations) - Clustering-based representation learning framework that encourages the emergence of semantically meaningful features by forcing the model to identify which augmentations correspond to the same underlying image after swapping weights and augmentations across different views.

2.4 Evaluation

To achieve adequate performance in a self-supervised scenario, it is instrumental to first understand what evaluation metrics will be used. Since the studied SSL approach is tailored for a downstream task of image classification, the evaluation metrics are similar to those used in a supervised environment: accuracy, F1-score, which can be better visualized by plotting a confusion matrix, and the Area Under the ROC Curve (AUC). Accuracy measures, as a percentage, the relationship between the number of correct predictions and the total number of predictions, as described in the following representation:

$$\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} \tag{2.1}$$

- True Positive (TP) - Number of instances where the model correctly predicts the positive class as positive.
- True Negative (TN) - Number of instances where the model correctly predicts the negative class as negative.
- False Positive (FP) - Number of instances where the model incorrectly predicts the positive class when the actual true label is negative.
- False Negative (FN) - Number of instances where the model incorrectly predicts the negative class when the actual true label is positive.

Additionally, the F1-score is defined as the harmonic mean of precision and recall. Its formula can have different weights, with one of the most common measures being the F1-score, calculated with the ensuing formula:

$$F_1 = \frac{TP}{TP + \frac{FP+FN}{2}} \tag{2.2}$$

All mentioned metrics aim to reach the numeric value one, measuring how well the model distinguishes between separate classes. The most commonly used metric to evaluate performance in the medical self-supervised scenario is the F1-score, even though the remaining metrics are also calculated and considered for evaluation.

2.5 Architecture

According to *Liu et al.* [27], self-supervision can be summarized into three main categories, as described in Figure 2.5.

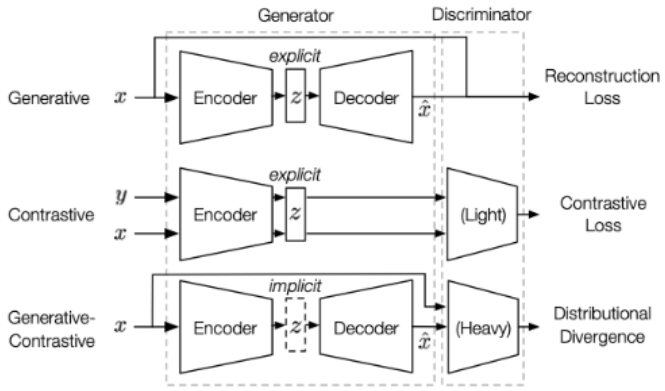


Figure 2.5: Comparison between the three SSL architectures [27]

- **Generative**: Trains an encoder to transform a single input into an explicit vector, passed to a decoder that attempts to reconstruct the input from the generated vector.
- **Contrastive**: Trains an encoder to transform inputs (generally two) into an explicit vector, measuring the similarity between different vectors.
- **Generative-Contrastive** (Adversarial): Trains an encoder to generate fake samples that are passed to a discriminator.

These alternatives have their assets and liabilities, and their understanding is essential to justify their choice for each use case. Previous works [27] have shown the contrastive approach presents better results for image classification tasks, as its nature complies with the image classification task by discarding the decoder and assuming the downstream task will be classification.

2.5.1 Generative Self-Supervised Learning

In this approach, a reconstruction loss is employed, allowing an encoder-decoder network to learn how to reconstruct the provided input.

The success of generative SSL is its ability to recover the original data distribution without assuming which downstream tasks will be used in further steps, making this model versatile in its application in classification and generation models. Despite its flexibility in different scenarios, generative SSL has been found to have poor performance in classification tasks compared to contrastive learning, as the latter's nature complies with the image classification task by discarding the decoder and assuming the downstream task will be classification.

According to *Liu et al.* [27], contrastive algorithms like MoCo [17], SimCLR [7], and SwAV [5], have presented overwhelming performances in several CV benchmarks, while the Natural

Language Processing (NLP) domain still requires contrastive learning models to conduct text classification.

2.5.2 Contrastive Self-Supervised Learning

Statistically, contrastive methods are discriminative, while generative models are, as the name implies, generative. For example, using the joint distribution P(X,Y) of input X and target Y, discriminative models aim to model P(Y|X=x), while generative models calculate P(Y|X=y). Earlier, generative models were considered the only option for representation learning; however, with the success of algorithms like MoCo [17] and SimCLR [7], contrastive models have gained popularity.

The objective of contrastive learning is to learn representations by comparing the similarity between two images using latent representations - a simplified vector representation of the input data containing essential information to model the input. Traditionally, cosine similarity is used to measure similarity, with similar samples determined by augmenting the original image and dissimilar samples determined by comparing with other images. This can be represented through a Noise Contrastive Estimation (NCE) objective:

$$L_{\text{NCE}} = -\log \frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)} + e^{f(x)^T f(x^-)}} \tag{2.3}$$

where x^+ is a positive sample (similar to x), x is a negative sample (dissimilar to x), T is a hyper-parameter called temperature coefficient, and f can be any similarity function. When multiple dissimilar pairs are involved, the InfoNCE loss is computed through the following formula:

$$L_{\text{infoNCE}} = -\log \frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)} + \sum_{k=1}^{K} e^{f(x)^T f(x^k)}} \tag{2.4}$$

MoCo [17] and SimCLR [7] are two commonly used instance discrimination-based methods used to differentiate between instances of a different class, learning representation that can be later finetuned on a downstream task; both methods aim to minimize contrastive loss but differ in how samples are maintained [17].

SimCLR [7] was proposed by *Chen et al.* back in 2020, achieving a new state-of-the-art in self-supervised scenarios [7] with its 76.5% top-1 accuracy, improving upon previous works by 7% [19]. SimCLR [7] employs an end-to-end system where the negative and positive samples are selected from the same batch and optimized using backpropagation in an integrated manner. This means that a single image is transformed in multiple ways before a comparison is made to maximize the agreement with the original image and minimize it with dissimilar images.

In contrast, MoCo [17] abandons the traditional end-to-end training framework [27], storing negative samples in a queue and processing positive samples in each training batch. Additionally, a momentum encoder is used to maintain consistency between the current and previous keys [8], decoupling the batch size from the number of negative samples, significatively enhancing the negative sample efficiency [27].

Both approaches have been continuously studied for improvements, with SimCLR [7] enhancing its end-to-end instance discrimination component by increasing its batch size to 8196 [27; 8], thus providing a more significant amount of negative samples. Additionally, it includes practical techniques to improve its performance, such as incorporating a learnable nonlinear transformation between the representation and the contrastive loss, extending the training steps, and using deeper neural networks [27]. Furthermore, the MoCo [17] framework was upgraded to MoCo v2 [8] on a study by *Chen et al.*, using the ImageNet [10] training set, adding an Multi-Layer Perceptron (MLP) head on top of the linear classifier produced by the CNN, a network architecture to extract the high-level representations from images, affecting the unsupervised training stage, improving accuracy from 60.6% to 66.2%. Additionally, the author extends the augmentation used by *He et al.* [17] by adding blur augmentation [7], improving the accuracy to 63.4%, resulting in a 2.8% increase.

Contrastive learning assumes classification as the downstream application; hence it only utilizes the encoder and drops the decoder compared to generative models, making contrastive models lightweight and well-suited for discriminative downstream tasks.

Since it is still an arising field, contrastive SSL contains issues yet to be solved, including:

1. **Lack of good results in NLP** [27]. Contrastive Learning does not scale to NLP pretraining, and research shows generative approaches are more suited to this task.

2. **Negative Sampling**. It is currently a requirement for most contrastive learning, being a biased and time-consuming procedure. Algorithms such as BYOL [16] and SimSiam [9] have been developed to avoid the need for negative samples, but [27] states there are still improvements to be made.

3. **Root of Data Augmentation success**. Studies [7; 17; 38; 28] have shown data augmentation of the input images improves contrastive learning's performance. However, no conclusion has been reached regarding this apparent boost.

2.5.3 Adversarial (Generative-Contrastive) Self-Supervised Learning

Generative-contrastive or adversarial representation learning is derived from generative learning, trying to address some issues by reconstructing the original data distribution instead of the samples by minimizing the distributional divergence [27]. This approach can be effectively demonstrated using Generative Adversarial Networkss (GANs) [15], where a generator creates fake samples, and a discriminator tries to distinguish them from the real ones, resulting in a min-max optimization problem that can be described as follows:

$$\min_G \max_D \mathbb{E}_{P_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{p_z(z)}[\log(1 - D(Z))] \tag{2.5}$$

Generative-contrastive SSL excels in generating, transforming, and manipulating images. In contrast, it is outperformed in feature extraction by contrastive learning approaches, even though

studies such as BiGAN [12] and BigGAN [13] have attempted to address this concern. Besides, *Liu et al.* [27] shows GANs are prone to collapse during training, leading to developing techniques such as spectral normalization [29] and W-GAN [3], where it remains challenging to train an adversarial network effectively.

2.6 Self-Supervised Learning in Medical Image Analysis

SSL is particularly relevant in medical contexts due to the cost of annotations and the need for specialized, time-consuming, and, therefore, expensive annotation by trained specialists. This section describes examples of SSL applications in medical scenarios and datasets that will be used.

Since Chest Radiography is the most common examination in the world, helping medical professionals through screening, diagnosis, and management of hazardous diseases [20]. Thus, searching for automatic image classification approaches might improve the global population's health through enhanced medical workflow prioritization.

One of the main issues of this imaging modality is the need for specialized training for proper interpretation, consequently falling in the use case of an area requiring further SSL studies. Besides, contrary to natural image classification, chest X-ray interpretation presents a unique problem [35]. First, identifying irregularities in just a few pixels may be sufficient for disease diagnosis. Second, because they are bigger, grayscale, and have consistent spatial patterns across images, chest X-rays differ from natural images in terms of their characteristics. Finally, compared to natural photos, there are significantly fewer unidentified chest X-ray images. These variations might make it more challenging to interpret chest X-rays using contrastive learning techniques, which were first created for natural image categorization.

Previous contrastive learning methods for X-ray images have limited applications. Most approaches encompass SL methods with labeled data [26; 32], or semi-supervised [30], using both labeled and unlabeled data, but those approaches cannot keep up with the increasing amounts of unlabelled data being added to the medical workflow daily. *Chaitanya et al.* [6] uses a localized and a global loss function during pretraining to extract contrastive pairs from the MRI and CT datasets. However, the proposed method significantly relies on the volumetric characteristics of MRI and CT scans, therefore not greatly applicable to chest radiography.

2.6.1 Datasets

This dissertation aims to evaluate the performance of the MoCo-CXR [35] algorithm through a comparative medical imaging study, requiring an initial selection of a dataset for pretraining. Two datasets were analyzed for selection, CheXpert [20], and MIMIC-CXR [23], both containing 2D chest radiography images with medical observations. Since our goal was to perform an in-depth analysis of MoCo-CXR, [35] CheXpert [20] presents itself as a better option due to being the focus of the studied algorithm.

CheXpert [20] (**Che**st e**Xpert**) is a large dataset of 224,316 chest radiographs of 65,240 patients. Labels for 14 different observations are obtained via an automatic labeler that leverages the text-free radiology reports available in the dataset. This data was collected from chest radiographic studies performed between October 2002 and July 2017 at Stanford Hospital, California, and has already been used for several studies [35].

MIMIC-CXR [23] is a large dataset of 227,835 imaging studies for 65,379 patients of the Beth Israel Deaconess Medical Center Emergency Department between 2011 and 2016. Each imaging study contains at least one image, commonly a frontal and lateral view of the patient, whose identity has been hidden to protect patient privacy, adding to a total of 377,110 images that were made public and freely available, aiming to facilitate and encourage a wide range of research in several areas of artificial intelligence, including computer vision, hence proving a valuable tool for our task of image classification.

2.6.2 Published Works

Although lacking many studies, the self-supervision paradigm for medical image analysis contains some existing examples, namely the work by *Azizi et al.* [4], which applies self-supervised pretraining followed by supervised finetuning on image classification on two different tasks, the first with dermatology examples and the second with chest X-rays. Furthermore, *Azizi et al.* [4] introduce a novel method called Multi-Instance Contrastive Learning (MICLe) to construct more informative pairs for SSL (see Figure 2.6), outperforming robust supervised state-of-the-art pretraining approaches on ImageNet [10], resulting in an improvement of 6.7% in top-1 accuracy and 1.1% in mean AUC.

The approach consists of three phases visually described in Figure 2.7. Phase one uses the SimCLR [7] algorithm for self-supervised pretraining on ImageNet [10]. In phase two, self-supervised methods are applied to the unlabeled data to create labels. The final phase is a supervised finetuning process, functioning as the final downstream task, while the first two stages are self-supervised pretext tasks.

The study highlights a contrastive learning approach in which image augmentations provide the encoder with two views of the same image, leading to a maximized agreement between the resulting representations [39]. If multiple images exist, a MICLe approach is used, where two different images are used to create a similar pair of examples, which is the case of the CheXpert [20] dataset. Furthermore, it is concluded that the self-supervised pretraining using both ImageNet [10] and Chexpert [20] data significantly improves performance on a distribution-shifted dataset, which is paramount to clinical applications [4]. Ultimately, the author mentions the scalability of SSL due to its lack of annotations and that the next step would be to determine the limit of SSL for immense datasets.

The global Covid-19 pandemic has put immense pressure on public health systems, thus making early patient screening crucial to prevent the spread of the infection and reduce the workload on healthcare providers. A study by *Li et al.* [24] proposes a self-knowledge distillation-based

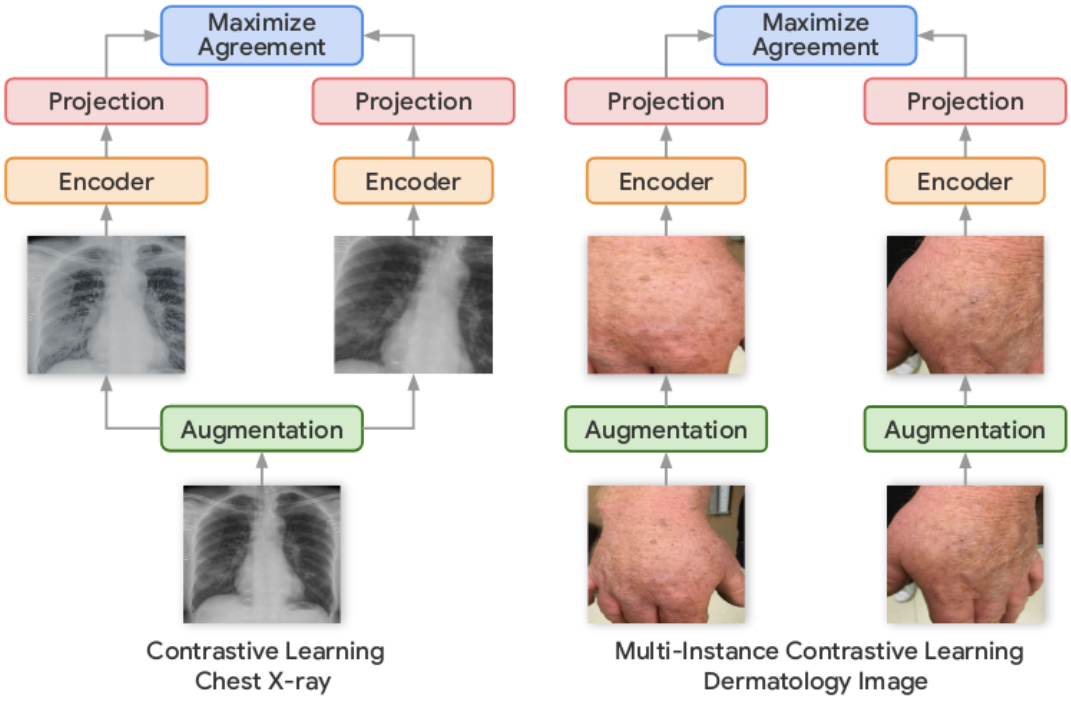


Figure 2.6: Contrastive learning and MICLe [24]

SSL method for Covid-19 detection, attempting to replace the use of the standard detection system, the RT-PCR test, which has a high false-negative rate and is time-consuming, with Chest X-rays, which are low cost, have short scan time, and low radiation. This study achieves promising results, with an F1-score score of 0.988, an AUC of 0.999, and a 0.957 accuracy applied on an extensive open Covid-19 chest X-ray dataset [32].

According to *Caron et al.* [5], their method for Covid detection is based on triplet networks, a variation of the Siamese Network [25], used to learn discriminative representations from the Chest radiography images. Figure 2.8 describes the suggested approach consisting of three networks, where the target network's weights are an Exponential Moving Average (EMA) of the weights of the online network, and the encoders in the Self-Knowledge Distillation (SKD) network and online network share the weights [37]. The method consists of two components: SSL and a SKD component. The SKD component can be classified as "regularizing the training using soft targets that carry the "dark knowledge" of the same network" [24], learning better representation from different radiography images based on the similarity between visual features, assuming that images with similar features have similar probabilities generated by the predictor in the SSL component.

Furthermore, *Li et al.* concludes that the proposed method outperforms state-of-the-art techniques, being especially effective when using ResNet50 [18] network as an encoder, making use of self-knowledge of images based on similarities of visual features. However, one challenge regarding the proposed methods is the lack of testing for datasets unrelated to Covid-19 and for the classification of multiple annotations since the study only considers a binary classification of a

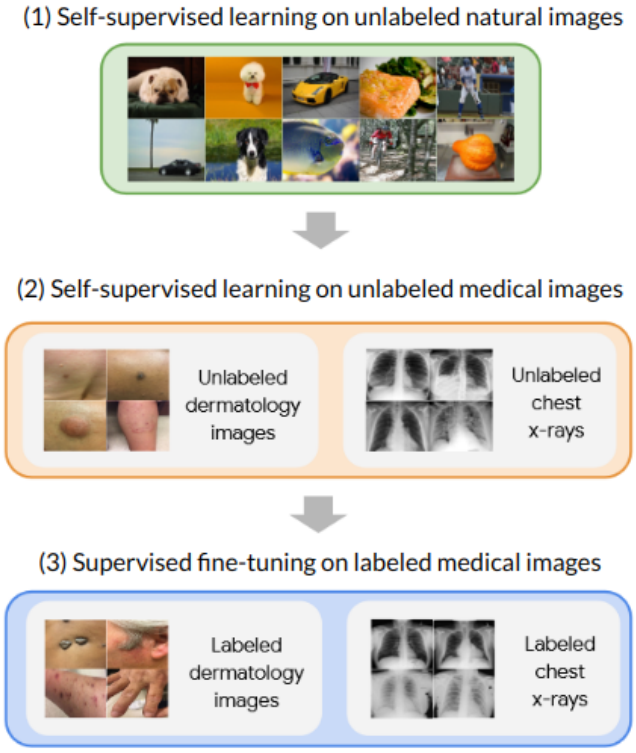


Figure 2.7: Three phases used for an SSL approach performed on two separate use cases, dermatology and chest X-rays [4]

patient being covid-positive.

To create models with better representations and initializations for the diagnosis of diseases in chest X-rays, *Sowrirajan et al.* proposes MoCo-CXR [35], which is a modification of the contrastive learning technique MoCo [17] applied to the CheXpert [20] dataset. The author addresses the main differences between chest-X ray interpretation from natural image classification that might affect the applicability of contrastive algorithms. For instance, MoCo [17] uses numerous data augmentation techniques to produce positive image pairs from unlabeled data; however, random cropping and blurring adjustments might lead to removing disease-related portions of the image. In contrast, color jittering and random grayscaling would not affect photos that are already grayscale. Additionally, it is still unknown whether retraining the models using MoCo [17] can outperform the conventional automated chest X-ray interpretation method, which entails fine-tuning pretrained models on ImageNet [10] with labeled chest X-ray images. This is due to the limited availability of chest X-ray images compared to natural images and their larger size.

In their study, *Anton et al.* [2] further explore the effectiveness of SSL models in the medical image scenario by assessing the generalisability of seven self-supervised models across nine medical datasets, including those related to chest radiography like CheXpert [20] and MIMIC-CXR [23], with the Bootstrap Your Own Latent [16] (BYOL) method achieving slightly better results. However, the researchers acknowledge the need for additional investigation into hyperparameter variance to draw conclusive findings.

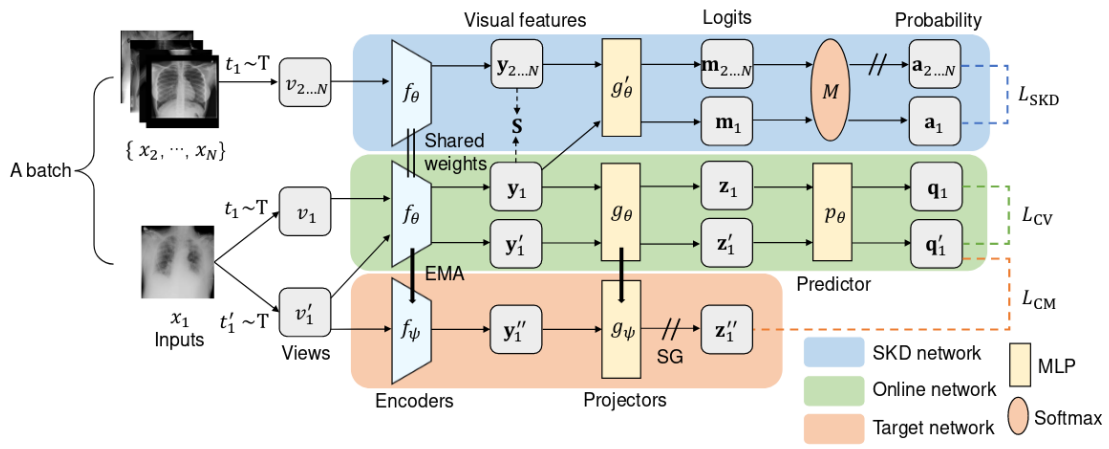


Figure 2.8: Overview of the method proposed by *Li et al.* [24]

Additionally, the study suggests that domain-specific pretraining is advantageous since SSL models trained on chest X-rays achieved better performance. With a dataset domain shift, however, performance suffers noticeably because even a small change in the domain can significantly affect the accuracy of the classification. Furthermore, analyzing the encoded features in the study leads to the conclusion that domain-specific pretraining yields a more targeted feature extraction compared to conventional ImageNet [10] pretraining. While this can greatly enhance performance for tasks within the same domain, it comes at the cost of reduced generalizability.

This study distinguishes itself from other research in the field by conducting the first extensive comparison of pretrained SSL models for their applicability to medical images. Additionally, it represents one of the initial efforts to assess the transferability of SSL models explicitly pretrained on ImageNet [10] compared to those pretrained on a medical domain-specific task across various distinct medical image datasets, providing a way to measure and quantify the benefits of both approaches directly.

In this work, two models are taken into account that may be used to analyze SSL medical imaging scenarios. First, using ResNet50 as the primary feature extractor, self-supervised models, including SimCLR-v1, [7] MoCo-v2, [8] PIRL, [40] SwAV, [5] [5] and BYOL [16] are pretrained on the ImageNet [10] training set. Second, two domain-specific SS pretrained models are used: MIMIC-CheXpert [23] and MoCo-CXR [35]. Both were trained on respective chest radiography datasets and used a DenseNet121 backbone. MoCo-CXR [35] can also use a ResNet18 architecture for feature extraction.

One of the key limitations of this study regarding the CheXpert [20] dataset is the binary conversion of labels, as the analysis solely focuses on the Pleural Effusion pathology, representing 40.34% of all images in the dataset. Consequently, it overlooks most of the dataset, including 13 other pathologies.

2.6.3 Conclusions

Considering existing SSL approaches in the medical image analysis field, more specifically in the analysis of chest radiography, existing studies are limited. However, several works propose new methods yet to be fully explored, such as MICLe, self-knowledge distillation, and MoCo-CXR [35] that can be improved using different pretraining tasks, encoders, and finetuning, that might offer different results in different datasets.

Chapter 3

Methodology

This section describes the method used in this study, particularly the model architecture, the most relevant implementation details, and a description of the dataset and associated labels. Finally, all conducted experiments are described.

3.1 Model Overview

As mentioned in Chapter 1, this dissertation intends to conduct an in-depth investigation into MoCo-CXR [35], a self-supervised contrastive learning framework that uses data augmentations to generate views of an image to learn its intrinsic characteristics in an unsupervised fashion [35], enabling the detection of different pathologies on the CheXpert [20] dataset. A MoCo-based [17] implementation was chosen due to its efficiency at learning representations from large amounts of unlabeled data, which aligns with our selection of the CheXpert [20] dataset, and its proven success with smaller batch sizes when compared to its counterpart algorithms, such as SimCLR [7].

The chosen approach for this study encompasses two phases, as described in Figure 3.1. First, a pretraining task based on the work by *Sowrirajan et al.* [35] allowing the creation of MoCo-CXR [35] pretrained models with different augmentations, and finally, an image classification task based on the SSL medical image study by *Anton et al.* [2], where two different methodologies are applied, linear probing and finetuning.

Following the authors' approach, MoCo [17] pretraining is performed on the entire CheXpert [20] training dataset with pre-initialized ImageNet [10] weights due to its possible convergence benefits [31] and no extra cost associated with its addition. The MoCo-CXR [35] approach is similar to any other self-supervised implementation, with the exception of the used augmentations, which discard random crop and Gaussian blur, which could affect the disease labels in a chest radiograph scenario. Furthermore, no color jittering and random greyscale were used since they do not represent meaningful augmentations, as they would not affect photos that are already grayscale. *Sowrirajan et al.* [35] uses horizontal flipping and random rotation of 10 degrees as

the main augmentations since they are the most commonly used by other studies on chest radiograph models [20]. In this study, we differ by evaluating models containing variations of these two augmentations.

The second phase consists of a downstream image classification task based on the work by *Anton et al.* [2]. The authors test two scenarios: linear probing and finetuning.

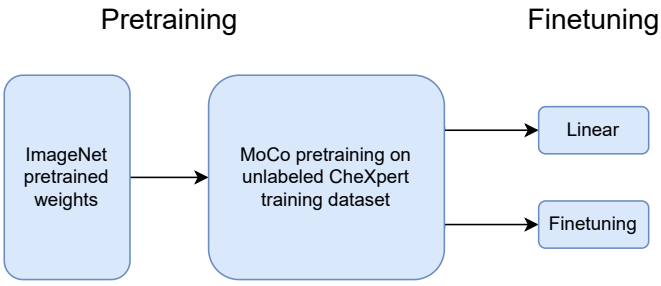


Figure 3.1: Architecture describing the pipeline for the 2 main phases of the used methodology by combining the approaches from *Sowrirajan et al.* [35] and *Anton et al.* [2]

As for the linear approach, the selected pretrained model is frozen and leveraged as a fixed feature extractor with a multinominal logistic regression fitted on top of the fixed features through the following representation:

$$\mathbb{P}(y = c_i|x) = \frac{e^{\omega_i \cdot x}}{\sum_{k=1}^{K} e^{\omega_k \cdot x}} \tag{3.1}$$

- x - Feature representation.
- $\{\omega_1, \ldots, \omega_K\}$ - Learned set of weights.
- $\omega_i \in \mathbb{R}^d$ - Where d is the dimensionality of the extracted features.
- $\{c_1, \ldots, c_K\}$ - Set of class labels.

For the finetuning approach, all pretrained parameters are refitted, along with an attached linear classification head

3.2 Implementation Details

This section discusses the applied methods during both phases of the proposed methodology, detailing the approach and preparation for all experiments and the environment used for the project.

3.2.1 Pretraining

In the pretraining phase, an Adam optimizer was selected with a learning rate of $1.25e^{-5}$, a weight decay of $1e^{-4}$, and a batch size of 16. The chosen values were adapted from those found on the MoCo-CXR [35] paper, considering the capabilities of our computational resources as well as the

effectiveness of the optimizer when using large-scale datasets such as CheXpert [20]. Additionally, pre-processing was performed with data normalization and an image resize (320x320 pixels). Furthermore, when training the model, it was possible to select the following augmentations:

- Random Horizontal Flip.
- Random Rotation of 10º.

When training different models, random horizontal flip and random rotation functioned as variable hyperparameters to create 4 different models to evaluate augmentation influence. The author [35] adopts an approach that incorporates all previously mentioned augmentations, asserting these are commonly employed in training scenarios involving chest radiographs [20; 33]. Regarding the backbones used, the emphasis is placed on two specific backbones: ResNet18 and Densenet121. These backbones are simultaneously used by *Sowrirajan et al.* and *Anton et al.*, aligning our implementation with the original work of these researchers [35; 2].

3.2.2 Downstream Task

The approach employed for the downstream task was derived from the work of *Anton et al.* [2]. This task comprises two distinct scenarios, namely the linear probing and finetuning methods, each with its own selection of hyperparameters. In both cases, when training with ResNet18, a dimensionality value of 512 is employed for the extracted features (d), while a value of 1024 is utilized for DenseNet121.

In the linear approach, the authors adopt a strategy inspired by *Rricson et al.* [14] by applying an ℓ_2 regularization constant to the validation set using 45 logarithmically spaced values ranging from $1e^{-6}$ to $1e^{-5}$. The logistic regression model is subsequently retrained on the combined training and validation sets, using the chosen ℓ_2 regularization constant, and evaluated on the test set. During training, no data augmentation is implemented except for bicubic resampling to 224 pixels, followed by a center crop of 224 × 224. Furthermore, due to resource limitations, batch sizes of 64 and 128 were selected for ResNet18, while a batch size of 32 was used for DenseNet121.

In the finetuning approach, the model undergoes training for a variable number of steps, beginning at 5000. The optimization is performed using Stochastic Gradient Descent (SGD) with Nesterov momentum, where the momentum parameter is set to 0.9. Initially, an early stopping mechanism was incorporated with a patience of 3, based on the classification accuracy on the validation set as the relevant metric, checking the accuracy every 200 steps. However, subsequent implementations experiment with removing this feature and extending the training duration instead of interrupting the process, improving the results.

For all models with the finetuned approach, a batch size of 64 and a weight decay of $1e^{-8}$ are used during training, while the selected learning rate was $1e^{-2}$ on ResNet18 and $4e^{-2}$ on DenseNet121. These values were initially derived from the original paper but have been modified considering our reduced resource constraints compared to those available to the authors.

3.3 Dataset

The main focus of this dissertation relies on experiments done on the CheXpert [20] dataset. This is the most commonly used dataset for medical image analysis when considering chest radiographs, thus being selected for this study. Moreover, the implemented approach aims to explore and deepen the research performed by *Sowrirajan et al.* [35] and *Anton et al.* [2], making it crucial to utilize a shared dataset with both approaches. CheXpert [20] contains 224,316 chest radiographs of 65,240 patients labeled for the presence of 14 common pathologies. It is worth mentioning that this dataset is not manually annotated by radiologists; labels are automatically extracted from the available medical text reports.

When using the CheXpert [20] dataset, we considered two distinct labeling methodologies: the default labels provided by CheXpert [20] and the labels generated by CheXbert, [34] a state-of-the-art improvement over the default labels. Existing label extraction approaches such as CheXpert [20] typically rely on sophisticated feature engineering based on medical domain knowledge or manually annotated labels. However, CheXbert [34] is positioned as a strategy that leverages the vast scale of rule-based systems and the high quality of expert annotations. This approach achieves superior performance by initially training a biomedically pretrained BERT [11] model on annotations from a rule-based labeler, followed by finetuning on a limited set of expert annotations augmented with automated back-translation.

CheXpert [20] was used for this project's pretraining phase. Although the CheXbert [34] paper presents itself as a state-of-the-art improvement over CheXpert, [20] it only does so by providing better labels, which are irrelevant in the pretraining stage since SSL methods do not consider labels to learn relevant features.

For the image classification downstream task, both CheXpert [20] and CheXbert [34] were used. Therefore, different linear and finetuned models were trained and compared side by side, concluding what the CheXbert [34] paper stated, that it provided better results with their improved labels. Thus, CheXbert [34] labels were confirmed as superior and, from this point on, used for the remainder of the experiments.

The data is divided into train, validation, and test sets when employing both approaches. Both training and validation use the same split of the dataset, encompassing 60% of the data, while the validation set, reserved for calculating the final accuracy, consists of the remaining 40% split of the dataset.

In later experiments, we tested our best-performing model on a test set annotated by professional radiologists. Since these labels are the closest we can get to real-world deployments, they were used to confirm the quality of our models. All three labeling approaches encompass observations for the presence of 14 common pathologies, as described in Table 3.1.

The same table shows CheXpert [20] and CheXbert [34] have similar percentages of labels, displaying the similarity between both labeling methods. In contrast, radiologist annotations contain distinct values for each pathology. The difference in size between the datasets may justify these disparities. Besides, Table 3.1 shows that some pathologies rarely occur in the dataset, such

Table 3.1: The percentage of positive labels in each dataset for each pathology. In both CheXpert [20] and CheXbert, [34] uncertain labels are converted into positive labels (U-Ones Methodology), while the Radiologists' dataset does not contain uncertainties. The radiologists' test set only contains 668 images.

Dataset Labels	CheXpert [20]	CheXbert [34]	Radiologists
Atelectasis	30.04	30.56	26.65
Cardiomegaly	15.70	15.43	26.20
Consolidation	19.03	18.07	5.24
Edema	29.20	29.21	12.72
Enlarged Cardiomediastinum	10.38	10.14	44.61
Fracture	4.33	4.12	0.90
Lung Lesion	4.78	4.89	2.10
Lung Opacity	49.76	46.28	46.41
No Finding	10.02	9.46	16.32
Pleural Effusion	43.78	43.39	17.96
Pleural Other	2.76	2.99	1.20
Pneumonia	11.10	10.85	2.10
Pneumothorax	10.11	9.09	1.50
Support Devices	52.40	50.66	47.16

as "Lung Lesion" and "Pleural Other". This means the dataset is imbalanced, and the models will have problems learning these features. Besides, since most of the values of the mentioned observations have negative labels, the model might appear to have a high accuracy value when in fact might only be predicting negative labels. Additionally, the table shows that multiple observations can be present for each chest radiography; thus, this is a multi-label classification problem.

Each observation is classified as either Positive (1), Negative (0), or Uncertain (-1). Currently, as described in CheXpert, [20] there are five main methodologies to deal with uncertainty labels:

- U-Ignore - Uncertain labels are ignored during training.
- U-Zeroes - Uncertain labels are mapped to 0.
- U-Ones - Uncertain labels are mapped to 1.
- U-SelfTrained - First, a model is trained using the U-Ignore approach to convergence. Afterward, the model is used to make predictions that re-label each uncertainty with the probability prediction outputted by the model.
- U-MultiClass - Uncertainty labels are treated as their own class.

Each of these methodologies has advantages and disadvantages, which have been thoroughly explored in the CheXpert [20] paper. In this work, we adopted the U-Ones approach, leaving the exploration of the other alternatives for future work.

3.4 Experimental Setup

This section describes all experiments performed in this study and the narrative that led to the decisions made in the process. All experiments were performed using an NVIDIA RTX 2080 TI graphics card with 11GB of VRAM. This graphics card is better than those used by most state-of-the-art studies [2; 35], removing some of their limitations.

3.4.1 Experiment 1: Pretraining Data Augmentation

We start by studying the impact of data augmentation on the MoCo-CXR [35] training process. To do so, we started by adapting the work from *Sowrirajan et al.* [35] and attempting to recreate their model, which used a random rotation of 10 degrees and horizontal flipping, which are the most commonly used augmentations in chest radiograph classification.

In this process, we decided to create four models to study the influence of each augmentation, alternating between rotation and horizontal flipping. Additionally, we decided to evaluate these models using two distinct backbones, ResNet18 and DenseNet121. This approach allows us to examine the performance of MoCo-CXR [35] across various backbones with varying numbers of layers. Consequently, this yields eight pretrained models, as outlined in Table 3.2. All these models were trained for 20 epochs, aligning with the methodology employed by [35].

Table 3.2: Models with different pretraining augmentations and backbones for MoCo-CXR [35].

Model	No Augmentation	Rotate	Flip	Rotate Flip
ResNet18	Model1	Model2	Model3	Model4
DenseNet121	Model5	Model6	Model7	Model8

3.4.2 Experiment 2: Batch Size Influence on Downstream Task

For the downstream image classification task, we follow the approach proposed by *Anton et al.* [2], who consider linear and finetuned schemes, as described in Section 3.2.2. Note that [2] evaluate all models solely on the presence or absence of "Pleural Effusion" by converting the multi-label annotations of CheXpert [20] into a binary format. This involves assigning a value of 1 for "Pleural Effusion" and disregarding all other labels by converting them to 0, following a methodology employed by [4].

This experiment focuses on analyzing the effect of different batch sizes on the classification task, which varies according to the selected backbone since deeper networks such as DenseNet121 require more computational power. Since we do not have enough computational power for extensive testing on DenseNet121 with multiple batch sizes, this study only considers the ResNet18 model on both linear probing and finetuning.

3.4.3 Experiment 3: Number of Finetuning Steps

Afterward, we studied the finetuning approach, achieving slightly improved results compared to those reported by the authors [2]. Upon examining their methodology, we discovered that the authors had limited computational resources, leading them to train the model for 5k steps with an implemented early stopping with patience of 3 using the classification accuracy on the validation set, evaluated at every 200 steps. In contrast, since we had more computational resources, we conducted a study regarding the influence of the number of steps in the finetuning process. Thus, we trained a model for 200k steps and plotted the metrics calculated on the validation set. We aimed to determine the optimal number of steps at which training should be concluded through an extensive run that lasted approximately 36 hours.

3.4.4 Experiment 4: Training Label Quality

As previously mentioned, the CheXpert [20] dataset does not contain manual annotations. Instead, they are automatically extracted from the available medical reports. Therefore, different labeling methods might lead to different labels. Additionally, *Smit et al.* [34] proposed that CheXbert [34] labels outperformed the standard labels provided by[20], as discussed in section 3.3. As a result, we designed this experiment with the intention of comparing these two labeling methodologies. To accomplish this, we used the radiologist test set to perform a comparative analysis between CheXpert [20] and CheXbert, [34] considering both linear and finetuned approaches, with the objective of determining which set of labels should be chosen for subsequent experiments.

3.4.5 Experiment 5: Evaluation on Different Pathologies

Due to the predominant focus on the "Pleural Effusion" observation in previous studies conducted on the CheXpert [20] dataset, and limited investigations into other pathologies, with only 5 out of the 14 pathologies being explored, we examine the performance of our models on all observations present in the dataset. However, we excluded "Support Devices" from our analysis as it is not considered a pathology, resulting in a total evaluation of 13 pathologies. Consequently, a comparative analysis was carried out on the CheXpert [20] dataset using radiology labels to assess the accuracy of each model, considering a binary labeling method for each pathology.

3.4.6 Experiment 6: Multilabel Classification

Afterward, we extend the analysis of the multilabel scenario. This resulted in a requirement for new evaluation criteria, as the labeling system was no longer binary and allowed for the presence of multiple positive pathologies simultaneously. To address this, we employed two different metrics to evaluate the performance of our models.

- **Exact match ratio** - The predicted value is considered correct if and only if all predicted labels coincide with the real values.

- **Label-wise accuracy** - Measures the accuracy of the model for each individual class separately rather than aggregating the accuracy across all classes.

These two different metrics evaluate different aspects of the model's performance. While the exact match ratio excels at evaluating the model's capability of being fully correct on all 13 predictions simultaneously, label-wise accuracy is more effective at finding the model's proficiency at predicting each individual label consistently. However, both metrics come with their own limitations: for instance, the first metric does not differentiate between a model that predicts all classes incorrectly and one that only fails to predict a single class among the 13; the second one is highly susceptible to imbalanced data, as classes with low occurrence may result in high accuracy if the model solely predicts the majority class.

3.4.7 Experiment 7: Radiologists' Test Set

This final experiment aims to assess our models' performance in real-world deployment scenarios by evaluating the top-performing model on the radiologists' test set. This evaluation was previously conducted for the "Pleural Effusion" pathology, and now we aim to extend the analysis to multiple observations, as explored in experiments 5 and 6. Therefore, this experiment serves as an extension of the analysis performed in experiment 4, covering a wider range of observations.

3.5 Conclusion

This work was based on the approaches proposed by [35] and [2], combining their methodologies applied to the CheXpert [20] dataset multiple labeling systems. CheXpert [20] and CheXbert [34] labels were utilized for training and validation, while the test set evaluation involved manually annotated labels by radiologists in subsequent experiments. This combined approach involves a SSL pretraining phase and an image downstream classification task, considering linear probing and finetuning. We explored the impact of various factors through multiple experiments, including pretraining augmentations, dataset labels, training steps, hyperparameters, and multiple simultaneous labels. These studies aimed to understand how each model characteristic influenced its performance comprehensively.

Chapter 4

Results and Discussion

This chapter presents the results of the 7 experiments described in Section 3.4, along with a concluding section summarizing the findings of those experiments. The initial 4 experiments focus on the "Pleural Effusion" pathology within the CheXpert [20] dataset, involving variations in pretraining augmentations and downstream image classification hyperparameters. Subsequently, the following 2 experiments extend the analysis to include the remaining 12 pathologies described in Section 3.3. These experiments explore the binary classification of each pathology or a multilabel scenario. The final experiment evaluates the previously obtained best-performing models on the radiologist's test set, providing insight into the real-world performance of our models if deployed.

4.1 Experiment 1: Pretraining Data Augmentation

In this experiment, our objective was to replicate the pretraining methodology employed in MoCo-CXR [35] while exploring the influence of various data augmentations. This involved creating 8 different models, as outlined in Section 3.4.1 on Table 3.2. The performance of these models was then evaluated through a downstream image classification task, as detailed in Section 3.2.2. This experiment was conducted using 2 different backbone architectures, ResNet18 and DenseNet121, resulting in the total of 16 models showcased in Tables 4.1 and 4.2. These experiments were conducted using the CheXpert [20] dataset and its original automatic labeling system, and the metric selected for evaluation was accuracy, consistent with the approach used by *Anton et al.* [2]. Additionally, for the finetuning approach, training was performed for 5k steps, employing an early stopping mechanism with a patience of 3 and evaluation at every 200 steps. This aligns with the methodology used in [2], allowing for comparative analysis.

Among both backbones, the best performance was obtained with the finetuned model pretrained solely with rotation, with the DenseNet121 rotation model outperforming all others, reaching an accuracy value of 76.90%. As for the linear classification task, the Flip augmentation model on the ResNet18 backbone achieved the highest performance, reaching an accuracy of 68.94%. In contrast, the optimal linear model for the DenseNet121 backbone was achieved without any additional augmentations.

When comparing the 4 different augmentations on the linear approach, the model with only the flip augmentation achieved consistently high results, although the differences appeared to be low and inconsistent between different runs. On the finetuned approach, there were high variances within different augmentations, which were not consistent when comparing both backbones.

When comparing both classification methods, we can conclude that finetuned models were overall better, as the best linear model was outperformed by the best-finetuned model on both backbones. Besides, the average accuracy value for each method was always superior on the finetuned method, with a particularly high value of 75.53% on the DenseNet121 method. The observed lower accuracy values in the linear model were expected, as the model only learns the last layer, while the finetuning approach requires re-learning the whole model.

Table 4.1: Accuracy on the CheXpert [20] validation set obtained by different pretraining data augmentations strategies with the ResNet18 backbone. Both Linear and finetuning approaches are considered.

Method/Augmentation	No Augmentation	Rotate	Flip	Rotate Flip	Avg
Linear	68.07	68.15	**68.94**	67.83	68.25
Finetuning	71.93	**75.85**	68.56	63.04	69.85

Table 4.2: Accuracy on the CheXpert [20] validation set obtained by different pretraining data augmentations strategies with the DenseNet121 backbone. Both Linear and finetuning approaches are considered.

Method/Augmentation	No Augmentation	Rotate	Flip	Rotate Flip	Avg
Linear	**67.53**	66.93	67.52	66.60	67.15
Finetuning	76.66	**76.90**	72.37	76.19	75.53

When looking at the author's results, we were unable to replicate them. While the accuracy of our linear model with rotation and flip was 67.83%, *Anton et al.* achieved the value of 74.76% (6.93% more accurate). Furthermore, when comparing the finetuned model, our model outperformed theirs by 1.15%. The difference between our results and the original authors might be due to inherent algorithm randomness factors and each graphics card can behave differently. Besides, the study only mentions the use of labels from the CheXpert [20] dataset, which contains several versions with different labels and image resolutions, which will affect the results and make an exact comparative analysis quite challenging.

4.2 Experiment 2: Batch Size Influence on Downstream Task

In this experiment, we investigated the impact of various batch sizes on the model's accuracy. To conduct this analysis, we opted for the ResNet18 backbone architecture as it requires less computational power. This enabled us to test and compare models with increased batch sizes, reaching up to 128 in the linear approach and 64 in finetuning. In contrast, if DenseNet121 was selected, the maximum batch size that could be tested in our environment was limited to 32.

During the testing of models using the finetuning method, we consistently paired an increase in batch size with a corresponding increase in the learning rate. The initial experiment based on the work by *Anton et al.* [2] contained a batch size of 64 and a learning rate of $1e^{-4}$ for ResNet18. Subsequently, all our conducted experiments involved doubling or halving the current batch size, with the same operation being performed to the learning rate.

Table 4.3: ResNet18 backbone testing with linear image classification for different pretraining conditions and hyperparameters. Values represent the model accuracy on the CheXpert [20] dataset as a percentage value.

Batch Size	No Augmentation	Rotate	Flip	Rotate Flip	Avg
16	67.92	**68.73**	68.25	67.94	68.21
32	68.16	**68.85**	68.20	67.78	68.25
64	68.07	68.15	**68.94**	67.83	68.25
128	68.24	68.22	**68.83**	68.10	68.35

On the linear approach, 4 different batch sizes were investigated (Table 4.3), with the highest accuracy value obtained with a batch size of 64 on the model with a horizontal flip augmentation, reaching 68.94%. No relevant differences are found either between different augmentations or different batch sizes, as the average value from the worst performing batch size (batch size 16 with 68.21% accuracy) is only inferior to the best performing model (128 with 68.35%) by 0.14%. Consequently, we conclude that the correlation between batch size and performance is not clear on the linear model, and no significant gains are found through batch size variance. Therefore, we decided to keep the batch size used by the author (batch size of 64) for a better comparative analysis.

Table 4.4: ResNet18 backbone testing with downstream image classification for different pretraining conditions and hyperparameters. Values represent the model accuracy on the CheXpert [20] dataset as a percentage value. Training was performed for 5k steps without early stopping.

Batch Size	No Augmentation	Rotate	Flip	Rotate Flip	Avg
16	72.69	**72.74**	72.04	72.46	72.48
32	74.43	74.43	73.80	**74.49**	74.29
64	76.21	**76.85**	76.73	76.15	76.48

On the finetuning approach, only 3 batch sizes were tested (Table 4.4), resulting in a noticeable improvement every time the batch size is doubled, as all pretraining augmentations obtain improved performance, with the top model containing a rotation augmentation on a batch size of 64. Consequently, we can conclude an increase in batch size results in a performance improvement for 5k steps training. However, since we could not further increase the batch size due to resource constraints, we are unsure what the optimal batch size for ResNet18 finetuned training should be. Thus, such investigation could be a future work improvement of our study in scenarios with fewer resource constraints.

4.3 Experiment 3: Number of Finetuning Steps

Finetuning models obtained inconsistent results between different runs and augmentations. We hypothesized this might be due to the early stopping mechanism implemented by the author due to their lack of computational resources, which interrupted the run when no further gains were made after 3 verifications were done every 200 steps. This means no improvements after 600 steps would cause the run to terminate.

Tables 4.5 and 4.6 reveal that activating the early stopping mechanism significantly diminishes the performance of the model. This effect is more pronounced in the case of ResNet18 finetuning, where the average accuracy without early stopping was 76.48% (6.63 increase with early stopping removal). The difference is less noticeable in the DenseNet121 approach with only a 1.3% variation, as the early stopping mechanism was triggered solely by the flip augmentation model, concluding training at 3k steps (60% of the total run). In contrast, Table 4.5 displays 3 models that ended earlier, with the rotate and flip augmentations model finishing after only 1k steps (20% of training completed), resulting in a significantly lower average value (63.04%). These reduced average accuracy values, coupled with the lowest observed accuracy, coincided with the run that terminated earlier (rotation and flip model on ResNet18 with 1k steps and 63.04% accuracy). This prompted us to conduct further experiments to determine the optimal number of steps.

Table 4.5: ResNet18 backbone finetuning comparison between early stopping addition and removal. Values represent the model accuracy on CheXpert [20] labels as a percentage value.

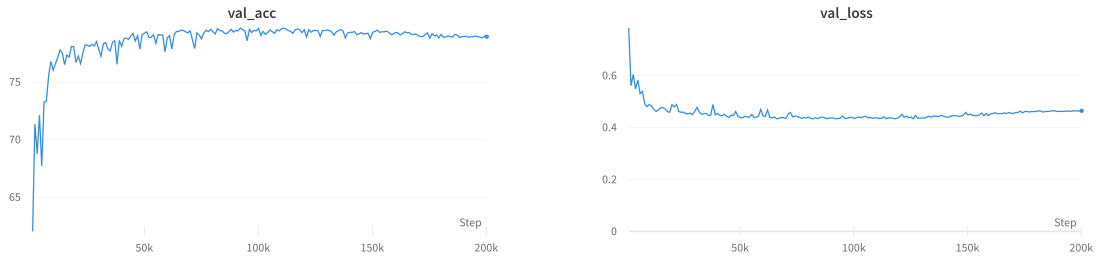
Finetuning	No Augmentation	Rotate	Flip	Rotate Flip	Avg
w/o early stop	76.21	**76.85**	76.73	76.13	76.48
w/ early stop (steps)	71.93 (2k)	**76.85 (5k)**	68.56 (2.6k)	63.04 (1k)	69.85

Table 4.6: DenseNet121 backbone finetuning comparison between early stopping addition and removal. Values represent the model accuracy on CheXpert [20] labels as a percentage value.

Finetuning	No Augmentation	Rotate	Flip	Rotate Flip	Avg
w/o early stop	76.66	**76.90**	76.47	76.19	76.56
w early stop (steps)	76.66 (5k)	**76.90 (5k)**	72.37 (3k)	76.19 (5k)	75.53

This way, we opted to conduct additional training runs on the ResNet18 backbone with rotation and flip augmentations extending the duration to 10k, 25k, and even 50k steps. These extended training runs led to substantial performance improvements, with accuracy values reaching 77.92%, 78.86%, and 79.87%, respectively. Since the performance was continuously increasing, we decided to perform a long run for 200k steps (see Figure 4.1). The ResNet18 model was selected since both backbones were expected to perform similarly, and ResNet18 would take significantly less time to execute and achieve the same goals.

After analyzing Figure 4.1, we can observe a gradual increase in performance that eventually reaches a plateau around the 100k steps mark, suggesting that the model has reached its learning limit. The authors acknowledge their limitations regarding the lack of computational resources,



(a) Accuracy on validation set

(b) Loss on validation set

Figure 4.1: Accuracy and loss plotted on a 200k steps run of a ResNet18 rotation and flip model

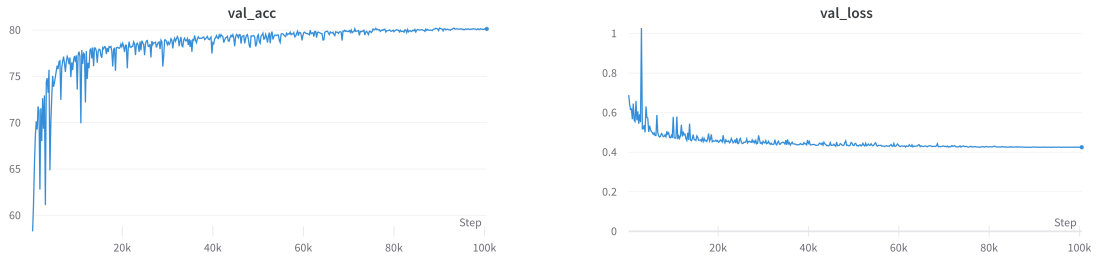
and these experiments were able to find the true limitations of this method. Consequently, further experiments were performed with 100k steps when attempting to reach the optimal finetuning model, represented by Figure 4.2, obtaining 80.54% accuracy and when evaluated on the validation set.

This way, by studying the influence of the number of finetuning steps, we were able to conclude that it indeed plays a significant role. Consequently, we determined that the default 5k steps performed by the authors should be increased to 100k, improving the accuracy of the rotate and flip augmentations model on the ResNet18 backbone by 4.41% (from 76.13% to 80.54%).

4.4 Experiment 4: Training Label Quality

In this experiment, a comparison between CheXpert [20] and CheXbert [34] labeling systems was performed. To do so, the ResNet18 rotate and flip model was selected. Table 4.7 displays CheXbert's [34] labels as performing slightly better under radiologist test set labels, aligning with the claims of the author that proposed CheXbert [34].

Therefore, since CheXbert [34] labels performed better and presented a closer approach to the state-of-the-art, previous experiments were remade with CheXbert [34] labels. Therefore, experiment 1 was repeated considering the lessons learned from Experiment 3, running the finetuned



(a) Accuracy on validation set

(b) Loss on validation set

Figure 4.2: Accuracy and loss plotted on a 100k steps run of a ResNet18 rotation and flip model.

Table 4.7: Difference between CheXpert [20] and CheXbert [34] on a ResNet18 model with rotation (10º) and horizontal flip pretraining augmentations . These results were calculated on the radiologist's test set and are evaluated using the accuracy metric. The finetuned models were trained for 100k steps.

	CheXpert	CheXbert
Linear	81.19	**81.29**
Finetuning	79.89	**80.66**

model for 100k steps, resulting in Tables 4.8 and 4.9 exploring ResNet18 and DenseNet121 backbones, respectively.

After discovering CheXbert [34] labels, we decided to remake previous experiments with the new labels, therefore calculating the new linear and finetuning models on tables 4.8 and 4.9.

In the finetuning approach, there was a significant increase in accuracy, which can be seen both through the average increase of accuracy on all pretraining augmentations, as ResNet18 went from 76.48% to 80.47%, while DenseNet121 raised from 76.56% to 80.54%. Additionally, the best model on ResNet18 changed from 76.85% to 80.57%, replacing the best model from solely the rotation augmentation to the one also containing horizontal flip. On the DenseNet121 backbone, although the best performance was also significantly superior (from 76.90% to 80.56%), the best model remained the rotation pretraining augmentation. Although all finetuned models reach similar performances, the ResNet18 model containing rotation and flip augmentation performs the best, being selected as the best-found model and used in further experiments.

In the linear approach, the results do not increase in the same degree. While ResNet18's average slightly increases from 68.25% to 68.46%, the best-performing model solely achieves 68.72% accuracy (in contrast with the previous 68.94%).

Regardless, when comparing the different augmentations in both linear and finetuning approaches, the differences in performance are not consistent, and the influence of each augmentation is not clear and highly influential. This is especially true in the finetuning process, where accuracy values are noticeably closer with no significant deviations from the average value. This might be due to the long training process of 100k steps that re-trains the whole network after the pretraining phase.

Note that even though this experiment focuses on comparing results between the 2 labeling approaches, the models are trained and evaluated on the same validation set but with different labels, meaning results are not directly comparable. Besides, both approaches use different amounts of finetuning training steps with no early stopping, hence the significant increase in accuracy. The comparisons should highlight general differences between different approaches and register the new scores obtained with more recent labels and optimized finetuning. Table 4.7 is where the comparison between both labeling approaches is done, since there the comparison is performed on the same test set manually labeled by the radiologists.

Table 4.8: ResNet18 linear probing and finetuning accuracy with multiple pretraining augmentations using CheXbert [34] labels. Hyperparameters used are the same as previous experiments.

Method/Augmentation	No Augmentation	Rotate	Flip	Rotate Flip	Avg
Linear	68.56	**68.72**	68.03	68.51	68.46
Finetuning	80.50	80.28	80.51	**80.57**	80.47

4.5 Experiment 5: Evaluation on Different Pathologies

In this experiment, 13 pathologies were evaluated, in contrast to previous experiments, which only evaluated "Pleural Effusion". "Support Devices" was excluded from the evaluation as it is not classified as a pathology, resulting in the assessment of the remaining 13 pathologies in the CheXpert [20] dataset.

Table 4.10 presents the results obtained with the ResNet18 model with augmented rotation and flip on all these pathologies with either linear probing or finetuning on CheXbert [34] labels. This model was selected since it was the best-performing model in pretraining. For this experiment, the multiple metrics on the table are useful for finding issues in the imbalanced dataset, which can be seen when the F1-score is low (7 pathologies have an F1-score lower than 1%), and accuracy is high.

The best accuracy results for linear and finetuned models were obtained for the "Pleural Other" pathology with the same value of 98.26%. However, when looking at Table 3.1, we can see that this pathology only contains 2.99% positive labels in CheXbert, [34] being a minority class in an imbalanced dataset. Looking at the F1-score, we can see that the value is only 0.38%, which alerts us of this problem.

When evaluating through the F1-score, "Pleural Effusion" was the best-performing model. Although it did not excel in other metrics, this pathology performed well. Moreover, when comparing the accuracy between different models, the "Lung Opacity" pathology obtained the worst results on either linear probing and finetuning, even though it is the most common pathology on both CheXpert [20] and CheXbert [34] labels (49.76% and 46.28% label occurrence). A high percentage of positive labels does not inherently make models better at predicting the correct labels.

Finetuning accuracy values are better overall than linear since all models were trained for 100k steps. This aligns with the conclusions reached in Experiment 3, further proving the importance of optimized finetuning steps.

This experiment can be further extended in the future with different hyperparameter tuning per pathology. This experiment serves as a simple initial step to evaluate the problems within the dataset. This data should be analyzed in conjunction with Table 3.1, which confirms that some pathologies contain a very low amount of positive labels, such as the "Pleural Other" observation, which makes the model biased. These challenges might be mitigated by using resampling and loss-weigthing techniques.

Table 4.9: DenseNet121 linear probing and finetuning approaches with multiple pretraining augmentations using CheXbert [34] labels. Hyperparameters used are the same as previous experiments.

Method/Augmentation	No Augmentation	Rotate	Flip	Rotate Flip	Avg
Linear	67.04	67.46	**66.85**	66.17	66.88
Finetuning	80.54	**80.56**	80.53	80.54	80.54

4.6 Experiment 6: Multilabel Classification

Afterward, a multilabel classification experiment was performed, where the model had to predict all 13 pathologies simultaneously. This model was evaluated for the finetuning approach and was tested for different amounts of steps on CheXbert [34] labels.

The label-wise accuracy values were quite similar in all experiments, showing that the amount of correctly predicted labels only changed by a small margin of 0.9% between the worst and best models. In contrast, by analyzing the exact-match ratio, we can see that this slight change in the number of correctly predicted labels significantly increases the value of this metric. This factor shows the model is improving at optimizing the overall precision rather than paying attention to the accuracy of every individual label, not being effective at capturing the nuances and patterns specific to each label. This might be due to the imbalance of this dataset since the majority class might dominate the predictions, resulting in a model that struggles with predicting the minority classes. This is another situation that might be improved with data resampling or even loss weighting to increase the weight of a minority class.

4.7 Experiment 7: Radiologists' Test Set

The best model found in each experiment was tested on the radiologists' dataset as a final experiment. This way, Table 4.12 displays linear and finetuning performance in the same way as Table 4.10, on ResNet18 backbone with flip and rotation pretraining augmentations, as described in Experiment 7.

Regarding the F1-score, "Pleural Effusion" was the best-performing model, increasing the metric value from 75.45% to 80.66% (5.21% increase). When comparing the F1-score obtained for all pathologies, no immediate correlation was found when comparing both labeling systems, as some F1-scores were increased while others lowered their value.

When considering both finetuned and linear accuracy, the best model changed from "Pleural Other" to "Fracture", with this last pathology increasing its accuracy values from 96.05% to 99.10%. The accuracy values follow the same behavior as the F1-score, with both alternate increases or decreases in performance compared to prior CheXbert [34] validation set results.

Table 4.10: Performance of trained ResNet18 rotate and flip pretrained model on 13 different pathologies on CheXbert [34] labels. The first 2 columns represent different metrics for the finetuned model trained for 100k steps, while the third column contains the accuracy for linear probing setting. Hyperparameters used are equivalent to those on the "Pleural Effusion" task. The last column represents the percentage of positive labels from CheXbert pathologies according to Table 3.1.

Pathology	Finetuning		Linear	Labels
	Accuracy	F1-score		
Atelectasis	84.74	0.33	84.74	30.56
Cardiomegaly	88.35	44.63	86.26	15.43
Consolidation	93.93	0.16	93.93	18.07
Edema	80.42	52.30	76.83	29.21
Enlarged Cardiomediastinum	96,57	0.31	96.57	10.14
Fracture	96.05	0.48	96.08	4.12
Lung Lesion	95.82	0.00	95.82	4.89
Lung Opacity	66.59	65.28	62.44	46.28
No Finding	90.82	32.55	90.44	9.46
Pleural Effusion	80.57	**75.45**	68.51	43.39
Pleural Other	**98.26**	0.38	**98.26**	2.99
Pneumonia	97.79	0.00	97.79	10.85
Pneumothorax	92.44	27.69	91.92	9.09

4.8 Conclusions

These experiments led to some interesting conclusions regarding the self-supervised scenario of the MoCo-CXR [35] paradigm. First, we started by evaluating the effect of MoCo-CXR [35] pretraining augmentations on downstream image classification tasks, concluding the effect of each augmentation would not significantly increase the model's final performance. Afterward, an analysis was performed studying the effect of different batch sizes on both linear and finetuning approaches, obtaining inconclusive results, thus selecting the values proposed by the author for further experiments.

During experiment 3, a study was conducted to investigate the optimal amount of finetuning steps on image classification tasks and their impact on model performance. This resulted in a new discovery extending the work by [2], concluding the training should resume for 100k steps instead of 5k, and the early stopping mechanism should be removed to improve accuracy (by approximately 4.41%). Subsequently, a study on the difference between CheXpert [20] and CheXbert [34] labels was performed by evaluating their results on a test set manually annotated by chest radiology professionals, concluding CheXbert [34] labels are indeed an improvement over those provided by the original published paper.

While the first 4 experiments targeted a single observation, "Pleural Effusion", the most commonly explored pathology in the dataset, the last 3 experiments investigated 12 other existing pathologies within the CheXpert [20] dataset. This includes the exploration of a multilabel sce-

Table 4.11: Multilabel finetuning classification performance with ResNet18 with rotation and flip augmentations on the pretraining phase.

Steps	Exact-match ratio	Label-wise Accuracy
5k	15.15	88.76
10k	16.33	89.08
100k	**20.44**	**89.66**

nario with a model predicting all pathologies simultaneously. The last experiment evaluates the best-found model in the radiologist's test set, assessing the performance of our models in a real-world deployment scenario.

Through all these experiments, we reached our top performance for the "Pleural Effusion" observation with a ResNet18 backbone model pretrained with a 10º Rotation and Horizontal flipping for 20 epochs and finetuned for 100k steps with an accuracy of 79.89% on the test set provided by radiologists.

Regardless, there are still many experiments to be performed in this scenario, such as a deeper analysis of different augmentations and hyperparameter tuning. Besides, our methodology can be further tested in different datasets outside the chest radiograph scenario, extending the original study from *Anton et al.* [2] and attempting to learn in which scenarios this methodology offers the best results.

Table 4.12: Performance of trained Resnet18 rotate and flip pretrained model on 13 different pathologies on the radiologists' test labels. The first 2 columns represent different metrics for the finetuned model trained for 100k steps, while the last column contains the accuracy of the linear model. Hyperparameters used are equivalent to those on the pleural effusion linear and finetuned tasks. The last column represents the percentage of positive labels on the radiologist's test set according to Table 3.1

Pathology	Finetuning		Linear	Labels
	Accuracy	F1-score		
Atelectasis	73.50	1.28	73.35	26.65
Cardiomegaly	78.89	33.74	73.80	26.20
Consolidation	94.76	0.00	94.76	5.24
Edema	88.92	50.11	87.28	12.72
Enlarged Cardiomediastinum	55.54	0.71	55.39	44.61
Fracture	**99.10**	0.00	**99.10**	0.90
Lung Lesion	97.90	0.00	97.90	2.10
Lung Opacity	76.35	67.47	69.16	46.41
No Finding	86.53	40.41	84.28	16.32
Pleural Effusion	79.89	**80.66**	81.29	17.96
Pleural Other	98.80	0.00	98.80	1.20
Pneumonia	97.90	0.00	97.90	2.10
Pneumothorax	97.46	15.01	98.50	1.50

Chapter 5

Conclusions

The DL scenario for medical imaging has been greatly dominated by SL approaches, requiring large amounts of labeled data, an expensive task that interrupts the medical workflow. To work around these limitations, SSL has emerged as an alternative for CV tasks such as image classification. Moreover, SSL has different architectures with several models with optimal use cases, concluding that contrastive learning is the model that is best suited for image classification tasks.

Regarding the application of SSL algorithms in the medical field, studies are limited; however, approaches such as MICLe for datasets with multiple views of the same image have presented promising results in the area, specifically in the analysis of chest X-rays. The limited existing use cases for medical image classification make room for improvement by trying different approaches, mixing the positive results obtained from these different works on different datasets, and performing a comparative analysis through the use of the traditional metrics for classification problems, such as accuracy and F1-score.

To further explore this scenario, we conducted a study on MoCO-CXR, [35] with the initial objective of examining the impact of each intrinsic characteristic of the algorithm, aiming to identify and, if possible, mitigate any shortcomings associated with the algorithm. MoCo-CXR [35] is an adaptation of the MoCo [17] method specifically designed for the task of learning representations from chest radiographs on the CheXpert [20] dataset. It aims to extract meaningful representations from large amounts of data in an unsupervised fashion through a contrastive learning approach. The selected dataset, CheXpert, [20] contains 224,316 chest radiographs of 65,240 patients automatically labeled for the presence of 14 common pathologies.

To perform this study, we started by adapting the work from MoCo-CXR [35] on the pretraining phase and integrate it with the downstream image classification proposed by [2], who consider 2 different evaluation methodologies, linear probing, and finetuning. On linear probing, the selected pretrained model is frozen and leveraged as a fixed feature extractor with a multinomial logistic regression fitted on top of the fixed features. When finetuning, all pretrained parameters are refitted, along with an attached linear classification head.

The proposed methodology was then applied to the CheXpert [20] dataset through 2 different labeling systems, CheXpert [20] and ChexBert [34]. The former refers to the default labeling

system proposed by CheXpert's authors [20], while the latter is a state-of-the-art improvement. While both approaches are generated using a rule-based system that extracts information from radiology reports, CheXbert [34] undergoes additional processing with a pretrained BERT [11] model and finetuning with CheXbert [34] annotations. Furthermore, we make use of a third dataset with manually annotated labels from professional radiologists.

This way, 7 experiments were performed. The first 4 focused on the "Pleural Effusion" pathology, since it was the main focus of the authors [35; 2], while the last 3 experiments extend the study to the remaining 13 observations, excluding "Support Devices", as it was not considered a pathology.

We started by evaluating the influence of 4 MoCo-CXR [35] pretraining augmentations on downstream image classification models, both on linear and finetuning approaches, as proposed by [2]. In this experiment, ResNet18 and DenseNet128 backbone architectures were used and no significant difference was found between the selected augmentations. As a next step, we studied the influence of batch size on the same scenario, starting from 16 and reaching up to 128, with no significant correlation between batch size and performance being discovered.

Experiment 3 investigated the influence of the number of learning steps in downstream image classification finetuning, with the removal of the early stopping mechanism implemented by the authors, leading to the conclusion that further improvements of 4.5% accuracy could be achieved by training by 100k steps instead of 5k, reaching the optimal value for the algorithm. Furthermore, experiment 4 was conducted to confirm that CheXbert [34] labels would lead to improved results when compared to the default CheXpert [20] dataset labels, by testing the performance of the previous best model (ResNet18 with 10º rotation and horizontal flip augmentations). It was indeed proved that CheXbert [34] obtained slightly better accuracy improvements on both linear and finetune approaches. Therefore, we decided to use CheXbert [34] labels for the remaining experiments.

The last 3 experiments conducted a study on the influence of multiple labels, selecting the best "Pleural Effusion" performing model's pretraining augmentations and hyperparameters as a base for this study. The first of these experiments individually evaluated a linear and finetuning model for each pathology, reaching an accuracy of 98.26% on both classification tasks for "Pleural Other". Even though these accuracy values are high, they might be associated with some bias resulting from imbalanced datasets. Afterward, an experiment was made considering a single model with a multilabel task simultaneously predicting all pathologies, being evaluated with 20.44% exact-match ratio and 89.66% label-wise accuracy. The final experiment solely evaluated the best model for all pathologies on the radiologists' test set, with the objective of assessing the performance of our findings in a real-world deployment scenario.

Although we were able to conduct these 7 experiments and perform an in-depth exploration of MoCO-CXR, [35] there are still unexplored aspects that can identify and mitigate shortcomings of this methodology. A future study can be conducted investigating the influence of the number of epochs on the pretraining phase, as our selected value was 20, aligning with the work from MoCo-CXR [35] authors. Besides, the CheXpert [20] dataset is imbalanced, and several measures

can be taken to attempt to mitigate this issue, either by experimenting with resampling and loss weighting techniques on the minority classes. It would also be interesting to explore how the amount of pretraining images influences the performance of MoCo-CXR [35] on the downstream image classification task.

References

[1] F. Altaf, S. M. Islam, N. Akhtar, and N. K. Janjua. Going deep in medical image analysis: concepts, methods, challenges, and future directions. *IEEE Access*, 7:99540–99572, 2019.

[2] J. Anton, L. Castelli, M. F. Chan, M. Outters, W. H. Tang, V. Cheung, P. Shukla, R. Walambe, and K. Kotecha. How well do self-supervised models transfer to medical imaging? *Journal of Imaging*, 8(12):320, 2022.

[3] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[4] S. Azizi, B. Mustafa, F. Ryan, Z. Beaver, J. Freyberg, J. Deaton, A. Loh, A. Karthikesalingam, S. Kornblith, T. Chen, et al. Big self-supervised models advance medical image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3478–3488, 2021.

[5] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

[6] K. Chaitanya, E. Erdil, N. Karani, and E. Konukoglu. Contrastive learning of global and local features for medical image segmentation with limited annotations. *Advances in Neural Information Processing Systems*, 33:12546–12558, 2020.

[7] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[8] X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[9] X. Chen and K. He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[13] J. Donahue and K. Simonyan. Large scale adversarial representation learning. *Advances in neural information processing systems*, 32, 2019.

[14] L. Ericsson, H. Gouk, and T. M. Hospedales. How well do self-supervised models transfer? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5414–5423, 2021.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[16] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[17] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] O. Henaff. Data-efficient image recognition with contrastive predictive coding. In *International conference on machine learning*, pages 4182–4192. PMLR, 2020.

[20] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 590–597, 2019.

[21] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

[22] L. Jing and Y. Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):4037–4058, 2020.

[23] A. E. Johnson, T. J. Pollard, S. J. Berkowitz, N. R. Greenbaum, M. P. Lungren, C.-y. Deng, R. G. Mark, and S. Horng. Mimic-cxr, a de-identified publicly available database of chest radiographs with free-text reports. *Scientific data*, 6(1):317, 2019.

[24] G. Li, R. Togo, T. Ogawa, and M. Haseyama. Self-knowledge distillation based self-supervised learning for covid-19 detection from chest x-ray images. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1371–1375. IEEE, 2022.

[25] G. Li, R. Togo, T. Ogawa, and M. Haseyama. Tribyol: Triplet byol for self-supervised representation learning. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3458–3462. IEEE, 2022.

[26] J. Liu, G. Zhao, Y. Fei, M. Zhang, Y. Wang, and Y. Yu. Align, attend and locate: Chest x-ray diagnosis via contrast induced attention network with limited supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10632–10641, 2019.

[27] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):857–876, 2021.

[28] I. Misra and L. v. d. Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6707–6717, 2020.

[29] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[30] P. Pandey, A. Pai, N. Bhatt, P. Das, G. Makharia, P. AP, et al. Contrastive semi-supervised learning for 2d medical image segmentation. *arXiv preprint arXiv:2106.06801*, 2021.

[31] M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio. Transfusion: Understanding transfer learning for medical imaging. *Advances in neural information processing systems*, 32, 2019.

[32] T. Rahman, A. Khandakar, Y. Qiblawey, A. Tahir, S. Kiranyaz, S. B. A. Kashem, M. T. Islam, S. Al Maadeed, S. M. Zughaier, M. S. Khan, et al. Exploring the effect of image enhancement techniques on covid-19 detection using chest x-ray images. *Computers in biology and medicine*, 132:104319, 2021.

[33] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017.

[34] A. Smit, S. Jain, P. Rajpurkar, A. Pareek, A. Y. Ng, and M. P. Lungren. Chexbert: combining automatic labelers and expert annotations for accurate radiology report labeling using bert. *arXiv preprint arXiv:2004.09167*, 2020.

[35] H. Sowrirajan, J. Yang, A. Y. Ng, and P. Rajpurkar. Moco pretraining improves representation and transferability of chest x-ray models. In *Medical Imaging with Deep Learning*, pages 728–744. PMLR, 2021.

[36] A. Taleb, C. Lippert, T. Klein, and M. Nabi. Multimodal self-supervised learning for medical image analysis. In *Information Processing in Medical Imaging: 27th International Conference, IPMI 2021, Virtual Event, June 28–June 30, 2021, Proceedings*, pages 661–673. Springer, 2021.

[37] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

[38] Y. Tian, D. Krishnan, and P. Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020.

[39] M. Tschannen, J. Djolonga, M. Ritter, A. Mahendran, N. Houlsby, S. Gelly, and M. Lucic. Self-supervised learning of video-induced visual invariances. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 13806–13815, 2020.

[40] A. Verma, V. Murali, R. Singh, P. Kohli, and S. Chaudhuri. Programmatically interpretable reinforcement learning. In *International Conference on Machine Learning*, pages 5045–5054. PMLR, 2018.