Master Thesis

A Comparative Study of Visual Transformers for Thoracic Disease Classification

Demet Demirkiran

Master Thesis

Thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science of Artificial Intelligence
at the Department of Data Science and Knowledge Engineering
of the Maastricht University

Thesis Committee:

Alexia Briassouli
Mirela Popa

Maastricht University
Faculty of Science and Engineering
Department of Data Science and Knowledge Engineering

ADD MONTH, 2022

Contents

Abstract

A common process for diagnosing Thoracic disease is the acquisition and interpretation of Chest X-rays (CXR). However, this requires highly specialized and experienced medical professionals. The quality of the diagnosis and its timeliness directly impact the quality of care the patient receives. The recent emergence of Computer-Aided Diagnostic systems attempts to improve the workflow of doctors, generally by training a Convolutional Neural Network to classify the images. We have undertaken a comparative study on different classification architectures, and losses to determine what would aid the best in this endeavor. We compare 6 different architectures, three of which are based on recent transformer models. We also evaluate metric losses to see if they will aid in the improvement of the classification losses. In this work, it was found that for classification loss by itself that the model ResNet+Self_Attention achieves the highest result with 76.50% and 78.45% for the datasets ChestX-ray14 and CheXpert, respectively. Meanwhile, with joint classification and metric losses, the model ViT Hybrid achieves the highest classification result with 76.93% on ChestX-ray14. With the results found from this study it can be seen that Vision Transformers are suitable for Computationally Assisted Diagnosis of thoracic disease. Furthermore, the addition of metric losses together with transformer models increases the performance by at least 1%.

Chapter 1

Introduction

For Thoracic disease classification Chest X-rays (CXR), are one of the most commonly used diagnostic methods. However, to be able to read CXR images highly specialized and experienced medical professionals are needed. The celerity and accuracy of diagnosis depend on several factors, such as the available information, the experience of the doctor, the nature of the disease, and the patient. Depending on the pathology presented by the patient, the speed at which a diagnosis is reached could be vital for achieving a positive outcome. Therefore, the speed and quality of a diagnosis are directly related to the care and treatment a patient receives.

An emergent trend in literature is the deployment of systems capable of assisting the doctor in the diagnostic task. Some of these systems directly analyze the images and predict the pathology presented in the pictures. Commonly, Convolutional Neural Networks are trained and deployed, as they have become the state-of-the-art in several computer vision tasks. However, computer vision literature moves at a blistering pace, and newer, better, or more efficient architectures and techniques are published rapidly. Recent literature in image classification has exploited advances from Natural Language Processing to achieve better performance on visual tasks [49, 8]. It is of interest to know if some of these recent advances can also benefit Computer-Aided Diagnostics.

To that end, we have undertaken a comparative study to determine if recently released state-of-the-art architectures, specifically attention-based models, work comparably or better than standard classification networks such as ResNet [13]. Furthermore, we also study whether classification losses are the best option or if the addition of auxiliary loss terms, such as metric losses, provides better performance. In this study, we deploy 6 different models, ResNet50 [13], ResNet50+Self_Attention [49], CBAM [53], C-Tran [24], ViT [8] and ViT Hybrid [8]. We consider several standard classification losses (Binary Cross-Entropy Loss (BCE), Cross-Entropy Loss (CEL) [51], Weighted Cross-Entropy Loss (WCEL) [51], and Focal Loss [25]), as well as various metric losses (Contrastive Loss [11], Triplet Loss [41], and Proxy-NCA loss [31]).

The previous objective can be succinctly described by the following research questions:

1. Are Vision Transformers suitable for Computationally Assisted Diagnosis of thoracic disease?

2. Do attention-based methods outperform current state-of-the-art systems for thoracic disease classification?

3. Does learning with metric loss functions, such as Contrastive or Triplet loss, improve the overall classification performance?

This work is structured as follows, Chapter 2 summarizes related works and recent trends in literature. Chapter 3 provides an overview of our methodology and the necessary concepts. Chapter 4 presents the results of our experiments. Finally, Chapter 5 provides the conclusion and recommendations for future work.

Chapter 2

Related Works

In this chapter, a summary of research on the chest X-ray classification problem is presented. The simpler single-label case is discussed, followed by the datasets that enable the extension of the problem to the multi-label domain. Afterward, the state-of-the-art is studied, which introduces the vision transformer model.

The chapter is structured as follows. Section 2.1 gives an overview of single-label Chest X-ray (CXR) classification, including the different architectures and methods used. Then, Section 2.2 delves into multi-label CXR classification. Therein, popular models and techniques are described. The datasets used in this study are introduced in Sections 2.2.1 and 2.2.2. It is followed by Section 2.2.3, wherein the three main approaches identified in the literature (for the multi-label case) are presented. These approaches being the CNN, Metric learning, and attention-based approaches. Lastly, Section 2.2.4 gives a small overview of the latest state-of-the-art approach to image classification.

2.1 Single Label Chest X-ray Classification

Being able to interpret CXR images requires long and specialized training. Due to this, it is desirable to develop a computer-assisted diagnostic(CAD) system capable of accurately predicting a pathology from a CXR. The intention is not to replace the medical expertise of radiologists but to aid them and to be able to speed up their diagnostics. A specific sub-category of radiology deals with thoracic disease, which is the focus of this work.

One of the earliest works for computer-assisted diagnosis was by Hall *et al* [12]. In their paper, the authors of [12] developed a method for classifying textural patterns occurring in the lungs of patients suffering from pneumoconiosis. Their results showed that such a system would be feasible for X-ray classification. Another study uses Gradient Pattern Coding [20] to identify specific patterns that correlate to the severity of pneumoconiosis. In the paper published by Savol *et al* [40], an adaptive object growing algorithm for CAD of small rounded opacities in patients CXR for automated detection of early stages

of pneumoconiosis was proposed.

Instead of relying on hand-crafted solutions for CAD (CXR in this case), it is possible to employ machine learning techniques to leverage the available data. Therefore, yielding systems capable of benefiting doctors and patients. Unfortunately, unbalanced datasets or medical professionals unable to simplify their diagnostic criteria into formulas may complicate the deployment of these techniques. We consider X-ray image analysis to be a computer vision task. Convolutional Neural Networks (CNNs) and deep Learning (DL) have become popular tools to solve computer vision problems. Currently, in the domain of medical imaging, researchers have turned their attention towards models applied for classification to exploit their immense data computing ability.

In single-label CXR classification, the models are used to learn to model and find the diagnoses for only one ailment at a time. This form of modeling does not consider that a person may have more than one ailment at a time. Doing this reduces the complexity of the problem. The models typically used in classification for CXR's are Convolutional Neural Networks (CNNs), ResNet50 [13], VGG16 [42], DenseNet [15], Inception models [46] [44], and Xception [6]. These models, deployed in combination with other DL methods such as transfer learning [49], and together with losses such as binary cross entropy (BCE), cross-entropy loss (CEL), weighted cross-entropy loss (WCEL), and focal loss achieve excellent performance in various computer vision tasks.

An example of research using the models mentioned above is [3]. There, the authors compare 8 state-of-the-art models that have become benchmarks in the field of DL with CNNs. This research, motivated by the recent Covid-19 pandemic, strives to aid those in the medical field. In [3], it was found that a fine-tuned version of ResNet50 [13], MobileNetV2 [39] and Inception ResNetV2 [44] show a training and validation accuracy of more than 96% accuracy. Similarly, Madaan *et al* [28] presented a model called XCovNet to aid in the early detection of Covid-19 in patients using 2 separate datasets of CXR images. They successfully detected when a patient was sick with Covid-19, other ailments, or was not sick at all with a success rate of 98.44%.

CoroDet [16] was designed for detecting Covid-19 from CXR and CT scan images. CoroDet has developed accurate diagnostics under four evaluation scenarios. These include binary (Covid-19 and Healthy), ternary (Covid-19, healthy, and non-Covid-19 viral pneumonia), and 4 class classification (Covid-19, healthy, non-Covid-19 viral pneumonia, and non-Covid-19 bacterial pneumonia). It achieves a classification accuracy of over 90% on all scenarios. Similarly, Ibrahim *et al* [18] deploy a pretrained model as their base model. Furthermore, both papers use the three classification scenarios mentioned above. The difference that Ibrahim *et al* [18] have in their paper is that they compare all of the health statuses that patients can have in an exhaustive N-by-N manner. For example, in the binary scenario, they make four comparisons. These are COVID-19 and healthy, bacterial pneumonia and healthy, non-COVID-19 viral pneumonia and healthy, and COVID-19, and bacterial pneumonia. The accuracy found in this paper for every diagnosis across all scenarios exceeded 91%.

A popular area of research, in the context of the Covid-19 pandemic, was comparing architectures to discover the best-performing network. Rawat *et al* [38], and KC *et al* [22] compare 4 [46, 14, 6, 15] and 8 [42, 46, 13, 14, 39, 15, 58] different CNN models, respectively. Out of which, the best performing networks are MobileNet [14] with 99.1%, and DenseNet121 with 98.69% accuracy for Rawat *et al* [38] and KC *et al* [22], respectively. This difference in results can be attributed to dataset differences between the authors. A similar direction of research for bacterial endemic Tuberculosis was done by Rahman *et al* [36]. Nine different deep CNNs [13, 37, 46, 42, 15, 17, 14] were used for transfer learning for classifying Tuberculosis and non-Tuberculosis cases. Three different experiments were performed: segmentation of CXR images using two different U-net models, classification using CXR images, and segmented lung images. The results show that DenseNet201 achieved the best performance (accuracy of 98.6%) with predominantly segmented lung images.

A direction pursued by Toğaçar *et al* [47] involved additional work in the preprocessing step via the Fuzzy Color and stacking techniques. Three different datasets of three different classifications they used were then fed into SqueezeNet [17] and MobileNetV2 [39] to learn features. Afterward, Support Vector Machines (SVMs) were trained for classification. They attempted to separate diseased and healthy lungs based on their appearance. The authors achieved excellent classification results with their Covid-19 prediction reaching 100% accuracy.

Stephen *et al* [43] proposed a novel CNN architecture instead of using transfer learning. Due to the difficulty of collecting a sufficiently large pneumonia dataset, the authors implemented several data augmentations methods. This was done to improve the accuracy of their proposed model. With their proposed system, they achieve a 93.73% validation accuracy. The main difference between this work and that discussed above is that it focused on pediatric images.

A limitation of the previously discussed work is that it focuses on a singular disease, whereas a patient can exhibit multiple ailments simultaneously. To undertake the complexity presented by multiple ailments, researchers have developed new more complex datasets and models.

2.2 Multi-Label Chest X-ray Classification

In this section, various solutions to the multi-label CXR classification problem are studied. Patients may present multiple ailments at the same time. If their condition has worsened, it can affect other areas of their health. Alternatively, unrelated accidents or misfortune could present cases with several diseases. Unfortunately, using multi-labeled images adds additional complications to classifying images. CNN-based systems are data-hungry, and the absence of large enough annotated datasets can impede their deployment. Two recent works have identified this deficiency and attempted to re-mediate it. ChestX-ray8 [51] and CheXpert [19] are two datasets commonly used as benchmarks for multi-label classification of CXR. They are introduced in the section below.

Additionally, state-of-the-art proposals for multi-label classification will also be discussed in this section.

2.2.1 Dataset: ChestX-ray8 and ChestX-ray14

In [51], a new dataset, ChestX-ray8 updated to ChestX-ray14, is presented. These two versions of the dataset contain either 8 or 14 different thoracic ailment classifications. The updated version of the dataset consists of 112,220 frontal view X-rays. The authors employed various Natural Language Processing (NLP) techniques to determine the class labels as well as handle negation and uncertainty in the radiology reports.

To release this dataset, the authors proved that it is learnable and classifiable with deep learning methods. To this end, the authors implemented a Deep Convolutional Neural Network (DCNN) architecture utilizing three different pooling methods. The DCNN architecture used in this paper is adapted by using pre-trained models(AlexNet [23], ResNet50 [13], GoogleNet [45], and VGGNet-16 [42]) and removing their fully-connected layers or final classification layers. In their place, the authors added transition, global pooling, prediction, and loss layers. The dataset in question was split into training(70%), validation(10%), and test sets(20%). The DCNN classifies the disease in the images using the pretrained models. Three different types of pooling are compared to each other to see how they perform.

2.2.2 Dataset: CheXpert

For this dataset, CheXpert [19], the authors compiled 224,316 chest radiographs of 65,240 patients. A labeler was designated to automatically detect the presence of 14 set observations in radiology reports while capturing any written uncertainties. They investigated different approaches exploiting the uncertainty labels to train CNNs. The CNNs output the probability of these observations based on input radiographs. The validation set of 200 chest radiographs was manually annotated by 3 board-certified radiologists. Each radiologist annotated different uncertainty types for different pathologies. Their best model was evaluated on the test set of 500 CXR images, annotated by a consensus of 5 board-certified radiologists. The validation and test sets were cross-validated to examine the accuracy of the classification methods used in this study. The model used in classification for this work after testing various networks was, DenseNet121 [15]. Wang *et al* [51], and these authors use different methods to calculate the validity of their datasets.

2.2.3 State of the Art

In this section, we summarize recent work focusing on the multi-label classification of thoracic disease. We have identified three different approaches to solving this problem. These are generalized CNN, metric learning, and attention-based approaches.

CNN-Based Approaches When looking at CNN-based multi-label classification, we can see a popular research direction is for authors to make custom CNN architectures. As seen in Rajpukar *et al* [37], the authors have developed a 121-layer CNN network trained on ChestX-ray14 [51]. The authors compare their results to the diagnoses given by radiologists. At the time of writing, the official test split of ChestX-ray14 [51] has not been released. Similar to other fields of image classification, medical image classification has also turned to transfer learning to better tune their models. In Raghu *et al* [35], we see a comprehensive study done about transfer learning for medical images. They found that the different outcomes from transfer learning were because of the over-parametrization of standard models. Following Rajpukar *et al* [37], Baltruschat *et al* [5] used ChestX-ray14 [51] as their dataset in the research they conducted. This paper by Baltruschat *et al* [5] compares ResNet50 [13] with and without transfer learning, along with an extended ResNet50 architecture, and a network that integrates non-image data during the classification. They also investigate other ResNet models such as ResNet38 [54] and ResNet101 [13].

A slightly different approach is taken by Yan *et al* [57], where they propose a model that has squeeze-and-excitation blocks, along with multi-map transfer, and max-min pooling for classification. They base their network on DenseNet [15], while using ChestX-ray14 [51] as their dataset. Likewise, Pham *et al* [33] present their own CNN network based on DenseNet121 [15], using transfer learning and CheXpert [19] as their dataset. A somewhat different model is proposed by Agu *et al* [1] in AnaXNet, which has detection and anatomical dependency modules. For this model, they propose to focus on the anatomical information observable from the CXR's. Their anatomical dependency module utilizes graph convolutional networks, enabling learning on label dependency and the correlation between the anatomical regions in CXR's.

Another approach was taken by Mo *et al* [30], and Tran *et al* [48] is to focus on the losses instead of the models in an attempt to either optimize or propose their losses to solve the multi-label classification problem. Mo *et al* [30] present an entropy weighting loss to observe inter-label dependencies and employ classes with fewer examples in comparison to others. They test the proposed loss with DenseNet121 [15] as their base model on ChestX-ray14 [51] as their dataset. Tran *et al* [48] propose a distribution-balanced loss, which is a reshaped version of Binary-Cross Entropy loss (BCE) to learn more difficult diagnoses by down-weighting the loss assigned to the majority classes. To make sure the validity of their proposed loss they compare it to several losses such as BCE and Focal loss while using DenseNet-121 [15], Dense-169 [15], and ResNet-101 [13] as their classification models.

In combination with creating their model and loss, Xu *et al* [56], propose Cxnet-M3, a deep quintuplet network. They design a novel loss by combining deep metric learning and deep learning based on multi-labels. Transfer learning is employed to optimize the loss function. The dataset used is in this work is ChestX-ray14 [51].

Metric Learning Approaches Metric learning approaches have recently been employed for medical image analysis following the success of such approaches in other computer vision tasks. FLANNEL [34], was proposed by Qiao *et al* to manage the class imbalance inherent to medical image datasets. Since some diseases are inherently rarer than others, datasets tend to be unbalanced. For example, pneumonia is a more common disease than chronic eosinophilic pneumonia (chronic chest infection caused by white blood cells filling up the lungs). To address this imbalance, the authors of [34], proposes to use the focal loss, adapted to the multi-class classification task, as a means of balancing the dataset. 5 state-of-the-art CNNs are deployed as base models in an ensemble structure. FLANNEL was found to vastly outperform all other state-of-the-art CNN architectures(InceptionV3 [46], Vgg19 [42], ResNeXt101 [55], ResNet152 [13], Densenet161 [15], and other additional baselines) that the authors considered. In particular, for the COVID-19 detection task, without much-added model complexity or parameters.

Annarumma *et al* [2], propose a notion of 'radiological similarity', where two CXRs are similar if they present the same abnormalities. They present two loss functions to deal with the multi-label classification problem, which are combined into a single function. They use a custom 745,000 CXR dataset where the labels are extracted from free-text radiological reports through a natural language processing system.

Attention-Based Approaches The task of diagnosing thoracic disease from a CXR is usually done with the full image as input. However, this presents several challenges. For example, CXR abnormalities are generally in small regions. Furthermore, some CXR images will have poor alignment because of their irregular borders. Attention is deployed to ameliorate these issues within the framework of CNN models. An example is the work by Wang *et al* [50], where a novel model, ChestNet, consisting of two branches is proposed. These two branches being a classification branch and an attention branch. The classification branch serves as a uniform feature extraction classification network, and the attention branch exploits the relationship between the labels and the locations of pathological abnormalities. This allows the model to focus on the pathologically abnormal regions, providing better feedback for classification. The authors use the ChestX-ray14 [51] for this work. Guan *et al* [10], propose three-branch attention-guided CNN to be able to handle the issues mentioned above. These three branches first learn global features. Then, they use an attention heat map to create a mask to crop the region needed from the global image. Afterward, this region is used for training, and lastly, they concatenate the last pooling layers of the global and local branches. They use these methods in conjunction with ResNet50 [13] and DenseNet121 [15] models. In an alternative work [9], the authors propose a category-wise residual attention learning framework. Within this framework, the authors aim to suppress the contributions of irrelevant classes by down-weighting their representations. Simultaneously, the more relevant classes will have higher weights. This framework consists of

2 separate parts, the feature embedding, and the attention module. The first is trained with high-level features, while the latter focuses on finding the different values of different categories. The authors use ChestX-ray14 [51] for both of their works.

In TieNet [52], Wang *et al* [52] have proposed a model to produce attention-based image and text representations. This multi-level attention model is end-to-end trainable and highlights meaningful image areas and words in the radiograph texts. They use three separate datasets in their endeavor, these being the ChestX-ray14 [51], OpenI [7], and their hand-labeled dataset. Liu *et al* [26], propose a Contrastive Attention model, to be able to find and explain abnormal regions. The contrastive information acquired represents the features of the abnormal regions found in the CXRs more successfully. The datasets used in this work are, IU-X-ray [7] and MIMIC-CXR [21]. An attention-driven weakly supervised model with a hierarchical attention mining framework is proposed by Ouyang *et al* [32] therein, they merge activation and gradient-based attention. The authors have implemented explicit ordinal attention constraints that allow for training in a weakly supervised manner. Therefore, allowing the creation of visual attention-driven model explanations using localization. This proposed framework has been implemented on the datasets, ChestX-ray14 [51] and CheXpert [19].

2.2.4 Vision Transformers

This section presents the latest state-of-the-art model for image classification, the Visual Transformer (ViT). This model was first introduced by Dosovitskiy *et al* [8]. ViT works by splitting images into patches of a specific size, linearly embedding each of these patches then adding position embeddings. Afterward, they pass the resulting vector sequence into a standard Transformer encoder. An additional learnable classification token is appended to the sequence to perform classification. When compared to publicly available models, ViT surpasses the best ResNet model by a thin margin when pretrained on a sufficiently large dataset. However, the benefit of the ViT over other models grows with the dataset.

Lanchantin *et al* [24], propose the model Classification Transformer (C-Tran) for multi-label image classification that takes advantage of Transformers to exploit the relationship between visual features and labels, which can grow quite complex. This model predicts a set of target labels given an input of masked labels from the combinations of visual characteristics from a convolutional neural network with a Transformer encoder. A label mask training objective, which uses a ternary encoding technique to express the state of the labels as positive, negative, or unknown during training, is a critical component of their strategy.

In the above sections, we have discussed several ways to approach the problem of multi-label image classification. From the research done, we have seen that overall, ResNet50 [13] or other classification models have the best performance towards this problem. Therefore, we will be taking this model as our base model and expanding it with attention modules as they seem like a

promising research direction. Furthermore, we will explore the possibilities that state-of-the-art transformer models present for multi-label CXR classification.

Chapter 3

Methods

In this chapter, we discuss our approach for evaluating the suitability of transformer models for multi-label chest X-ray (ML-CXR) classification. The chapter is structured as follows. Section 3.1 gives an overview of the system architecture. The baseline convolutional backbone is presented in Section 3.2. Afterward, Section 3.3 extends the baseline model with two specialized attention blocks. Followed by Section 3.4, wherein we discuss the transformer models in our work. Sections 3.4.1 and 3.4.2 provide detail on the Visual Transformer [47] and Classification Transformer [24], respectively. Section 3.5 discusses and defines the various losses employed in this work.

3.1 System Architecture

An overview of the system architecture is presented in Figure 3.1. Three different processing modes are identified. Firstly, is the baseline model composed exclusively of the convolutional backbone. Second, the previous model is extended by the inclusion of an attention module. Third, we deploy more complex attention-based architectures known as transformers [49] as our model. Each of these approaches is trained with a variety of losses and their performance compared. In the section below, we describe the simplest model considered in our work.

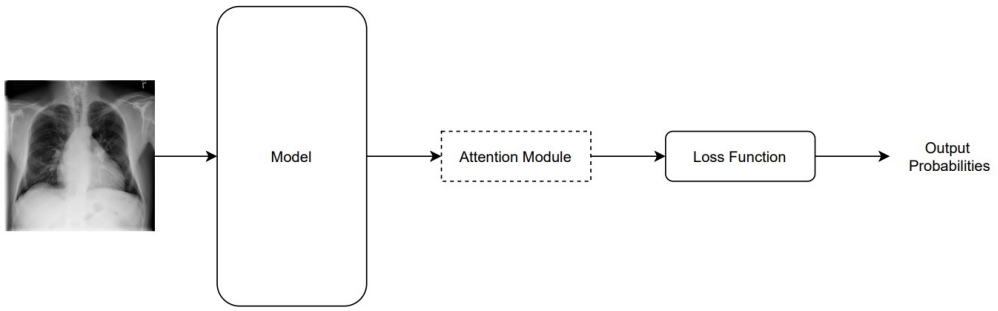

Figure 3.1: An outline of how the different networks we implemented work, whether these networks have attention added is dependent on the network chosen at the moment. The classification and metric losses implemented can be used in combination or separately for the loss function.

3.2 Convolutional Backbone

ResNet has quickly become one of the default convolutional models employed as a baseline. ResNet, first proposed by Kaiming *et al* [13], is trained for image classification on the ImageNet dataset. The framework proposed by the authors was a deep residual training framework. A residual block allows the layers in their network to fit a residual mapping. Therefore, enabling the training of deeper networks while reducing gradient problems. This was done because the authors noticed that over-fitting was not the cause of degradation of the training accuracy when deep networks started to converge and that the number of layers added to an already deep network correlated to a higher training error.

The network in question consists of stacked convolutional layers. The number of these layers can be increased or decreased as a design choice. The convolutional output is fed into a global average pooling layer and a 1000-way fully-connected layer with softmax. The model in question is depicted in Figure 3.2.

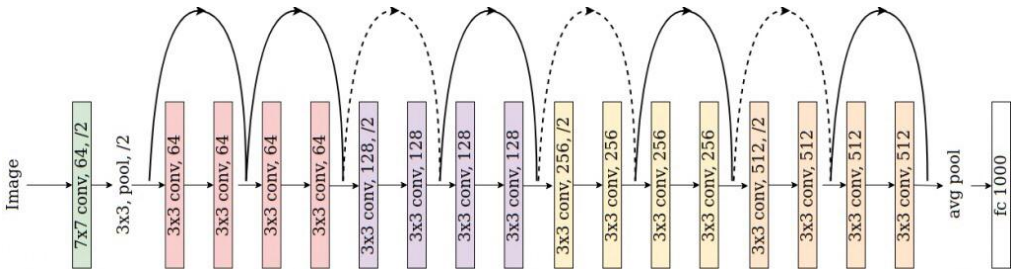

Figure 3.2: ResNet34 Architecture, as seen in Kaiming *et al* [13]

The ResNet architecture has been defined with 18, 34, 50, 101, and 152 layers. The main reason that these networks work so well is that the many layers learn residual functions. The inclusion of identity mappings allows the network to learn, from layer to layer, with references to the previous layer. To create this residual mapping between each layer, they denote the desired mapping between the layers as $H_(x)$, when the stacked nonlinear layers are represented as another mapping, $F_(x) := H(x) - x$. From these formulas we can see that $H_(x)$ can be realized as $F_(x)+x$. To get the desired mapping between the layers, $H_(x)$, 'shortcut connections' are introduced. These 'shortcut connections', are connections that skip one or more layers in the network. Having the 'shortcut connections', allows for the authors to realize their new formulation of $H_(x)$. Hence, performing identity mapping and adding the outputs of these connections to the outputs of the stacked layers. This desired mapping in the network can be seen in the residual block in Figure 3.3. The authors present extensive empirical evidence demonstrating that residual networks are easier to tune and benefit from much higher depth.

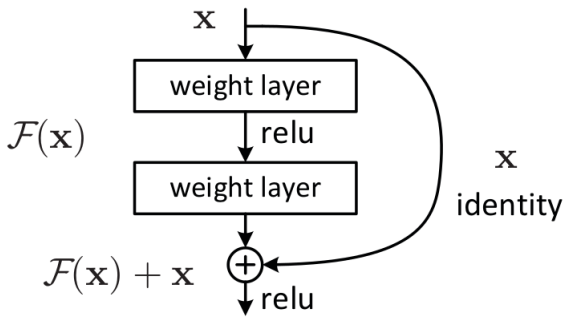

Figure 3.3: The residual block, as seen in Kaiming *et al* [13]

In this work, we use the model ResNet50 [13], as our convolutional backbone. As stated above ResNet50 [13] has become one of the benchmark architectures in the field of computer vision. Therefore, we follow this trend in literature [10] and also employ it as our benchmark model. Furthermore, ResNet50 [13] was the best architecture in classification for the dataset ChestX-ray14 [51], and that in the paper that published the model, Vision Transformer [8], ResNet50 [13] was one of their comparison points to see the models accuracy.

3.3 Attention

Attention was first implemented in the field of natural language processing as an improvement in neural machine translation [4]. Afterward, it was adapted

into other fields of research in neural networks, such as image processing [29], speech processing [27], among others. The first attention model was proposed by Bahdanau *et al* [4]. In our work, we use two implementations of attention. These are self-attention, and the Convolutional Block Attention Module (CBAM). These two concepts will be discussed in the subsections 3.3.1, and 3.3.2 below.

3.3.1 Self-Attention

Self-attention as a concept has become more popular in the field of deep learning after the publication of the paper by Vaswani *et al* [49]. Also called intra-attention, self-attention is an attention technique that connects different points in a single sequence to compute a representation of it. An input tensor, x, is fed through three different convolution blocks that produce the key, query, and value tensors, respectively. These three outputs are vectorized and used to calculate the attention mask. This mask is modeled such that a query and a pair of key-value pairs are mapped to an output. The weighted sum is computed for the output of these values, and a compatibility function of the query for each correlating key is assigned to each value as a weight. The authors propose two different types of self-attention, a Scaled Dot-Product, and a Multi-Head Attention. In this work we limit our studies Scaled Dot-Product self-attention due to constrains on computational resources.

3.3.2 CBAM

Woo *et al* [53] proposed a Convolutional Block Attention Module (CBAM), which is a simple attention module for feed-forward CNNs. The proposed model in question operates along the channel and spatial dimensions of an input feature map. This process is performed in sequential order. This module can be integrated into any CNN architecture with negligible overhead and is trained end-to-end alongside basic CNNs. The authors validate their proposed model with experiments done on several detection datasets. The authors state that their contributions are as follows, the proposed model can be used to boost the performance of CNNs, extensive studies done on the effectiveness of their model, and they verified the performance of their model on various benchmarks.

When looking at the last convolutional layer of a CNN, a feature tensor x can be extracted. This tensor can be considered as a set of spatial feature maps. Alternatively, it can be interpreted as a matrix of d-dimensional feature vectors in its tensor form. A series of convolutional operations re-weigh their contributions. Two separate vectors are computed. The channel features are max and average pooled, respectively. By feeding these new vectors through a multi-layer perceptron, new representations can be learned. These new representations are summed and fed through a sigmoid function. After this step, the output vector is multiplied by the feature tensor x. A similar operation is performed along the spatial dimension. Lastly, spatial attention re-evaluates the elements in the original set of spatial feature maps. CBAM is used to improve the performance of learned descriptors in general. Each attention module re-weights the channels

or spatial information on its own. Key point identification can be done using spatial attention maps. The CBAM architecture can be seen in Figure 3.4 below. In the Figures 3.5 and 3.6, we depict the inner working of the Channel Attention Module and Spatial Attention Module used in CBAM.

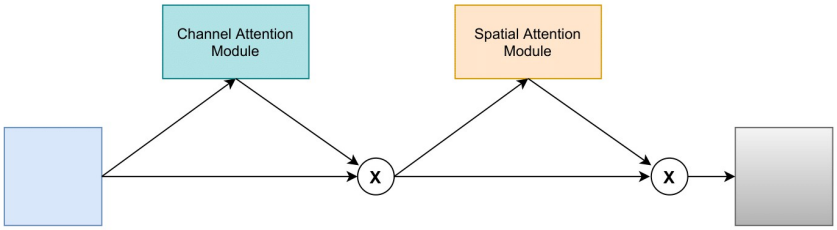

Figure 3.4: CBAM Architecture, as inspired by the model in Woo *et al* [53].The blue block is the input and the gray block on the right hand side is the weighted output.

Figure 3.5: The overview of the Channel Attention Module, where the input tensor is pooled using the max (dark purple, top) and average (light purple, bottom) operations. They are then fed through a convolutional layer, the channel-weighted map and a sigmoid operation. The results of each module are multiplied by the original input tensor.

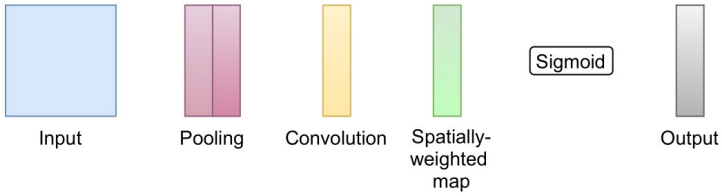

Figure 3.6: The overview of the Spatial Attention Module, where the input tensor is pooled using the max (darker purple, right) and average (light purple, left) operations. They are then fed through a convolutional layer, the spatial-weighted map and a sigmoid operation. The results of each module are multiplied by the original input tensor.

3.4 Transformer Models

Transformers are a state-of-the-art approach to deep learning. This trend started with the paper, Attention Is All You Need from Vaswani *et al* [49]. The proposed model in the paper was an attention-based encoder-decoder model they called Transformer. This model was applied in the field of Natural Language Processing. The encoder maps an input sequence into an abstract continuous representation that holds all of the learned information of that input. The first step of the encoder is feeding the inputs into an embedding layer. Therein, there is no concept of a time step. As such, all of the inputs are passed together, with embeddings determined simultaneously. As computers don't understand words the input will be in the form of numeric integers, vectors, or matrices. The idea is to embed every word to a point in the space where words of similar meaning are physically closer to each other. The space they are present in is called the embedding space. In this embedding space, the words are mapped to a vector. Afterward, positional information is injected into the vectors with the positional encoder. The positional encoder is a vector that gives context based on the position of the words in a sentence. Vectors with positional information are passed through a multi-headed attention layer and then a fully connected feed-forward network. In the multi-headed attention module, self-attention allows each word in the input to be mapped to other words in the input. To this is achieved by following the procedure described in Section 3.3.1. The result is a score matrix, with the weight allocated to each value determined by the query's compatibility function with the relevant key. This matrix determines the level of attention given to words correlating to other words. The higher the score matrix, the more attention is given. The Figures 3.7 and 3.8 show the transformer model and the multi-headed attention respectively.

The decoder then takes this information, along with the previous output generated, and step by step generates a single output from the combination of these two as inputs. The decoder has two multi-headed attention layers as well as a point-wise feed-forward layer with residual connections after each sub-layer.

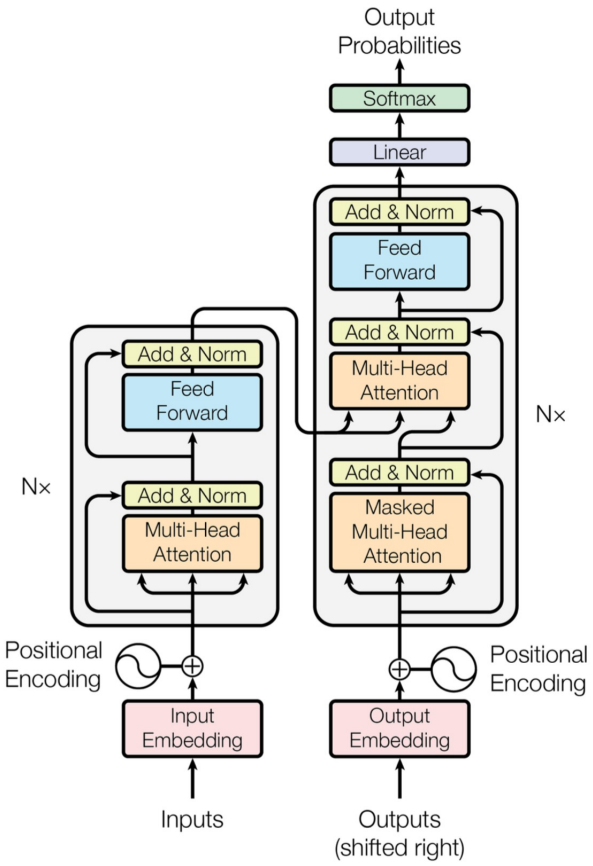

Figure 3.7: Transformer model from Vaswani *et al* [49]

The first multi-headed attention layer is masked to facilitate learning using only the previous output instead of all the output words. Without the mask, there would be no learning. The sub-layers act similarly as they do in the encoder. However, each multi-headed attention layer has a different job. These layers are followed by a linear layer which acts as a classifier together with a softmax layer to produce probabilities per word. The decoder stops generating information once it gets an end token as an output.

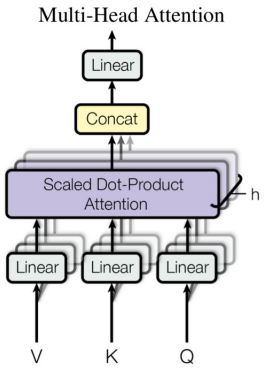

Figure 3.8: Multi-Headed attention model from Vaswani *et al* [49]

3.4.1 Vision Transformer

The concept of transformers has been extended into accepting images as inputs in the work of Dosovitskiy'*et al* [8]. The authors have adapted the transformer to take in sequences of image 'patches' as inputs instead of sequences that work for inputs in the original form of transformers. These 'patches' are fixed-size split sections of images. To find 'patches' for 2D images x where $x \in R^{H \times W \times C}$,is reshaped into a sequence of 2D patches $x_p \in R^{N \times (P^2 C)}$. Where (H, W) is the resolution of the original image x, C is the number of channels, (P, P) is the resolution of each image patch, and $N = HW/P^2$ is the resulting number of patches. This is also used as the input length variable for ViT. For ViT, the proposed method of patching the images works in the same way that language datasets do for the original version of transformers. Following the concept of the original transformer models, this adapted model still uses positional embeddings to provide positional data. The authors note that they use standard 1D position embeddings as they did not see a significant improvement in performance in the model over the 2D positional embeddings. The architecture is depicted in Figure 3.9.

The authors also propose another model in this work, a hybrid ViT approach. This model offers another approach, without the use of patches from images, using feature maps from a CNN as its input sequence. These patches, extracted from said feature maps, are obtained by flattening the dimensions of the output feature maps and adapting them to the dimensions needed for the vision transformer. The rest of the model architecture remains the same as the proposed ViT model.

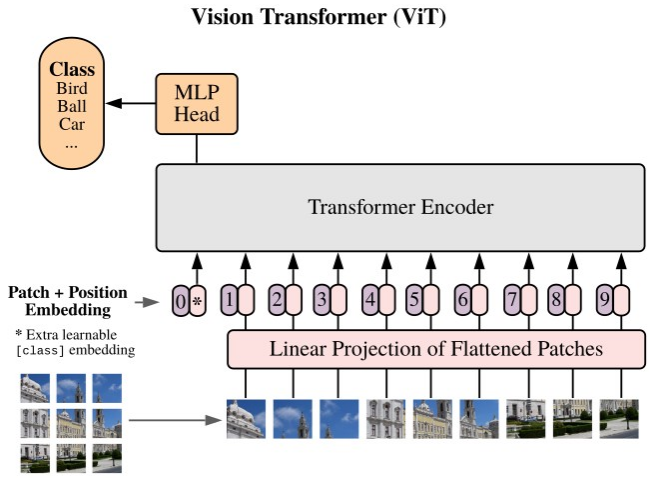

Figure 3.9: The architecture of Vision Transformer as seen in Dosovitskiy *et al* [8]

3.4.2 C-Tran

Lanchantin *et al* [24] proposed a model in their work that they called the Classification Transformer (C-Tran), which is a model for multi-label image classification that uses transformers to exploit the correlations among the labels and visual features. A noteworthy difference in their implementation of the transformers is that the author's proposed method of label masking uses ternary encoding to represent labels as either positive, negative, or unknown during the training phase. The proposed model explicitly represents the labels during training. Therefore, allowing for more generalization among images that have either extra annotations or partially annotated labels. Another fundamentally different way in which C-Tran works is by partially masking labels during the training phase and fully masking the labeled data during the inference phase. In short, the claims C-Tran makes in improvement are in the fields of flexibility, accuracy, and interactivity. Figure 3.10 and 3.11 show the architecture of the C-Tran model for the training and test phases respectively.

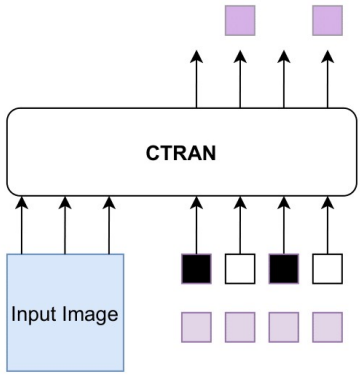

Figure 3.10: The architecture of C-Tran for the training phase as seen in Lanchantin *et al* [24]. During the training phase it can be seen that the model randomly masks labels, the top row of the blocks are masks, where the blacked out boxes are masked. Then this C-Tran model predicts the labels for the randomly masked labels.

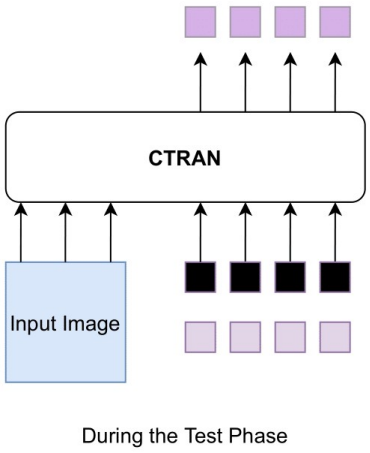

Figure 3.11: The architecture of C-Tran for the test phase as seen in Lanchantin *et al* [24]. During the test phase all of the labels are masked, as to not interfere with testing. Then this C-Tran model predicts the labels for every instance.

3.5 Losses

In the following sections 3.5.1 and 3.5.2, the loss functions employed in our work are discussed. We perform a comparative study across classification and

metric losses both separately and jointly. This is done to identify if there is an advantage to deploying these losses for chest disease classification.

Features extracted from models need to be fine-tuned in the data domain to obtain the most relevant representations, which is achieved by using loss functions to train for specific tasks, such as classification or metric learning. These are defined below.

3.5.1 Classification Loss

In this section, we discuss the classification losses used throughout our research. These losses are either proposed in work relevant to thoracic disease classification or frequently used in other fields.

Binary Cross Entropy Loss Binary Cross Entropy (BCE) loss is one of the most common loss functions used in multi-label classification, and therefore was the first loss function considered for evaluation. The BCE loss is defined as follows,

$$L_{BCE} = -1/N * \sum_{i=1}^{N} t_i \cdot \log(\hat{t}_i) + (1 - t_i) \cdot \log(1 - \hat{t}_i), \tag{3.1}$$

where N is the number of output size scalar values, while t_i is the target value, and $\hat{t}_i$is the i-th scalar value of the output of the model.

Cross Entropy Loss We consider Cross-Entropy Loss (CEL) next, as it has become standard to use in multi-label classification. The equation is as follows,

$$L_{CEL} = -\sum_{i=1}^{NC} y_i \cdot \log(p_i), \tag{3.2}$$

where NC is the number of classes, y_i indicates if the image belongs to class i, and p_i is the predicted probability that the image is a part of class i. In this loss function, it can be seen that the value of the loss function itself may be affected by the ease with which images are classified. Therefore, indicating that images that are easily classified may significantly affect the gradient. In turn, this will control the probability of learning from images that are perceived to be harder than average.

Weighted Cross Entropy Loss The Weighted Cross-Entropy Loss (WCEL) function is as its name suggests a weighted version of CEL. We follow the original definition used in [51],

$$L_{WCEL}(f(\overrightarrow{x}), \overrightarrow{y}) = \beta_N \sum_{y_i=1} -\log(f(x_i)) + \beta_N \sum_{y_i=0} -\log(1 - f(x_i)), \tag{3.3}$$

where β_P, β_N are balancing factors used to impose the learning of the positive samples. These balancing factors are of a positive and negative class, respectively. β_P is equal to $(\mid P \mid + \mid N \mid)/ \mid P \mid$, and β_N is equal to $(\mid P \mid + \mid N \mid)/ \mid N \mid$, where $\mid P \mid$, and $\mid N \mid$ are equal to the total number of '1's and '0's in a set of image labels.

Focal Loss We explored more loss functions to be able to give a broad view of how they acted in our research, especially their interaction with the metric losses. To this end, we came upon Focal Loss (FL) [25] as a viable loss for the ML-CXR problem. FL was first proposed to address the class imbalance of datasets. FL expands upon CEL, by adding a factor of modulation. Therefore, enabling this loss to shift the focus from weights of easier found images towards harder ones, these weights in question being the negatively affecting ones. The equation of this loss is as follows,

$$L_{FL} = -\sum(1 - p_i)^{\gamma} \log(p_i), \tag{3.4}$$

where γ is the parameter that controls the weight which enhances the relevance of more informative samples. When γ is set to be above 0, it will start reducing the importance of correctly classified images.

3.5.2 Metric Loss

Metric losses attempt to optimize the distance between embeddings by reducing it if they belong to the same class and increasing it if not. Therefore these types of losses teach the network to model similarity. Meanwhile, classification losses provide explicit semantic information regarding inputs and outputs. These types of losses are more standard in image retrieval as they yield highly separable representations.

Contrastive Loss Proposed by Hadsell *et al* [11], this loss function was originally used for dimensionality reduction. They required a mapping function that would preserve the neighborhoods of the relationships between data points, similarly as clustering does for data points. They also needed this mapping function to be able to generalize new information. To this end, they proposed the following generalized formula,

$$L_{gen\_cont}(W, (Y, \overrightarrow{X_1}, \overrightarrow{X_2})^i) = (1 - Y)L_S(D_w^i) + YL_D(D_w^i). \tag{3.5}$$

Where D_w is the parameterized distance function between the values of X_1 and X_2, for the outputs. The values X_1 and X_2 are the representation of a pair of input vectors. For the method of dimensionality reduction method proposed by Hadsell *et al* [11], along with their proposed contrastive loss works over a pair of samples instead of the sum of the samples. Y is the binary label that is assigned to the pair of input vectors X_1 and X_2. For $Y = 0$, it means that the pair of input vectors X_1 and X_2 are similar, when $Y = 1$ the pair of input

vectors are dissimilar. Therefore, we can state that $(Y, \overrightarrow{X_1}, \overrightarrow{X_2})^i$ is the sample pair of the i-th order. For the partial loss function in this loss, L_s is used when there is similarity, and L_D is used for dissimilarity.

The authors also proposed an exact loss function as given in the equation below,

$$L_{exact\_cont}(W, Y, \overrightarrow{X_1}, \overrightarrow{X_2}) = (1 - Y)\frac{1}{2}(D_w)^2 + Y\frac{1}{2}\{\max(0, m - D_w)\}^2. \quad (3.6)$$

The constraint in the formula given above is that $m > 0$ is a margin. This margin determines the radius around the outputs. Through the author's research, it was seen that dissimilar pairs only aid the loss when the distance between them is in this radius of m. It was also seen that the existence of dissimilar pairs is pivotal to this loss, as without them when there is minimization a collapsed solution will occur. In our work, we employ the exact formulation of the contrastive loss.

Triplet Loss Triplet loss is the brainchild of Schroff *et al* [41]. This loss function was designed for the field of facial recognition. This loss function tries to classify similarity in value of the distance measure. Therefore, face images are closer together and dissimilar images are further apart. This loss works such that an image is defined as an anchor. The anchor has an associated positive (negative) image whose distance will be reduced (increased). The formula for this loss is as follows,

$$L_{triplet} = max(0, m + D(x_a, x_p) - D(x_a, x_n)), \quad (3.7)$$

where x_a denotes the anchor, x_p the cases where there is a positive similarity to the anchor, and x_n denotes where there is a negative similarity. Triplet loss focuses on optimizing the differences between these losses for their best case for each probability, unlike contrastive loss.

Proxy-NCA Loss Another distance metric loss proposed by Movshovitz-Attias *et al* [31] is Proxy-NCA loss. This loss endeavors to optimize triplet loss, because of the slow convergence that these loss functions have. They take the points used in triplet from proxy points instead of images. The proxies in question used instead of points in this loss are approximates of the data points in the images. This loss works better empirically along with the fact that the proxies allow for a tight upper bound for the loss function. The authors claim state-of-the-art outcomes and convergence. The loss in question is formulated as follows,

$$L_{Proxy-NCA}(x_a, x_p, Z) = -\log(\exp(-d(x_a, x_p)) \div \sum_{z \epsilon Z} \exp(-d(x_a, x_n)), \quad (3.8)$$

where similarly to triplet loss x_a denotes the anchor, x_p denotes the cases where there is a positive similarity to the anchor, and x_n denotes where there is a negative similarity proxy. These elements all being in a set Z.

Chapter 4

Experiments and Results

In this chapter, we discuss our approach for evaluating the suitability of convolutional models, among which are transformers, for multi-label chest X-ray (ML-CXR) classification. The chapter is structured as follows. Section 4.1 enumerates the models employed in our experiments. The data and evaluation metrics are presented in Section 4.2. Section 4.3 describes the implementation details. Finally, a discussion of our results is located in Section 4.4.

4.1 Implemented Models

For our research, we have implemented six different models. These models are the ResNet50 [13] (as the baseline model), ResNet50 [13]+ Self_Attention [49], CBAM [53], C-Tran [24], ViT [8], and the Hybrid ViT [8]. These models have been selected due to both their prevalence in literature and their relationship to our research questions. While undertaking the task of implementing these models, we first had benchmark their performance for the two medical datasets we have worked on, ChestX-ray14 [51] and CheXpert [19].

4.2 Data and Evaluation Metrics

4.2.1 Data

In the scope of this thesis, we used the two datasets described in Sections 2.2.1, and 2.2.2. For the ChestX-ray8 (Section 2.2.1), we have worked with the updated version of this dataset, ChestX-ray14 [51]. For the CheXpert [19] dataset, we used the smaller version of the dataset available from the Stanford Machine Learning Group. This has been done because the complete dataset was harder to access and process due to its size. Furthermore, the original dataset would have proved too large considering our computational resources. This version of CheXpert [19] contains over 12 GB of images.

Regarding the annotations of ChestX-ray14 [51], they include a "No Finding" label when no ailment is visible. However, this label was not reported in the original paper until Table 16 [51]. We have included the "No Finding" label in our calculations. This is done to accurately compare both datasets, without implementing dataset-specific training and testing. Furthermore, we also added an "Uncertainty" class to the dataset CheXpert [19]. Considering that this dataset is labeled using uncertainty for a different disease. Each ailment in the dataset has the following values determining if the confidence of the annotation. Confidently present, confidently absent, and uncertainly present are annotated as 1, 0, or -1, correspondingly. Additionally, the certainty may be absent for CheXpert [19]. This allows us to calculate a measure for uncertainty for each epoch value tested for the dataset in question.

The labels for the ChestX-ray14 [51] dataset in order are as follows: Atelectasis, Cardiomegaly, Effusion, Infiltration, Mass, Nodule, Pneumonia, Pneumothorax, Consolidation, Edema, Emphysema, Fibrosis, PT (Pleural Thickening), Hernia, No Finding.

The labels for the CheXpert [19] dataset in order are as follows; No Finding, Enlarged Cardiomediastinum, Cardiomegaly, Lung Opacity, Lung Lesion, Edema, Consolidation, Pneumonia, Atelectasis, Pneumothorax, Pleural Effusion, Pleural Other, Fracture, Support Devices, Uncertainty.

In the Figure 4.1, the frequency of the types of disease is shown. From looking at this data we can see that for the ChestX-ray14 [51] dataset the most data is for 'No Finding', then 'Infiltration', 'Atelectasis', 'Effusion', and 'Nodule'. For the CheXpert [19] dataset, the top 5 classifications in the dataset are 'Support Devices', 'Lung Opacity', 'Pleural Effusion', 'Edema', and 'Atelectasis'. When looking at both the datasets it can be seen that they both have the classifications of 'Atelectasis' and '(Pleural) Effusion'. Since we are dealing with CXR's any effusion classification is for the pleural cavity, ChestX-ray14 [51] just abbreviated this condition.

4.2.2 Evaluation Metrics

To evaluate the classification accuracy of our training script, we use methods from the sci-kit-learn library for Python. From this library, we use the functions for calculating the Receiver Operating Characteristic curve, and report on the Area-Under-Curve (AUC). This metrics has been selected based on our literature review. The research in which these datasets were published measures their performance with this metric.

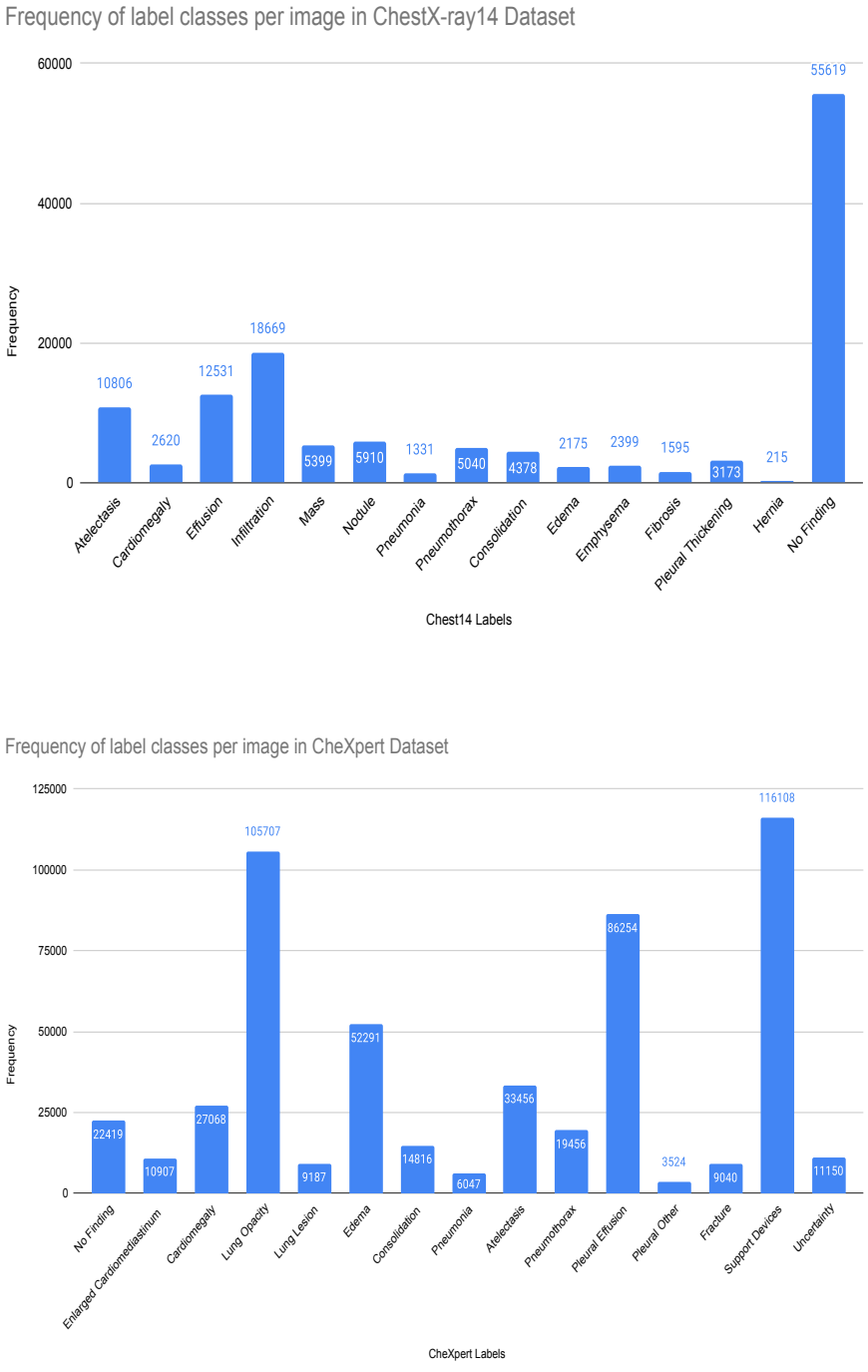

Figure 4.1: The plots of data frequency for the ChestX-ray14, and CheXpert datasets respectively.

4.3 Implementation details

In this section, we cover the implementation details for our work. Experiments have been performed using PyTorch running on consumer-level hardware. Specifically, a laptop with an Intel Core i7 processor, NVIDIA GeForce RTX 2070 graphics card, and 16GB RAM.

To ensure fair model comparison, we conducted baseline experiments to determine hyper-parameters such as batch size and learning rate. This is relevant because the different families of models studied in this work require significantly different resources. For example, the ResNet50 architecture is relatively light when compared to C-Tran or ViT. This fact, in conjunction with the limited hardware, required the normalization of certain hyper-parameters across all runs.

First, we considered the batch size for the six implemented models in our research. Early experiments showed that the transformer models, C-Tran, ViT, and Hybrid ViT, cannot handle a large batch size on the available hardware. C-Tran can handle up to 24 batches, whereas with the ViT model family we cannot handle more than 20 images per batch. For an acceptable run time per model, we have selected a batch size of 16 images.

Secondly, the selection of the optimal learning rate was done by performing a sweep across several possible values. These values are momentum, decay, a base learning rate, and step size. For every number of steps indicated in the configuration file, the learning rate is reduced by the rate of decay. For our implementation, we start out with a base learning rate of 0.0001 and reduce every 3 steps with a decay rate of 0.1. From thorough testing, this was found to be the best way to schedule the learning rate.

Finally, to determine the optimal amount of epochs, we run tests on our simplest network, ResNet50. This network was chosen because it took the least amount of time to run per epoch and also was the easiest to run consecutive tests on. We ran ResNet50 for 20 epochs for every individual loss considered in this work. From these results, we have observed that, on average, these losses do not yield significant improvements after the fifth epoch. Therefore, we limited our losses to 5 epochs for all models. This decision turned out to be advantageous as even for five epochs, the transformers took over a day to run completely. The best results for the individual losses for each dataset can be seen below in the Tables 4.1 and 4.2.

It should be noted for the training sets run for each different loss for each of the models the training times differed greatly. ResNet50, ResNet+Self_Attention and CBAM took from an hour to at most two and a half hours per epoch. The transformer models, C-Tran, ViT and ViT Hybrid took from a minimum of 4 hours to over 5 hours to run per epoch. This limitation was in part due to the models themselves, in other ways due to the amount of data, and the hardware restrictions we had working with a laptop.

After studying Tables 4.1 and 4.2, we can make the following remarks. For the dataset ChestX-ray, the best average classification is achieved by the loss WCEL after 2 epochs, with an average AUC of 76.12%. For the dataset CheX-

Loss Name	Best Average	Epoch
BCE	0.7495	2
CEL	0.7576	2
WCEL	0.7612	2
Focal	0.7454	3
Contrastive	0.5007	4
Triplet	0.5001	5
Proxy-NCA	0.4882	5

Table 4.1: Best average AUC of ResNet50 for each individual loss for the dataset ChestX-ray14.

Loss Name	Best Average	Epoch
BCE	0.7083	2
CEL	0.7766	4
WCEL	0.7735	4
Focal	0.7220	3
Contrastive	0.5522	2
Triplet	0.5912	5
Proxy-NCA	0.4862	5

Table 4.2: Best average AUC of ResNet50 for each individual loss for the dataset CheXpert.

pert, the best average classification result is achieved with the loss CEL after 4 epochs, having an average AUC score of 77.66%.

While looking at these losses in both Tables 4.1 and 4.2, it is evident that the metric losses by themselves do not perform well. A performance drop of approximately 20% exists between the average AUC of metric and classification losses across both datasets. Therefore, metric losses will not be studied individually in the following sections. We report the results on either classification losses alone or joint metric and classification loss.

4.4 Results

4.4.1 Preliminary Results

In this section, we discuss the results obtained after training ResNet50. Keep in mind, that all of our results use the same metrics asset in their configuration files. We evaluate the CNN on two different datasets, ChestX-ray14 and CheXpert.

For the ChestX-ray14 dataset, the results we have achieved are contained in Table 4.3. The first table summarizes the performance of ResNet50 for various classification losses. From it, we observe that the best classification loss is WCEL with an average score of 76.12% after 2 epochs. This aligns with

our expectations since this loss was designed to address the class imbalance in ChestX-ray14. Therefore, it makes sense that it would outperform the other loss functions. When considering the combination of classification and metric losses, summarized in Table 4.3, the best average performance is obtained with the CEL and Contrastive loss after 3 epochs. This result found for the combined losses is an average AUC of 76.67%. While metric losses do not negatively impact the performance, they also provide minimal gains. Additionally, we observe that metric losses have a negative interaction with BCE and Focal loss. Both contrastive and triplet loss present a significant drop in AUC not present in neither CEL nor WCEL. We conjecture that the mining procedure is at fault and that our selection of positives and negatives may not adequately represent the similarity between classes. We adjudicate the performance drop to the mining process because Proxy-NCA sidesteps this issue and retains good average performance. This is because the Proxy-NCA loss learns proxy embeddings which can be considered as cluster centers in feature space. In this manner, the positives and negatives are learned in relation to, and updated with, the proxies instead of our fixed label similarity metric.

When comparing both of the best losses from the two separate tables for this dataset, we can see that the combined loss achieves slightly better classification results with the difference of 0.55%. Additionally, the classification losses for both of the best losses from the separate tables are different. For classification loss on its own WCEL being better falls in line with our initial testing for this dataset. While in the joint case CEL performs better, this is a relatively minor difference. We speculate that the metric losses are not sufficiently descriptive of the intrinsic differences in multi-label images. Conceivably, the criterion used for determining positive and negative samples (thresholded Jaccard distance of the labels) does not provide a sufficiently good indicator of similarity.

For CheXpert [19], the results are summarized in Table 4.4. Therein, we observe that for purely classification losses, the best result is achieved by CEL after 3 epochs. The classification percentage achieved by this loss is 76.22%. The best result achieved by the combined losses is 80.82% AUC after 3 epochs, using CEL and Triplet loss. From these best losses, we can see that the combined loss performs better with a difference of 4.6%. The classification losses for both losses are also in line with initial testing, as CEL performs better for this dataset. The observation made for the dataset ChestX-ray14 holds for CheXpert as well. This observation is that BCE and Focal loss do not perform well in combination with metric losses, barring Proxy-NCA. Because this observation holds for the two different datasets, we can extrapolate that this trend will continue throughout our experiments.

The Class-specific AUC using the CEL losses for ResNet is depicted in Figure 4.2. This figure is shown both for brevity, and because the ResNet model along with the attention models implemented work better with CEL. We plot the performance on the CheXpert dataset since these models obtain better performance in comparison to ChestX-ray14. The acronyms used in the graphs are 'E.C.' for 'Enlarged Cardiomediastinum', 'P.E.' for 'Pleural Effusion', and 'S.D.' for 'Support Devices'.

Loss Name			Best Average	Epoch
BCE			0.7495	2
	+	***Contrastive***	0.5575	5
	+	***Triplet***	0.6257	5
	+	***Proxy-NCA***	0.7516	2
CEL			0.7576	2
	+	***Contrastive***	0.7667	3
	+	***Triplet***	0.7564	2
	+	***Proxy-NCA***	0.7647	2
WCEL			0.7612	2
	+	***Contrastive***	0.7604	2
	+	***Triplet***	0.7588	2
	+	***Proxy-NCA***	0.7636	2
Focal			0.7543	2
	+	***Contrastive***	0.5266	5
	+	***Triplet***	0.5377	2
	+	***Proxy-NCA***	0.7532	2

Table 4.3: Results for ResNet50 for the joint losses along with the epoch that yields the best AUC average for the ChestX-ray14 dataset.

Loss Name			Best Average	Epoch
BCE			0.7216	3
	+	***Contrastive***	0.4866	2
	+	***Triplet***	0.5790	5
	+	***Proxy-NCA***	0.7597	5
CEL			0.7622	3
	+	***Contrastive***	0.783	3
	+	***Triplet***	0.8082	3
	+	***Proxy-NCA***	0.7764	3
WCEL			0.7427	3
	+	***Contrastive***	0.7856	5
	+	***Triplet***	0.7878	3
	+	***Proxy-NCA***	0.7805	3
Focal			0.6973	2
	+	***Contrastive***	0.5335	4
	+	***Triplet***	0.5395	3
	+	***Proxy-NCA***	0.7103	3

Table 4.4: Results for ResNet50 for the joint losses along with the epoch that yields the best AUC average for the CheXpert dataset.

From the figure, we notice a large drop in performance for the 'Enlarged Cardiomediastinum' class along with the 'Pneumonia' class. It can also be seen that for the class 'Lung Lesion' the loss CEL by itself has a drop in the classi-

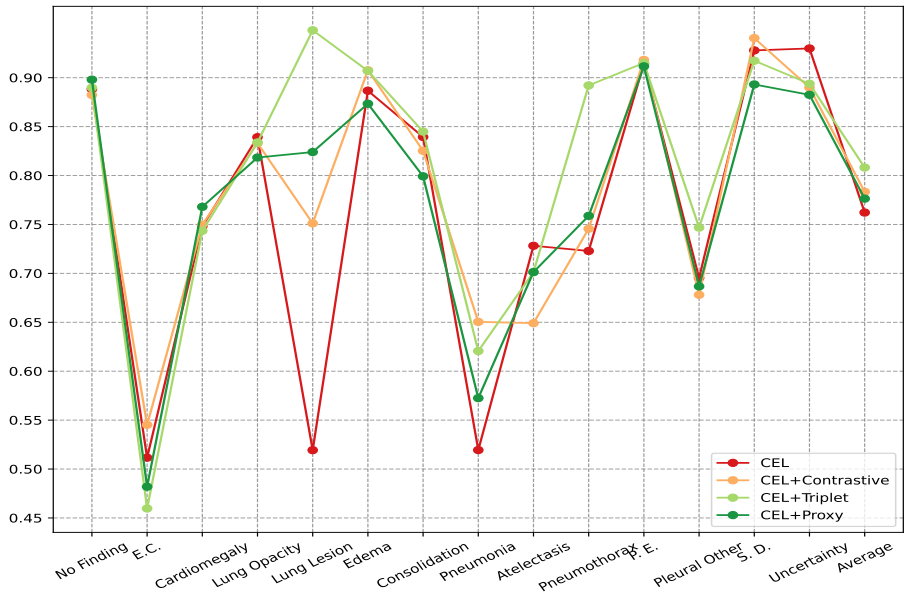

Figure 4.2: Plot of all CEL losses for the ResNet50 model for the CheXpert dataset.

fication accuracy. Adding the metric losses increases the classification accuracy by 23.18%, 42.9.%, and 30.47% for Contrastive, Triplet, and Proxy-NCA loss, respectively. In comparison, the other losses retain a classification performance above 70%. Particularly CEL with Triplet loss, which performs above 90%.

An interesting observation is that CEL has a more unstable behavior across the different labels when compared to the weighted variant. CEL exhibits much stronger AUC fluctuations across different classes. However, WCEL has a higher failure percentage for 'Lung Lesion', and all of the losses in WCEL for this classification drop downwards on their line. For the other losses (and combinations) except for BCE, the most challenging class is 'Enlarged Cardiomediastinum'. For BCE, the most challenging class is 'Lung Lesion'. Examples of how these classes that have a high failure percentage can be seen in the Figures 4.3 and 4.4 below from images taken from the CheXpert dataset.

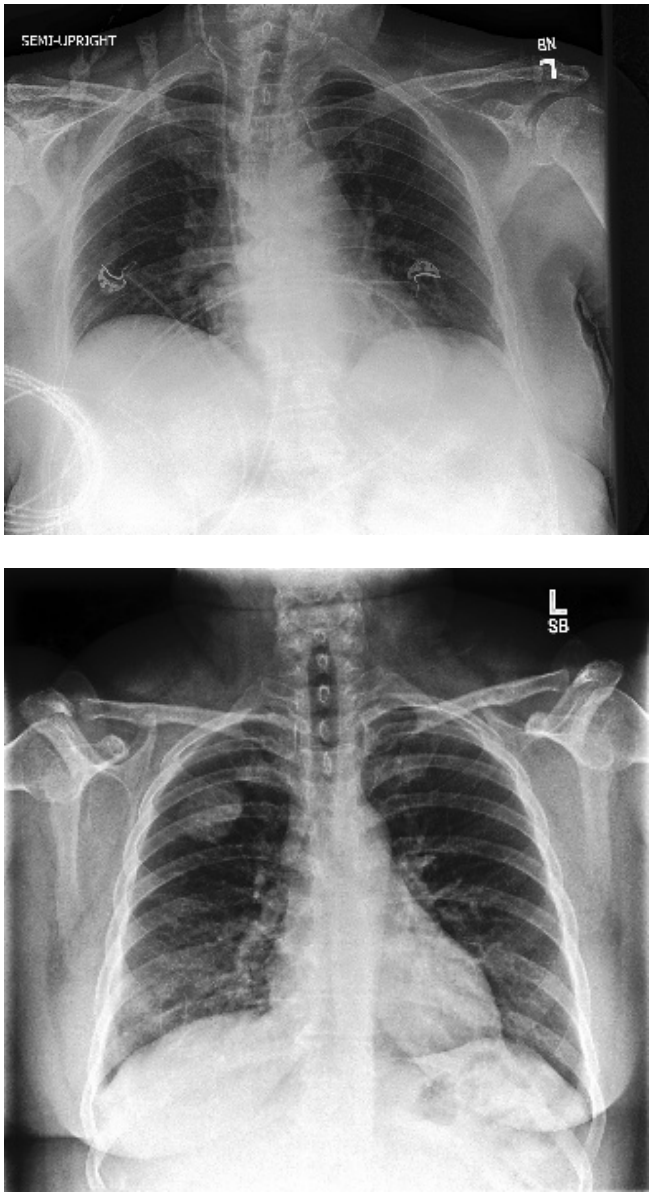

Figure 4.3: Images from the CheXpert dataset. An example of 'Enlarged Cardiomediastinum' above, 'Lung Lesion' below.

Figure 4.4: Images from the CheXpert dataset. An example of 'Pleural Other' above, 'Pneumonia' below.

4.4.2 Impact of Attention Models on CXR

In this section, we study the impact of attention modules on classification performance. We examine the results attained by ResNet50+Self_Attention and CBAM on both CXR datasets.

Table 4.5 showcase the results of ResNet50+Self_Attention on the ChestX-ray14 dataset. From these tables, we notice that, when only using the classification losses, the best performance is obtained by CEL with an AUC average of 76.50% after 2 epochs. Combined losses present the best performance with CEL and Contrastive loss, with an average of 76.80% after 3 epochs. In both cases, CEL provides the best performance. However, the classification loss alone converged an epoch sooner than the combined loss.

For CheXpert, the results are collected in Table 4.6. We notice that, when classification loss is employed on its own, the best average classification is achieved with the percentage of 78.45% after 3 epochs for CEL. For joint losses, the best average AUC is achieved with 79.02% after 3 epochs, for CEL and Contrastive loss. Furthermore, both of the losses for the dataset CheXpert that performed the best for this model, ResNet50+Self_Attention, have the classification loss CEL. The convergence of both best losses also took the same amount of epochs. When analyzing Table 4.6, we can see the issue that BCE and Focal have for the combination losses.

Furthermore, in Tables 4.5 and 4.6, we notice the trend predicted in the previous section. The BCE and Focal losses do not work well in combination with the metric losses.

Loss Name			Best Average	Epoch
BCE			0.7565	2
	+	***Contrastive***	0.5639	5
	+	***Triplet***	0.6136	3
	+	***Proxy-NCA***	0.7555	2
CEL			0.7650	2
	+	***Contrastive***	0.7680	3
	+	***Triplet***	0.7614	2
	+	***Proxy-NCA***	0.7562	3
WCEL			0.7565	2
	+	***Contrastive***	0.7633	2
	+	***Triplet***	0.7595	2
	+	***Proxy-NCA***	0.7599	2
Focal			0.7489	2
	+	***Contrastive***	0.5389	2
	+	***Triplet***	0.6290	2
	+	***Proxy-NCA***	0.7565	4

Table 4.5: Results for ResNet50+Self_Attention for the joint losses along with the epoch that yields the best AUC average for the ChestX-ray14 dataset.

Loss Name			Best Average	Epoch
BCE			0.7008	3
	+	***Contrastive***	0.5394	5
	+	***Triplet***	0.5855	2
	+	***Proxy-NCA***	0.7664	3
CEL			0.7845	3
	+	***Contrastive***	0.7902	3
	+	***Triplet***	0.7790	2
	+	***Proxy-NCA***	0.7336	2
WCEL			0.7545	3
	+	***Contrastive***	0.7273	3
	+	***Triplet***	0.7653	3
	+	***Proxy-NCA***	0.7592	3
Focal			0.7396	2
	+	***Contrastive***	0.5638	3
	+	***Triplet***	0.6047	4
	+	***Proxy-NCA***	0.7137	3

Table 4.6: Results for ResNet50+Self_Attention for the joint losses along with the epoch that yields the best AUC average for the CheXpert dataset.

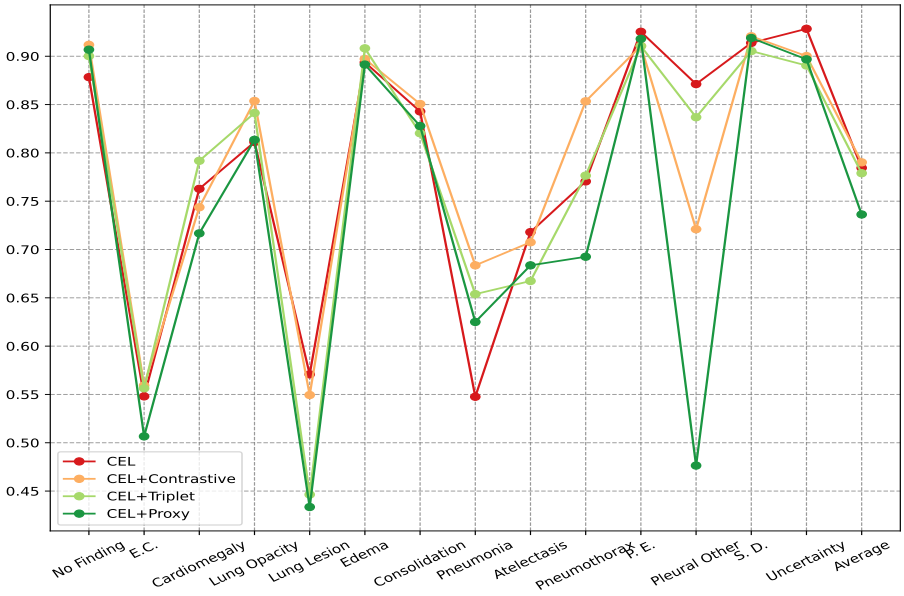

Figure 4.5: Plot of all CEL losses for the ResNet50+Self_Attention model for the CheXpert dataset.

The Class-specific AUC using the CEL losses for ResNet+Self_Attention is depicted in Figure 4.5. From the figure, we notice a large drop in performance for the classes 'Lung Lesion', 'Pleural other', 'Enlarged Cardiomediastinum', and 'Pneumonia'. For these classes, and unlike the baseline plot (Figure 4.2), all of the losses have a dip in their performance. From the losses, it can be seen that the combination of CEL and Proxy-NCA has the most decrements in performance. Meanwhile, the other three losses do not exhibit such poor performance in the 'Pleural Other' class. When looking at the graph plotted for WCEL for this model, it can be seen that WCEL has even steeper fluctuations in AUC than CEL does, along with the same problematic classes. However, in the graph for WCEL, there is a lower accuracy percentage than CEL. It can be seen for both BCE and Focal loss, that their graphs have stronger fluctuations without a specific pattern. However, their AUC for 'Lung Lesion' is even worse, dropping to roughly 15%.

Loss Name			**Best Average**	**Epoch**
BCE			0.7494	2
	+	***Contrastive***	0.6073	3
	+	***Triplet***	0.6087	5
	+	***Proxy-NCA***	0.7563	2
CEL			0.7617	3
	+	***Contrastive***	0.7692	3
	+	***Triplet***	0.7643	3
	+	***Proxy-NCA***	0.7664	3
WCEL			0.7646	3
	+	***Contrastive***	0.7649	2
	+	***Triplet***	0.7650	2
	+	***Proxy-NCA***	0.7646	3
Focal			0.7565	2
	+	***Contrastive***	0.5708	3
	+	***Triplet***	0.5872	5
	+	***Proxy-NCA***	0.7538	5

Table 4.7: Results for CBAM for the joint losses along with the epoch that yields the best AUC average for the ChestX-ray14 dataset.

Regarding CBAM, for the ChestX-ray14 dataset, the results we have achieved can be seen in Table 4.7. In the aforementioned table, the best classification loss is WCEL which achieved a classification percentage of 76.46% after 3 epochs. When looking at the combined loss in Table 4.7, we can see that the best loss is 76.92% after 3 epochs, achieved by the combination of CEL and Contrastive loss. The best result is achieved by the combined loss with a difference of 0.46%. The best performance is obtained on both cases with the CEL loss, and they both converged in 3 epochs.

For CheXpert, the results can be seen in Table 4.8. We first look at the table with results for purely the classification loss, from there we can see that

Loss Name			Best Average	Epoch
BCE			0.7340	3
	+	***Contrastive***	0.5802	2
	+	***Triplet***	0.5961	2
	+	***Proxy-NCA***	0.7466	3
CEL			0.7803	5
	+	***Contrastive***	0.7674	3
	+	***Triplet***	0.7594	3
	+	***Proxy-NCA***	0.7870	5
WCEL			0.7618	3
	+	***Contrastive***	0.7447	2
	+	***Triplet***	0.7626	2
	+	***Proxy-NCA***	0.7514	5
Focal			0.6974	2
	+	***Contrastive***	0.5636	4
	+	***Triplet***	0.6157	2
	+	***Proxy-NCA***	0.7463	3

Table 4.8: Results for CBAM for the joint losses along with the epoch that yields the best AUC average for the CheXpert dataset.

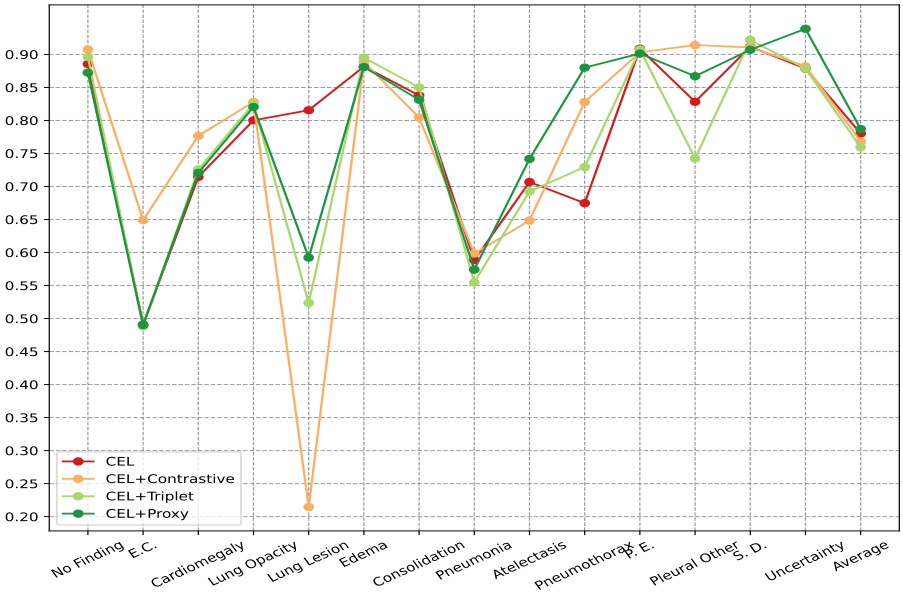

Figure 4.6: Plot of all CEL losses for the CBAM model for the CheXpert dataset.

the best result is achieved by the loss CEL in 5 epochs for a percentage of 78.03%. Meanwhile, the combined CEL and Proxy-NCA loss achieved an AUC

of 78.70% in 5 epochs. When comparing both of the best results from the two tables, we conclude that the combined loss performs better, with a difference of 0.67%. Furthermore, the issue with BCE and Focal loss persists across both datasets.

Figure 4.6 shows the class-specific performance for CBAM for the CheXpert dataset. Therein, it can be noticed that 'Lung Lesion' performs the worst out of all of the classes in the dataset. Furthermore, all of the losses have a downward trend for this class. While for all of the four losses, they seem to follow each other trajectory closely for the class 'Lung Lesion', CEL by itself has the best classification average over the combinations with metric losses. The opposite occurs for the class 'Pneumothorax'. When comparing this model's figure to Figure 4.5 it can be seen that this model with attention does not have as many classes that it fails in. When looking at the way WCEL behaves in comparison to CEL, it can be seen that WCEL has a higher rate of failure for the class 'Lung Lesion' for all of the losses, whereas CEL's failure in classification is much milder in comparison. However, WCEL's classification base starts higher at around 30%. WCEL has more prominent dips in the figure for the class 'Pleural Other'. For the other two losses(and combinations of), BCE and Focal loss, they both behave more erratically over their graphs. BCE fails at the classes, 'Enlarged Cardiomediastinum', 'Lung Lesion', 'Pneumonia', and 'Pleural Other'. Whereas, Focal loss fails in the classes 'Lung Lesion', 'Atelectasis', and 'Uncertainty'. Since 'Atelectasis' is one of the most frequent classes in the dataset, the failing should originate from the model in question.

4.4.3 Transformer Performance Comparison

In this section, we analyze the performance of transformer models and compare them against each other across both of our selected datasets. These models are C-Tran, ViT, and ViT Hybrid.

For C-Tran, on the ChestX-ray14 dataset, the results are summarized in Table 4.9. From this table, we observe that the best performance is achieved with Focal and Proxy-NCA loss. This combined loss achieves a classification average of 61.41% after training for 5 epochs. C-TRAN exhibits much lower performance than the other models examined in this work, regardless of the loss. This likely indicates that the overall architecture, designed for partial annotations, does not translate well to the use case of CXR classification.

For CheXpert, the results can be seen in Table 4.10. The best result is obtained with BCE and Proxy-NCA loss, having an average AUC of 63.13%, after 5 epochs. When comparing these two losses, it becomes apparent that the combined loss performs better with a difference of 6.6%.

The results from the datasets ChestX-ray14 and CheXpert show that this model does not perform well with CXR data. As mentioned above, that C-TRAN architecture was designed to handle missing (or incomplete) labels. While the original paper shows similar performance to other architectures, it does not transfer well to the CXR classification task.

Loss Name			Best Average	Epoch
BCE			0.5290	4
	+	***Contrastive***	0.5414	5
	+	***Triplet***	0.5088	3
	+	***Proxy-NCA***	0.5190	3
CEL			0.5083	3
	+	***Contrastive***	0.4989	4
	+	***Triplet***	0.5458	5
	+	***Proxy-NCA***	0.5043	5
WCEL			0.5031	4
	+	***Contrastive***	0.5107	2
	+	***Triplet***	0.5054	3
	+	***Proxy-NCA***	0.5156	5
Focal			0.5195	4
	+	***Contrastive***	0.5221	5
	+	***Triplet***	0.5102	2
	+	***Proxy-NCA***	0.6141	5

Table 4.9: Results for C-Tran for the joint losses along with the epoch that yields the best AUC average for the ChestX-ray14 dataset.

Loss Name			Best Average	Epoch
BCE			0.4706	5
	+	***Contrastive***	0.5096	5
	+	***Triplet***	0.5289	5
	+	***Proxy-NCA***	0.6313	5
CEL			0.5335	4
	+	***Contrastive***	0.4951	2
	+	***Triplet***	0.4877	3
	+	***Proxy-NCA***	0.5211	2
WCEL			0.5126	4
	+	***Contrastive***	0.5425	4
	+	***Triplet***	0.5054	3
	+	***Proxy-NCA***	0.5386	4
Focal			0.5653	4
	+	***Contrastive***	0.5547	5
	+	***Triplet***	0.5093	2
	+	***Proxy-NCA***	0.5432	2

Table 4.10: Results for C-Tran for the joint losses along with the epoch that yields the best AUC average for the CheXpert dataset.

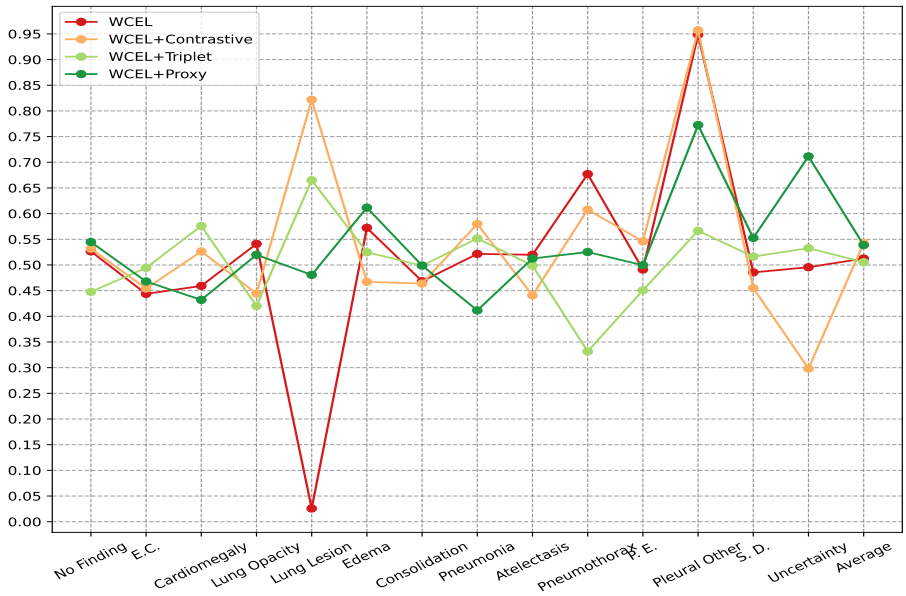

Figure 4.7: Plot of all WCEL losses for the C-Tran model for the CheXpert dataset.

The Class-specific AUC using the WCEL losses for C-TRAN is depicted in Figure 4.7. This figure is shown both for brevity, and because the transformer models work better with WCEL. We plot the performance on the CheXpert dataset since these models obtain better performance in comparison to ChestX-ray14.

From the figure in question, we notice a large drop in performance for the 'Lung Lesion' class. Therein, the classification percentage goes down to almost 0 when using WCEL. In comparison, the other losses retain a classification performance above 45%. Particularly WCEL with Contrastive loss, which performs above 80%. The other classes where the model struggles (AUC under 35%) are 'Pneumothorax' and 'Uncertainty'. The latter is likely due to its prevalence and lack of defining features. An interesting observation is that WCEL has a more stable behavior across the different labels when compared to the unweighted variant. CEL exhibits much stronger AUC fluctuations across different classes. For the other losses (and combinations) the most challenging class is 'Pleural Other'.

The results of ViT on the ChestX-ray14 dataset are shown in Tables 4.11. The best result is achieved CEL with Contrastive loss. This combined loss achieves an average of 75.92% in 3 epochs. This combined loss performs better when compared to just classification loss. However, the performance of CEL and WCEL with metric losses is comparable.

Loss Name			Best Average	Epoch
BCE			0.7166	3
	+	***Contrastive***	0.5661	5
	+	***Triplet***	0.5203	5
	+	***Proxy-NCA***	0.7418	2
CEL			0.7338	3
	+	***Contrastive***	0.7592	3
	+	***Triplet***	0.7534	3
	+	***Proxy-NCA***	0.7505	3
WCEL			0.7290	4
	+	***Contrastive***	0.7552	3
	+	***Triplet***	0.7532	2
	+	***Proxy-NCA***	0.7552	2
Focal			0.7189	3
	+	***Contrastive***	0.5562	2
	+	***Triplet***	0.5002	2
	+	***Proxy-NCA***	0.7341	3

Table 4.11: Results for ViT for the joint losses along with the epoch that yields the best AUC average for the ChestX-ray14 dataset.

Loss Name			Best Average	Epoch
BCE			0.7202	5
	+	***Contrastive***	0.5598	4
	+	***Triplet***	0.5230	4
	+	***Proxy-NCA***	0.7204	5
CEL			0.7359	3
	+	***Contrastive***	0.7657	4
	+	***Triplet***	0.7557	5
	+	***Proxy-NCA***	0.7196	3
WCEL			0.7557	4
	+	***Contrastive***	0.7337	3
	+	***Triplet***	0.7230	4
	+	***Proxy-NCA***	0.7653	4
Focal			0.7101	3
	+	***Contrastive***	0.6042	4
	+	***Triplet***	0.5854	4
	+	***Proxy-NCA***	0.6398	5

Table 4.12: Results for ViT for the joint losses along with the epoch that yields the best AUC average for the CheXpert dataset.

For CheXpert, the results can be seen in Table 4.12. The best performance is 76.57% after 4 epochs and is obtained by the same loss as in ChestX-ray14. However, larger variances are noticeable across experiments with CEL and WCEL.

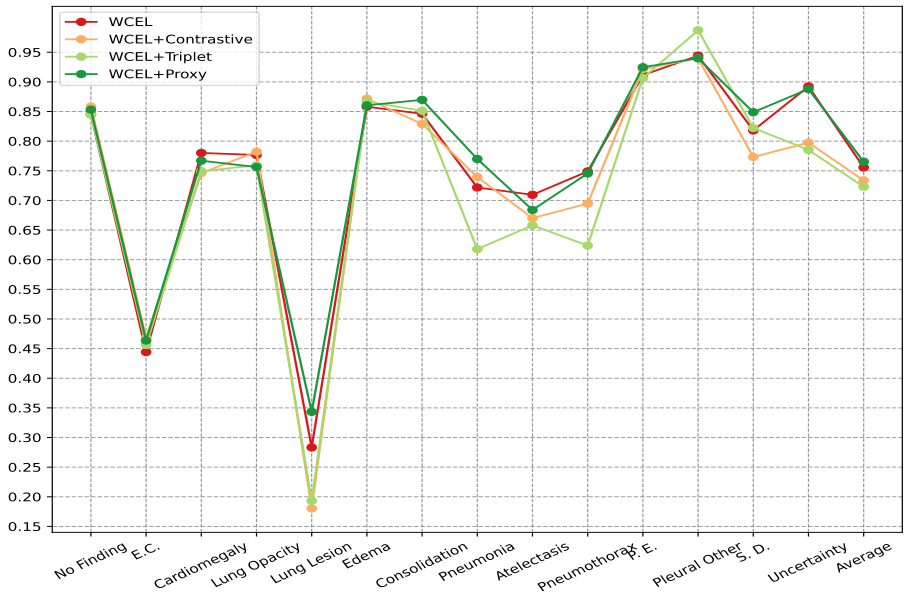

Figure 4.8: Plot of all WCEL losses for the ViT model for the CheXpert dataset.

We conjecture that for this model, the combination of class-balancing regularization and embedding learning harms the overall model performance.

A pattern we see that starts to emerge when looking at this Figure 4.8 is that 'Lung Lesion' begins to fail for the WCEL losses here again. Another class with low performance with the WCEL loss is 'Enlarged Cardiomediastinum'. For all the losses for this model, ViT, the classification 'Lung Lesion' fails. This image category is one of the rarest, which helps explain its consistently low performance. The other trend that follows for all of the losses is 'Enlarged Cardiomediastinum' also falls short of being classified well, with this classification falling below 45% in general.

In our experiments, BCE and Focal loss exhibit a large variance in performance across the different classes. Even when combined with metric losses, several classes present unacceptable performance. These are 'Enlarged Cardiomediastinum', 'Lung Lesion', 'Consolidation', 'Pneumonia', and 'Pleural Other'. These classes are underrepresented in the dataset.

For ViT Hybrid for the ChestX-ray14 dataset, the results we have achieved can be seen in Table 4.13. We can see that the best result is achieved by WCEL and Triplet loss after 3 epochs, with an average AUC of 76.93%. Similarly, this joint loss achieves the best performance on CheXpert with an average AUC of 80.15%, in 5 epochs.

When looking at Figure 4.9, we can see that with this model, the WCEL loss achieves comparable performance across all classes. The addition of metric losses has a minor influence on most classes. This phenomenon is not as consistent

Loss Name			Best Average	Epoch
BCE			0.7470	3
	+	***Contrastive***	0.5736	5
	+	***Triplet***	0.5180	4
	+	***Proxy-NCA***	0.7463	4
CEL			0.7589	4
	+	***Contrastive***	0.7543	4
	+	***Triplet***	0.7354	5
	+	***Proxy-NCA***	0.7547	3
WCEL			0.7390	5
	+	***Contrastive***	0.6395	5
	+	***Triplet***	0.7693	3
	+	***Proxy-NCA***	0.7199	5
Focal			0.7568	3
	+	***Contrastive***	0.5085	4
	+	***Triplet***	0.5429	4
	+	***Proxy-NCA***	0.7411	4

Table 4.13: Results for ViT-Hybrid for the joint losses along with the epoch that yields the best AUC average for the ChestX-ray14 dataset.

Loss Name			Best Average	Epoch
BCE			0.7650	5
	+	***Contrastive***	0.7061	3
	+	***Triplet***	0.5149	2
	+	***Proxy-NCA***	0.7523	3
CEL			0.5430	5
	+	***Contrastive***	0.7792	3
	+	***Triplet***	0.7751	3
	+	***Proxy-NCA***	0.7752	3
WCEL			0.7821	3
	+	***Contrastive***	0.7744	5
	+	***Triplet***	0.8015	5
	+	***Proxy-NCA***	0.7866	5
Focal			0.7422	2
	+	***Contrastive***	0.7048	5
	+	***Triplet***	0.5395	5
	+	***Proxy-NCA***	0.7512	5

Table 4.14: Results for ViT-Hybrid for the joint losses along with the epoch that yields the best AUC average for the CheXpert dataset.

with the other transformer models. Furthermore, the base performance of ViT Hybrid on CheXpert is larger than that of ViT. Likely, the ResNet backbone is more suitable for this task and this amount of data.

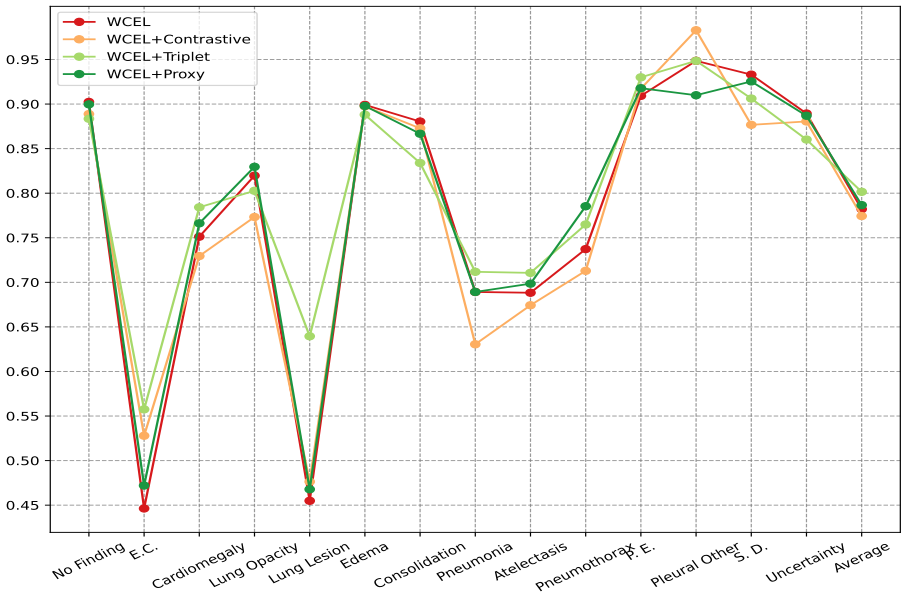

Figure 4.9: Plot of all WCEL losses for the ViT Hybrid model for the CheXpert dataset.

4.4.4 Loss Comparison

In this section, we discuss the loss functions and their overall performance. We first look at classification, metric, and joint losses, in that order. Regarding the ChestX-ray14 dataset, it can be seen that the best result (using only classification losses) is obtained by the ResNet+Self_Attention model trained with CEL. The best performance with a combined loss is achieved by the ViT Hybrid model using WCEL and Triplet losses.

Across the majority of the experiments (3 out of 6 models), the CEL loss yields the best results, followed by WCEL on 2 out of 6 models. For the combined losses across the 6 models we have evaluated, the best overall joint loss is CEL with Contrastive. This combined loss achieved the best performance on 4 out of 6 models. Combining losses with WCEL only outperforms other models when training the ViT Hybrid model. When comparing models trained with the stand-alone and combined losses, we observe that in every instance the combined losses obtain better results. However, the improvement produced by adding these additional loss terms varies strongly. The smallest difference is reported for ResNet+Self_Attention with a difference of 0.30%. Meanwhile, the most difference is observed for C-Tran with 8.51%. Furthermore, the other transformer models, ViT and ViT Hybrid, also benefit from the addition of these metric loss terms. Their AUC improves by 2.59% and 1.04%, respectively. In Table 4.15 the best results for each model along with the losses and epoch

where the loss was reached can be seen.

Models	Loss	Epoch	Average AUC
ResNet50	CEL + Contrastive	3	0.7667
ResNet50+Self_Attention	CEL + Contrastive	3	0.7680
CBAM	CEL + Contrastive	3	0.7692
C-Tran	Focal + Proxy-NCA	5	0.6141
ViT	CEL + Contrastive	3	0.7592
ViT Hybrid	WCEL + Triplet	3	0.7693

Table 4.15: The best results for each model along with the loss that produced the best result, and the epoch at which the result was produced for the dataset ChestX-ray14.

When considering the CheXpert dataset, and its corresponding results, ResNet+Self_Attention exhibits the best performance when training only for classification. This result is consistent with that of ChestX-ray14. However, unlike those experiments, the highest performance by combined loss is achieved by ResNet with CEL and Triplet loss. Another trend that is similar across both datasets is the prevalence of CEL across the majority of experiments. For the stand-alone classification losses, CEL obtains the highest performance on 3 out of 6 models. Additionally, metric losses combined with CEL present the best AUC on 4 out of 6 models. In general, CEL yields the best results when paired with the Contrastive loss, with Triplet and Proxy-NCA tied for second. The combination with WCEL loss only outperforms the alternatives when training ViT Hybrid. The difference between the classification losses by themselves and the combined losses is more pronounced in the CheXpert dataset. The smallest variations are observed on ResNet+Self_Attention with a difference of 0.57%. Meanwhile, the largest gain is recorded on C-Tran with 6.6%. Once again, this is a similar trend to ChestX-ray14. It can also be seen that the other transformer models also benefit from the inclusion of an additional metric learning term. The performance of ViT and ViT Hybrid increase by 1% and 1.94%, correspondingly. On average, the best results of CheXpert exceed those of ChestX-ray14. In Table 4.16 the best results for each model along with the losses and epoch where the loss was reached can be seen.

When looking at the overall performance of the models, it can be seen that the CEL family of losses performs the best. The only exception to this statement is the C-Tran model. We conjecture that the poor performance presented by C-Tran is likely due to the architectural differences and training protocol of the model.

Models	Loss	Epoch	Average AUC
ResNet50	CEL + Triplet	3	0.8082
ResNet50 + Self Attention	CEL + Contrastive	3	0.7902
CBAM	CEL + Proxy-NCA	5	0.7870
C-Tran	BCE + Proxy-NCA	5	0.6313
ViT	CEL + Contrastive	4	0.7657
ViT Hybrid	WCEL + Triplet	5	0.8015

Table 4.16: The best results for each model along with the loss that produced the best result, and the epoch at which the result was produced for the dataset CheXpert.

Tables 4.17 and 4.16, we have compiled the results of the best loss per model studied in this work for ChestX-ray14 and Chexpert, correspondingly. These results are from the best on average AUC per model. Additionally, Tables 4.15 and 4.18 show the class-specific AUC for the best models.

Upon observation of the classification accuracy for the best average AUC across the networks we notice a trend for the dataset ChestX-ray14 [51]. 'Pneumothorax' and 'Edema' are two classes that appear in every single model's top 5 highest scoring classes. 'Cardiomegaly' appears 5 out of 6 of the models. Meanwhile, 'Emphysema' and 'Hernia' appear 4 out of 6 times. When looking at the 5 classes with the worst performance from Table 4.15, it can be seen that the class labels 'Pneumonia' and 'Nodule' are consistent for all models. 'Infiltration' and 'No Finding' appear 5 out of the 6 models in their least classified. It can also be seen for the model C-Tran [24] that class labels that do not appear in the top or bottom 5 of the other models appear. For the class labels, 'Emphysema', 'Edema', and 'Hernia' appearing with more frequency across the models even thought they are some of the classes that have the least amount of examples, our intuition is that this occurs because of the contrast they cause in CXR when compared to normal CXR images. For 'Emphysema' the ribs spread and a colour contrast occurs from the air that spreads into the lung, and the fact that the diaphragm gets pushed down. 'Edema' in the lugs is when there is any sort of fluid build up in the chest cavity, this build up will turn out a grayish white on the CXR. 'Hernia' can be seen by the movement of either the diaphragm or the stomach into the chest cavity, along with the build up of fluid and gasses.

When looking at Table 4.16 trends become noticeable. 'Support Devices' and 'Pleural Effusion' appear in 5 out of the 6 models in their top 5 highest scoring classes. 'Edema' and 'Uncertainty' appear in 4 out of 6 models. 'No Finding' appears in 3 out of 6 models. When looking at the 5 classes with the worst performance in the table, it can be seen that the class labels 'Enlarged Cardiomediastinum', 'Pneumonia', 'Atelectasis', and 'Lung Lesion' appears in 5 out of the 6 models. It can also be seen again for the model C-Tran [24] that class labels that do not appear in the top or bottom 5 of the other models appear. An interesting finding is that the class label 'Uncertainty', one of the

least frequent in the dataset, consistently appears in the top 5 highest-scoring. Our intuition for this behavior is as follows. This class is easier to predict for the models because it is not attached to any specific semantic meaning.

	Models					
Class Labels	**ResNet50**	**ResNet50+Self_Attention**	**CBAM**	**C-Tran**	**ViT**	**ViT Hybrid**
Atelectasis	0.7427	0.7490	0.7487	0.5807	0.7326	0.7510
Cardiomegaly	0.8603	0.8554	0.8623	0.5250	0.8399	0.8763
Effusion	0.8061	0.8062	0.8084	0.7341	0.7966	0.8172
Infiltration	0.6488	0.6563	0.6542	0.6244	0.6586	0.6645
Mass	0.7794	0.7862	0.7940	0.5734	0.7642	0.7934
Nodule	0.7153	0.7216	0.7182	0.5538	0.6957	0.7241
Pneumonia	0.6924	0.6911	0.6871	0.5525	0.6998	0.7080
Pneumothorax	0.8224	0.8275	0.8239	0.6512	0.8064	0.8199
Consolidation	0.7208	0.7232	0.7241	0.6707	0.7157	0.7365
Edema	0.8186	0.8225	0.8168	0.6730	0.8136	0.8219
Emphysema	0.8744	0.8825	0.8598	0.6066	0.8344	0.7883
Fibrosis	0.7730	0.7722	0.7803	0.5538	0.7756	0.7596
Pleural_Thickening	0.7265	0.7334	0.7213	0.5977	0.7361	0.7241
Hernia	0.8093	0.7806	0.8225	0.6317	0.8164	0.8310
No Finding	0.7100	0.7129	0.7167	0.6828	0.7028	0.7244
Average AUC	0.7667	0.7680	0.7692	0.6141	0.7592	0.7693

Table 4.17: The results for all the classes for the best average result found for each model for the dataset ChestX-ray14.

	Models					
Class Labels	**ResNet50**	**ResNet50+Self_Attention**	**CBAM**	**C-Tran**	**ViT**	**ViT Hybrid**
No Finding	0.8898	0.9118	0.8724	0.7029	0.8687	0.8835
Enlarged Cardiomediastinum	0.4597	0.5593	0.4902	0.7558	0.4725	0.5575
Cardiomegaly	0.7435	0.7437	0.7206	0.7468	0.7475	0.7842
Lung Opacity	0.8338	0.8537	0.8204	0.7359	0.7695	0.8027
Lung Lesion	0.9485	0.5494	0.5923	0.4764	0.5665	0.6395
Edema	0.9070	0.8972	0.8809	0.6780	0.8306	0.8881
Consolidation	0.8449	0.8506	0.8314	0.8431	0.8238	0.8339
Pneumonia	0.6206	0.6836	0.5741	0.4444	0.7804	0.7118
Atelectasis	0.7013	0.7075	0.7418	0.7132	0.6809	0.7106
Pneumothorax	0.8921	0.8534	0.8800	0.5487	0.6759	0.7649
Pleural Effusion	0.9151	0.9107	0.9013	0.7063	0.9139	0.9300
Pleural Other	0.7468	0.7210	0.8670	0.2017	0.8712	0.9485
Support Devices	0.9174	0.9207	0.9070	0.5828	0.8456	0.9062
Uncertainty	0.8938	0.9002	0.9388	0.7027	0.8731	0.8602
Average AUC	0.8082	0.7902	0.7870	0.6313	0.7657	0.8015

Table 4.18: The results for all the classes for the best average result found for each model for the dataset CheXpert.

Chapter 5

Conclusion

In this work, we have implemented, trained, and tested several popular or state-of-the-art CNNs: ResNet50, ResNet50+Self_Attention, CBAM, C-Tran, ViT, and ViT Hybrid. We have studied the impact on the final performance of various classification losses. Additionally, we extended our study and evaluated the losses with an additional metric loss term. The objective of this thesis is a comparative study on how well Vision Transformers perform on the Chest X-ray classification task in comparison to simpler models, including more traditional convolutional backbones employing attention. We have also considered whether the inclusion of metric losses can improve the overall classification performance. This remains a relatively understudied topic due to the additional complexity presented by the multi-label problem.

In our experiments, Vision Transformers provide comparable performance to the baseline models. Furthermore, we conclude that the joint deployment of classification and metric losses improve the average Area Under the Curve (AUC) metric. Nevertheless, the beneficial effect depends highly on both the combination of losses, as well as the dataset and model to be trained. Under the best circumstances, the combination of these losses can increase the AUC by up to 2%. For the Vision Transformer models, we have reported that the combined loss was particularly helpful for improving the classification performance. While it is surprising that the more powerful models do not perform significantly better than older architectures, Vision Transformers are particularly data-hungry. Thereby, we conjecture that for sufficiently large medical datasets, Vision Transformers will outperform our baseline models.

From our experiments, we successfully answer our research questions. We have observed that the ViT Hybrid architecture achieves the best performance on the ChestX-ray14 dataset. Meanwhile, the baseline ResNet50 model obtained the best results on CheXpert. These results are for the classification percentages of the combined losses. For the case of just the classification losses being implemented for the models, it was seen that for the ChestX-ray14 and CheXpert datasets that the model ResNet50+Self_Attention performed the best. For all of the models except C-Tran, and for the singular case of just the classification

loss being used for ViT, it can be seen that classification accuracy of 75% and over was found.

Future research in the context of this work would be to re-train and test the models with the full version of the dataset CheXpert. This would allow us to confirm our current conjectures regarding transformer models and their reliance on massive datasets. We limited ourselves to the smaller version of the dataset due to hardware and time constraints. Another interesting avenue of research is studying better ways of mining positive and negative samples that could benefit metric learning techniques depending on it. Alternatively, other proxy-based losses can be studied, as we have demonstrated that they reduce the need to mine relevant samples. Additionally, and based on the success of CEL and WCEL, we consider that adapting the losses to the specific architecture could provide significantly enhanced performance. Finally, we propose to extend this work to classification in other medical imaging domains (MRI, ultrasound).

Bibliography

[1] Nkechinyere N. Agu, Joy T. Wu, Hanqing Chao, Ismini Lourentzou, Arjun Sharma, Mehdi Moradi, Pingkun Yan, and James Hendler. Anaxnet: Anatomy aware multi-label finding classification in chest x-ray. In *MICCAI*, 2021.

[2] Mauro Annarumma and G. Montana. Deep metric learning for multi-labelled radiographs. *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, 2018.

[3] Khalid El Asnaoui, Youness Chawki, and A. Idri. Automated methods for detection and classification pneumonia based on x-ray images using deep learning. *ArXiv*, abs/2003.14363, 2020.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2015.

[5] Ivo M. Baltruschat, Hannes Nickisch, Michael Grass, Tobias Knopp, and Axel Saalbach. Comparison of deep learning approaches for multi-label chest x-ray classification. *Scientific Reports*, 9, 2019.

[6] François Chollet. Xception: Deep learning with depthwise separable convolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807, 2017.

[7] Dina Demner-Fushman, Marc D. Kohli, Marc B. Rosenman, Sonya E. Shooshan, Laritza M. Rodriguez, S. Antani, George R. Thoma, and Clement J. McDonald. Preparing a collection of radiology examinations for distribution and retrieval. *Journal of the American Medical Informatics Association : JAMIA*, 23 2:304–10, 2016.

[8] A. Dosovitskiy, L. Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, M. Dehghani, Matthias Minderer, G. Heigold, S. Gelly, Jakob Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2021.

[9] Qingji Guan and Yaping Huang. Multi-label chest x-ray image classification via category-wise residual attention learning. *Pattern Recognit. Lett.*, 130:259–266, 2020.

[10] Qingji Guan, Yaping Huang, Zhun Zhong, Zhedong Zheng, Liang Zheng, and Yi Yang. Diagnose like a radiologist: Attention guided convolutional neural network for thorax disease classification. *ArXiv*, abs/1801.09927, 2018.

[11] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2:1735–1742, 2006.

[12] ErnestL Hall, RichardP Kruger, TurnerA Franklin, et al. Automated measurements from chest x-rays for lung disease classification. *Information Processing Society of Japan Research Report Medical Information Processing (MED)*, 1975(15 (1975-MED-007)):22–30, 1975.

[13] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[14] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, M. Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *ArXiv*, abs/1704.04861, 2017.

[15] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

[16] Emtiaz Hussain, M. Hasan, Md Anisur Rahman, Ickjai Lee, Tasmi Tamanna, and M. Parvez. Corodet: A deep learning based classification for covid-19 detection using chest x-ray images. *Chaos, Solitons, and Fractals*, 142:110495 – 110495, 2020.

[17] Forrest N. Iandola, M. Moskewicz, Khalid Ashraf, Song Han, W. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡1mb model size. *ArXiv*, abs/1602.07360, 2016.

[18] Abdullahi Umar Ibrahim, M. Ozsoz, Sertan Serte, Fadi Al-turjman, and P. Yakoi. Pneumonia classification using deep learning from chest x-ray images during covid-19. *Cognitive Computation*, pages 1 – 13, 2021.

[19] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn L. Ball, Katie S. Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz,

Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. *CoRR*, abs/1901.07031, 2019.

[20] JR Jagoe. Gradient pattern coding—an application to the measurement of pneumoconiosis in chest x rays. *Computers and Biomedical Research*, 12(1):1–15, 1979.

[21] Alistair E. W. 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 large publicly available database of labeled chest radiographs. *ArXiv*, abs/1901.07042, 2019.

[22] KC Kamal, Zhendong Yin, Ming-Yang Wu, and Zhilu Wu. Evaluation of deep learning-based approaches for covid-19 classification based on chest x-ray images. *Signal, Image and Video Processing*, pages 1 – 8, 2021.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, May 2017.

[24] Jack Lanchantin, Tianlu Wang, Vicente Ordonez, and Yanjun Qi. General multi-label image classification with transformers. In *CVPR*, 2021.

[25] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:318–327, 2020.

[26] Fenglin Liu, Changchang Yin, Xian Wu, Shen Ge, Ping Zhang, and Xu Sun. Contrastive attention for automatic chest x-ray report generation. In *FINDINGS*, 2021.

[27] Andrew L. Maas, Awni Y. Hannun, Daniel Jurafsky, and Andrew Y. Ng. First-pass large vocabulary continuous speech recognition using bi-directional recurrent dnns. *CoRR*, abs/1408.2873, 2014.

[28] Vishu Madaan, A. Roy, C. Gupta, Prateek Agrawal, Anand Sharma, Cristian-Sorin Bologa, and R.-C. Prodan. Xcovnet: Chest x-ray image classification for covid-19 early detection using convolutional neural networks. *New Generation Computing*, pages 1 – 15, 2021.

[29] Andreas K. Maier, Christopher Syben, Tobias Lasser, and Christian Riess. A gentle introduction to deep learning in medical image processing. *CoRR*, abs/1810.05401, 2018.

[30] Shaocong Mo and Ming Cai. Deep learning based multi-label chest x-ray classification with entropy weighting loss. *2019 12th International Symposium on Computational Intelligence and Design (ISCID)*, pages 124–127, 2019.

[31] Yair Movshovitz-Attias, Alexander Toshev, Thomas Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 360–368, 2017.

[32] Xi Ouyang, Srikrishna Karanam, Ziyan Wu, Terrence Chen, Jiayu Huo, Xiang Sean Zhou, Qian Wang, and Jie-Zhi Cheng. Learning hierarchical attention for weakly-supervised chest x-ray abnormality localization and diagnosis. *IEEE Transactions on Medical Imaging*, 40:2698–2710, 2021.

[33] Hieu Pham, Tung T. Le, Dat Thanh Ngo, Dat Q. Tran, and Ha Q. Nguyen. Interpreting chest x-rays via cnns that exploit hierarchical disease dependencies and uncertainty labels. *Neurocomputing*, 437:186–194, 2021.

[34] Zhi Qiao, Austin Bae, Lucas Glass, Cao Xiao, and Jimeng Sun. Flannel (focal loss based neural network ensemble) for covid-19 detection. *Journal of the American Medical Informatics Association : JAMIA*, 28:444 – 452, 2021.

[35] Maithra Raghu, Chiyuan Zhang, Jon M. Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. In *NeurIPS*, 2019.

[36] Tawsifur Rahman, A. Khandakar, M. A. Kadir, Khandaker R. Islam, K. F. Islam, R. Mazhar, T. Hamid, M. Islam, Z. B. Mahbub, M. Ayari, and M. E. Chowdhury. Reliable tuberculosis detection using chest x-ray with deep learning, segmentation and visualization. *IEEE Access*, 8:191586–191601, 2020.

[37] Pranav Rajpurkar, Jeremy A. Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Yi Ding, Aarti Bagul, C. Langlotz, Katie S. Shpanskaya, Matthew P. Lungren, and A. Ng. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *ArXiv*, abs/1711.05225, 2017.

[38] Ram Murti Rawat, Shivam Garg, Naman Jain, and Gagan Gupta. Covid-19 detection using convolutional neural network architectures based upon chest x-rays images. *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 1070–1074, 2021.

[39] M. Sandler, Andrew G. Howard, Menglong Zhu, A. Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[40] AM Savol, CC Li, and RJ Hoy. Computer-aided recognition of small rounded pneumoconiosis opacities in chest x-rays. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2(5):479–482, 1980.

[41] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.

[42] K. Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015.

[43] O. Stephen, M. Sain, U. J. Maduh, and Do-Un Jeong. An efficient deep learning approach to pneumonia classification in healthcare. *Journal of Healthcare Engineering*, 2019, 2019.

[44] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016.

[45] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[46] Christian Szegedy, V. Vanhoucke, S. Ioffe, Jonathon Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[47] Mesut Toğaçar, B. Ergen, and Zafer Cömert. Covid-19 detection using deep learning models to exploit social mimic optimization and structured chest x-ray images using fuzzy color and stacking approaches. *Computers in Biology and Medicine*, 121:103805 – 103805, 2020.

[48] T. T. Tran, Huy-Hieu Pham, T. V. Nguyen, T. T. Le, H. T. Nguyen, and H. Q. Nguyen. Learning to automatically diagnose multiple diseases in pediatric chest radiographs using deep convolutional neural networks. *ArXiv*, abs/2108.06486, 2021.

[49] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *ArXiv*, abs/1706.03762, 2017.

[50] Hongyu Wang and Yong Xia. Chestnet: A deep neural network for classification of thoracic diseases on chest radiography. *ArXiv*, abs/1807.03058, 2018.

[51] 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. *CoRR*, abs/1705.02315, 2017.

[52] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, and Ronald M. Summers. Tienet: Text-image embedding network for common thorax disease classification and reporting in chest x-rays. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9049–9058, 2018.

[53] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In-So Kweon. Cbam: Convolutional block attention module. In *ECCV*, 2018.

[54] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *CoRR*, abs/1611.10080, 2016.

[55] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431*, 2016.

[56] Shuaijing Xu, Xiaoyilei Yang, Junqi Guo, Hao Wu, Guangzhi Zhang, and Rongfang Bie. Cxnet-m3: A deep quintuplet network for multi-lesion classification in chest x-ray images via multi-label supervision. *IEEE Access*, 8:98693–98704, 2020.

[57] Chaochao Yan, Jiawen Yao, Ruoyu Li, Zheng Xu, and Junzhou Huang. Weakly supervised deep learning for thoracic disease classification and localization on chest x-rays. *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, 2018.

[58] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8697–8710, 2018.