Evaluating Pre-training Mechanisms in Deep Learning Enabled Tuberculosis Diagnosis

School of Computer Science & Applied Mathematics
University of the Witwatersrand

Zororo Zaranyika
2544888

Supervised by Professor Richard Klein

May 30, 2024

Ethics Waiver Number: CSAM-2023-1W

A research report submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in partial fulfilment of the requirements for the degree of Master of Science by Course Work and Research Report in the field of Computer Science

Abstract

Tuberculosis (TB) is an infectious disease caused by a bacteria called Mycobacterium Tuberculosis. In 2021, 10.6 million people fell ill because of TB and about 1.5 million lives are lost from TB each year even though TB is a preventable and curable disease. The latest global trends in TB death cases are shown in 1.1. To ensure a higher survival rate and prevent further transmissions, it is important to carry out early diagnosis. One of the critical methods of TB diagnosis and detection is the use of posterior-anterior chest radiographs (CXR). The diagnosis of Tuberculosis and other chest-affecting diseases like Pneumoconiosis is time-consuming, challenging and requires experts to read and interpret chest X-ray images, especially in under-resourced areas. Various attempts have been made to perform the diagnosis using deep learning methods such as Convolutional Neural Networks (CNN) using labelled CXR images. Due to the nature of CXR images in maintaining a consistent structure and overlapping visual appearances across different chest-affecting diseases, it is reasonable to believe that visual features learned in one disease or geographic location may transfer to a new TB classification model. This would allow us to leverage large volumes of labelled CXR images available online hence decreasing the data required to build a local model. This work will explore to what extent such pre-training and transfer learning is useful and whether it may help decrease the data required for a locally trained classifier. In this research, we investigated various pre-training regimes using selected online datasets to understand whether the performance of such models can be generalised towards building a TB computer-aided diagnosis system and also inform us on the nature and size of CXR datasets we should be collecting. Our experiment results indicated that both supervised and self-supervised pre-training between the CXR datasets cannot significantly improve the overall performance metrics of a TB. We noted that pre-training on the ChestX-ray14, CheXpert, and MIMIC-CXR datasets resulted in recall values of over 70% and specificity scores of at least 90%. There was a general decline in performance in our experiments when we pre-trained on one dataset and fine-tuned on a different dataset, hence our results were lower than baseline experiment results. We noted that ImageNet weights initialisation yields superior results over random weights initialisation on all experiment configurations. In the case of self-supervised pre-training, the model reached acceptable metrics with a minimum number of labels as low as 5% when we fine-tuned on the TBX11k dataset, although slightly lower in performance compared to the supervised pre-trained models and the baseline results. The best-performing self-supervised pre-trained model with the least number of training labels was the MoCo-ResNet-50 model pre-trained on the VinDr-CXR and PadChest datasets. These model configurations achieved recall scores of 81.90% and a specificity score of 81.99% on VinDr-CXR pre-trained weights while the PadChest weights scored a recall of 70.29% and a specificity of 70.22%. The other self-supervised pre-trained models failed to reach scores of at least 50% on both recall or specificity with the same number of labels.

Declaration

I, Zororo Zaranyika, hereby declare the contents of this research report to be my own work. This report is submitted for the degree of Master of Science by (Coursework and Research Report) in the field of Computer Science at the University of the Witwatersrand. This work has not been submitted to any other university, or for any other degree.

Zororo Zaranyika
May 30, 2024

Contents

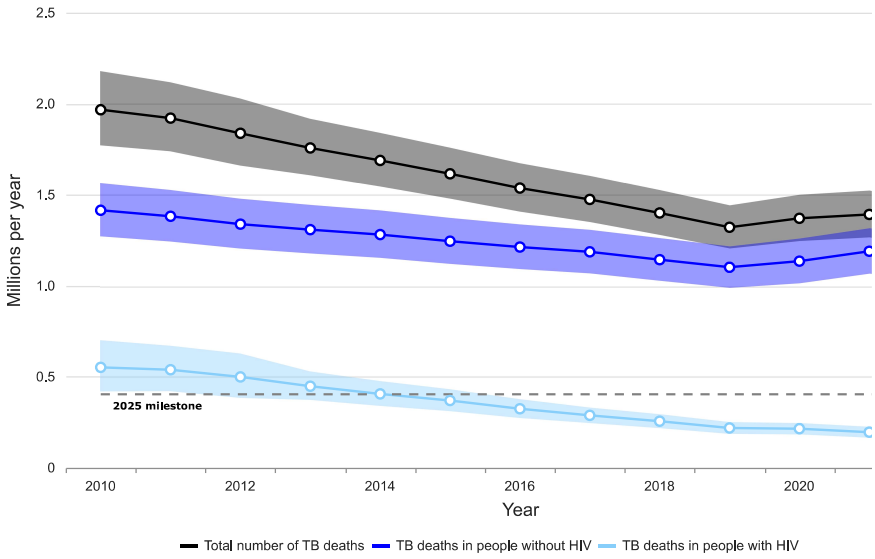
List of Figures

List of Tables

Chapter 1

Introduction

In medical diagnosis, the use of X-rays is popular to examine lung infections such as Tuberculosis (TB), Pneumonia, and other chest-affecting diseases. The procedures of examining chest X-ray images (CXR) by medical practitioners are cheaper in comparison to other techniques including Computed Tomography (CT) and Magnetic Resonance Imaging (MRI). The implementation of the X-ray is simple and cost-effective. Hence, most preliminary lung screening considers X-ray images. Several methods of carrying out computer-based TB screening in chest X-ray images have been considered by researchers of which deep learning is one of them. Deep learning (DL) methods such as CNNs continue to dominate this field and they can effectively carry out the diagnosis of chest-affecting abnormalities with a higher degree of precision when compared to the medical experts. It should be noted that though these methods are considered accurate, they cannot replace the work of medical practitioners such as radiologists or pulmonologists. The results from DL TB classifiers can be used for screening and triage. They have proven to help provide a substitute or alternative diagnosis solution [Biju *et al.* 2022]. It should be noted that the acceptable level of performance on any TB classifier is a sensitivity of over 90% and a specificity of over 70% as per World Health Organisation (WHO) guidelines[WHO and others 2021].



(b) Rate per 100 000 population

Figure 1.1: Global annual mortality trends from 2010 to 2022 as a result of TB [WHO and others 2022]

To increase the detection accuracy of TB from chest X-rays, various techniques can be used to enhance classification accuracy. Researchers have suggested ways of improving deep learning models by improving algorithms or combining several top-ranked models to form hybrid architectures. One of the ways of dealing with limited data is the use of augmentation techniques. To further improve the accuracy of detection, segmented chest X-ray images can substantially improve the overall model prediction level by forcing the model to focus and predict on a section of the lungs, not the whole chest X-ray. Therefore it becomes important to isolate other regions from the CXR image [Biju *et al.* 2022].

There is a major constraint in the availability of locally sourced and reliable labelled CXR datasets due to the higher cost of getting labels. With the abundance of rich international datasets at our disposal we need to understand whether transfer learning between these datasets can inform us on dataset attributes and size we should be focusing on when building CAD models for TB diagnosis. Transfer learning has become dominant in computer vision and deep learning for solving image classification problems, especially in cases where image labels are difficult to find. The idea behind transfer learning is to use a pre-trained model that has solved a similar task to perform a new classification task or use the weights from such model. The weights could have been trained on huge datasets such as ImageNet. We then perform the process of fine-tuning on another model by further training on a smaller targeted dataset. The weights will save time and computer resources during the training of the new model. Deep learning models pre-trained on other datasets have been shown to bring great potential and accuracy in various lung disease classification tasks [Ravi *et al.* 2022].

With the above ideas highlighted, we aim to answer the following questions with the idea of providing insight into CXR dataset collection for building a TB diagnosis framework.

- Suppose we pre-train a model on chest X-ray datasets containing any other chest-affecting condition seen in the population, how does the model perform when we save the weights and fine-tune the same model on another dataset to classify TB? How does the model perform in terms of accuracy, the F1-Score, the area under the curve, precision, specificity and sensitivity?

- In a scenario where we have different chest X-ray TB datasets that differ in terms of geographical origin and size if we pre-train models and save the weights using supervised techniques on a dataset that originates from a different region, do we get better metrics or predictions if we fine-tune on another dataset that was sourced from a different geographical location? If this experiment yields good results, what properties make the pre-trained weights perform better when we consider the origins of these datasets?

- We will investigate the impact of pre-training given a small number of labelled TB CXR images on the model performance in the diagnosis of TB using self-supervised learning models by gradually removing the number of labels from the dataset. This will allow us to determine the minimum number of labels required to get optimal prediction performances. Self-supervised learning approaches have proven to be successful in natural image classification hence it is important to investigate how these methods can perform and generalise when encountered with CXR datasets with limited labels.

The purpose of this research is to understand whether various pre-training regimes such as pre-training on international datasets, different chest-affecting diseases and using limited labelled datasets in self-supervised models will help us in building a CAD system for TB detection. We strongly believe that the consistent structure of CXR images will have an overlapping visual representation that is likely to span across different chest-affecting ailments hence pre-training on these online datasets will likely to enhance the model learning via transfer learning as we transition from one dataset to another. Additionally, we should be able to understand how transfer learning can inform us on the size and attributes of datasets that we should collect to facilitate the building of a TB diagnosis CAD system. The structure of this research report is as follows, In Chapter 1 we introduced the problem domain and the questions that we aim to answer. In Chapter 2, we described the important background concepts that are necessary to understand the research problem area. Chapter 3 introduces the design of the research, the tools and the methods that were used for implementation of the models. In Chapter 4 we presented the results of our experiments. Finally, Chapter 5 gives a summary view of all the findings, the contributions of this research, and future work.

1.1 Literature Review

In this section we summarise the work that has been done using deep learning methods to classify TB. Various methods have been developed to minimise the diagnostic overload on medical institutions to ensure an accurate and early diagnosis of TB. Various methods for tuberculosis detection were highlighted in this section [Mohan *et al.* 2022].

Recent studies confirmed that deep learning methods support TB detection using X-ray images. Several publicly available data sources were consolidated to form a database with 700 TB-positive and 3500 TB-negative CXR images. These sources included the National Library of Medicine (USA), the NIAID TB dataset, the Belarus dataset and the RSNA CXR dataset [Rahman *et al.* 2020b]. The findings of the study indicated precision, accuracy and recall values of 96.62%, 96.47%, 96.47% respectively. These values were obtained from experiments in which image segmentation was not applied. When they applied image segmentation they obtained scores of 98.57%, 98.6%, 98.56% for precision, accuracy and recall respectively. These results indicated that segmentation can increase the classifier performance.

In another study by An *et al.* [2022], they built a mobile-friendly model for TB detection ideal for personal computers or embedded medical devices. As part of this research they used The Shenzhen, China dataset (CHN) and the Montgomery dataset [Jaeger *et al.* 2014] and employed a custom-built deep learning model based on ResNet consisting of 5 basic blocks and Efficient Channel Attention (ECA) blocks, Global Average pooling (GAP), Drop Out, Fully Connected Layer (FC) (1024), Drop out Layer and another Fully Connected (2) and a SoftMax classification layer. The performance of E-TBNet was low compared to other mobile models like MobileNetV2, and SqueezeNet1. The authors argued that their model did not perform well due the absence of pre-training. The authors mentioned that the E-TBNet was easy to deploy on the mobile device which can present a possible solution to under-resourced areas.

As part of the research by Zaidi *et al.* [2022], they built a custom CNN that does not use a pre-trained CNNs but rather focuses on training on a bigger NIH chest X-ray dataset which is made up of over 112 000 chest X-ray images from more than 30 000 unique patients. Using Custom Convolutional Layers, Inception ResNetV2 (combining high performance characteristics from Inception network and residual connections of ResNet network). The architecture consisted of a classification layer which was used in a 3 phase pattern thus two neuron classification layer for either TB or health, another 10 neurons layer to predict the 10 TB manifestations including infiltration, pleural thickening, fibrosis, pneumonia, pneumothorax, consolidation, emphysema, nodule, atelectasis, effusion as well as another 15 neuron classification layer for the rest of the fourteen manifestations. These are atelectasis, effusion, nodule, pneumonia, pneumothorax, consolidation, emphysema, fibrosis, pleural thickening, infiltration, cardiomegaly, mass, edema, and hernia. After the computation of precision, recall, and F1-Score on the model for the three different classification phases, the model predicted

average scores of over 80%, while the overall model achieved scores of: (95%, 88%, 92%) for precision, recall, and F1-Score respectively which was higher in performance in comparison to three radiologists [Zaidi *et al.* 2022].

In an attempt to deal with limited labels in datasets for chest X-ray Tuberculosis classification, Park *et al.* [2022] suggested a method that makes use of unlabelled X-ray images in TB diagnosis called Distillation for self-supervised and self-train learning (DISTL). The model makes use of semi-supervised learning, self-supervised or self-training with knowledge sharing between the teacher network and the student network using a Vision transformer (ViT). The core idea of DISTL is to train an initial model with a limited number of labelled data and then use the initial model as the teacher with small initial data for correction, the student network is trained with increasing unlabelled data over time (T). The aim is to simulate real-world clinical environments in which new unseen data increases over time Park *et al.* [2022]. The framework was used in performing three CXR tasks including the diagnosis of Tuberculosis by gradually increasing the number of unlabelled data at different time steps (T). In comparison to other CNN-based algorithms, the Vision Transformer-based model demonstrated a significant increase in performance. In the final iteration at T=3, the DISTL-based framework was used in classifying CXR images between TB positive and normal classes, it attained the classification performance with the following metrics thus a sensitivity of 95.0%, specificity of 97.3%, an AUC of 0.98 and an accuracy of score of 94.0%.

Motivated by the idea of building a highly accurate CAD TB Screening model using X-rays targeting low-income countries with limited medical professionals, Dasanayaka and Dissanayake [2021] proposed a model based on the following models, a Deep Convolution Generative Adversarial Network (DC GAN), U-Net (Segmentation), Deep Convolutional Neural Network (DCNN), VGG-19 and InceptionV3. They claimed that their proposed model achieved an accuracy of 97.1%, a sensitivity is 97.9%, and a specificity of 96.2% attributed to the use of SOTA models such as U-Net, VGG-19, the DC GAN, and InceptionV3 coupled with segmentation, augmentation and hyper-parameter tuning techniques.

From the various research papers highlighted above, it is apparent that a substantial amount of work must be targeted to building various deep-learning algorithms that can potentially change the landscape of automated TB classification. These algorithms mainly focus on enhancing classification performance metrics. While the development and refinement of the architectures themselves are important, it is equally important to consider how the data and pre-training regimes affect the accuracy and generalisability of these models. Little work has been done considering the effects of the origin of the data, transferability between datasets, and self-supervised pre-training especially when it comes to low data environments where minimal data may exist for actual deployment in a clinical location.

1.2 Problem Statement

In this section, we discuss the problem that this research report seeks to address. Suppose we want to build a deep learning model for the diagnosis of TB using chest X-ray images when we have limited labelled images, especially in South Africa. In these scenarios, it is important to consider model architectures as well as the underlying algorithms. Should we be focusing only on self-supervised approaches, or should we consider models that use both self-supervised and supervised algorithms? It is equally important to investigate the sizes and the underlying data attributes of the CXR datasets that we should be using for the various pre-training scenarios. Should we be using these international datasets in the first place? Does transfer learning work when we consider pre-training on international datasets and then fine-tuning on local datasets?

In this work we explored how models behave differently when presented with datasets that differ in terms of disease but originate from the same population. We pre-trained on CXR datasets that contain multiple labels for chest-affecting conditions (ChestX-ray14) and then fine-tuned on a model that predicts TB using a different dataset (TBX11K). The ChestX-ray14 consists of multiple labels including chest affecting diseases such as Pneumonia and Emphysema. Our aim in this question was to find out if there is any model performance benefit by transfer learning from other chest-affecting conditions to detect TB.

We investigated the scenario where a model designed to classify the presence or absence of TB in X-ray images is pre-trained on a larger international CXR dataset, does it yield different result metrics when we save the weights and use those weights in fine-tuning on a smaller dataset? Our aim in this question was to investigate if there is any benefit in using larger datasets for pre-training especially in TB classification thus to find out if the size of the pre-training CXR dataset impacts the final results when fine-tuning on a smaller dataset. In this investigation we pre-trained on each of the following: the CheXpert dataset with 224 316 CXR images and the MIMIC-CXR dataset (377 110 CXR images). We fine-tuned on the TBX11K dataset which contains 11 200 CXR images.

We considered pre-training and testing our models with datasets that originate from different countries to check whether transfer learning will yield better classification results based on data geographical differences. We wanted to investigate if the diversity and origin of the CXR images affect the performance outcomes of a TB classification model. In this scenario we used the VinDr-CXR dataset of Vietnamese origin and the PadChest dataset of Spanish origin for pre-training and fine-tuned on the TBX11K dataset.

We further investigated how self-supervised models perform when fine-tuned with limited labelled data. We applied self-supervised pre-training on international datasets using a contrastive learning-based algorithm (MoCO) and a distillation-based self-supervised learning algorithm (DINO). We saved the model weights and fine-tuned this model with another subset of the same dataset or a different dataset. The aim is to inves-

tigate whether self-supervised algorithms (DINO and MoCO) can learn any important patterns in the images that will help us to solve the challenge of TB classification.

In all the above scenarios, we want to assess the evaluation metrics on both an "in-distribution test set" (sampled from the same dataset) and an "out-of-distribution test set" (sampled from another dataset constructed in a different country). This research is the first part of a bigger assignment to build a TB diagnosis CAD system and will assist in informing us on the nature, data attributes and size of the South African chest X-ray data that we should collect. Our assignment was to investigate how these pre-training regimes can assist us in collecting datasets and building models for the South African context. These models should aim to achieve the same levels of TB prediction accuracy as those from international datasets.

1.3 Research Aims and Objectives

In the previous chapter, we gave an overview of the main questions that we aimed to answer. Building on the problems presented in this section we briefly describe and state the contributions that this research will make as well as identify and state respective objectives that will enable us to achieve our aim. We highlight some of the limitations that we encountered in this research report.

1.3.1 Research Aims

This research aimed to investigate and conduct experiments on the various pre-training configurations stated in section 1.3. We aimed to provide insight into the size and nature of CXR data collection and to devise a TB reference framework that can be used when pre-training TB classification models. This could be useful for building a CAD system that works for TB diagnosis in any population. In this report, we aimed to provide information related to classifying TB both in the case of labelled and limited labelled CXR images by the use of deep-learning techniques.

1.3.2 Objectives

Having identified the scope of our research aims, we intend to accomplish the following tasks:

1. To find reputable and labelled chest X-ray datasets preferably with labels for TB, Pneumonia or any other related chest-affecting disease.
2. Choose and decide on reputable models to run experiments on.
3. Conduct pre-training on the relevant TB classification models.
4. Fine-tune the models on datasets that differ in terms of size and origin.
5. Run experiments and record results to answer the hypothesised questions.

1.4 Research Questions

The performance of the best computer vision classifiers is affected by how much transfer learning has been done [Newell and Deng 2020]. Transfer learning is not a new idea and has been used extensively in most high-performing algorithms. Due to the limited nature of labels in CXR datasets and the high cost of getting labels for these datasets in the South African context, we seek to investigate how these international datasets can benefit local models via transfer learning. This should help us to understand the data attributes and size we should be collecting to solve the problem of TB diagnosis. The question that we tried to address in this research was: What kind of chest X-ray dataset should we use in terms of the labels, attributes and size? What algorithms should we be using to build TB deep learning model classifiers? Building on the ideas above, we narrowed down the bigger question by stating the following sub-questions:

In the context of TB diagnosis, if we pre-train a model on any other condition that affects the chest such as Pneumonia, Lung Cancer, COVID-19, Pneumothorax or Pneumoconiosis. We wanted to find out to what extent does the pre-training improve the results of the model relative to random initialisation and ImageNet initialisation. In this case, we wanted to understand whether transferring knowledge obtained from a related task will result in improved accuracy when conducting a TB diagnosis.

Secondly, suppose we have two labelled chest X-ray datasets that significantly differ in size or origin. Suppose we pre-train the model on the bigger dataset from another location (Europe) and then freeze the weights. What kind of prediction accuracy do we get if use the pre-trained weights and fine-tune on a smaller TB chest X-ray dataset that was collected from a different region (Asia). Ultimately we wanted to investigate whether transfer learning could yield any significant performance when we consider pre-training on a dataset from one region of the world and fine-tuning on a smaller dataset from another region.

In the case that these datasets transfer well between themselves, it is important to investigate how self-supervised learning can assist us in TB CXR image classification. Therefore, the last question that this research aimed to investigate was the impact of pre-training given a poorly labelled chest X-ray dataset. Provided a chest X-ray dataset that consists of mostly unlabelled images, to what extent does contrastive and distillation-based self-supervised pre-training improve the accuracy of a downstream TB classifier and to what extent do we see transfer between different regions of the world? How will this impact the TB classification performance of the model when we transfer weights and knowledge by fine-tuning the model on a different dataset with a smaller number of labels? The dataset for fine-tuning can be sourced from a different region or can be set aside as a test set from the same dataset?

1.5 Conclusion

In this chapter, we have introduced the current state of affairs as far as using CXR images and deep learning for TB diagnosis. We highlighted recent literature in how CXR images are used together with deep learning for the diagnosis of chest-affecting diseases including TB. We have indicated the main research questions that this research seeks to answer. Finally, we have summarised the aims and objectives of this research report.

Chapter 2

Background and Related Work

In the previous chapter we presented the research area and the problem we want to solve. This chapter focuses on key concepts that will enable us to carry out this research. In section 2.1, we highlight some of the recent deep learning models that have been developed in recent years to facilitate TB diagnosis. In section 2.2 we present some of the publicly available datasets that have been used in this study. In section 2.4 we discuss the concept of image segmentation and explore the U-net architecture. Lastly, section 2.4 describes self-supervised learning and highlights two examples of self-supervised algorithms.

2.1 Deep learning models in TB Diagnosis

Deep learning has gained a lot of attention, especially in tasks that involve medical image classification and radiology. The use of Convolutional Neural Networks plays a crucial function in learning deep features and feature extraction. Pre-trained CNN models have become useful in the classification of TB using chest X-ray images. The basic building blocks of CNN contain three base layers which are, firsly the convolutional layer, followed by the pooling layer and the linear layers [Krishna Puttagunta and Ravi 2021].

Deep Convolutional Neural Networks are popular due to their ability to get robust results when performing image classification tasks. The convolutional layers of the network and the filters assist in learning deep image features. Transfer learning has become useful in TB classification using CNN when datasets are small. Lately, transfer learning has been effectively used in various areas such as medical imaging, manufacturing and baggage screening in logistics. Transfer learning allows the training of CNNs with smaller datasets and significantly reduces the time required for training the models. Transfer learning has been proven to be a better alternative to training from scratch [Rahman *et al.* 2020b].

Prominent pre-trained deep learning CNNs include DenseNet201, InceptionV3, ResNet18, SqueezeNet, VGG-19, ResNet-50, ResNet101, ChexNet, and MobileNetV2. These are widely used for TB classification through chest X-ray images. Most of these CNNs except ChexNet were originally pre-trained on the ImageNet dataset. ResNet was originally developed to solve the vanishing gradient problem. The ResNet model has several variations such as 18, 50, 101, and 152 depending on the number of layers in the network. ResNet has recorded great success in medical image classification using transfer learning. The ordinary deep CNN layers learn by observing the low or high-level image features when training the model while on the contrary, ResNet learns image residuals[Rahman *et al.* 2020b].

Recently, Xu and Yuan [2022], suggested a CAD TB diagnosis system named VGG-19-CoordAttention. Their method entailed using a modified VGG-19 base model which uses the coordinated attention mechanism. Their model was compared with a Vision-Transformer, VGG-19, ResNet-50 and MobileNet-V2 to determine the most effective model to classify TB. The training involved freezing each of the models for about 30 training epochs and then unfreezing for the remaining 90 training runs. The Shenzhen dataset was used for the experiments and they got an accuracy of 92.73%, a precision of 92.73%, an F1-score of 92.82%, and an area under the curve of 0.9771. The results affirmed that using the attention mechanism was an effective way of improving TB diagnosis.

Showkatian *et al.* [2022] implemented a Deep Convolutional Neural Network (DCNN) model called ConvNet which was trained from the ground up to detect TB using CXR images. They conducted transfer learning using ResNet-50, Inception-V3, Xception, VGG-19 and VGG-19. The metrics from these models were used to evaluate the effectiveness of their suggested technique against each model. They used a dataset containing CXR images obtained from the Montgomery and Shenzhen datasets. The proposed deep learning architecture scored a precision of 88.0% and an overall score 87.0% for sensitivity, F1-score, accuracy and area under the curve (AUC).

In attempting to do multi-disease classification, Kim *et al.* [2022] implemented EfficientNet-V2M coupled with transfer learning. Their target was to categorize CXR images into three classes including pneumothorax, pneumonia and normal. Pre-processing was applied to images collected from the ChestX-ray14 dataset. The EfficientNet-V2M model achieved a mean accuracy of 82.15%, a sensitivity of 81.40%, and a specificity of 91.65%. In another experiment with a dataset from Cheonan Soonchunhyang University Hospital, their model scored an accuracy of 82.20% for the classification of images into four classes thus pneumothorax, normal, pneumonia, and tuberculosis.

In these related studies, we note that most authors have used the power of improved algorithms to enhance the prediction accuracy of their models. Our study makes use of similar model architectures but it highlights the fact that not much is known in terms of how these models scale when pre-trained on TB datasets that differ in terms of origin, structure and attributes. We aim to leverage on the abundance of online CXR datasets

of which CXR images are a good candidate for use in deep learning as they present consistency features. The features are likely to be present in datasets that originate from the same geographical regions and images from the same family of chest-affecting diseases. We believe that if models can learn from these datasets, hence local models can benefit from transfer learning with smaller CXR datasets. This can result in saving computer resources by reducing training times when we consider building models for local use.

2.2 Existing Chest X-ray Datasets

In this section we briefly describe some of the chest X-ray image datasets that are publicly available. In many cases, researchers cannot publish some of the datasets due to ethical and privacy reasons. Despite these challenges, a lot of data is still required due to the data-hungry nature of deep learning. Most models are trained and tested on an ensemble of various datasets. The datasets consist of chest X-ray images and, in other cases other information (metadata) to give details about the patient. Metadata may include age, race, gender, and dimensions of the images [Johnson *et al.* 2019].

2.2.1 ChestX-ray8 and ChestX-ray14

This dataset is known as ChestX-ray8 comprised of 108 948 frontal CXR images obtained from 32 717 patients. Eight class labels were identified in the dataset including atelectasis, mass, pneumonia, infiltration, nodule, pneumothorax, cardiomegaly, and effusion. It is indicated that the labels were mined by a Natural Language Processing algorithm and can be prone to some errors [Wang *et al.* 2017]. The ChestX-ray14 dataset was an enhancement of the ChestX-ray8 consisting of six additional chest problems including fibrosis, pleural thickening, consolidation, emphysema, hernia and edema. This dataset is made up of 112 120 CXR images with 51 708 images containing one or more problems and the other 60 412 images did not contain any of the fourteen chest conditions. The images were acquired from 30 805 patients [Wang *et al.* 2017].

2.2.2 Padchest (Pathology Detection in Chest Radiographs)

The Padchest dataset is amongst others considered the largest and most well-labelled public datasets, consisting of 168 861 chest X-ray images collected from 67 000 individuals who visited the Spanish-based San Juan's Hospital. The radiology reports identified up to 174 labels and 19 diagnosis varying to over 104 different chest anatomy locations.The dataset was made in collaboration with contributions from 18 radiologists [Bustos *et al.* 2020].

2.2.3 CheXpert

CheXpert is one of the largest datasets available to the public consisting of 224 316 chest X-ray images collected from 65 240 individuals. A total of fourteen common

chest abnormalities were noted and this data was consolidated from the Stanford Hospital between the years 2002 and 2017. Every CXR image in the dataset was labelled for the presence or absence of 14 conditions as either negative or positive or unknown using a systematic labelling algorithm that was used to take out the labels from the radiologist free text reports. The labels in the dataset include no finding, enlarged cardiomediastinum, cardiomegaly, lung lesion, edema, lung opacity, consolidation, pleural effusion, pneumothorax, pleural Other, pneumonia, atelectasis, fracture and support devices[Irvin *et al.* 2019].

2.2.4 MIMIC-CXR Dataset

This data is made up of 377 110 CXR images from a total of 227 835 patients. This dataset is said to be one of the biggest online CXR datasets with free-text radiologist reports. It contains fourteen chest abnormalities labels including enlarged cardiomediastinum, lung opacity, lung lesion, edema, consolidation, pneumonia, atelectasis, support devices, pneumothorax, cardiomegaly, pleural effusion, fracture, pleural Other and no finding. The dataset was consolidated by the Boston-based Beth Israel Deaconess Medical Center from the USA from 2011 up-to 2016 [Johnson *et al.* 2019].

2.2.5 VinDr-CXR Dataset

VinDr-CXR dataset contains annotations curated by radiologists and is publicly available. It is made up of 18 000 chest X-ray images that contain the location in the chest as well as the classification of the abnormality. In total the dataset included 28 labels including consolidation, edema, pneumonia, rib fracture, Pleural effusion, infiltration, interstitial lung disease, lung cavity, lung cyst, chronic obstructive, no finding, nodule/mass, other diseases, other lesion, pulmonary disease, pulmonary fibrosis, tuberculosis, clavicle fracture, emphysema, enlarged PA, lung opacity, lung tumor, mediastinal shift, Pleural thickening, Pneumothorax, aortic enlargement, atelectasis, calcification and cardiomegaly. Each of these was categorised into either a local label or a global label. This dataset was consolidated by the combined effort from two of the largest medical institutions from Vietnam, the Hanoi Medical University Hospital and Hospital H108 [Nguyen *et al.* 2020].

2.2.6 TBX11K Dataset

The TBX11K dataset has a total of 11 200 chest X-rays images which measure 3000 x 3000 in spatial resolution. These are composed of 5 000 annotated as healthy, 5 000 sick (not TB), and 1 200 images with manifestations of TB. 1200 are made up of 924 labelled active TB, 212 latent TB, 54 cases that are both labelled as active and latent TB, and 10 labelled as uncertain. This dataset was collected in collaboration with various hospitals from China [Liu *et al.* 2020].

2.3 Image Segmentation and the U-net Model

Image Segmentation plays an important role as a pre-processing technique. It is usually important to divide images into a fragmented form. This technique allows segmenting the whole chest X-ray image to isolate the region of interest which is the lung region as shown in figure 2.1.

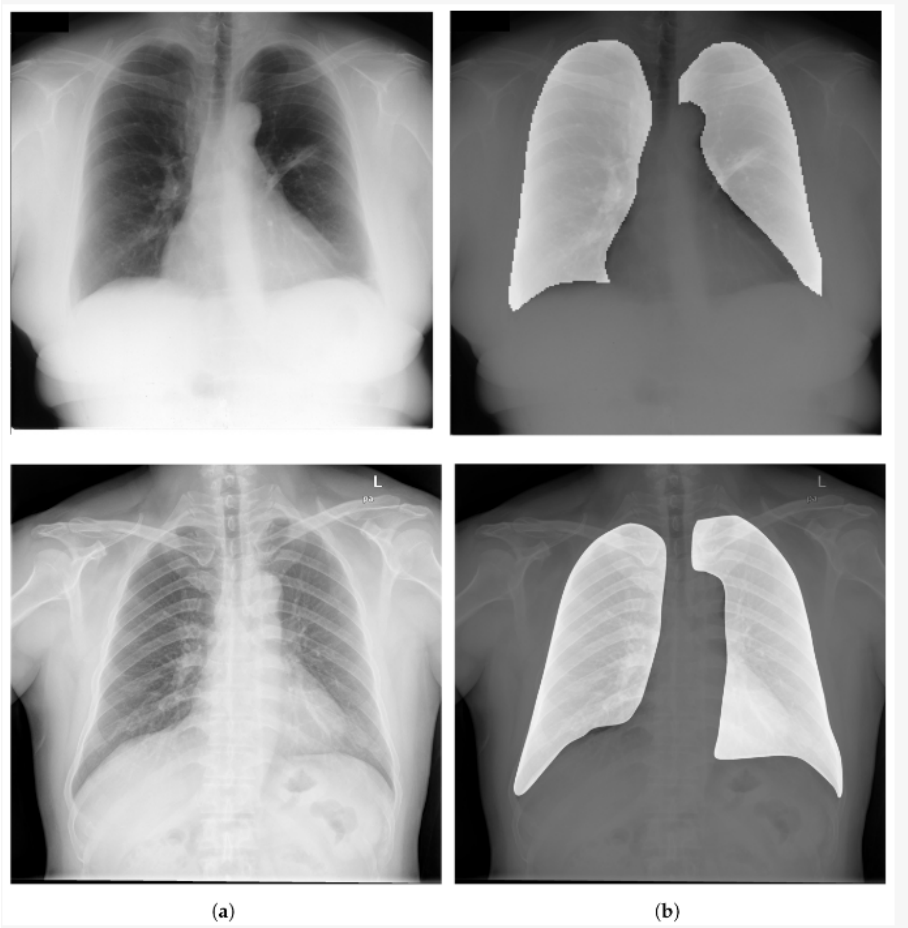


Figure 2.1: Segmentation applied to chest X-ray images

Several studies have implemented chest X-ray image segmentation and demonstrated improved efficacy in the detection and classification process. The U-Net model is the most efficient way of performing image segmentation on chest X-rays whilst some researchers have mentioned the use of the Fully Convolutional Networks (FCN) as an alternative to performing segmentation. The building blocks of FCN include the convolution layer, the pooling layer, and the activation functions [Long *et al.* 2015].

The effectiveness of segmentation using the original U-net model is depicted in figure 2.2 after a study by Rahman *et al.* [2020b]. The original U-net is composed of an upward-expanding path and a downward-moving contracting path. In the contracting, there are repeating two 3×3 convolutions without padding, each layer is followed by a ReLU and a 2×2 MAP operation with a stride of 2 for down-sampling. The expanding path will have an up-sampling of the feature map followed by a 2×2 up convolution that divides the number of channels, a concatenation with the corresponding cropped feature map from the contracting path, and two 3×3 convolutions, each is followed by a ReLU activation function.

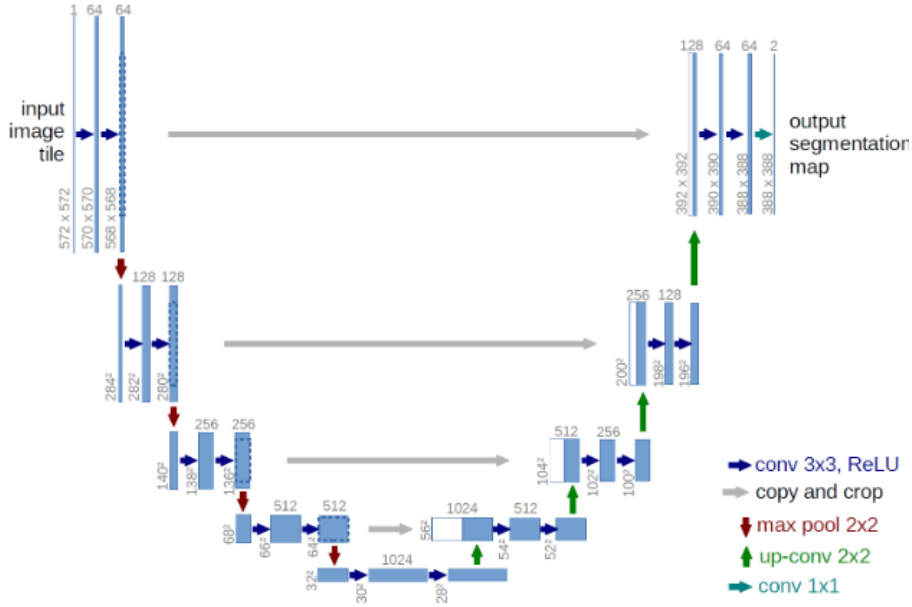


Figure 2.2: The Original U-net Architecture

2.4 Self-Supervised Learning in Image Classification

Supervised learning needs labels to solve image classification problems. However, the task of labelling images is time-consuming and expensive. In Self-Supervised Learning (SSL), models use unlabelled data to learn meaningful feature representations which can be transferred to downstream tasks such as image classification. Self-supervised learning aims to gather visual image features without using manual annotations. There are three variations of SSL: clustering-based methods, contrastive-based or distillation-based methods. The clustering methods loop through the whole dataset to create image representations of cluster assignments which are used as target points when training the model thus they use a clustering algorithm to group images with the same visual features. On the other hand, contrastive methods make use of a loss function to make a comparison of pairs of image representations, thus bringing together images with the same transformation views while pushing away image representations with different sources [Newell and Deng 2020].

2.4.1 Self-Distillation With No Labels (DINO)

DINO is a self-supervised algorithm that involves two networks, a teacher and a student. Both networks use a single Vision Transformer and they both use inputs from dual views on the same image. These will be large patches consisting of the global crops of the image as well as a number of small patches that show a local representation of the image known as local crops. Subsequently, the crops are moved over to the student network while the global crops are conveyed to the teacher such that both networks will agree on the semantic representation within the images using self-distillation. Ultimately the idea is to show and evaluate that local and global crops portray the same image object [Caron *et al.* 2021].

2.4.2 Momentum Contrast (MoCo)

MoCo utilizes a momentum encoder, a slow-moving average neural network and this ensures a consistent representation of negative pairs extracted from a memory bank. This approach, known as Momentum Contrast (MoCo), was introduced as a method for constructing consistent large dictionaries in unsupervised learning using a contrastive loss. The dictionary is managed as a queue of data samples, where the encoded representations of the current mini-batch are added to the queue, while the oldest samples are removed. By decoupling the dictionary size from the small-batch size, the queue can grow in size. To maintain consistency, a gradually evolving key encoder is implemented as a moving average of the query encoder in a momentum-based approach as the dictionary keys are derived from the preceding several mini-batches [He *et al.* 2020].

2.5 Conclusion

In this chapter we have highlighted some recent advances in using deep learning methods for TB diagnosis. We have explored some of the publicly available datasets considered for our investigations. Self-supervised learning algorithms have been highlighted and the base structure of the dominant segmentation models such as U-Net have been explored. The following section describes the research methodology that we followed in this research.

Chapter 3

Research Methodology

This chapter discusses the research methodology adopted in this report. We highlight some of the main pre-processing techniques we applied to our experiments' datasets. We present some of the popular deep CNN algorithms that were used in our experiments. In section 3.2, we describe the tools and procedures that were used in carrying out this research. We mention the reporting strategies that were adopted for recording results and metrics from the experiments. We highlight some of the potential limitations and ethical considerations of this research in sections 3.3 and 3.4 respectively.

3.1 Research Design

This research was focused on experimenting and evaluating the various datasets previously mentioned using the pre-training regimes presented above. In each experiment, we identified several datasets that are suitable to answer the questions effectively based on literature. Some of the datasets suitable for pre-training include CheXpert, PadChest, ChestX-ray14, MIMIC-CXR, VinDr-CXR and TBX11K. We used these datasets to perform our pre-training and fine-tuning by performing sub-sampling and ignoring labels in the case of self-supervised experiments. In the pre-training section, state-of-the-art deep convolutional neural networks were implemented for feature extraction and performing the classification tasks on the various chest X-ray image datasets.

3.2 Methods

In this section we discuss our approach for evaluating the effectiveness of the various pre-training regimes. The proposed method consists of four main experiments. Each experiment has its own sub-tasks including data gathering, data preparation, data preprocessing, pre-training, fine-tuning, and evaluating the results. The core idea of these experiments is to investigate if the international datasets can transfer between themselves and to find out if the TB classification model performances can be achieved when encountering datasets from different regions. This research will enable us to determine

the extent to which these pre-training and transfer learning experiments are useful in informing us if there is a decrease in the required data for locally built TB classifiers. It should be noted that due to the unavailability of South African datasets at the time of writing this report, we resorted to using international datasets. The major assumption is that if transfer learning between these international datasets yields significant results, we can conclude that the model may yield acceptable results when we transfer to datasets from a different region. This assumption is based on the fact that CXR images tend to maintain structure and will possibly maintain an overlapping set of features across chest-affecting diseases. In case these datasets do not give good results between themselves, we can conclude that these models will not scale when encountered with new datasets from different regions.

In our experiments, we used popular deep-learning models as suggested by the literature. We considered the following deep-learning CNN models including ResNet-50, VGG-19 and DenseNet-121 for pre-training. We used SOTA SSL algorithms including DINO and MoCo for the self-supervised pre-training experiments.

In this research, all programming tasks were carried out in Python. Various Python libraries and packages were used for the experiments including Pytorch, Tensorflow, the Weights and Biases package (for visualisation and tracking of the experiments).

3.2.1 Experiments

The details of our experiments are highlighted in this section and a summary is provided in table 3.1. In our experiments, we performed both supervised and self-supervised pre-training on various international datasets and fine-tuned on other geographically different international datasets. It should be noted that there are many permutations for selecting the datasets for the experiments. The order in which the datasets were selected for either pre-training or fine-tuning is not particularly important as the main idea is to investigate the effectiveness of pre-training and transfer learning between these datasets. We strongly believe that CXR images present overlapping visual features between chest-affecting diseases and these features can be learned easily by deep learning models. We want to find out if the online datasets are likely to save us training time and resources when we perform transfer learning. Therefore, the examples given in table 3.1 are not exhaustive but are only for illustrating the core ideas. The experiments are stated as follows:

1. This experiment served as a baseline for the other experiments to help us understand the inconsistencies that may arise from the datasets. We fine-tuned using the TBX11k dataset on the following base models ResNet-50, VGG-19, and DenseNet-121 which were either initialised with ImageNet weights or random weights.

2. The focus of this experiment was to conduct pre-training and evaluation on different chest X-ray datasets that differ in terms of origin and size (pre-training on an international dataset and then testing on another dataset that comes from a different region). For this experiment, we performed pre-training on the following geographically different datasets including VinDr-CXR (Vietnam), PadChest

(Spanish), and Chest-Xray14 (American). We used all the labels in each dataset as highlighted in section 2.2. The models were evaluated by fine-tuning on the TBX11K dataset which contains TB labels for images from China. In each experiment, the following deep learning models were used, that is ResNet-50, VGG-19 and DenseNet-121.

3. In this experiment, we wanted to evaluate the performance of a TB classifying model when pre-training is done on different chest-affecting diseases. We conducted pre-training on datasets with labels for multiple chest-affecting conditions and fine-tuned the model on the TBX11K dataset. In this experiment, we implemented ResNet-50, VGG-19 and DenseNet-121. We experimented with each on the following datasets: VinDr-CXR, CheXpert, ChestX-ray14, PadChest and MIMIC-CXR. We used all the labels in each dataset as described in section 2.2 with special interest to the dataset with labels for Lung cancer, Pneumonia and Emphysema. To evaluate the effectiveness of transfer learning on each of the datasets, we fine-tuned on the TBX11K dataset.

4. This experiment considers the effect of different number of labels on the performance. We use in-distribution self-supervised pre-training using DINO [Caron *et al.* 2021] and MoCo [He *et al.* 2020]. Using the same dataset for pre-training and training, we consider the case where pre-training is done with data originating in the same region. We split the TBX11K training dataset into pre-training (without labels) and training (with labels) sets. We sweep over various split proportions to establish the effect of limited labels on performance.

5. This experiment mimics the experiment above, but considers out-of-distribution pre-training. We therefore consider the case where pre-training data comes from a separate region for the training and testing dataset. We use the VinDr-CXR, CheXpert, and PadChest datasets for SSL pre-training with DINO and MoCo. We then train and test with the TBX11K dataset. Note that this experiment also differs from experiment 4 in that the entire training set is used. We performed various split ratios in multiple sweeps to determine the effect of limited labels on model performance.

3.2.2 Evaluation Metrics

In this section, we give a high-level description of the metrics that we have referenced in our experiments. The following statistical measures including true positive, false positive, false negative, and true negative. The performance of a deep learning classifier is measured using a confusion matrix [Ravi *et al.* 2023]. In the context of TB classification, we defined the metrics as follows:

- True positive (TP): TB positive CXR image labelled as TB positive.
- True negative (TN): TB negative CXR image labelled as TB negative.
- False positive (FP): TB negative CXR image miss-labelled as TB positive.
- False negative (FN): TB positive image miss-labelled as TB negative.

Table 3.1: Experiments Summary

No	Experiment	Base Model(s)	Pre-training	Fine-tuning
1	Baseline Experiment(s)	ResNet-50, VGG-19, DenseNet-121	ImageNet,Random weights	TBX11K
2	Pre-training and testing on geographically different datasets	ResNet-50,VGG-19,DenseNet-121	VinDr-CXR,PadChest	TBX11K
3	Pre-training on different chest affecting conditions and testing on TB	ResNet-50,VGG-19,DenseNet-121	CheXpert,ChestX-ray14,PadChest,MIMIC-CXR	TBX11K
4	Pre-training on unlabelled CXR images using self-supervised learning (same location)	ResNet-50 (pre-trained on DINO/MoCo algorithms)	TBX11K (various training split ratios)	TBX11K
5	Pre-training on unlabelled CXR images using self-supervised learning (different locations)	ResNet-50 (pre-trained on DINO/MoCo algorithms)	PadChest, VinDr-CXR, CheXpert, MIMIC-CXR (various training split ratios)	TBX11K

Accuracy measures the ability of the TB deep learning model to classify all healthy (TB negative) chest X-ray image samples as No TB or healthy whilst the images that are labelled TB positive indicate the presence of TB in the lungs. In the case of imbalanced dataset classes, accuracy is an unreliable metric as a model that always predicts the majority class could still achieve high accuracy. Note that the TBX11K dataset is imbalanced as there are 1 200 TB images and 10 000 non-TB images, therefore the importance of accuracy is limited in our context. Equation 3.1 is used to compute the model accuracy.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.1}$$

Precision measures the proportion of images that actually had TB out of all the test images classified as TB-positive. It can be expressed with the equation 3.2:

$$Precision = \frac{TP}{FP + TP} \tag{3.2}$$

Recall/Sensitivity measures the ability of a classifier to label a TB-positive chest x-ray image sample as TB-positive. This may be defined as the True Positive Rate, that is of all test images with a TB-positive label, what proportion were detected by the model. It can be expressed with the equation 3.3:

$$Recall/Sensitivity = \frac{TP}{TP + FN} \tag{3.3}$$

Specificity measures the ability of a classifier to mark a non-TB (TB Negative) chest X-ray image sample as TB negative, that is of all the test images with a TB-Negative label, what proportion were correctly classified. It can be defined as the True Negative Rate and is expressed with the equation 3.4:

$$Specificity = \frac{TN}{TN + FP} \tag{3.4}$$

The F1-score measures the weighted average of recall and the precision of a classifier. It can be expressed with the equation 3.5:

$$F1 = \frac{TP}{TP + 0.5(FP + FN)} \tag{3.5}$$

The Receiver Operating characteristics curve (ROC) is used to visualise and assess the performance of a classifier. To plot the ROC curve, we choose a threshold that gives us a specific false positive Rate (FPR) and then measure the true positive rate (TPR). This is done for multiple FPRs to get the ROC curve. The ROC curve can be visualized by a plot of the TPR alongside the FPR in which the TPR takes the y-axis and the FPR takes the x-axis on a Cartesian plane. The AUC is then the area under the curve. An AUC of 0.5 represents a random classifier while an AUC of 1.0 is a perfect score.

The most widely used evaluation metric in classification models is accuracy. It should be noted that accuracy is affected heavily by differences in class numbers (class imbalances) in the dataset. Another weakness of highly relying on accuracy is that there is potential for two deep learning models to come up with the same accuracy score but can perform differently depending on the type of positive and negative predictions. Sensitivity or Recall is a crucial metric when it comes to medical image classification which leads to diagnosis. It is important to avoid a situation in which you misdiagnose a TB-infected individual as TB-negative as this results in further increasing the chances of spreading Tuberculosis as they will miss out on treatment. In general, we may consider a FP more acceptable than a FN to avoid missing patients that actually do have TB. The minimum diagnostic accuracy as recommended by the World Health Organisation is a sensitivity of over 90% and a specificity of over 70% when using computer-aided TB detection methods [WHO and others 2021].

3.3 Limitations

Various datasets were sourced from different online sources. They had different image formats including JPEG, PNG or DICOM formats. Image scaling and pre-processing may have compromised the image quality for some of the experiments hence compromising the final performance results. Due to the unavailability of South African-sourced datasets at the time of conducting this research, these experiments must be repeated to find out if the SA-based dataset can generalise on these models.

3.4 Ethical Considerations

All of the chest X-ray image datasets we used for this research's experiments are publicly available. The relevant authors have already authorised these data sources at the time of publishing the datasets. For the purposes of this research, the requirement was to complete and submit the ethics waiver and it was approved by the relevant ethics committee (Reference number: CSAM-2023-1W).

Chapter 4

Experiments and Results Discussion

In the previous chapter, we presented the methodology applied in the research. This chapter presents the results of the experiments highlighted in table 3.1. We initially present the results of the baseline experiments carried out using the TBX11K dataset. Secondly, we present the results of the pre-training from the supervised experiments. We finally give an account of the results from the self-supervised pre-training experiments across the various chest X-ray datasets.

4.1 Baseline Experiment Results

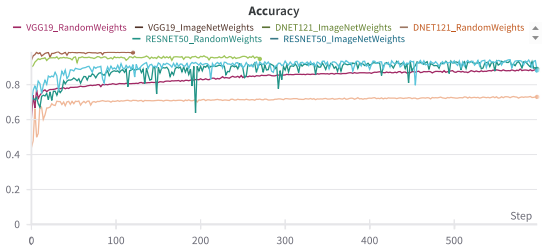
We conducted TB classification experiments using the TBX11K dataset with the configurations summarised in 4.1. Three deep learning models including ResNet-50, DenseNet-121 and VGG-19 were each used as backbones. Each of these models was initialised with either random weights or ImageNet weights. A classification layer was added on each model and a softmax activation function was used. We used the Categorical Cross entropy loss function and initialised the learning rate at 0.001 with a batch size of 32. All images were standardised and resized to 224 x 224. We opted to use the training and validation sets provided by the authors for our experiments. The ratios were as follows, 80% training and 20% validation. The target classes were health, sick and TB.

We set the training epochs up to 600 in each configuration and we implemented early stopping on validation loss with a patience level of 50 on each experiment configuration shown in 4.1.

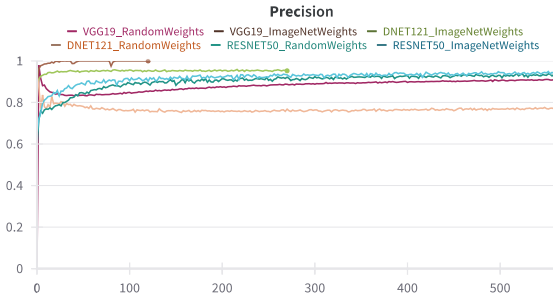
The VGG-19 model that was initiliased with ImageNet weights proved to be robust as shown in figure 4.1. We achieved an accuracy of 98.44%, a precision of 98.44%, an recall of 98.44%, and a specificity of 99.78%. This same configuration surpassed the other models with an F1-Score of 98.44% and an AUC of 0.99. These validation results show that ImageNet weights initialisation can improve model training in classification tasks by extracting important features/patterns from natural images which can be useful in chest X-ray image classification via transfer learning.

Table 4.1: TB Classification Baselines Results. ACC denotes Accuracy, PRE denotes Precision, SPE represents Specificity, F1 is the F1-Score

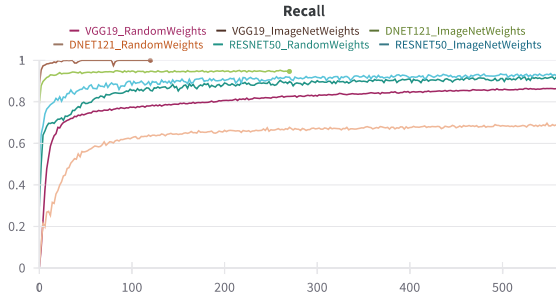
Model	**Weights**	**ACC**	**PRE**	**Recall**	**SPE**	**F1**	**AUC**
ResNet-50	ImageNet	88.06	88.51	87.72	**99.89**	88.12	0.96
ResNet-50	Random	88.95	89.67	88.17	99.22	88.91	0.98
VGG-19	ImageNet	**98.44**	**98.44**	**98.44**	99.78	**98.44**	**0.99**
VGG-19	Random	88.62	90.81	86.05	99.39	88.37	0.96
DenseNet-121	ImageNet	94.75	94.74	94.53	99.72	0.95	**0.99**
DenseNet-121	Random	72.99	76.67	67.86	96.82	0.72	0.89



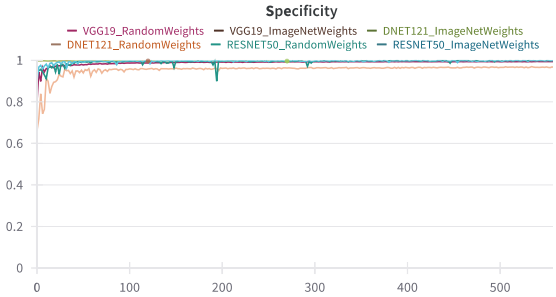
(a) validation accuracy



(b) validation precision



(c) validation recall

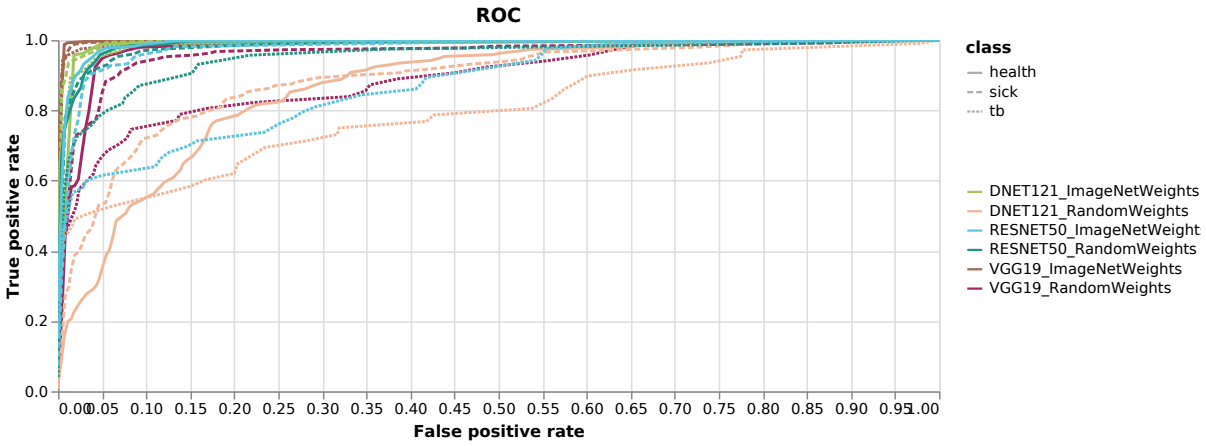


(d) validation specificity

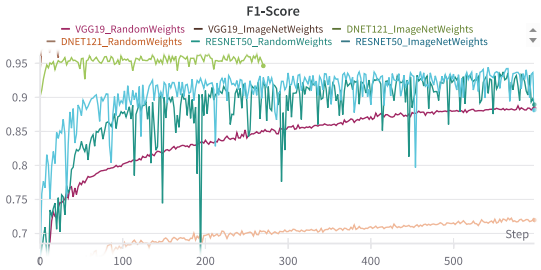
Figure 4.1: TBX11k Baseline Results

The Receiver Operating Characteristic (ROC) curve shown in figure 4.2 shows a visual representation of the effectiveness of each model in identifying the different classes in the dataset. The models that were initialised with ImageNet weights scored significant Area Under the Curve (AUC) of close to 1. This further supports the idea that transfer

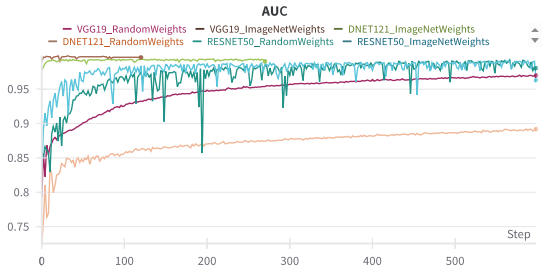
learning using ImageNet weights gives the models an edge over random initialisation when performing the chest X-ray classification task.



(a) validation ROC curve



(b) validation precision



(c) validation F1-Score

Figure 4.2: TBX11k training progress showing validation ROC curve F1-Score, AUC for models on which pre-training was carried out on the VinDr-CXR dataset

These baseline experiments created a foundation that guided us on the possible outcomes from our supervised and self-supervised experiments. The baseline experiments suggested a significant benefit in transfer learning in this case using ImageNet pre-trained weights in comparison to random weights. Based on the results, we speculated that pre-training between big chest X-ray datasets can potentially benefit the classification of TB. The following experiments were aimed at investigating if there is any benefit in pre-training on one chest X-ray dataset and fine-tuning on different TB chest X-ray datasets. We focused on datasets sourced from different regions of the world and with a particular interest in the labels for chest-affecting conditions.

4.2 Supervised Pre-training and Classification

In these experiments, we conducted pre-training on chest X-ray datasets with labels for multiple chest-affecting conditions. We used the weights from the pre-training to fine-tune on the TBX11k dataset which had definite TB labels. Deep learning models including ResNet-50, DenseNet-121 and VGG-19 were each used as backbones. Each of these models was initialised with either random weights or ImageNet weights. The weights of the pre-training exercise were saved. The learning rate was set to 0.0001 with a batch size of 32. The images were resized to 224 x 224. The target classes were health, sick and TB. We used the following ratios 80% for training, 20% for validation. The naming convention of our weights will start by denoting the dataset on which the pre-training was done followed by either Random or ImageNet to indicate the initialisation weights on each model. This naming convention will be followed throughout the supervised learning experiments and plots.

4.2.1 VinDr-CXR Pre-training - TBX11K Fine-tuning

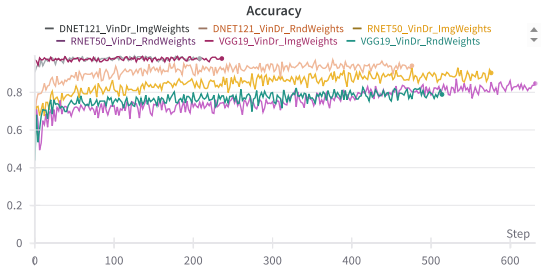
In this experiment, we pre-trained using ResNet-50, DenseNet-121 and VGG-19 on the VinDr-CXR dataset. We initialised the weights with ImageNet or random weights. We saved the weights from pre-training and fine-tuned on the TBX11k dataset to predict the three classes in the dataset. We set early stopping based on validation loss with a patience level of 50 and we set the initial training epochs up to 1000 on each model configuration, table 4.2 shows the summary of the results.

Table 4.2: VinDr-CXR - TBX11K Results. ACC denotes Accuracy, PRE denotes Precision, SPE represents Specificity, F1 is the F1-Score

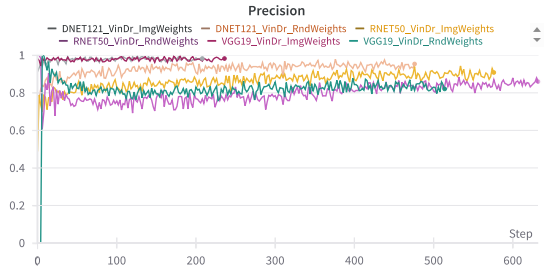
Model	**Weights**	**ACC**	**PRE**	**Recall**	**SPE**	**F1**	**AUC**
ResNet-50	VinDr-CXR ImageNet	90.28	90.81	89.24	99.83	90.00	0.98
ResNet-50	VinDr-CXR Random	84.72	86.02	83.33	99.13	84.66	0.95
VGG-19	VinDr-CXR ImageNet	**97.92**	**98.26**	**97.92**	99.83	98.09	**0.99**
VGG-19	VinDr-CXR Random	78.82	82.06	74.65	98.26	78.00	0.92
DenseNet-121	VinDr-CXR ImageNet	97.91	98.25	**97.92**	99.65	**99.09**	**0.99**
DenseNet-121	VinDr-CXR Random	94.1	95.37	93.06	**100.00**	94.20	**0.99**

Pre-training on the VinDr-CXR Dataset using ImageNet weights proved more efficient than the randomly initialised weights across all three models when we consider the metrics measured during the fine-tuning on the TBX11K dataset. The specificity and

recall scores were over 90% and 70% respectively which is expected for any model that goes into production by the World Health Organisation.

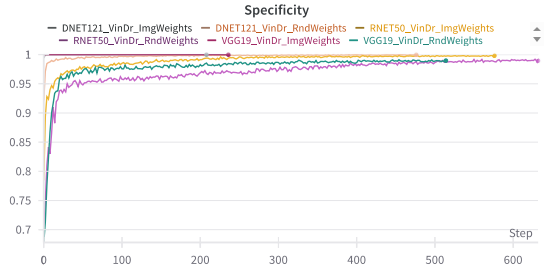


(a) validation accuracy



(b) validation precision

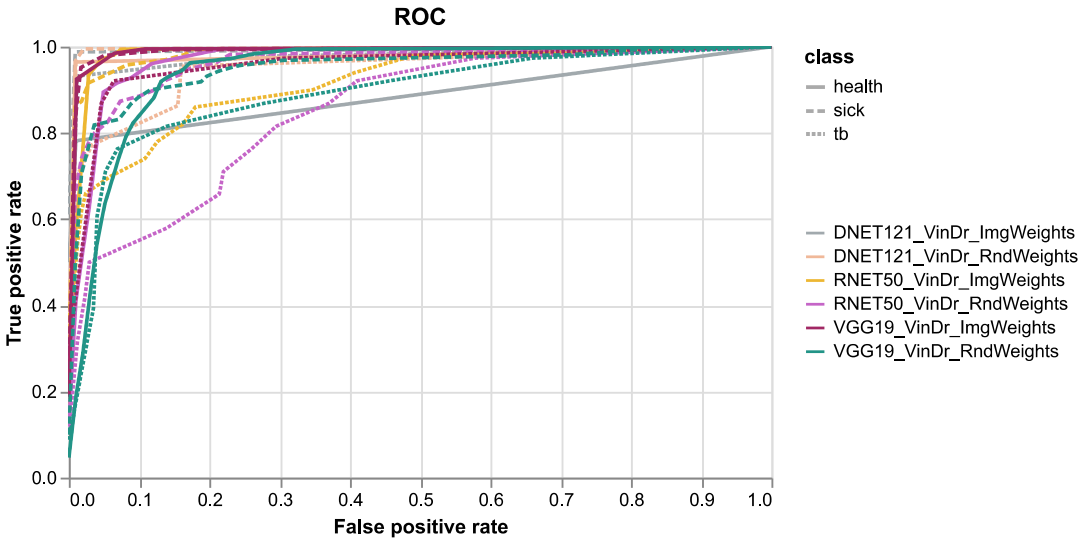
(c) validation recall



(d) validation specificity

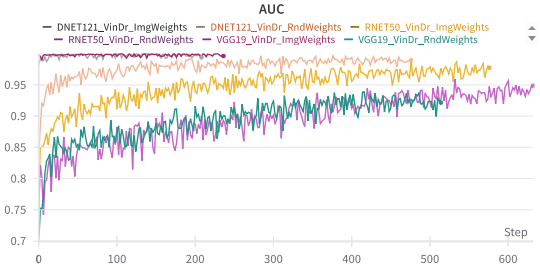
Figure 4.3: TBX11k training results indicating the progression of Accuracy, Recall, Precision, Specificity for models on which pre-training was carried out on the VinDr-CXR

We observed that the models configured to use the ImageNet weights tend to converge faster than the randomly initialised models as shown in figure 4.3. The DenseNet-121 model showed higher levels of accuracy, recall and specificity. We remain reluctant to base our conclusions on the results of the DenseNet-121 model as it showed signs of over-fitting and diverging using both ImageNet and random weights pre-trained on the VinDr-CXR dataset. Both model configurations converged after 200 epochs while VGG-19 and ResNet-50 converged after 500 epochs as shown in figure 4.4.



(a) validation ROC curve

(b) validation F1-Score



(c) validation AUC

Figure 4.4: TBX11k training showing the ROC curve, F1-Score and AUC for models on which pre-training was carried out on the VinDr-CXR dataset

In figure 4.4, we highlight that the DensetNet-121 model showed signs of divergence as indicated by both the F1-Score and AUC plots. It should be noted that even after adjusting the batch size and the learning rate we failed to reach convergence.

Based on these results, we can speculate that the attributes, labels, geographical location and size of the VinDr-CXR chest X-ray dataset did not improve the outcome of

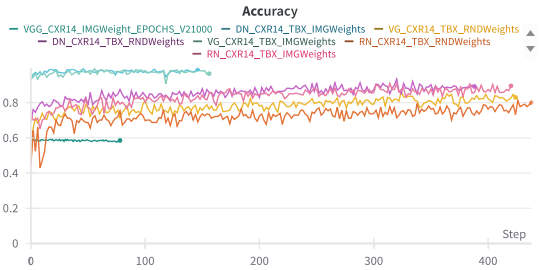
the results. If we compare the results from table 4.1 and table 4.2, we can observe the decline in AUC, precision and recall scores from the baseline models. Supervised pre-training on the VinDr-CXR dataset seems to have degraded performance relative to the baseline, therefore the transfer from VinDr-CXR to TBX11K was detrimental versus the baseline.

4.2.2 ChestX-ray14 Pre-training - TBX11K Fine-tuning

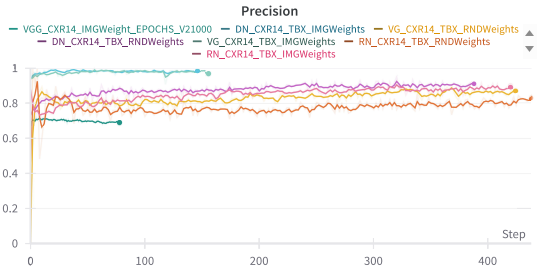
In this experiment we pre-trained ResNet-50, DenseNet-121 and VGG-19 on the ChestX-ray14 dataset. We initialised the models on ImageNet or random weights. We saved the weights accordingly and fine-tuned on the TBX11k dataset to predict the three classes in the dataset. The summary results are shown in table 4.3. After training on the TBX11K dataset with ResNet-50, we noted that the ImageNet-based weights were more robust by achieving an AUC of 0.91 versus 0.89 on the random weights. The same pattern continues across to the VGG-19 and DenseNet-121 models. Figure 4.6 shows the progress of validation scores during training on the TBX11k dataset. From the plots, the DensetNet-121 and VGG-19 models trained on the ChestX-ray14 ImageNet weights show signs of divergence. Both models stopped training before the 100 epoch mark, hence the results from those models will not be considered valid for making any conclusions.

Table 4.3: ChestX-ray14 - TBX11K Results. ACC denotes Accuracy, PRE denotes Precision, SPE represents Specificity, F1 is the F1-Score

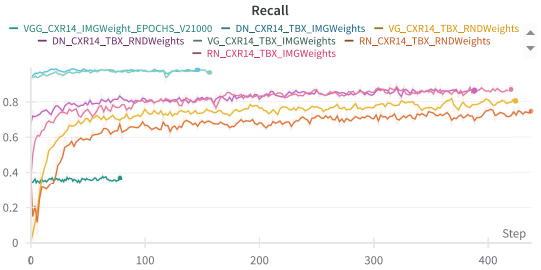
Model	**Weights**	**ACC**	**PRE**	**Recall**	**SPE**	**F1**	**AUC**
ResNet-50	ChestX-ray14 ImageNet	89.58	90.07	88.19	97.22	**89.12**	**0.97**
ResNet-50	ChestX-ray14 Random	79.86	83.46	75.35	99.48	79.2	0.92
VGG-19	ChestX-ray14 ImageNet	97.26	97.28	97.28	97.28	97.28	0.99
VGG-19	ChestX-ray14 Random	82.99	87.55	80.56	99.31	0.95	0.84
DenseNet-121	ChestX-ray14 ImageNet	**98.41**	**98.43**	**98.43**	**99.98**	**98.25**	**0.99**
DenseNet-121	ChestX-ray14 Random	89.24	90.91	86.81	98.78	88.81	0.96



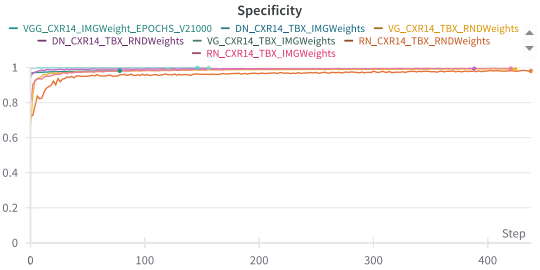
(a) validation accuracy



(b) validation precision

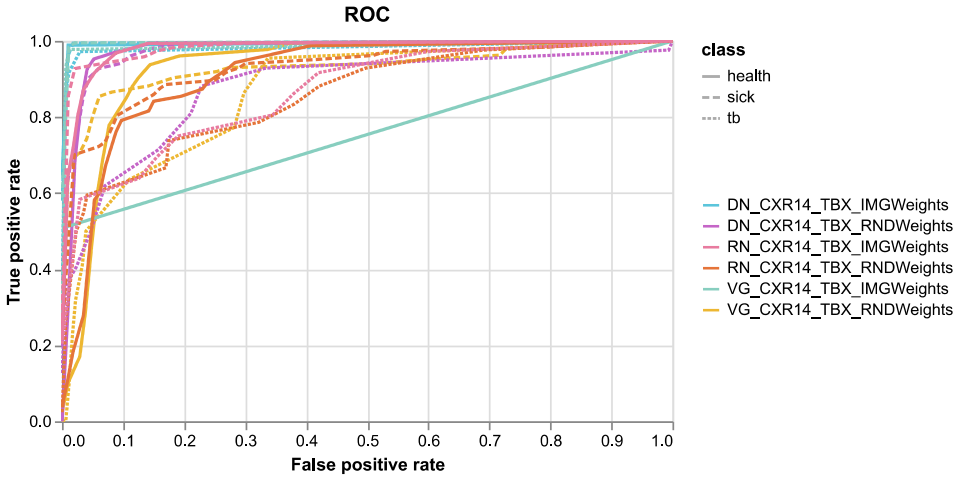


(c) validation recall

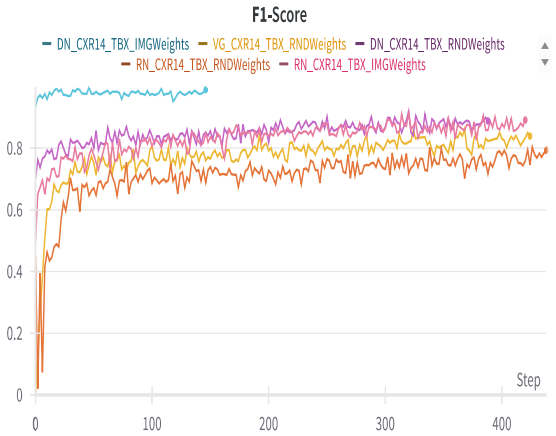


(d) validation specificity

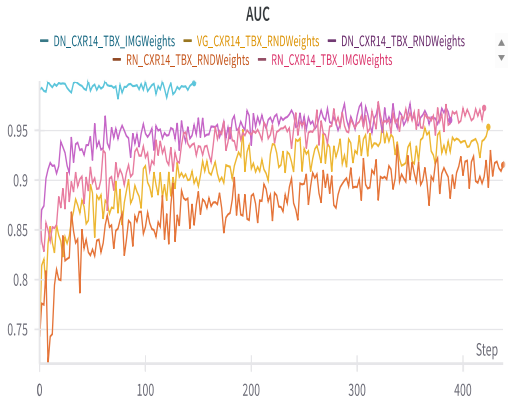
Figure 4.5: TBX11k training results indicating the progression of accuracy, precision, recall, and specificity on weights obtained from the ChestX-ray14 dataset pre-training



(a) The ROC curve



(b) F1-Score



(c) AUC

Figure 4.6: TBX11k training showing validation ROC plot, AUC, and validation F-Score using the pre-trained weights from the ChestX-ray14 dataset

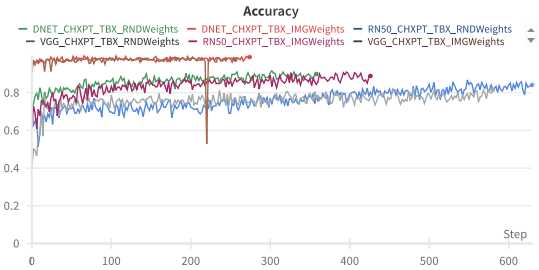
We recorded a decrease in model performance and metric effectiveness across the models from the VinDr-CXR pre-trained weights relative to the ChestX-ray14 pre-trained weights. We can speculatively state that the VinDr-CXR dataset appears to be more robust compared to the ChestX-ray14 due to differences in the geographical source location of the dataset and the quality of the labels. It was mentioned that the ChestX-ray14 may be prone to errors due to unverified labels extracted by NLP algorithms [Wang *et al.* 2017]. This may have contributed to the low performance of the model pre-trained on ChestX-ray14. Additionally, there is a decline in performance when we compare the results from the baseline experiments in table 4.1 against the results from table 4.3. It should be noted that the VinDr-CXR dataset is slightly smaller than the ChestX-ray14 dataset, we expected the bigger dataset to provide more robust weights resulting in more effective metric results.

4.2.3 CheXpert Pre-training - TBX11K Fine-tuning

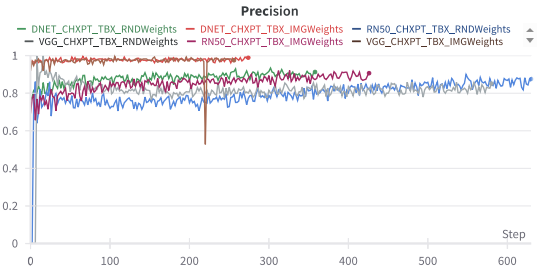
In this experiment we pre-trained three models including ResNet-50, DenseNet-121 and VGG-19 on the CheXpert dataset. We initialised each model on either ImageNet or random weights during pre-training. We saved the weights and fine-tuned on the TBX11k dataset to predict the three classes in the dataset. The summary of the different configurations of these experiments and results are shown in table 4.4.

Table 4.4: CheXpert-TBX11K results. ACC denotes Accuracy, PRE denotes Precision, SPE represents Specificity, F1 is the F1-Score

Model	**Weights**	**ACC**	**PRE**	**Recall**	**SPE**	**F1**	**AUC**
ResNet-50	CheXpert ImageNet	89.45	90.62	89.32	**100.00**	89.97	**0.99**
ResNet-50	CheXpert Random	85.55	87.02	83.07	99.96	85.18	0.98
VGG-19	CheXpert ImageNet	97.4	97.65	**97.27**	**100.00**	97.46	**0.99**
VGG-19	CheXpert Random	75.13	87.42	50.65	99.73	64.14	0.78
DenseNet-121	CheXpert ImageNet	**98.05**	**97.92**	94.4	99.99	**97.98**	0.97
DenseNet-121	CheXpert Random	93.23	94.71	90.89	99.99	92.76	0.94

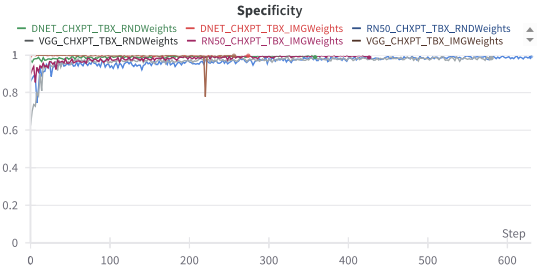


(a) validation accuracy



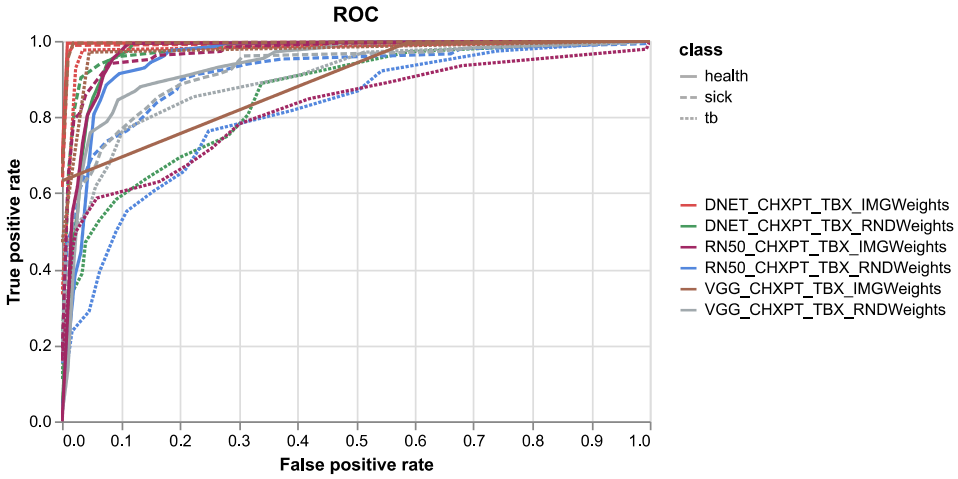
(b) validation precision

(c) validation recall

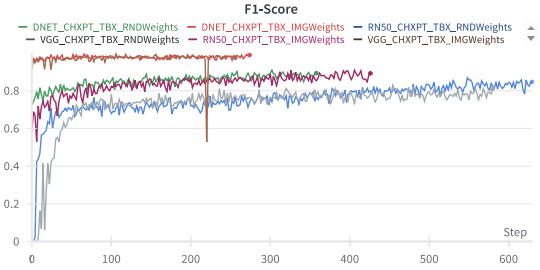


(d) validation specificity

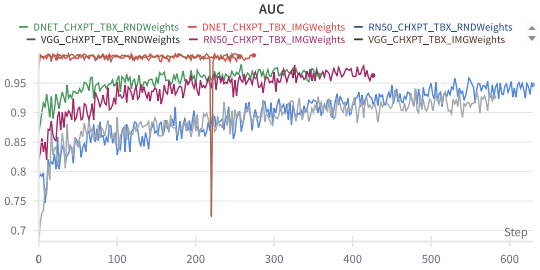
Figure 4.7: TBX11k training results indicating the progression of accuracy, recall, precision and specificity with model weights from CheXpert pre-training



(a) The ROC curve



(b) F1-Score



(c) AUC

Figure 4.8: TBX11k training showing validation ROC plot, AUC, and the validation F-Score using the pre-trained weights from the ChestX-ray14 dataset

After pre-training on the CheXpert dataset, we used the weights from the pre-training to fine-tune on the TBX11k dataset. We implemented early stopping based on validation loss with a patience level of 50 on each model during training. The VGG model with ImageNet weights converged with an accuracy of 97.4% and the DenseNet-121 model converged with an accuracy of 98.05%. We noted that models initialised with the random weights from CheXpert pre-training converged later on with ResNet-50 and

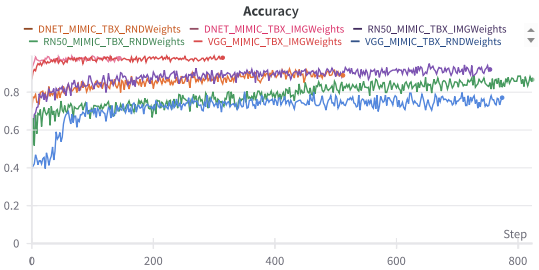
VGG-19 converging after 580 and 630 respectively. Despite a lower recall score of 50.65% from VGG-19 on CheXpert-Random weights which was below the 70% mark, the other models scored over 70% on recall. We recorded specificity scores of over 90% across all the different weights configurations as shown in table 4.4 and plots shown in figure 4.7. The ROC curve in figure 4.8 indicated that models initilised with CheXpert random weights scored relatively lower on the AUC metric, especially for DenseNet-121 and ResNet-50 even though the scores were over 0.5 for the TB class when compared to models that were initialised with CheXpert ImageNet weights. These attained AUC scores of close to 0.9 for the CheXpert ImageNet weights. We noted that the models were robust in predicting the health and sick classes over the TB class. Despite the shortcomings in the other model configurations, the ResNet-50 model initialised with ImageNet outperformed the VinDr pre-trained ResNet-50 baseline model. This outcome may support the idea that pre-training on datasets that contain labels for other chest-affecting diseases may improve the efficiency and performance of deep-learning models in TB classification tasks.

4.2.4 MIMIC-CXR Pre-training - TBX11K Fine-tuning

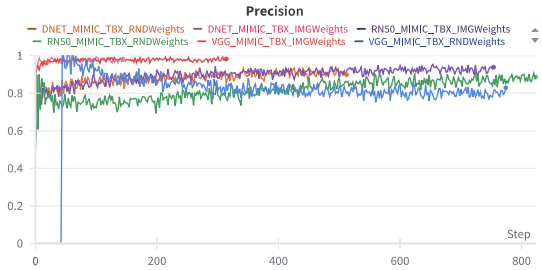
The MIMIC-CXR dataset presents a good opportunity for pre-training and transfer learning as it is one of the largest publicly available datasets with labels of over 14 chest-affecting conditions. We first extracted the pixel data from the Digital Imaging and Communications in Medicine (DICOM) format in this experiment. We scaled down each image to the 224 x 224 dimensions which is the standard image size required for training deep learning models. We initialised all models with either the ImageNet or random weights and pre-trained ResNet-50, DenseNet-121 and VGG-19 on the converted MIMIC-CXR chest X-ray dataset. We saved the weights and fine-tuned on the TBX11k dataset to predict the three classes in the dataset. The summary results were presented in table 4.5.

Table 4.5: MIMIC-CXR-TBX11K results. ACC denotes Accuracy, PRE denotes Precision, SPE represents Specificity, F1 is the F1-Score

Model	**Weights**	**ACC**	**PRE**	**Recall**	**SPE**	**F1**	**AUC**
ResNet-50	MIMIC-CXR ImageNet	92.01	93.84	89.93	**100.00**	91.84	**0.99**
ResNet-50	MIMIC-CXR Random	86.46	88.52	82.99	99.48	85.66	0.96
VGG-19	MIMIC-CXR ImageNet	98.26	**98.26**	97.92	**100.00**	**98.09**	**0.99**
VGG-19	MIMIC-CXR Random	77.08	82.92	69.10	97.92	75.4	0.88
DenseNet-121	MIMIC-CXR ImageNet	**98.31**	97.92	**97.92**	**100.00**	97.92	**0.99**
DenseNet-121	MIMIC-CXR Random	88.89	89.86	86.11	99.31	87.94	0.97

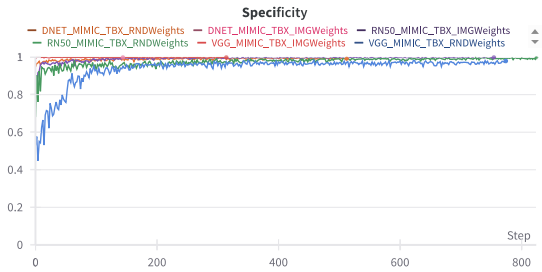


(a) validation accuracy



(b) validation precision

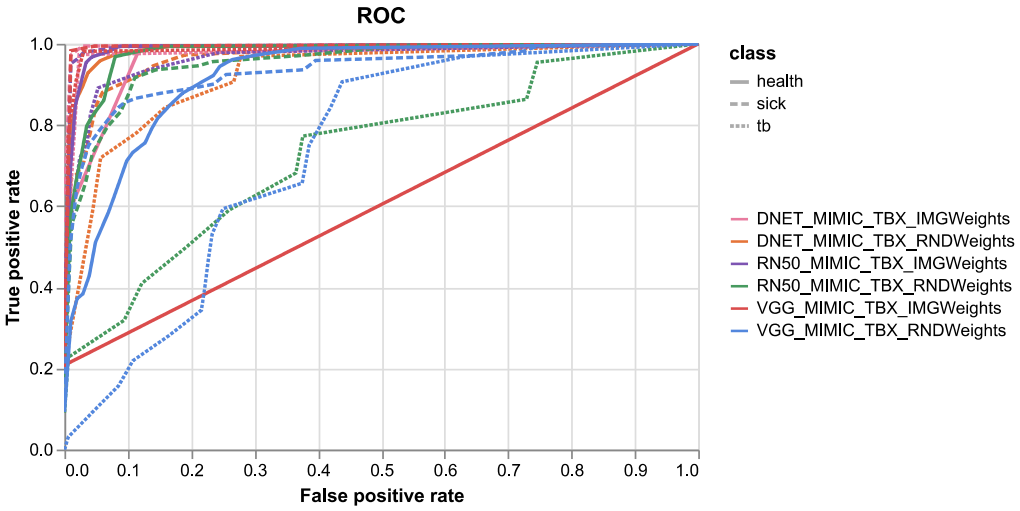
(c) validation recall



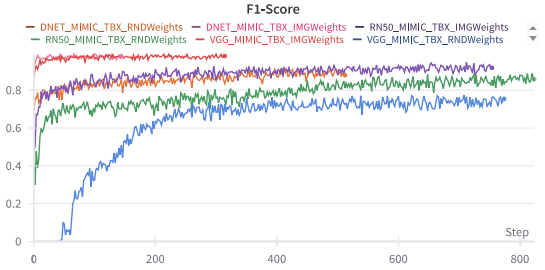
(d) validation specificity

Figure 4.9: TBX11k training results indicating the progression of accuracy, recall, precision, specificity for models on which pre-training was carried out on the MIMIC-CXR dataset

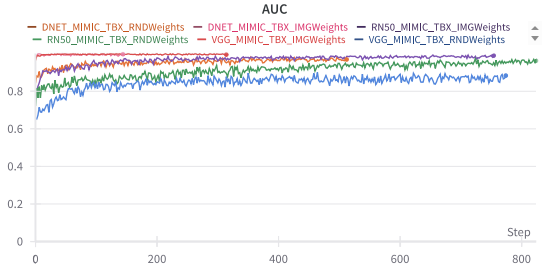
We set the training epochs to 1000 and implemented early stopping on validation loss with a patience level of 50 on each Model configuration. After training for about 145 epochs on the TBX11k dataset, the DensetNet-121 model initialised with MIMIC-CXR ImageNet weights converged to achieve specificity and recall values of 99.99% and 94.4% respectively. VGG-19 initialised with MIMIC-CXR random weights performed slightly lower with a recall of 69.53%. We observed that the ResNet-50 initialised with MIMIC-CXR ImageNet weights and the DenseNet initialised with random weights surpassed the corresponding baseline models. As per figure 4.9, VGG-19, DenseNet-121 and ResNet-50 pre-trained on the MIMIC-CXR ImageNet weights proved robust when we observed the progression of accuracy, recall and specificity graphs.



(a) The ROC curve



(b) F1-Score



(c) AUC

Figure 4.10: TBX11k training showing validation ROC plot, AUC, and the validation F-Score using the pre-trained weights from the MIMIC-CXR dataset

The ROC curve in figure 4.10 shows the results of training on various configurations from MIMIC-CXR pre-training and fine-tuning on the TBX11k dataset. Three models performed significantly lower when we compared the AUC scores. The scores for TB classification were lower when compared to the AUC scores for the health and sick classes. This pattern was observed in the following model configurations: ResNet-50 which was initilised with the MIMIC-CXR random weights, VGG-19 initilised with

MIMIC-CXR random weights and the VGG-19 initilised with MIMIC-CXR ImageNet weights. The weights obtained by pre-training on VinDr-CXR, ChestX-ray14 and CheXpert datasets yielded lower performance results relative to the weights from the MIMIC-CXR dataset. This was observed in all the configurations of ResNet-50 and VGG-19 models. We can speculate that the MIMIC-CXR dataset provided more robust weights for transfer learning than the other datasets. This may be attributed to the size of the dataset (377 110 images) and the balance of labels for each class. With four of the models failing to match the corresponding baseline models, we cannot say that pre-training of the MIMIC-CXR dataset is beneficial in improving the results of a TB classifier. We also noted that the results of this experiment did not surpass those from the baseline experiment hence pre-training on MIMIC-CXR and fine-tuning on TBX11k dataset was not beneficial. However, metrics from MIMIC-CXR pre-training and TBX11k fine-tuning are within the recommended ranges by the WHO.

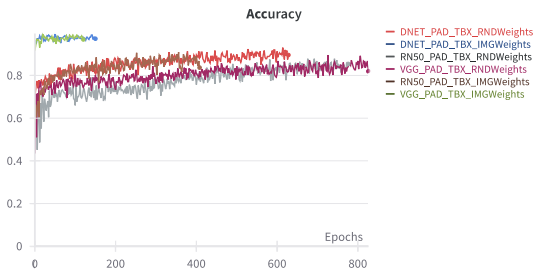
4.2.5 PadChest Pre-training - TBX11K Fine-tuning

The Padchest dataset is of Spanish origin and it surpasses the MIMIC-CXR dataset in terms of the number of conditions and labels that were identified in the dataset with a total of 174 findings and 19 chest diagnoses Bustos *et al.* [2020]. We speculated that the dataset makes a good candidate for pre-training and will enable us to answer whether there is a benefit of pre-training on datasets with labels for multiple conditions. Additionally, this dataset comes from a different geographical location, and we speculated that fine-tuning on the TBX11k dataset can yield better performance metrics. We pre-trained ResNet-50, DenseNet-121 and VGG-19 on the PadChest chest X-ray dataset. We initialised the models on ImageNet or random weights during pre-training. We saved the weights accordingly and fine-tuned on the TBX11k dataset to predict the three classes in the dataset. Table 4.6 summarised the results of the different experiment configurations that were executed.

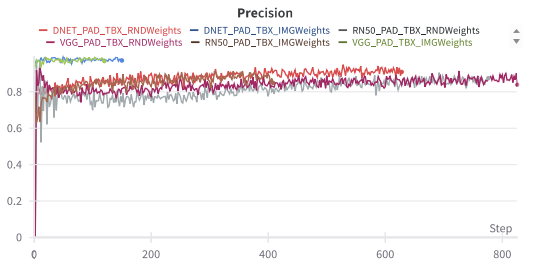
Table 4.6: PadChest - TBX11K results. ACC denotes Accuracy, PRE denotes Precision, SPE represents Specificity, F1 is the F1-Score

Model	**Weights**	**ACC**	**PRE**	**Recall**	**SPE**	**F1**	**AUC**
ResNet-50	PadChest ImageNet	83.68	84.98	84.72	98.96	82.71	0.94
ResNet-50	PadChest Random	85.42	87.46	80.56	99.13	86.07	0.95
VGG-19	PadChest ImageNet	96.88	96.88	96.88	**99.83**	96.88	**0.99**
VGG-19	PadChest Random	81.94	83.88	79.51	98.61	81.64	0.94
DenseNet-121	PadChest ImageNet	**96.88**	**96.87**	**96.87**	**100.00**	**96.88**	**0.99**
DenseNet-121	PadChest Random	89.58	90.94	87.15	99.65	89.01	0.97

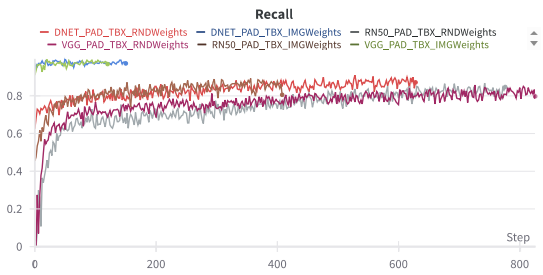
We observed early convergence on the VGG-19 initialised on PadChest ImageNet weights as shown in figure 4.11 and 4.12. We observed a recall of 96.88%, a specificity of 99.83% and an AUC of 0.99 after training respectively. These models show signs of divergence as noted in the various plots where there was early termination in training before the 100 epoch mark. There we cannot rely on the results from these two model configurations for further inference. The other models converged after 400 epochs, with a particular focus on models initialised on the PadChest random weights. The recall and specificity scores for these models were all recorded to be above the required minimum but were still below the baseline scores as shown in table 4.6. The progression of these metrics during TBX11k fine-tuning was recorded in figure 4.11.



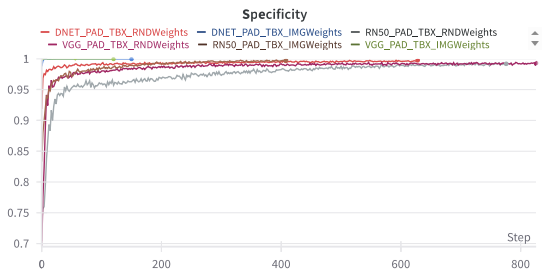
(a) validation accuracy



(b) validation precision

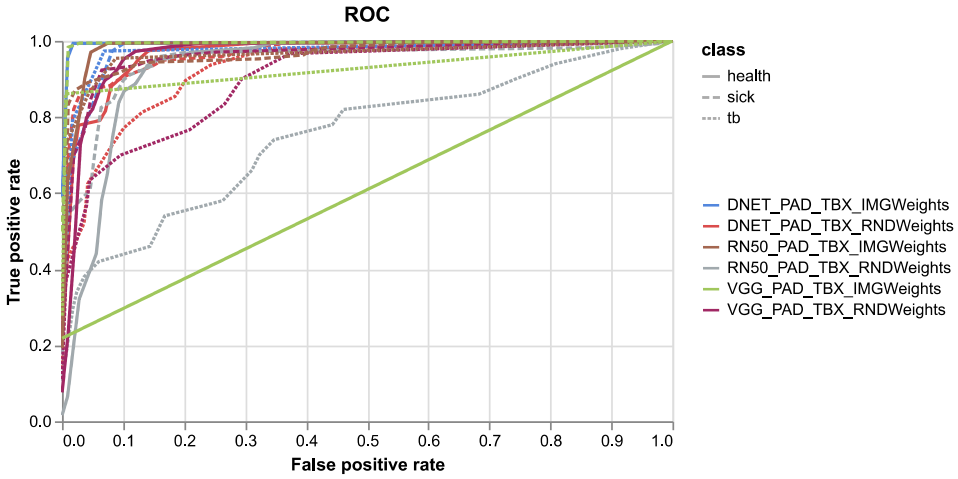


(c) validation recall

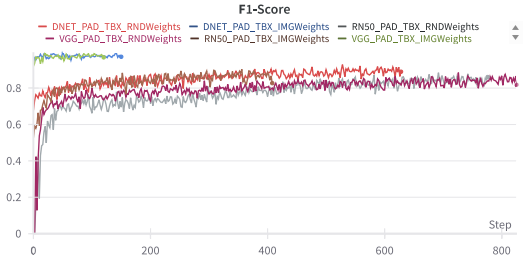


(d) validation specificity

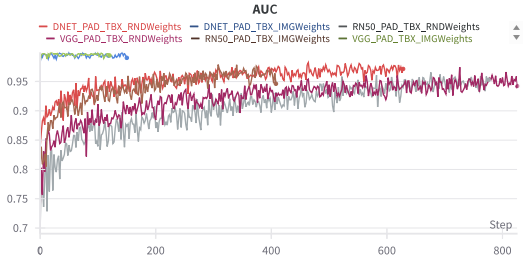
Figure 4.11: TBX11k training results indicating the progression of accuracy, recall, precision, and specificity for models on which pre-training was carried out on the PadChest dataset



(a) The ROC curve



(b) F1-Score



(c) AUC

Figure 4.12: TBX11k training showing validation ROC plot, AUC, and the validation F-Score using the pre-trained weights from the PadChest dataset

As per the ROC curve in figure 4.12, both versions of the Resnet-50 models scored AUC of 0.94 and 0.95 for the TB label classification. The ResNet-50 model initialised with random Padchest weights indicated better performance relative to the ImageNet weights in terms of precision, specificity and AUC score metrics. When we compare the outcomes of these experiments in table (4.6) with baseline experiments shown in table (4.1), the baseline results are robust hence there is a weak argument to support that

pre-training on the PadChest dataset and fine-tuning on the TBX11k CXR dataset is of any benefit. These results indicate that pre-training on the PadChest did not lead to better performance. We suspected that the large number of class labels in the dataset might have caused the models to fail especially in the case of models that were initialised with ImageNet weights.

4.2.6 Conclusion: Supervised Pre-training

In the previous section we established that pre-training on various CXR datasets is unlikely to enhance the performance of deep learning models. We noted that pre-training on the MIMIC-CXR dataset yielded better results in comparison to the results obtained by pre-training on the other datasets. These differences in pre-training results may be attributed to differences in CXR image quality, the dataset labels, and size. Across all the datasets we used for pre-training, using ImageNet initialised weights improved the training speed compared to random weights initialisation. We noted that ImageNet weights can improve the performance of the TB classifier with a higher margin than random weights. We can conclude that supervised pre-training on CXR datasets and fine-tuning on TB datasets cannot improve the performance of the TB classifier. However, there is a general benefit in the choice of weights initialisation, the size of the dataset, image quality, and the quality labels when building TB classifier. Despite our results, CXR dataset collection should aim to increase diversity by collecting images from various population groups to prevent biases from solely training on datasets from the same geographical areas.

4.3 Self Supervised Pre-training and Classification

Other areas of study have shown that self-supervised learning techniques have great potential in performing downstream tasks by finding meaningful patterns with limited labels. In this section, we performed self-supervised pre-training on chest X-ray image datasets using the SSL algorithms DINO or MoCo. We used the CheXpert, VinDr-CXR and PadChest datasets for pre-training. The original images in the CheXpert, VinDr-CXR and PadChest were resized by a scaling factor of 0.25, 0.25 and 0.15 respectively before the pre-training. The weights for each SSL algorithm and dataset configuration were saved.

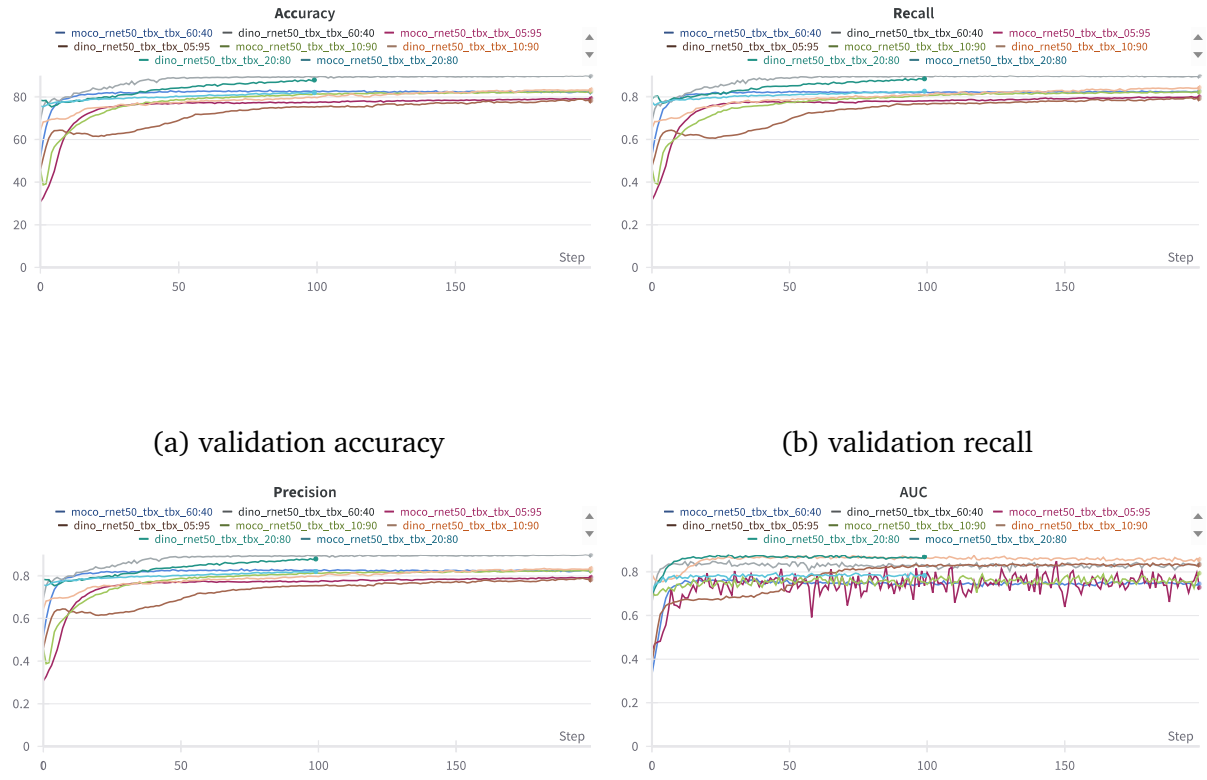
The second phase involved creating another classification model using the SSL pre-trained weights to perform TB image classification using the TBX11k dataset with the same classes mentioned before. We incrementally changed the size of our training-validation-test set ratio to establish the minimum number of labelled images needed to achieve an acceptable level of performance for TB classification. We performed both in-data pre-training using TBX11K dataset and out-of-data pre-training using different datasets highlighted previously. We discuss the experiment results from the different datasets and training/validation split ratios in the following sections.

4.3.1 In-distribution Self-Supervised Pre-training

In this experiment we pre-trained on the TBX11k dataset using either DINO or MoCo. We used 80% of the TBX11k dataset for pre-training and the remaining TBX11k (20%) dataset for the downstream TB classification task. The first classification task had the following split ratio, 60% of the images for the training set and 40% for the validation set. In another experiment, we split the images into the following ratios: 20% of the images for the train set and 80% for the validation set. We further performed other experiments by reducing the train set ratios as shown in table 4.7. In total, we created eight sub-experiments that differ in terms of data split ratio and algorithm dataset used. We created models with the ResNet-50 model as the backbone and used the weights previously saved for each algorithm-dataset configuration. We performed fine-tuning for up to 200 epochs until convergence in some configurations. Table 4.7 summarizes the results from the different configurations.

Table 4.7: SSL: In-distribution transfer learning - Different Split Ratios on TBX11K results. ACC denotes Accuracy, PRE denotes Precision, SPE represents Specificity, F1 is the F1-Score

Model	**Split-Ratio**	**ACC**	**PRE**	**Recall**	**SPE**	**F1**	**AUC**
DINO-ResNet-50	60:40	**89.40**	**89.60**	**89.26**	**89.66**	**89.42**	**0.89**
MoCO-ResNet-50	60:40	82.33	83.45	82.45	82.03	82.95	0.78
DINO-ResNet-50	20:80	88.55	88.70	88.49	84.38	88.59	0.88
MoCO-ResNet-50	20:80	82.54	82.66	82.96	84.49	82.81	0.79
DINO-ResNet-50	10:90	84.02	84.28	81.71	81.70	82.98	0.83
MoCO-ResNet-50	10:90	82.19	82.30	82.14	82.14	82.22	0.77
DINO-ResNet-50	5:95	78.88	78.91	78.99	78.91	78.95	0.82
MoCO-ResNet-50	5:95	80.96	79.79	80.20	79.80	79.79	0.78



(a) validation accuracy

(b) validation recall

(c) validation precision

(d) validation AUC

Figure 4.13: Progression of Accuracy, Recall, Precision and AUC across various split-ratios either pre-trained on MoCo/Dino with TBX11k weights. The models were fine-tuned on a subset of the TBX11k dataset with 80% pre-training set and 20% train-validation set

In table 4.7, we presented the results from the in-distribution pre-training and transfer learning using the TBX11K dataset in which we showed the effect of the gradual decrease in the train split ratios. We compared the performance of weights obtained from ResNet-50 pre-trained on the MoCo algorithm and ResNet-50 pre-trained on the DINO algorithm. As per the results from table 4.7 the 60:20 split on the images that were set aside from the self-supervised pre-training achieved an accuracy of 89.40%, a recall of 89.26%, an F1-Score of 89.42% and an AUC of 0.88 on the TB class using the DINO-ResNet-50 model. The same split ratio with ResNet-50 MoCo weights yielded an accuracy of 82.33%, a recall of 82.45%, an F1-Score of 82.95% and an AUC of 0.78 on the TB class. We further decreased the split ratios by decreasing the percentage of the training images to a ratio of 5% training and 95% validation. With the smallest split ratio, the model scored an accuracy of 78.88%, a precision of 78.91% on the DINO-ResNet-50 while the MoCo-ResNet-50 scored an accuracy of 80.96%, a specificity of 79.80%, a recall of 80.20% and we noted an AUC of 0.78 for the TB class.

In this set of experiments we noted a general decline in the performance metrics as we gradually decrease the train set ratio. This can be observed in the various plots in figure 4.13 in which all low-ranking plots indicate the smaller train ratio with higher-ranking plots indicating higher train splits. We observed that the DINO-ResNet-50 weights scored higher performance metrics when compared to the MoCo-ResNet weights with the same split ratios. We also observed that in the experiment with the smallest number of training labels (5%), the MoCo-ResNet-50 pre-training configuration scored 80.20% on recall and 79.80% on specificity. We can speculate that perhaps MoCo pre-training scales better with a few labels than the Dino pre-training configuration with 78.99% and 78.91% on recall and specificity respectively. Although the results metrics are lower than the supervised baseline we cannot rule out the potential of in-data self-supervised pre-raining.

4.3.2 Experiment 1: Out-of-distribution Self-Supervised Pre-training

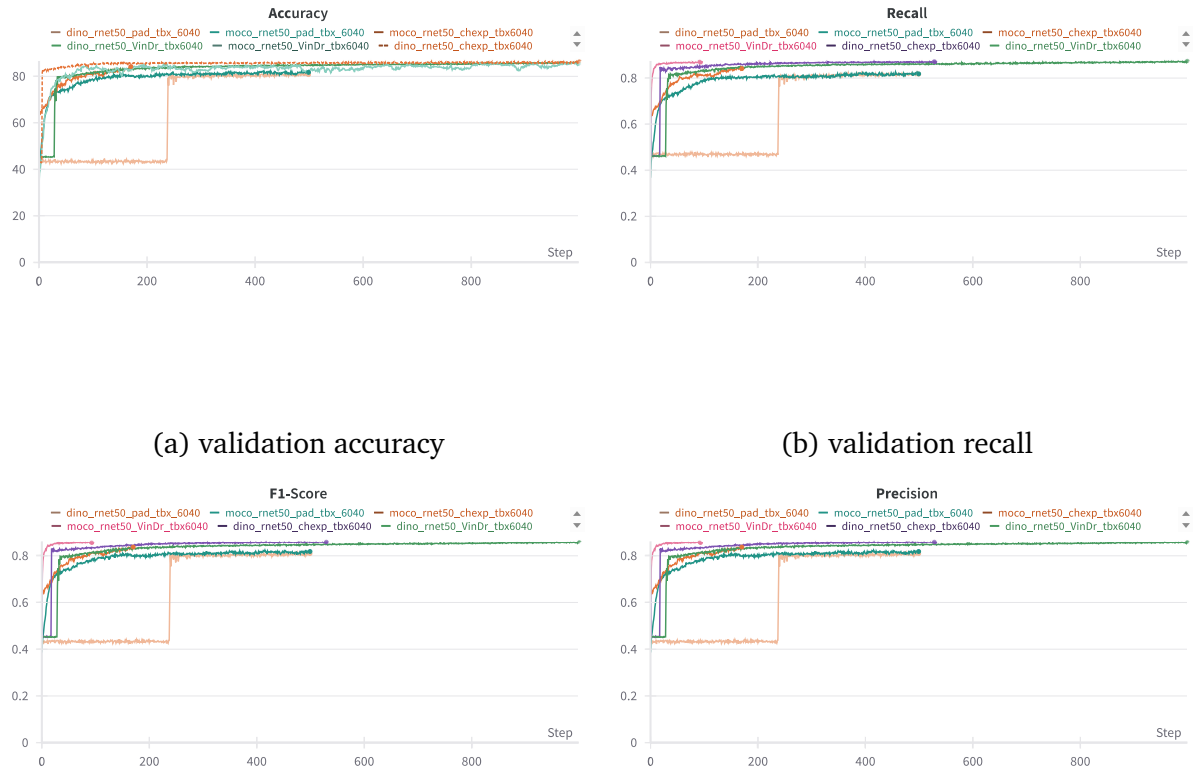
In this experiment we pre-trained the ResNet-50 model for up to 100 epochs on either of the SSL algorithms DINO and MoCo until convergence. We performed pre-training on each of the datasets VinDr, CheXpert and PadChest. For the downstream TB classification task, we used the TBX11k dataset that we split into the following ratios, 60% of the images for the train set and 40% for the validation set. We created ResNet-50 models initialised with the weights previously saved for each algorithm - dataset configuration. We performed fine-tuning for up to 500 epochs until convergence. Table 4.8 summarises the results from different configurations.

Table 4.8: SSL 60:40 Split Ratio on TBX11K results. ACC denotes Accuracy, PRE denotes Precision, SPE represents Specificity, F1 is the F1-Score

Model	**Weights**	**ACC**	**PRE**	**Recall**	**SPE**	**F1**	**AUC**
DINO-ResNet-50	DINO-CheXpert	**85.49**	**85.16**	**85.60**	**85.10**	**85.38**	0.70
MoCO-ResNet-50	MoCo-CheXpert	84.26	83.15	83.20	83.25	83.17	0.54
DINO-ResNet-50	DINO-VinDr-CXR	83.15	84.38	84.68	84.40	84.53	0.64
MoCO-ResNet-50	MoCo-VinDr-CXR	84.60	84.69	84.87	84.49	84.78	**0.81**
DINO-ResNet-50	DINO-PadChest	80.73	81.72	81.77	81.50	81.74	0.55
MoCO-ResNet-50	MoCo-PadChest	82.10	81.95	81.94	81.52	81.89	0.54

In this experiment, we fine-tuned on the TBX11k dataset with 60% of the training set. The best-performing model was the DINO-ResNet-50 Model which was initialised with weights from the CheXpert dataset pretraining. A recall score of 85.60% and a specificity score of 85.10% were recorded for this model configuration. This was outperformed by the in-distribution models with the same 60% training split if we

compare results in table 4.7 and table 4.8. In all the models and weights configurations, a recall score of 85.60% indicated that self-supervised pre-training on unlabelled chest X-ray image datasets yielded substantial results when we used the same weights to carry out downstream classification tasks with a particular interest in the classification of TB.



(a) validation accuracy

(b) validation recall

(c) validation F1-Score

(d) validation Precision

Figure 4.14: Progression of accuracy, recall, precision and F1-Score on MoCo/Dino pre-trained on VinDr-CXR, CheXpert and PadChest. The models were fine-tuned on the TBX11k dataset with 60% training set and 40% validation set

In figure 4.14 we showed the progression of the main metrics including accuracy, recall, precision and the F1-Score during the fine-tuning process. The models seem to show scalability across the different weights from the datasets. We noted that the ResNet-50 model pre-trained using the DINO algorithm on the PadChest took a long time to reach the optimal training level which was reached after 200 epochs of training and the ResNet-50 model that was pre-trained on the MoCo algorithm using the VinDr-CXR dataset out-performed the other models in terms of quick convergence.

4.3.3 Experiment 2: Out-of-distribution Self-Supervised Pre-training

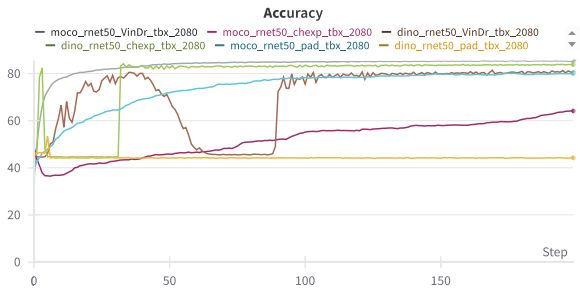
In this experiment we pre-trained the ResNet-50 model for up to 200 epochs on either of the SSL algorithms DINO and MoCo using the three datasets highlighted before. The weights for each configuration were saved. For the downstream TB classification task, we used the TBX11k dataset with the following ratios: 20% for the train set, and 80% for the validation set. This significantly reduced the number of image labels on the downstream task. We created ResNet-50 models and initialised the weights for each algorithm-dataset configuration. We trained on the TBX11k dataset for up to 500 epochs in some runs. Table 4.9 shows the results from the different training configurations.

Table 4.9: SSL 20:80 Split Ratio on TBX11K results. ACC denotes Accuracy, PRE denotes Precision, SPE represents Specificity, F1 is the F1-Score

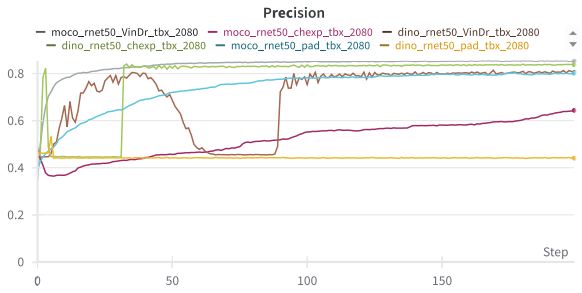
Model	**Weights**	**ACC**	**PRE**	**Recall**	**SPE**	**F1**	**AUC**
DINO-ResNet-50	DINO-CheXpert	84.26	84.01	81.19	81.76	82.58	0.61
MoCO-ResNet-50	MoCo-CheXpert	63.67	63.70	63.73	63.79	63.71	0.66
DINO-ResNet-50	DINO-VinDr-CXR	80.20	81.31	80.26	75.72	80.78	0.47
MoCO-ResNet-50	MoCo-VinDr-CXR	**85.19**	**85.25**	**85.33**	**86.96**	**85.29**	**0.82**
DINO-ResNet-50	DINO-PadChest	45.37	45.34	45.03	45.33	45.18	0.48
MoCO-ResNet-50	MoCo-PadChest	80.25	80.26	80.19	80.26	80.22	0.62

Table 4.9 shows the summary of the results and the training progress is shown by the plots in figure 4.15. We observed a decline in performance by the ResNet Model pre-trained on the CheXpert dataset using MoCo as the backbone algorithm. This model setup achieved a sensitivity of 63.73% and a specificity of 63.79%. There was a marked decline in metrics performance measured for the DINO-ResNet-50 model pre-trained on the PadChest dataset with an accuracy of 45.37%, a recall of 45%, and a sensitivity of 45.33%. There was no further improvement in the results even after increasing the training time and performing hyper-parameter tuning. The MoCo-ResNet-50 model which was pre-trained on the VinDr-CXR dataset out-performed all the other models with a recall of 85.33%, a specificity of 86.96% and an AUC of 0.82, however, these results showed a small increase from the previous experiment in which we used a training set of 60%.

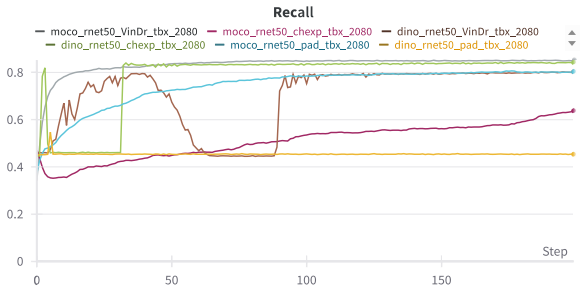
The plots in figure 4.15 indicate that the ResNet-50 model with the MoCo backbone pre-trained on the CheXpert dataset initially showed a steady rise in performance followed by another decline and then converged after 200 epochs with a recall of 63.73%, a specificity of 63.79% and AUC of 0.66. This was an unusual phenomenon when com-



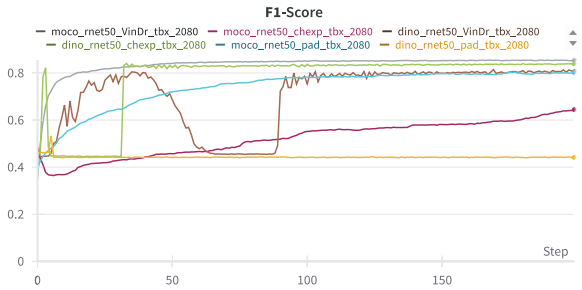
(a) validation accuracy



(b) validation Precision



(c) validation recall



(d) validation F1-Score

Figure 4.15: accuracy, recall, precision, and F1-Score across the models either on MoCo/Dino pre-trained on VinDr-CXR, CheXpert, CheXpert weights. The models were fine-tuned on the TBX11k dataset with 20% training set and 80% validation set

pared to the other model configurations. We can speculate that the model took longer to adjust to the hyper-parameters or the reduction in the training set.

4.3.4 Experiment 3: Out-of-distribution Self-Supervised Pre-training

In this experiment we pre-trained the ResNet-50 model for up to 100 epochs with the DINO and MoCo algorithms on the three datasets VinDr-CXR, CheXpert and PadChest. The weights from each model configuration were saved and used as the backbone for the classifier. In the downstream TB classification task, we used the TBX11k dataset with the following split ratios, 10% of the images for the training set, and 90% for the validation set. We implemented ResNet-50 models and initialised the weights previously saved for each algorithm-dataset configuration. We performed fine-tuning for up to 200 epochs until convergence. Table 4.10 summarises the results from the different experiment configurations.

Table 4.10: SSL 10:90 Split Ratio on TBX11K results. ACC denotes Accuracy, PRE denotes Precision, SPE represents Specificity, F1 is the F1-Score

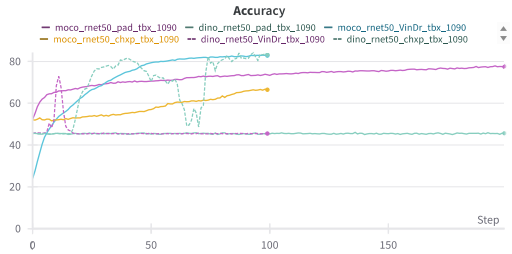
Model	**Weights**	**ACC**	**PRE**	**Recall**	**SPE**	**F1**	**AUC**
DINO-ResNet-50	DINO-CheXpert	81.6	**81.52**	**83.03**	**82.99**	82.27	0.48
MoCO-ResNet-50	MoCo-CheXpert	65.04	65.01	65.21	65.41	65.11	0.50
DINO-ResNet-50	DINO-VinDr-CXR	44.67	45.07	45.01	45.07	45.04	0.39
MoCO-ResNet-50	MoCo-VinDr-CXR	**82.67**	82.6	82.25	82.61	**82.42**	**0.80**
DINO-ResNet-50	DINO-PadChest	44.67	45.45	45.40	45.35	45.42	0.46
MoCO-ResNet-50	MoCo-PadChest	77.43	77.57	77.80	77.46	77.68	0.52

The 10:90 data split ratio indicated a drastic decline in the number of training images when we consider that deep learning models are generally data-hungry. Despite a general decline across all the models in this group of experiments, we observed a marked decline in performance from two of the model configurations.The ResNet-50 initialised with DINO-VinDr-CXR weights scored a recall of 45.01% and a specificity of 45.07%, while the ResNet-50 initialised with the DINO-PadChest weights recorded a recall of 45.40% and specificity of 45.35%. The inability of these models to learn and generalise was observed in the training process as shown in the various plots in figure 4.16 in which the graphs indicate straight lines for both models.

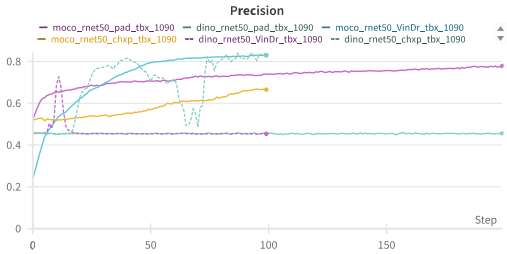
Besides these two models, we still noted a substantial resilience from the other models. We noted that the ResNet-50 model initialised with the DINO-CheXpert weights, ResNet-50 initialised with MoCo-VinDr-CXR weights and the MoCo-ResNet-50 on the PadChest dataset performed well relative to the other models. With this minimal number of training images, we noted that these three model configurations recorded a recall of 83.03%, 82.25% and 77.80% respectively. These results further support our initial assumption that self-supervised pre-training is a viable method for building deep-learning TB classifiers in scenarios when we do not have enough labelled data.

4.3.5 Experiment 4: Out-of-distribution Self-Supervised Pre-training

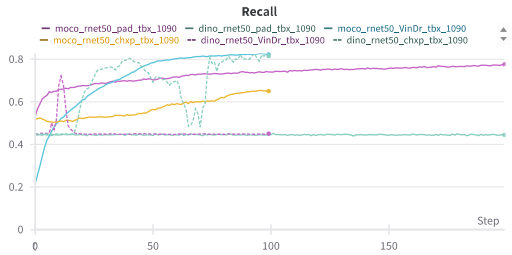
In this experiment we reduced the number of images to about 420 which is 4% for the training set. In the downstream TB classification task, we used the TBX11k dataset with the following ratios, 5% of the images for the train set and 95% for the validation set. We created ResNet-50 models loaded the weights previously saved for each algorithm-dataset pre-training configuration and performed training for up to 200 epochs until convergence. Table 4.11 summarises the results from the different configurations and figure 4.17 shows plots that track the training process.



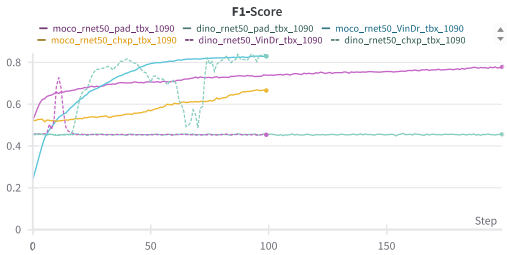
(a) validation accuracy



(b) validation Precision



(c) validation recall



(d) validation F1-Score

Figure 4.16: Progression of accuracy, recall, precision and F1-Score across all the models when we fine-tuned on the TBX11k dataset with 10% training set and 90% validation set

Results from table 4.11 show a decline in performance metrics across all model configurations with metrics of below 50% when compared to the previous experiment when we used 10% images for training. Only two model configurations maintained higher metrics including the ResNet-50 model initialised with MoCo-VinDr-CXR weights and ResNet-50 initialised with weights obtained from MoCo-PadChest dataset. We recorded a recall of 81.90%, a specificity of 81.99% and an AUC of 0.78 on the TB class for the best-performing MoCo-VinDr-CXR pre-trained model. From these results, we observed that self-supervised pre-training can still yield significant results with the smallest number of labelled images when performing TB classification using chest X-ray images.

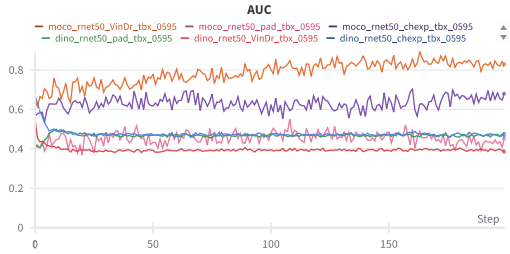
The plots in figure 4.17 further support our observations for the last split ratio in which we observed that four of our models failed while two MoCo-based ResNet-50 models continue to show resilience with limited training images which indicate 5% of our TBX11k dataset.

Table 4.11: SSL 5:95 Split Ratio on TBX11K results. ACC denotes Accuracy, PRE denotes Precision, SPE represents Specificity, F1 is the F1-Score

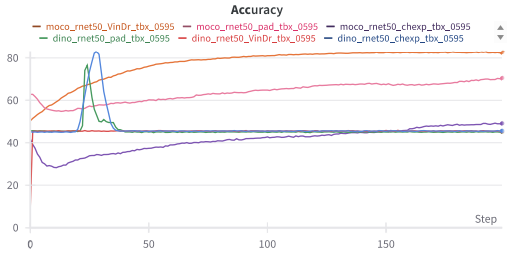
Model	**Weights**	**ACC**	**PRE**	**Recall**	**SPE**	**F1**	**AUC**
DINO-ResNet-50	DINO-CheXpert	44.80	44.81	44.86	44.82	44.83	0.46
MoCO-ResNet-50	MoCo-CheXpert	49.27	48.83	48.80	48.88	48.81	0.63
DINO-ResNet-50	DINO-VinDr-CXR	44.93	44.63	44.72	44.63	44.67	0.39
MoCO-ResNet-50	MoCo-VinDr-CXR	**82.18**	**81.98**	**81.90**	**81.99**	**81.94**	**0.78**
DINO-ResNet-50	DINO-PadChest	45.46	45.49	45.50	45.53	45.49	0.48
MoCO-ResNet-50	MoCo-PadChest	69.51	70.21	70.29	70.22	70.25	0.48

4.3.6 Conclusion: Self-Supervised pre-training

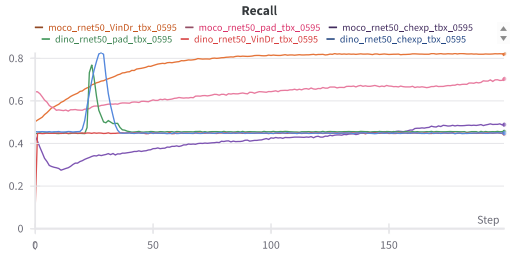
In this section we aimed to investigate how various self-supervised training regimes affect the performance of the TB classifier when it comes to TB chest X-ray diagnosis across various datasets. We observed that self-supervised pre-training can yield significant results with the smallest number of labels as low as 420 in the training set for the task at hand. We observed that self-supervised pre-training within the same dataset yielded higher performance when compared to out-of-data self-supervised pre-training. Although the performance metrics values for the self-supervised approaches are lower than those observed in the supervised baseline experiments, this section has established that we can build viable TB classifiers even when we have limited labels in a dataset. It should be noted that the pre-training experiments took a lot of computing resources and took a longer time to converge. However, the fine-tuning downstream task was faster relative to the pre-training experiments with average convergence achieved after 200 epochs. We cannot rule out that possible improvements in model structure and hyper-parameter tuning in under-performing models will enhance the results.



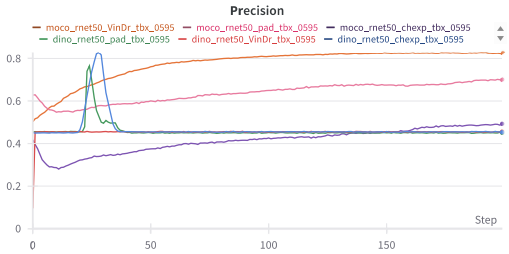
(a) validation AUC



(b) validation accuracy



(c) validation recall



(d) validation Precision

Figure 4.17: Training progression of accuracy, recall, precision and F1-Score across ResNet-50, VGG-19, and DenseNet-121 either pre-trained on MoCo/Dino with VinDr-CXR, CheXpert, PadChest weights. The models were fine-tuned on the TBX11k dataset with 5% training set and 95% validation/test set

Chapter 5

Conclusions and Future Work

In this research report we conducted various experiments spanning over different datasets and deep learning models to investigate whether transfer learning can yield significant results in predicting Tuberculosis. In this chapter, we submit the concluding remarks on this research, and we give pointers to potential work in using deep learning as a tool for TB diagnosis.

5.1 Conclusions

Deep learning has become important for analysing and processing chest X-ray images. Though supervised methods have dominated the field in building image classifiers, work still needs to be put into exploring the viability of self-supervised learning methods. In this report, we have conducted various experiment configurations to understand how pre-training on various CXR datasets impacts the TB classification.

We have established that pre-training using international datasets that contain labels for other chest-affecting diseases will not improve the classification accuracy of TB when fine-tuned on other CXR datasets. We observed that choosing to initialise the model with ImageNet weights contributes to a meaningful increase in performance accuracy compared to initialising with random weights when pre-training on international datasets.

This research has established that pre-training can significantly fast-track the process of TB classification by reducing the training time thus saving computing resources. The comparison between models configured to train using pre-trained ImageNet weights and randomly initialised weights has confirmed that ImageNet pre-trained models tend to yield better performance when we observe the various metrics. The results indicated that pre-training on large CXR datasets does not lead to improvement in model performance when we fine-tune on a smaller TB dataset. Besides these results, we noted that the size and quality of labels in a dataset play an important role, especially in the performance metrics we observed by comparing pre-training results from VinDr-CXR and ChestX-ray14 datasets. We established that self-supervised pre-training on CXR images has the potential to yield the same results which are equally accurate when per-

forming TB classification in CXR images with the smallest number of labels. We noted that in-data pre-training yields significant performance results more than out-of-data pre-training. The self-supervised pre-training results failed to match the result of the supervised baseline result, therefore more work should be directed towards building models based on self-supervised algorithms such as DINO or MoCo. The abundance of unlabelled datasets especially in the South African context presents a great opportunity for self-supervised pre-training.

5.2 Future Work

Future research in this field should aim to understand which of the multi-labels identified in the various CXR datasets contribute highly to the presence or absence of TB. Most online datasets do not have definitive TB labels. This will assist us in developing new datasets or algorithms that can extract and group these labels to determine their contribution to the presence or absence of tuberculosis. With the recent massive uptake of Large Language Models (LLMs) in various domains, we believe these models can automate the generation of chest X-ray reports, summarising chest X-ray reports and extracting labels from chest X-ray images. This will allow us to get more insight and fast-track the process of TB diagnosis. Furthermore, there is potential in using multi-modal data, such as combining chest X-ray images with clinical data or genetic information to enhance the accuracy of TB diagnosis using LLMs.

There is an urgent need to build and consolidate South African datasets. This will assist in establishing whether the quality and labels on these datasets are useful in developing deep learning models to solve the problem of TB diagnosis in South Africa. Performing self-supervision pre-training on international TB data has shown that this is a viable method of dealing with scarce chest X-ray labels hence we should focus on extending these techniques to South African-sourced datasets.

References

[Ait Nasser and Akhloufi 2023] Adnane Ait Nasser and Moulay A Akhloufi. A review of recent advances in deep learning models for chest disease detection using radiography. *Diagnostics*, 13(1):159, 2023.

[An *et al.* 2022] Le An, Kexin Peng, Xing Yang, Pan Huang, Yan Luo, Peng Feng, and Biao Wei. E-tbnet: Light deep neural network for automatic detection of tuberculosis with x-ray dr imaging. *Sensors*, 22(3):821, 2022.

[Biju *et al.* 2022] Roshima Biju, Warish Patel, K Suresh Manic, and Venkatesan Rajinikanth. Framework for classification of chest x-rays into normal/covid-19 using brownian-mayfly-algorithm selected hybrid features. *Mathematical Problems in Engineering*, 2022, 2022.

[Bustos *et al.* 2020] Aurelia Bustos, Antonio Pertusa, Jose-Maria Salinas, and Maria De La Iglesia-Vaya. Padchest: A large chest x-ray image dataset with multi-label annotated reports. *Medical image analysis*, 66:101797, 2020.

[Caron *et al.* 2021] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[Dasanayaka and Dissanayake 2021] Chirath Dasanayaka and Maheshi Buddhinee Dissanayake. Deep learning methods for screening pulmonary tuberculosis using chest x-rays. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 9(1):39–49, 2021.

[Devnath *et al.* 2022] Liton Devnath, Zongwen Fan, Suhuai Luo, Peter Summons, and Dadong Wang. Detection and visualisation of pneumoconiosis using an ensemble of multi-dimensional deep features learned from chest x-rays. *International journal of environmental research and public health*, 19(18):11193, September 2022.

[Dey *et al.* 2022] Subhrajit Dey, Rajarshi Roychoudhury, Samir Malakar, and Ram Sarkar. An optimized fuzzy ensemble of convolutional neural networks for detecting tuberculosis from chest x-ray images. *Applied Soft Computing*, 114:108094, 2022.

[Donahue *et al.* 2014] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation

feature for generic visual recognition. In *International conference on machine learning*, pages 647–655. PMLR, 2014.

[He *et al.* 2020] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross 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.

[Iqbal *et al.* 2022] Ahmed Iqbal, Muhammad Usman, and Zohair Ahmed. An efficient deep learning-based framework for tuberculosis detection using chest x-ray images. *Tuberculosis*, 136:102234, 2022.

[Irvin *et al.* 2019] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie 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.

[Jaeger *et al.* 2014] Stefan Jaeger, Sema Candemir, Sameer Antani, Yì-Xiáng J Wáng, Pu-Xuan Lu, and George Thoma. Two public chest x-ray datasets for computer-aided screening of pulmonary diseases. *Quantitative imaging in medicine and surgery*, 4(6):475, 2014.

[Johnson *et al.* 2019] Alistair EW Johnson, Tom J Pollard, Seth J Berkowitz, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Roger G Mark, and Steven Horng. Mimic-cxr, a de-identified publicly available database of chest radiographs with free-text reports. *Scientific data*, 6(1):317, 2019.

[Kim *et al.* 2022] Sungyeup Kim, Beanbonyka Rim, Seongjun Choi, Ahyoung Lee, Sedong Min, and Min Hong. Deep learning in multi-class lung diseases' classification on chest x-ray images. *Diagnostics*, 12(4):915, 2022.

[Krishna Puttagunta and Ravi 2021] Murali Krishna Puttagunta and S Ravi. Detection of tuberculosis based on deep learning based methods. In *Journal of Physics Conference Series*, volume 1767, page 012004, 2021.

[Liu *et al.* 2020] Yun Liu, Yu-Huan Wu, Yunfeng Ban, Huifang Wang, and Ming-Ming Cheng. Rethinking computer-aided tuberculosis diagnosis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2646–2655, 2020.

[Long *et al.* 2015] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[Mohan *et al.* 2022] Ramya Mohan, Seifedine Kadry, Venkatesan Rajinikanth, Arnab Majumdar, and Orawit Thinnukool. Automatic detection of tuberculosis using vgg19 with seagull-algorithm. *Life*, 12(11), 2022.

[Newell and Deng 2020] Alejandro Newell and Jia Deng. How useful is self-supervised pretraining for visual tasks? 2020 ieee. In *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7343–7352, 2020.

[Nguyen *et al.* 2020] Ha Q. Nguyen, Khanh Lam, Linh T. Le, Hieu H. Pham, Dat Q. Tran, Dung B. Nguyen, Dung D. Le, Chi M. Pham, Hang T. T. Tong, Diep H. Dinh, Cuong D. Do, Luu T. Doan, Cuong N. Nguyen, Binh T. Nguyen, Que V. Nguyen, Au D. Hoang, Hien N. Phan, Anh T. Nguyen, Phuong H. Ho, Dat T. Ngo, Nghia T. Nguyen, Nhan T. Nguyen, Minh Dao, and Van Vu. *VinDr-CXR: An open dataset of chest X-rays with radiologist's annotations*, 2020.

[Nguyen *et al.* 2022] Ha Q Nguyen, Khanh Lam, Linh T Le, Hieu H Pham, Dat Q Tran, Dung B Nguyen, Dung D Le, Chi M Pham, Hang TT Tong, Diep H Dinh, et al. Vindr-cxr: An open dataset of chest x-rays with radiologist's annotations. *Scientific Data*, 9(1):429, 2022.

[Park *et al.* 2022] Sangjoon Park, Gwanghyun Kim, Yujin Oh, Joon Beom Seo, Sang Min Lee, Jin Hwan Kim, Sungjun Moon, Jae-Kwang Lim, Chang Min Park, and Jong Chul Ye. Self-evolving vision transformer for chest x-ray diagnosis through knowledge distillation. *Nature Communications*, 13(1):3848, 2022.

[Rahman *et al.* 2020a] Tawsifur Rahman, Amith Khandakar, Muhammad Abdul Kadir, Khandaker Rejaul Islam, Khandakar F. Islam, Rashid Mazhar, Tahir Hamid, Mohammad Tariqul Islam, Saad Kashem, Zaid Bin Mahbub, Mohamed Arselene Ayari, and Muhammad E H Chowdhury. Reliable tuberculosis detection using chest x-ray with deep learning, segmentation and visualization. *IEEE Access*, 8:191586–191601, 2020.

[Rahman *et al.* 2020b] Tawsifur Rahman, Amith Khandakar, Muhammad Abdul Kadir, Khandaker Rejaul Islam, Khandakar F Islam, Rashid Mazhar, Tahir Hamid, Mohammad Tariqul Islam, Saad Kashem, Zaid Bin Mahbub, et al. Reliable tuberculosis detection using chest x-ray with deep learning, segmentation and visualization. *IEEE Access*, 8:191586–191601, 2020.

[Ravi *et al.* 2022] Vinayakumar Ravi, Vasundhara Acharya, and Mamoun Alazab. A multichannel efficientnet deep learning-based stacking ensemble approach for lung disease detection using chest x-ray images. *Cluster Computing*, pages 1–23, 2022.

[Ravi *et al.* 2023] Vinayakumar Ravi, Vasundhara Acharya, and Mamoun Alazab. A multichannel efficientnet deep learning-based stacking ensemble approach for lung disease detection using chest x-ray images. *Cluster Computing*, 26(2):1181–1203, 2023.

[Showkatian *et al.* 2022] Eman Showkatian, Mohammad Salehi, Hamed Ghaffari, Reza Reiazi, and Nahid Sadighi. Deep learning-based automatic detection of tuberculosis disease in chest x-ray images. *Polish Journal of Radiology*, 87(1):118–124, 2022.

[Wang *et al.* 2017] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017.

[WHO and others 2021] WHO et al. Who consolidated guidelines on tuberculosis: module 2: screening: systematic screening for tuberculosis disease. web annex c: Grade evidence to decision tables. 2021.

[WHO and others 2022] WHO et al. *Global tuberculosis report 2022*. WHO and others, 2022.

[Xu and Yuan 2022] Tianhao Xu and Zhenming Yuan. Convolution neural network with coordinate attention for the automatic detection of pulmonary tuberculosis images on chest x-rays. *IEEE Access*, 10:86710–86717, 2022.

[Zaidi *et al.* 2022] S Zainab Yousuf Zaidi, M Usman Akram, Amina Jameel, and Norah Saleh Alghamdi. A deep learning approach for the classification of tb from nih cxr dataset. *IET Image Processing*, 16(3):787–796, 2022.