

Rumi Rajbhandari

# AI Driven Healthcare: Automated detection of Chest-Xray Abnormalities

Master's thesis in Applied Computer Science

Supervisor: Assoc. Prof. Ali Shariq Imran

Co-supervisor: Mohib Ullah

December 2023



Rumi Rajbhandari

# **AI Driven Healthcare: Automated detection of Chest-Xray Abnormalities**

Master's thesis in Applied Computer Science  
Supervisor: Assoc. Prof. Ali Shariq Imran  
Co-supervisor: Mohib Ullah  
December 2023

Norwegian University of Science and Technology  
Faculty of Information Technology and Electrical Engineering





# Abstract

As the world population continues to grow, a significant challenge emerges to provide healthcare facilities to an ever-expanding population. Harnessing the power of AI, this challenge could be easily mitigated. One of the solutions to solve the lack of manpower in the field of medical sector is to incorporate AI to help in the decision making of the medical profession. In order to automate the identification of diseases in the field of chest x-rays, this thesis conducts a thorough investigation of image classification and object recognition techniques. With an emphasis on improving diagnostic capabilities, the work makes use of cutting-edge image classification and object identification models, to detect the anomalies in chest X-rays.

The study highlights the results of these experiments and discusses how crucial it is to constantly enhance and improve in order to correctly identify the disease as a whole. Evaluation and visual assessments of model outputs provide a deeper understanding of their effectiveness, laying the groundwork for future advancements in using machine learning for detecting deformities in chest X-rays.



# Acknowledgement

I would like to express my sincere gratitude to my supervisor Assoc. Prof. Ali Shariq Imran and my co supervisor Mohib Ullah for their continuous guidance and support throughout the thesis works. Their consistent guidance and supervision has helped me to create a right path for my thesis and also perform my best. I am truly grateful for their mentorship.

Additionally, I would like to express my gratitude to Norwegian University of Science and Technology for providing me with the necessary resources to conduct the experiments. This thesis would have been incomplete without the support provided by the university.

I would also like to extend my appreciation to my friends and family for their unwavering support throughout the entire process and their warm words of encouragement.

**Rumi Rajbhandari**





# Contents

<b>Abstract</b> . . . . .	<b>i</b>
<b>Acknowledgement</b> . . . . .	<b>iii</b>
<b>Contents</b> . . . . .	<b>v</b>
<b>Figures</b> . . . . .	<b>vii</b>
<b>Tables</b> . . . . .	<b>ix</b>
<b>Acronyms</b> . . . . .	<b>xi</b>
<b>1 Introduction</b> . . . . .	<b>1</b>
1.1 Introduction and Problem statement . . . . .	1
1.2 Justification, Motivation and Benefits . . . . .	2
1.3 Research Questions . . . . .	2
1.4 Contributions . . . . .	3
1.5 Paper Outline . . . . .	3
<b>2 Background</b> . . . . .	<b>5</b>
2.1 Object Detection . . . . .	5
2.2 Image Classification, Object Detection and Instance Segmentation . . . . .	5
2.3 Image Classification Models . . . . .	6
2.3.1 Mobilenet V2 . . . . .	6
2.3.2 Resnet-50 . . . . .	7
2.3.3 Efficient Net . . . . .	7
2.4 Object Detection Models . . . . .	8
2.4.1 Detectron2 . . . . .	8
2.4.2 RTMDet using MMDetection . . . . .	9
2.4.3 YOLOv8 . . . . .	10
2.5 Evaluation metrics . . . . .	12
2.5.1 Intersection over Union(IoU) . . . . .	12
2.5.2 True Positive, False Positive, False negative and True Negative . . . . .	12
2.5.3 Precision, Recall . . . . .	13
2.5.4 Average Precision (AP) . . . . .	13
2.5.5 Accuracy . . . . .	13
2.6 Optimizers . . . . .	14
2.6.1 Stochastic Gradient Descent(SGD) Optimizer . . . . .	14
2.6.2 Adam Optimizer . . . . .	14
<b>3 Related Work</b> . . . . .	<b>17</b>
3.1 Medical Image segmentation . . . . .	17

3.1.1	Fully Convolutional Neural Networks . . . . .	18
3.1.2	U-Net . . . . .	18
3.1.3	Transformer . . . . .	19
<b>4</b>	<b>Dataset Acquisition and Preprocessing . . . . .</b>	<b>21</b>
4.1	Dataset acquisition . . . . .	21
4.2	Database search . . . . .	21
4.3	Dataset Selection . . . . .	22
4.3.1	Dataset Size . . . . .	23
4.3.2	Data Diversity . . . . .	23
4.3.3	Domain of the dataset . . . . .	23
4.4	Dataset overview . . . . .	23
4.4.1	Dataset Visualization . . . . .	25
<b>5</b>	<b>Experiment . . . . .</b>	<b>27</b>
5.1	Image Classification . . . . .	27
5.1.1	Binary Image Classification . . . . .	28
5.1.2	Eight class Image Classification . . . . .	29
5.2	Object Detection . . . . .	30
5.2.1	Object Detection for 8 classes . . . . .	31
5.2.2	Object Detection for 2 and 3 classes . . . . .	35
<b>6</b>	<b>Results and Discussion . . . . .</b>	<b>39</b>
6.1	Object Detection for 8 classes . . . . .	39
6.2	Object Detection for 2 and 3 classes . . . . .	44
<b>7</b>	<b>Conclusion . . . . .</b>	<b>47</b>
7.1	Further Work . . . . .	47
	<b>Bibliography . . . . .</b>	<b>49</b>
<b>A</b>	<b>Additional Material . . . . .</b>	<b>55</b>
A.1	Code to convert Vindr dataset into COCO dataset Format . . . . .	55
A.2	Visualization of input images . . . . .	57

# Figures

2.1	Image Classification [8] . . . . .	6
2.2	Object detection [8] . . . . .	6
2.3	Instance segmentation [8] . . . . .	6
2.4	Residual block [10] . . . . .	7
2.5	Inverted residual block [10] . . . . .	7
2.6	Building block of ResNet[12] . . . . .	7
2.7	Detectron2 architecture (Base RCNN-FPN)[14] . . . . .	9
2.8	RTM-Det architecture[15] . . . . .	9
2.9	YOLOv8 architecture[26] . . . . .	11
2.10	IoU[28] . . . . .	12
4.1	Class Distribution . . . . .	25
4.2	Annotated sample image from VinDr . . . . .	26
5.1	Class Distribution for all the classes except No findings . . . . .	31
5.2	Class Distribution for eight classes . . . . .	32
5.3	Class Accuracy for SGD Optimizer . . . . .	34
5.4	Total Loss for SGD Optimizer . . . . .	34
5.5	Class Accuracy for Adam Optimizer . . . . .	34
5.6	Total Loss for Adam Optimizer . . . . .	34
6.1	Ground truth 1 . . . . .	40
6.2	RTMDet Output 1 . . . . .	40
6.3	Ground truth 2 . . . . .	40
6.4	RTMDet Output 2 . . . . .	40
6.5	Ground truth 3 . . . . .	40
6.6	RTMDet Output 3 . . . . .	40
6.7	Ground truth 4 . . . . .	41
6.8	YOLOv8 Output 1 . . . . .	41
6.9	Ground truth 5 . . . . .	41
6.10	YOLOv8 Output 2 . . . . .	41
6.11	Ground truth 6 . . . . .	41
6.12	YOLOv8 Output 3 . . . . .	41
6.13	SGD optimizer ground truth and output 1 . . . . .	42

6.14	SGD optimizer ground truth and output 2 . . . . .	42
6.15	Adam optimizer ground truth and output 1 . . . . .	43
6.16	Adam optimizer ground truth and output 2 . . . . .	43
6.17	2 classes object detection output 1 . . . . .	44
6.18	2 classes object detection output 2 . . . . .	44
6.19	3 classes object detection output 1 . . . . .	45
6.20	3 classes object detection output 2 . . . . .	45
A.1	Respective classes . . . . .	57
A.2	Input data visualization 1 . . . . .	58
A.3	Input data visualization 2 . . . . .	59

# Tables

4.1	Chest X-rays dataset . . . . .	22
4.2	Data classes in VinDR . . . . .	24
5.1	Ablation study for binary image classification . . . . .	28
5.2	Dataset count per class . . . . .	29
5.3	Ablation study for 8 classes image classification . . . . .	30
5.4	Ablation study on different models . . . . .	32
5.5	8 classes object detection in Detectron 2 . . . . .	33
5.6	Classwise AP for 8 classes using SGD and Adam Optimizer . . . . .	33
5.7	Result for 3 classes vs 3 classes . . . . .	36
5.8	Classwise AP for 2 and 3 classes . . . . .	36



# Acronyms

**Adam** Adaptive Moment Estimation Optimizer. v, 14, 15

**AP** Average Precision. v, 13

**BCE** Binary Cross Entropy. 11

**BN** Batch Normalization. 10

**COCO** Common Objects in Context. 3

**CSP** Cross Stage Partial. 10, 11

**CT** Computed Tomography. 1

**DFL** Distribution Focal Loss. 11

**FN** False Negative. 12

**FP** False Positive. 12

**FPN** Feature Pyramid Network. 8

**IoU** Intersection over Union. v, 12

**mAP** Mean Average Precision. 13

**MRI** . 1, 17

**NMS** Non-Maximum Suppression. 12

**PAFPN** Path Aggregation Feature Pyramid Network. 10

**ROI** Region of Interest. 2, 3, 8

**RPN** Region Proposal Network. 8

**RTMDet** Real Time Models for Object Detection. 9, 10

**SGD** Stochastic gradient descent. v, 14

**TN** True Negative. 13

**TP** True Positive. 12, 13

**WHO** World Health Organization. 1

**YOLO** You Only Look Once. 10



# Chapter 1

## Introduction

### 1.1 Introduction and Problem statement

Machine learning, especially deep learning, has become an inseparable part of our everyday life. From voice search technology to self-driven cars, it has influenced every domain imaginable. Its impact on healthcare is remarkable as well. Some of the applications of deep learning in health care include drug development, disease prediction, medical imaging and diagnostics, personalized treatment and so on. These applications have revolutionized the health care domain of the world by making it more efficient and accessible to wider group of population.

Using deep learning to interpret medical images like X-ray, Magnetic resonance imaging (MRI) scan, or Computed Tomography (CT) scan to perform diagnosis of the patients can be extremely useful in the current scenarios where doctor to population ratio is extremely low. The recommended doctor to population ratio by World Health Organization (WHO) is 10 doctors per 10,000 population whereas the current ratio in many countries is less than 1 doctor per 10,000 population [1]. The nation's medical health care system is heavily burdened by the lack of doctors, especially in the event of a pandemic, which will seriously impair people's quality of life and disrupt the health care system. Leveraging the power of deep learning to alleviate the burden faced by the doctor in these countries can not only lead to providing better health care facilities but also results in improving the quality of healthcare services. Using deep learning can also enhance the outreach services provided by health workers in remote areas of the world.

Accurately diagnosing any deformities in the medical images depends on the experience and the knowledge of the medical professional. According to Delrue et. al. [2], the examiner has to have following skills to accurately diagnose the disease.

- Knowledge of Anatomy and Physiology
- Radiograph Analysis Using a Fixed Pattern
- Familiarity with the clinical presentation, history, and correlation to other diagnostic outcomes

Without a full understanding of physical and physiological concepts, medical imaging can be misread. Psychological interpretation aspects should be taken into consideration as a potential source of errors while interpreting chest X-rays. In this case, machine learning can be used to lessen the possibility of a disease being misdiagnosed. By using image segmentation, the area with defects can be highlighted, bringing attention to the problematic area and lowering the likelihood of a false diagnosis. For improved health-care services, improving diagnosis accuracy and reducing diagnosis time are both vital. Both of these tasks could be accomplished with the use of machine learning, which would ultimately assist in saving lives.

Extensive research [3], [4],[5], [6] has been conducted to identify the Region of Interest (ROI) in the medical domain. However, the application of ROI detection in the medical field is in the early stages of development. Further researches are required to advance its maturity, in order to make it suitable for practical implementation in real-world scenarios.

## **1.2 Justification, Motivation and Benefits**

Having accessible healthcare is a fundamental human right and in order to make it accessible for everyone, we need to maximize resource utilization. Empowering the healthcare sector with AI provides a solution to address existing gaps in the medical sector. Especially in resource-constrained settings, the use of AI allows patients to be diagnosed faster, allowing doctors to treat patients more effectively and allocate more time for subsequent cases. The early diagnosis not only speeds up the treatment but also serves as a way to prevent possible complications as a consequence of later diagnosis. The use of AI in medical imaging is proving incredibly valuable, guiding physicians on where to focus and what to look for in diagnostic testing. By identifying potential issues in medical images, AI simplifies analysis by focusing attention on specific areas of concern. This not only allows the time needed to properly assess the patient but also provides greater support for an accurate and timely diagnosis.

In essence, the use of AI in healthcare optimizes the use of available resources, helping to diagnose medical conditions more efficiently and accurately. This transformative technology helps ensure that individuals receive treatment in a timely manner. This ultimately helps more people get healthcare quickly and reduces the risk of adverse outcomes as a result of delayed diagnosis.

## **1.3 Research Questions**

The research intends to address three crucial research questions related to medical imagery. They are:

- How do state-of-the-art models perform in terms of accuracy when it comes to the classification of chest X-ray images and what conclusions can be made for their applicability in medical image classification tasks?

- How effective and accurate are existing machine learning models for determining the ROI in medical images?
- What is the impact of altering the number of classes on determining the performance of object detection models and how does it affect the accuracy of the model?

## 1.4 Contributions

The main contributions of this thesis are mentioned below:

- Curation and creation of diverse sets of dataset from Vindr chest X-rays images to ensure a well rounded representation of various abnormalities and medical conditions. These subsets of the dataset consist of COCO-format images with different class counts.
- Presents ablation studies of different object detection models when trained with different datasets containing varying numbers of image classes.
- Evaluation of the performance of different object detection models for determining ROI in chest X-rays. examined the effects of changing the number of classes on object detection models, offering information on how the size of the class affects model accuracy.

## 1.5 Paper Outline

The rest of the paper is structured as follows: In Chapter 2, the theoretical information required for the research is covered along with an overview of image classification and object detection. The next chapter, chapter 3, includes literature review that consists of the state-of-the-art works on medical image classification and segmentation. Chapter 4 contains the details of the process of the dataset creation. Next in chapter 5, different experiments that are carried out in this thesis are discussed along with the comparison of its performance. Chapter 6 consists of results obtained from different models in different experiments. The discussion of various observations made throughout this research work and their explanations are presented in this chapter. At last, Chapter 7 summarizes the conclusions as well as provides direction for further research.



## Chapter 2

# Background

This chapter will provide the user with the basic understanding of different concepts and knowledge related to the project. It starts by discussing different topics related to computer vision and then proceeds to the discussion on different models employed in this project and a variety of evaluation metrics used for image classification and object detection. Reading this chapter will provide the user with all the necessary information to get a better understanding of the knowledge used in the project.

### 2.1 Object Detection

Object detection is a computer vision technique for detecting instances of semantic objects that belongs to a certain class such as animals, plants, etc in digital images and videos [7]. The aim of object detection is to develop a computational model that predict a set of bounding box and labels each object of interest. Humans can quickly identify and locate objects of interest when viewing images or videos. The goal of object detection is to recreate this intelligence using computer.

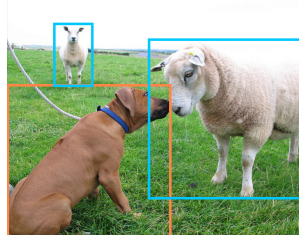
Despite the apparent similarity of various computer vision subfields, such as image classification, object detection, and image segmentation, each has its own goals and techniques.

### 2.2 Image Classification, Object Detection and Instance Segmentation

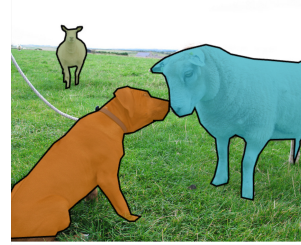
Image classification refers to the task of associating one or more labels or classes to a given image [9]. Single label classification deals with labeling the entire image with only one label or class whereas multi label classification deals with annotating the image with more than one label. If we look at the figure 2.1 we instantly classify the image as an image of sheep. This is exactly what image classification accomplishes - determine if the image falls under a specific class or not.



**Figure 2.1:** Image Classification [8]



**Figure 2.2:** Object detection [8]



**Figure 2.3:** Instance segmentation [8]

While image classification is capable of identifying the items in a given image, it is unable to determine their exact locations. That's when object detection comes into action. It is possible to determine the bounding box of the object using object detection. However, it offers no details regarding the object's shape. Instance segmentation can be utilized to determine the instances of the objects and marking their boundaries.

## 2.3 Image Classification Models

Models which are used during the image classification within the scope of this thesis are discussed in this section. We will examine the architecture or key features of each of the image classification models.

### 2.3.1 Mobilenet V2

Mobilenet V2 is a convolutional neural network architecture designed by Sandler et. al.[10] to build highly efficient mobile models for image classification and object detection. It is a memory-efficient predecessor over MobileNet[11] that aims to achieve better accuracy and lower the computational cost all while reducing the model size. It introduced the concept of inverted residuals, where a lightweight depthwise separable convolutions is followed by a linear bottleneck. It uses depthwise separable convolution of size 3X3 to decrease the computational cost by 8 to 9 times.

Figures 2.4 and 2.5 shows the difference between residual block and inverted residual block. The thickness of each block represents its proportionate number of channels. We can see that classical residuals links layers with a high number of channels, while inverted residuals connect the bottlenecks. By adopting an inverted residual block, the model's efficiency is increased and the computational load is reduced due to improved knowledge flow across layers.

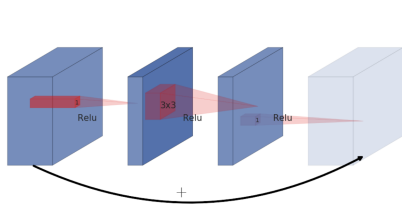


Figure 2.4: Residual block [10]

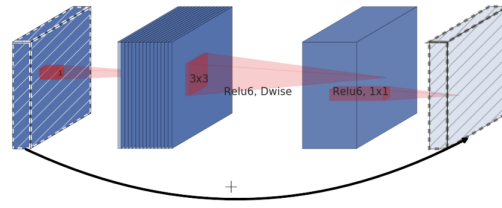


Figure 2.5: Inverted residual block [10]

### 2.3.2 Resnet-50

ResNet-50[12] is a deep convolutional neural network that is 50 layers deep. The one problem that is faced when increasing the depth of a neural network is that it leads to the degradation in performance caused by the training difficulties. Hence to overcome the degradation problem ResNet-50 was introduced. The main innovation in ResNet-50 is the residual blocks which consist of skip connections that can skip one or more layers. The skip connections allows the gradient to flow more easily during backpropagation hence diminishing the vanishing gradient problem.

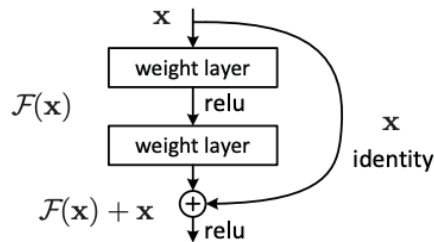


Figure 2.6: Building block of ResNet[12]

The building block for residual learning is represented as:

$$y = F(x, \{W_i\}) + x$$

Here,  $x$  is the input vector,  $y$  is the output vector and  $F(x, W_i)$  is the residual mapping to be learned. The output of the block in figure 2.6 is represented as  $F(x)+x$ . Stating the block's output as  $F(x)+x$ , rather than solely  $F(x)$ , enables the network to skip the contributions of a convolutional block if it doesn't contribute additional information to the network.

### 2.3.3 Efficient Net

In 2019 Tan et. al.[13], proposed a solution called compound scaling to tackle the challenge of finding the balanced network width, depth and resolution that has better accuracy and efficiency instead of manual tuning these parameters.

Compound scaling suggests that the depth, width and the resolution of should be scaled in a constant way which is represented as follows:

$$\begin{aligned}
 \text{Depth: } d &= \alpha^\phi \\
 \text{Width: } w &= \beta^\phi \\
 \text{Resolution: } r &= \gamma^\phi \\
 \text{s.t. } \alpha \cdot \beta^2 \cdot \gamma^2 &\approx 2, \\
 \alpha &\geq 1, \quad \beta \geq 1, \quad \gamma \geq 1
 \end{aligned}$$

Through the experiment, it was found that the best values of  $\alpha=1.2$ ,  $\beta=1.1$  and  $\gamma=1.15$ , considering a certain condition. Using these values as constants and scaling up the baseline network of EfficientNet-B0 with different value of  $\phi$  using the equations above, it was possible to obtain the improved EfficientNets.

## 2.4 Object Detection Models

In this section, we will take a look at the various models employed for object detection within the scope of this project. We will also have a general overview regarding the architecture and the key characteristics of the selected models used for object detection.

### 2.4.1 Detectron2

Detectron2[14] is the state-of-the-art next generation library developed by Facebook AI Research for object detection and object segmentation. It features a modular architecture that allows users to effortlessly customize and expand the functionality of the library. Detectron2 includes implementation of state-of-the-art object detection algorithms such as Faster R-CNN, Mask R-CNN and RetinaNet. It is renowned for its high performance in terms of both speed and accuracy.

Figure 2.7 represents the architecture of the Detectron2 model. It primarily comprises of three main blocks. The initial block, known as the backbone network, uses Feature Pyramid Network (FPN) with Resnet50 for FPN implementation. The role of this block is to extract feature maps from the input image. The output of this block is multi-scale feature maps which are then utilized as an input for the subsequent block. The second block of the architecture is called Region Proposal Network (RPN). Besides, the feature maps, the ground truth of the data is also fed into this block. The output of the block is proposal boxes. The third and final block is the Region of Interest (Box) Head, which receives feature maps from FPN, proposal boxes from RPN, and ground truth boxes. Utilizing proposal boxes for cropping, the Box Head processes ROI from the feature maps using ROIAlignV2, a precise pooling method for floating-point coordinate proposal boxes. The classification loss employs Softmax cross-entropy, while the localization loss uses L1



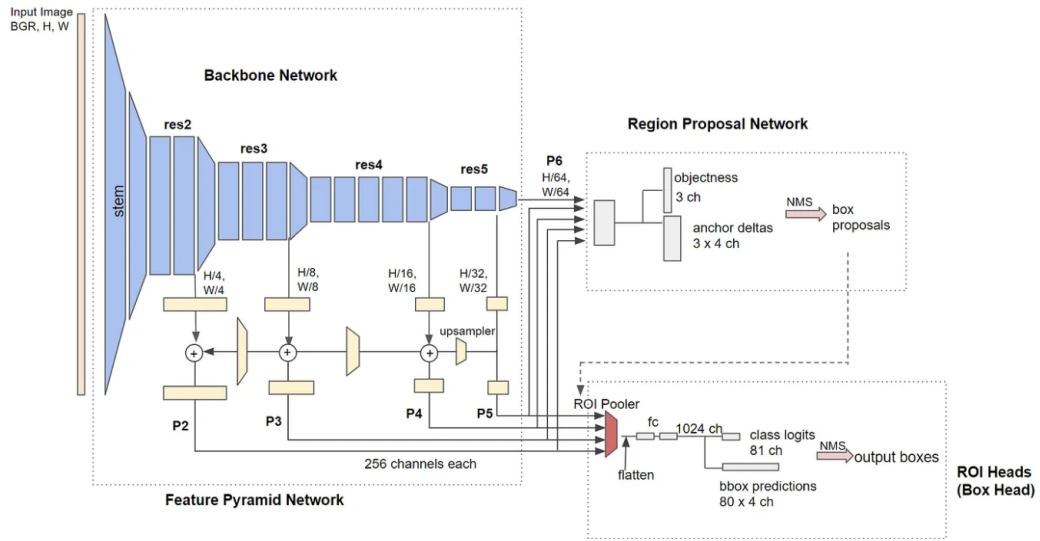


Figure 2.7: Detectron2 architecture (Base RCNN-FPN)[14]

loss. During testing, inference occurs by applying prediction deltas to proposal boxes, filtering them by score, performing non-maximum suppression to eliminate overlapping boxes, and ultimately selecting the top-k results.

### 2.4.2 RTMDet using MMDetection

RTMDet[15], an efficient Real-Time one stage detector, is used for detecting objects. An open source detection toolbox built by OpenMMLab called MMDetection[16] is used for utilizing the implementation of Real Time Models for Object Detection (RTMDet). Figure 2.8 illustrates the macro architecture of the RTMDet which consists of three components: the backbone, the neck, and the head.

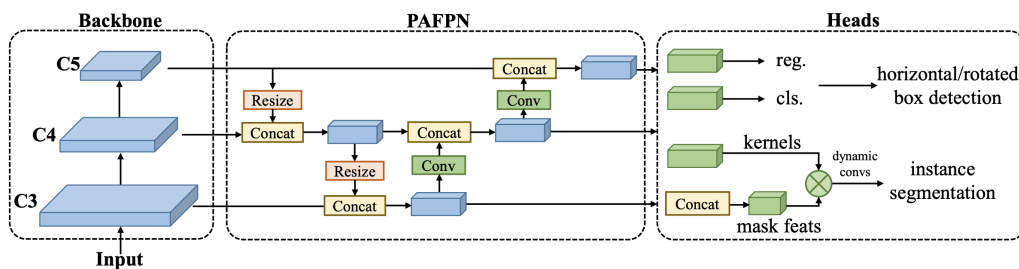


Figure 2.8: RTM-Det architecture[15]

1. **Backbone:** The backbone is responsible for extracting input image features.

Cross Stage Partial (CSP) blocks with large-kernel depth-wise convolution layers are used to build the backbone. CSP-blocks refer to Cross Stage Partial networks which are used to improve feature representation. The backbone of the model extracts multi-level features which are denoted as C3, C4 and C5.

2. **Neck(Path Aggregation Feature Pyramid Network (PAFPN)):** The output of the backbone, which is multi-scale features are passed into the neck and enhances them using a technique called PAFPN. To enhance the feature pyramid, the PAFPN block uses both top-down and bottom-up feature propagation techniques. The function of the neck is to guarantee that the model can recognise objects at different scales with accuracy, which helps with multi-scale feature representation.
3. **Head:** Using the feature maps derived from each scale, the detection head is responsible for predicting item bounding boxes and the categories which relate to them. RTMDet uses separate Batch Normalisation (BN) layers for every scale, while it uses shared convolution weights for the detection head across scales. This maintains precision while enabling parameter sharing between scales.

A strategy for dynamic soft label assignment, which is based on SimOTA [17], has been introduced, and the corresponding cost function can be presented as follows:

$$C = \lambda_1 C_{\text{cls}} + \lambda_2 C_{\text{reg}} + \lambda_3 C_{\text{center}}$$

Here  $C_{\text{cls}}$ ,  $C_{\text{center}}$  and  $C_{\text{reg}}$  represents classification cost, region prior cost and regression cost respectively.  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$  are the default weights of these three costs.

$$C_{\text{cls}} = \text{CE}(P, Y_{\text{soft}}) \times (Y_{\text{soft}} - P)^2$$

Earlier methods often use binary labels to calculate classification cost. This method enables a prediction with a high classification score but an inaccurate bounding box to produce a low classification cost, and vice versa. To address this challenge, RTMDet proposes incorporating soft labels into  $C_{\text{cls}}$  as shown in the above equation. In conclusion, RTMDet showcases a superior balance between accuracy and speed for object recognition tasks of different model sizes.

### 2.4.3 YOLOv8

Developed in 2015 at the University of Washington, You Only Look Once (YOLO)[18] has become a widely recognized model for object detection and image segmentation. It has been recognized for its exceptional speed and accuracy. Over the years, its popularity has led to continuous enhancements, striving to improve

performance and efficiency. Multiple iterations have been introduced, including YOLOv2[19], YOLOv3[20], YOLOv4[21], YOLOv5[22], YOLOv6[23], YOLOv7[24], and the most recent, YOLOv8[25].

The company that created YOLOv8, Ultralytics, has expanded on the achievements of its predecessors. YOLOv8 incorporates advancements aimed at enhancing overall performance, flexibility, and efficiency. This most recent version can handle a wide range of AI vision tasks with ease, including tracking, segmentation, pose estimation, detection, and image classification.

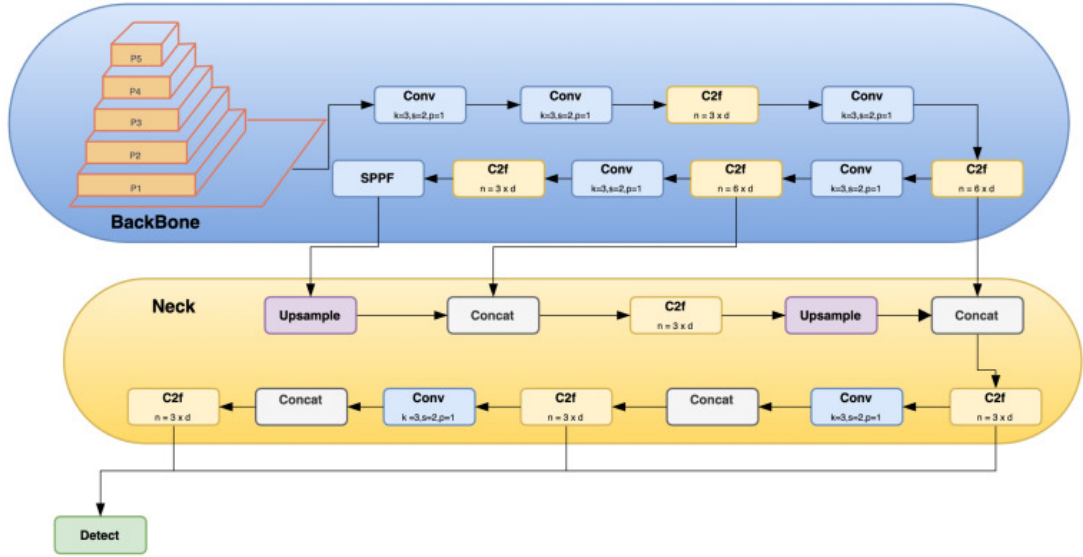


Figure 2.9: YOLOv8 architecture[26]

The YOLOv8 architecture comprises three main components: the backbone, neck, and head. The backbone generates a feature pyramid facilitating multi-scale object detection. YOLOv8 adopts the C2F model, based on CSP, in contrast to the YOLOv5's C3 module. The C2F block enhances CNN learning capacity while reducing computational demands. YOLOv8 introduces a decoupling structure in the head, deviating from the original coupling structure in YOLOv5.

YOLOv8 employed Binary Cross Entropy (BCE) loss for classification, while utilizing Ciou loss and Distribution Focal Loss (DFL) for regression[27]. The computation of DFL values was determined by the expression presented in equation 2.4.3.

$$\text{DFL}(s_i, s_{i+1}) = -((y_{i+1} - y) \log(s_i) + (y - y_i) \log(s_{i+1}))$$

The output  $s_i$  represents the sigmoid result for the network,  $y_i$  and  $y_{i+1}$  denote interval orders, and  $y$  is a label.

$$t = s^\alpha \times u^\beta$$

Notably, YOLOv8 is an anchor-free model, directly predicting object centers rather than anchor boxes. This results in fewer box predictions, leading to faster Non-Maximum Suppression (NMS). The alignment degree of Anchor level for each instance is calculated using an equation 2.4.3 where  $s$  represents the classification score,  $u$  is the IoU value, and  $\alpha$  and  $\beta$  are the weight hyperparameters. Negative samples include the remaining anchors, but positive samples include  $m$  anchors with the greatest value ( $t$ ) in each instance.

## 2.5 Evaluation metrics

### 2.5.1 Intersection over Union(IoU)

IoU is the metrics used to evaluate the overlap between two bounding boxes which are the bounding box of a ground truth and the bounding box of a predicted box. With the help of IoU we can tell if an detected object is True Positive or False Positive. The IoU can have values between 0 and 1. Having IoU value 0 indicates that the two boxes do not intersect and having IoU value 1 indicates the two boxes completely overlap.

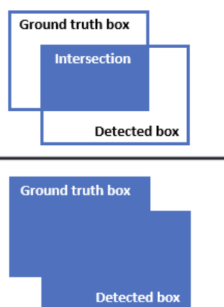
$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} = \frac{\text{Intersection}}{\text{Ground truth box} \cup \text{Detected box}}$$


Figure 2.10: IoU[28]

### 2.5.2 True Positive, False Positive, False negative and True Negative

- **True Positive (TP):** It is a case when a model predicts that a bounding box exists at a certain position (true) and it is actually correct (true).
- **False Positive (FP):** It is a case when a model predicts that a bounding box exists at a certain position (true) but it is wrong (false).
- **False Negative (FN):** It is a case when the model does not predict the existence of a bounding box at a certain position (negative) and it is wrong (false).

- **True Negative (TN):** It is the case when the model does not predict the existence of a bounding box (negative) and it is correct (true).

### 2.5.3 Precision, Recall

- **Precision:** It is the ratio of TP predictions to the total predicted positives. It gives the indication of how precise our model is.

$$\text{Precision} = \frac{\text{CorrectPredictions}}{\text{TotalPredictions}} = \frac{TP}{TP + FP}$$

- **Recall:** It is the ratio of TP prediction to the total actual positives. It gives the indication of how good the model is at recalling classes from images.

$$\text{Recall} = \frac{\text{CorrectPredictions}}{\text{TotalGroundTruth}} = \frac{TP}{TP + FN}$$

### 2.5.4 Average Precision (AP)

It is the one of the most popular metrics used for evaluating object detection models. From AP we can derive other metrics such as Mean Average Precision mAP, AP50, AP75 and AP[.5:.5:.95]. The AP is determined by computing the area under the Precision-Recall curve (AUC-PR). In practical applications, the AP is computed as an average across all recall values within the range of 0 to 1. The value of AP is calculated as follows:

$$\text{AP} = \frac{1}{R} \sum_{k=1}^R (P(k) \cdot \text{rel}(k)),$$

In the above equation,  $R$  is the number of recall values,  $P(k)$  denotes the precision at  $k$ -th recall, and  $\text{rel}(k)$  is an indicator function that is equal to 1 if the  $k$ -th retrieved item is a relevant item and 0 otherwise.

### 2.5.5 Accuracy

It is the ratio of number of correctly predicted bounding boxes to total number of bounding boxes which are presented as a percentage. The degree to which a model accurately locates and detects items in an image is measured by the object detection accuracy. The formula for calculation accuracy is presented below:

$$\text{Accuracy} = \frac{\text{Number of Correctly Predicted Bounding Boxes}}{\text{Total Number of Bounding Boxes}} \times 100\%$$

## 2.6 Optimizers

Optimizers are the algorithms that minimize the loss function by updating the parameter of the models. It helps to improve the performance of the deep learning models by updating its weight and biases in such a way that the model's representation gets closely aligned with the characteristics of the training data. The accuracy and performance of various models are impacted by different optimizers. While there isn't a universally applicable method for selecting the optimal optimizers for a deep learning model, experimenting with several optimizers and evaluating the outcomes can assist in determining which optimizer is best suited for a certain task.

### 2.6.1 Stochastic Gradient Descent(SGD) Optimizer

The term stochastic means "randomness", which is exactly what the optimizer is based upon. In SGD, random mini batches of training data are selected instead of the entire dataset, in order to improve the model's performance. In comparison with gradient descent, SGD needs more iterations to attain local minima. The computational expense is still lower than that of gradient descent, though. The formula for calculating SGD is given below.

$$\theta_{t+1} = \theta_t - \eta \nabla J(\theta_t, x_i, y_i)$$

Here,  $\theta_t$  represents the parameters of the model at  $t$  iteration,  $\eta$  is the learning rate and  $\nabla J(\theta_t, x_i, y_i)$  represents the gradient of the loss function  $J$  where  $x_i, y_i$  is mini-batch of data.

SGD iteratively adjusts the model parameter based on the gradient computed from the formula. As a result, the parameters converge and eventually reach the lowest loss across the training data set.

### 2.6.2 Adam Optimizer

ADAM stands for Adaptive Moment Estimation Optimizer. It is able to adaptively adjust the learning rate for each network weight individually. It combines the ideas from RMSprop and Momentum, which are the two other optimization algorithms. Adam has two moving averages for each parameter of the model: a moving average of the gradients(Momentum) and of the squared gradients(RMSprop).

$$\begin{aligned}
m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \\
v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \\
\hat{m}_t &= \frac{m_t}{1 - \beta_1^t} \\
\hat{v}_t &= \frac{v_t}{1 - \beta_2^t} \\
\theta_{t+1} &= \theta_t - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t,
\end{aligned} \tag{2.1}$$

The updated formula for Adam optimizer[29] relies on the first moment estimate and second moment estimate  $m_t$  and  $v_t$ . At time  $t$ , the gradient is denoted by  $g_t$ ,  $\theta_t$  represents model parameters,  $\hat{m}_t$  and  $\hat{v}_t$  are bias correctness estimate.  $\alpha$  is the learning rate and  $\beta_1$  and  $\beta_2$  are the decay rates. A small number  $\epsilon$  is included to prevent the issue of division by zero.





## Chapter 3

# Related Work

Deep learning networks have become the fundamental part of today's artificial intelligence system [30]. There are various types of deep learning networks, each designed for a specific task and purpose. For example, Convolutional Neural Networks(CNN) are used for image classification, object detection and other computer vision tasks. Recurrent Neural Networks are used for sequence modeling, natural language process, speech recognition and time series analysis. Transformer networks were first applied by Vaswani et al [31] to natural language processing tasks. Initial applications of transformer included machine translation, but they were later expanded to incorporate text generation [32], sentiment analysis [33], and question-answering [33].

### 3.1 Medical Image segmentation

Medical image segmentation is the process of separating organs or lesions from the background of images such as X-rays, MRIs, and so on. Hesamian et al. [6] have stated that it is one of the most difficult tasks in medical image analysis. Its usage can significantly reduce doctors' workload by assisting them in quantifying the effects of treatment and by verifying the size of diseased tumors [6]. It is a vital component of medical image analysis which also serves as the foundation for various clinical applications.

Early methods of medical image segmentation were based on classical image processing techniques such as the edge detection, thresholding and active contour[34]. These methods however, have their own limitations when it comes to handling varying image quality and complex anatomical structures.

Medical image segmentation has undergone a revolution as a result of the development of deep learning techniques, particularly convolutional neural networks (CNNs). Various ground breaking deep learning based techniques have been developed so far and have achieved state of the art in the field of medical image segmentation. Some of them are discussed below.

### 3.1.1 Fully Convolutional Neural Networks

CNN serves as the standard network model for computer vision. The efficient and cutting-edge deep learning system for semantic segmentation started from FCN [35]. But with the introduction of AlexNet[36], CNN truly took off and became mainstream. Since then, a large number of powerful and efficient convolutional neural networks have been proposed and have been able to obtain state-of-the-art performance for image classification, segmentation and detection. VGG[37], GoogleNet[38], ResNet[12], DenseNet[39] and EfficientNet[13] are a few of them. One of the main issues with these models, in the context of medical imaging, is that these models were trained using a big dataset, which is not always feasible when dealing with medical images like chest X-Rays due to the limited quantity of the accessible data[rpp].

To address the problem along with the other specific challenges such as reduced feature resolution, existence of objects at multiple scales and reduced localisation accuracy compactness which were not addressed by previous models like Imagenet and VGG, DeepLab[40] was introduced. DeepLab was able to improve the performance of accurately segmenting objects of various sizes. Different iterations of DeepLab which are DeepLabV1[40], DeepLabV2[41], DeepLabv3[42], DeepLabV3+[43], DeepLabV4[44], DeepLabV5 [45] and DeepLabV6[46] were able to achieve state-of-the-art results in semantic image segmentation tasks while also being more efficient in computation and memory requirements.

### 3.1.2 U-Net

Ronneberger et al. [47] designed U-net for biomedical image segmentation, which has been extensively used in medical image segmentation. Many variants of the U-net such as Residual and Attention U-Net along with encoder-decoder architectures such as ResNet and DenseNet have been able to achieve state of the art. All these architectures share a key similarity which is the presence of skip connections. Skip connection along with the attention block allows the model to provide focus on the key semantic features and dependencies. This inturn helps in the detection of the finer details.

All these aforementioned models have undoubtedly achieved state of the art in medical image segmentation but most of the work has been focused on MRI, positron emission tomography (PET) and X-ray scan images of brain, breast, liver and chest for anomaly and cancer detection. They have outperformed almost all previously known deep-learning models for lesion segmentation but it still has several limitations. One of the limitations is that it cannot accurately predict small objects and display anatomical characters in regions of interest in the predicted images. This limit should be taken into account in chest xray segmentation because the size of the abnormalities present in the image can vary dramatically depending on the age of the person [48].

To overcome the limitation of U-Net architecture and the presence of a generalized model in the domain of chest X-ray, Pal et al [48] proposed a modified

version of the U-Net model called UW net. Attention gates were implemented in the UW net model to improve the prediction accuracy of small lesion segmentation.

### 3.1.3 Transformer

Vaswani et al.[31] first proposed the idea that transformers may be utilized for tasks other than natural language processing. The idea that it can also be used for other tasks has resulted in the development of many new versions of transformers. In computer vision alone, it has been used for object detection [49], image restoration [50], medical image segmentation [51] and many more [52].

In 2020 Dosovitskiy et al. [53] presented a novel approach of applying a transformer model to image recognition, called as Vision Transformer, demonstrating excellent results in comparison to the state-of-the-art CNN while using fewer computational resources. The success of Vision Transformer was followed by several variants based on the transformer architecture. Both the medical[54][55][56] and non-medical [57][58][59] fields have shown success with its variation.

In 2021, Liu et al. [54] created the Swin transformer, a revolutionary architecture for computer vision tasks that demonstrated excellent performance in instance segmentation and semantic image segmentation. It uses a hierarchical approach that enables the model to capture both local and global information efficiently and also shifted windows which in turns reduces the computational complexity. In another study conducted by Ma et al.[60], Swin transformer has been shown to perform better than vision transformers in object detection and medical image segmentation. Furthermore, the pre-trained Swin transformer model on ImageNet was able to outperform CNN based models such as Resnet-50 in terms of the computer vision tasks.

In-depth studies on five different transformer-based models for identifying and segmenting lung areas were conducted by Ghali et al. [61]. The five models were ARSeg[62], TransM, Medical Transformer(MedT)[4], TransUNet[3] and UNeXt[5]. Upon evaluating the performance of the models using two loss functions (Dice loss and Combo loss) and using two evaluation metrics(F1 score and accuracy), it showed that all five of the models were able to achieve good performance in segmenting lung areas. But based on the F1 score, TransM was able to achieve the best score among all the other models. It was due to its ability in extracting rich feature maps using global and local branches. It was also able to outperform the U-Net model. All these five models were able to separate lungs irrespective of the varying lung shape which depended on age and gender showing great potential in future for medical image segmentation.

Besides having the aforementioned approaches of detecting objects–FCNN, U-Net and Transformer based approaches, there are many other models which follow their own unique approach for object detection. Yolo[18] for instance uses object detection as a single regression problem. It uses a single convolutional network which simultaneously predicts numerous bounding boxes and potential class

for those boxes. This unified approach of detecting objects makes the model extremely fast, with the ability to incorporate contextual information and has the generalization capacity.

Another model deviating from the aforementioned approaches is Detectron 2. Despite being primarily designed on R-CNN-based architectures, it is a modular and flexible framework that supports a number of backbone architectures such as ResNet[12], ResNetXt[63], and many other CNN variants. By default, Detectron2 includes the implementation of both Faster R-CNN [64] and Mask R-CNN [65]. The primary difference between Faster R-CNN and Mask R-CNN is that Faster R-CNN is used for predicting the coordinates of the bounding box and class probabilities whereas Mask R-CNN is used for predicting the segmentation masks for each detected object.

Faster R-CNN which was proposed by Ren et. al. in 2015 [64], comprises of two essential components: a Region Proposal Network(RPN) and a Fast R-CNN network. The RPN is responsible for generating proposals for regions that may contain objects within the images. The Fast R-CNN network which was proposed by Girshick et. al. [66] , takes the images and proposes candidate regions, then these candidate regions are passed through a popular pre-trained image classification model such as ResNet, VGG-16 to extract features. Once the features are extracted, it undergoes a Region of Interest(RoI) following layer which is then followed by label classification. Faster R-CNN is one of the best ways for object detection based on R-CNN series[67].

## Chapter 4

# Dataset Acquisition and Preprocessing

This chapter focuses on all the crucial elements of acquiring the dataset to identify deformities in chest x-rays. A diverse and quality dataset is essential for the purpose of training robust and accurate deep learning models for medical imaging. Since the success of deep learning models are heavily dependent on the dataset, lots of considerations were taken into account while selecting the chest X ray dataset for training. All those considerations along with the various challenges faced during the dataset acquisition are discussed in this chapter.

### 4.1 Dataset acquisition

In a real world scenario, obtaining dataset for medical imaging can involve sourcing data from various channels, including hospital records, public databases, or from various private collections through collaborations with healthcare institutes or organizations. However, this master thesis has an extremely specific focus, so it was decided to make use of an already existing, publicly accessible dataset that has already been annotated. In order to obtain the dataset that was ultimately chosen, the following procedures were taken.

### 4.2 Database search

To compile a list of all the publicly accessible datasets for chest X-rays, a thorough search of online sources was conducted. In order to find the dataset, relevant keywords such as “Chest X-rays”, “chest medical images”, “pulmonary images”, etc were used in well-known search engines and specialized dataset repositories. Some of them include Kaggle datasets, Google dataset search and publicly available domain-specific repositories. Table 4.1 presents the list of collected datasets.

Name	Number of Images	Image Size	Dataset Descriptions
JSRT	247	2048×2048	Annotated images for nodules
Montgomery	138	4020×4892×12	Tuberculosis diagnosis
NIH Chest-Xray Dataset	112,120	1024×1024	Limited images annotated with multiple thoracic diseases
Shenzhen	652	-	Tuberculosis diagnosis
Candid-PTX	19,237	1024×1024	Annotation available for pneumothoraces, acute rib fractures, and intercostal chest tubes
CheXpert	224,316	-	Multiple Diseases with detailed annotations including uncertainty labels; No bounding box
Vindr-CXR	18,000	Variable	14 types of thoracic abnormalities; each finding is localized with a bounding box; No radiologist report
Indiana State University	7,470	Variable	15 different abnormalities with bounding box; Contains radiologist report
MIMIC-CXR V2.0.0	377,110	Variable	Have annotations which describes the presence or absence of abnormalities
RSNA Pneumonia Detection Challenge	30,000	Variable	Labelled to indicated presence or absence of pneumonia

Table 4.1: Chest X-rays dataset

### 4.3 Dataset Selection

After enumerating the readily available chest X-ray datasets that are suitable for detecting abnormalities, the subsequent step is to filter out the dataset that is not required and select one dataset for training. During the meticulous process of selecting the dataset for training, various factors were taken into account. Among these considerations were:

### 4.3.1 Dataset Size

The size of the dataset is very crucial for training the deep learning model with larger datasets leading to better model performance and generalization. Therefore, datasets with less than one thousand images were excluded from consideration.

### 4.3.2 Data Diversity

Diverse dataset is essential to make sure that the deep learning model generalizes well across a range of diseases and diverse population. As a result, datasets focused exclusively on a single specific disease are excluded during this phase.

### 4.3.3 Domain of the dataset

A training dataset should not only include images and the corresponding labels but it should also incorporate bounding boxes along with the coordinates that specifies the problem of the image or the target area. Consequently, the datasets which do not contain annotations with bounding are removed. This step resulted in the exclusion of most of the datasets that were initially listed.

After going through all the dataset filtration process, it became evident that the Vindr Chest-Xray dataset aligns most closely with all the essential criteria mandated by the project. As a result of this, it was selected as the dataset for the project.

## 4.4 Dataset overview

Vindr-CXR is an open dataset of chest X-rays which are manually annotated by a group of 17 radiologists with at least 8 years of experience. The dataset is divided into two sets: 15,000 images of training set and 3,000 images of testing set. This dataset is made public by Vingroup Big Data Institute(VingBigData). All the images are in DICOM format and belong to one of the following 15 classes.

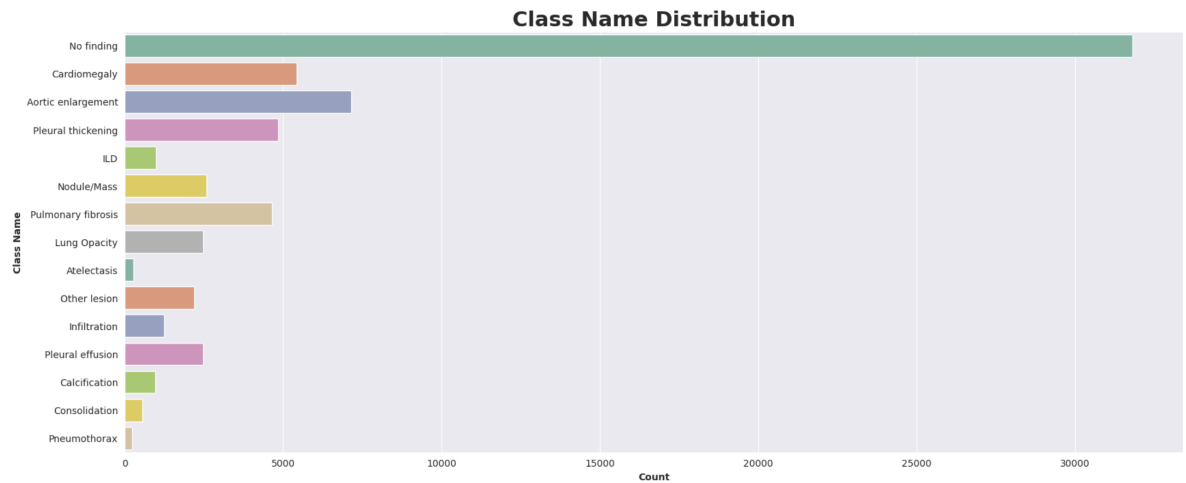
Class ID	Findings
0	Aortic enlargement
1	Atelectasis
2	Calcification
3	Cardiomegaly
4	Consolidation
5	ILD
6	Infiltration
7	Lung Opacity
8	Nodule/Mass
9	Other lesion
10	Pleural effusion
11	Pleural thickening
12	Pneumothorax
13	Pulmonary fibrosis
14	No finding

**Table 4.2:** Data classes in VinDR

In addition to the images, there is a file named “train.csv” that includes the metadata associated with the images. The file provides the following information:

- `image_id`: A unique identifier for the image
- `class_name`: The name of the class or the deformities found. In case of absence of deformities, it is labeled as "no findings"
- `class_id`: the ID of the identified object's class
- `rad_id`: a radiologist's ID who made the observation
- `x_min`: Minimum X coordinate of the bounding box of the detected object
- `y_min`: Minimum Y coordinate of the bounding box of the detected object
- `x_max`: Maximum X coordinate of the bounding box of the detected object
- `y_max`: Maximum Y coordinate of the bounding box of the detected object





**Figure 4.1:** Class Distribution

The dataset “train.csv” represents a considerable amount of information, with a total of 67,900 rows of data. This indicates that each medical image within the dataset consists of approximately 4.5 findings. When analyzing the distribution of image classes, an interesting finding emerges: 31,818 rows are labeled as “No finding”. This category represents instances where no abnormal medical conditions are detected in the image. This class accounts for a substantial 47% of the total row count in the “train.csv” file.

Among the various abnormalities present in the dataset, Aortic enlargement has the maximum number of data with a total of 7,162 instances in the dataset. This number is closely followed by Cardiomegaly and Pleural thickening with the respective counts of 5,427 and 4,842 instances. On the other hand, there are abnormalities with considerably fewer instances in the dataset. Pneumothorax has the least amount of occurrence with only 226 instances. Similarly, other abnormalities such as Atelectasis and Consolidation also have relatively small numbers with 279 and 556 instances respectively.

#### 4.4.1 Dataset Visualization

The images presented in 4.2 depict a subset of four random images selected from VinDr dataset. The images are plotted with regions of interest and the disease which is diagnosed for each region of interest is also plotted on the images. The information regarding the region of interest and the corresponding disease is obtained from the train.csv file. It is evident that the number of abnormalities present on the images vary from image to image. Some X-ray images exhibit only a couple of issues while others consist of more than four issues within a single image. This diversity highlights the intricate nature of medical imaging, where the condition of the presence of defects can range from straightforward to highly complex solutions.

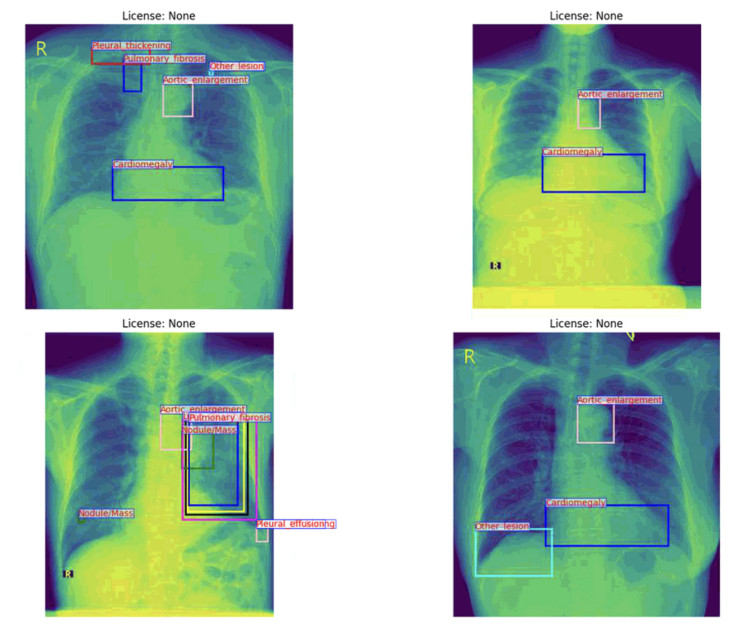


Figure 4.2: Annotated sample image from VinDr

Furthermore, the images also show a number of possibilities regarding the overlap between ROIs. In some cases, there is minimum to no overlapping between the regions of interest within a single image. However, in other images, a substantial portion of a ROI is overlapped by another ROI. Even though the abnormalities are completely different, overlapping between the ROI is observed. This introduces a layer of complexity as it becomes more difficult for machines to correctly identify between these defects.

As a result of these complexities, it poses a considerable challenge, especially while training the model. The presence of varying number of deformities, coupled with overlapping the ROI can impede the model's ability to generalize effectively.

## Chapter 5

# Experiment

In this chapter, you will find details about all the experiments conducted during this period, along with their respective results and conclusions. The process for experiment is thoroughly explained, covering the methods and results. The aim for this chapter is to provide a clear picture of what was done, what was found, and what the outcomes imply. This chapter serves as a thorough record of the research activities, providing insights and data in a comprehensible and straightforward way.

### 5.1 Image Classification

As mentioned in the dataset overview, the vindr dataset exhibits a certain degree of complexity. Image classification is chosen as the initial step for the research to effectively tackle this complexity. The motivation for performing image classification are listed below:

1. It helps in contextual understanding of the dataset, which facilitates in interpreting the image and provides direction for the subsequent segmentation efforts.
2. Image classification is computationally less expensive than image segmentation. Hence by performing image classification first, we can narrow down the focus of the project to only a handful of classes.
3. Additionally, image classification helps in filtering out the irrelevant data. When we examine the classification accuracy for each class, it provides a convincing argument for eliminating data that might not significantly contribute to the research.
4. Gradually increasing the level of complexity of the research in order to develop the expertise and improve methods.

### 5.1.1 Binary Image Classification

Binary image classification is carried out before carrying out multi class classification. It is because it is much simpler than multi class classification and understanding the output from this can help in getting a better understanding of what should be done in the future. The performance of binary image classification can also serve as the benchmark performance when the project moves to the multi class classification and what expectations we can have when the project reaches a more complicated step. In order to carry out binary image classification, the dataset has been meticulously separated into two distinct classes, which are images which consist of 1 or more abnormalities and images that do not consist of any abnormalities. This dataset serves as the backbone for carrying out the binary image classification which strives to differentiate between two different classes. Once the dataset has been prepared, three cutting-edge models were selected in the domain of image classification. The models chosen for this task are MobileNet-v2, Resnet-50 and EfficientNet. The reason behind selecting these 3 models was, during the course of the experiment, the performance of the models seemed to be better than other models and also due to their simple implementation. These models were trained for 50 iterations.

Model	Acc	Loss	Val acc	Val loss	Test acc	Test loss	Recall 0.90
Mobilenet v2	0.8622	0.2188	0.9093	0.04247	0.8786	0.4768	0.3943
Resnet-50	0.9175	0.1779	0.8754	1.0263	0.8555	1.911	0.4985
EfficientNet	0.5020	0.6931	0.5014	0.6931	0.5312	0.6928	0

**Table 5.1:** Ablation study for binary image classification

Overall, the performance of the binary image classification models was quite satisfactory. Among the three selected models, EfficientNet displayed the lowest accuracy, which was only 0.502. In comparison, MobileNet V2 obtained an accuracy of 0.8622 and a loss of 0.2188, outperforming EfficientNet. ResNet-50, on the other hand, produced the most astounding outcomes, having an accuracy of 0.9175 and a loss of 0.177.

Additionally, ResNet-50 exhibited better training accuracy and loss than MobileNet V2, but exhibited lower accuracy and loss during testing and validation, which may indicate overfitting. This is caused by the fact that ResNet-50 is inherently more sophisticated than MobileNet V2, which is comparatively lighter. Because ResNet-50 is more complicated, there are more parameters to learn, which means that a larger dataset is needed to get the best results. On the other hand, MobileNet V2 is more appropriate for situations with smaller datasets due to its lightweight design.

### 5.1.2 Eight class Image Classification

Following the completion of binary image classification, eight classes with the highest number of data have been selected to carry out image classification. The primary reason for carrying out eight class classification is to evaluate the model's ability to learn and understand the subtle complexities associated with different diseases that are presented in the images. Since all the defects are located in such a close proximity, conducting multi class classification becomes crucial. The underlying rationale behind incorporating a broader range of data is that by adding so the model is exposed to more complex features and patterns associated with diverse diseases. Achieving a good result through this process affirms that the dataset contains the depth and diversity necessary to train a model for sophisticated tasks such as object detection.

Table 5.2 illustrates the distribution of dataset for 8 class image classification. From the table, it is evident that the Nodule/Mass has the lowest data count at 826, while Aortic enlargement exhibits the highest data count of 3067. Following the meticulous preparation of the dataset for image classification, the dataset undergoes training for image classification. This database is also trained on three models which are used for training binary image classifications which are Mobile-net v2, Resnet-50 and EfficientNet. All these models were trained for 50 iterations. The outcome of the training process is summarized and presented in the Table 5.3

Class ID	Findings	Count
0	Aortic enlargement	3067
3	Cardiomegaly	2300
7	Lung Opacity	1322
8	Nodule/Mass	826
9	Other lesion	1134
10	Pleural effusion	1032
11	Pleural thickening	1981
13	Pulmonary fibrosis	1617

**Table 5.2:** Dataset count per class

Model	Acc	Loss	Val acc	Val loss	Test acc	Test loss
Mobilenet v2	0.2927	1.7319	0.3079	1.7490	0.3208	1.7364
Resnet-50	0.2984	1.7333	0.3009	1.7603	0.3009	1.7603
EfficientNet	0.2318	1.9944	0.2372	1.9828	-	-

**Table 5.3:** Ablation study for 8 classes image classification

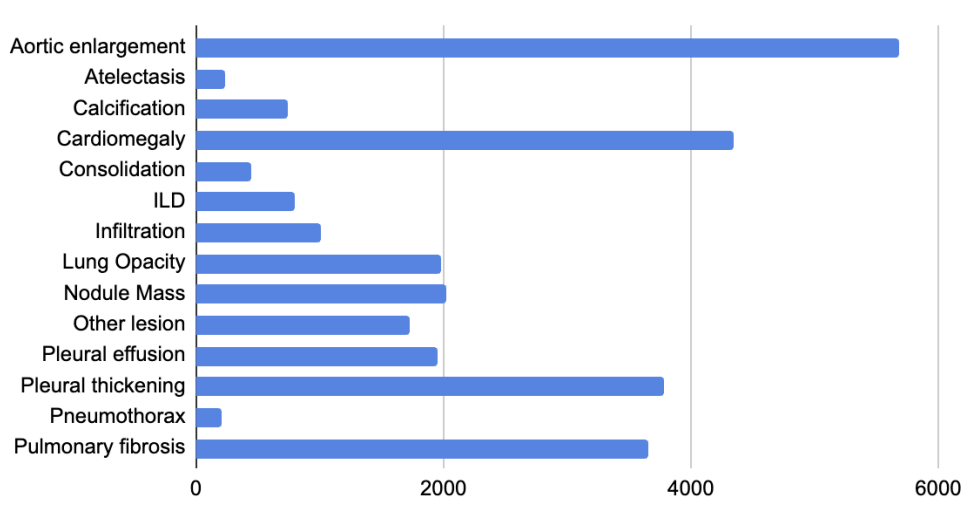
Assessing the performance of these three distinct models reveals that Resnet-50 achieved the highest accuracy at 0.2984. Following closely to Resnet-50 is MobileNet v2 with an accuracy of 0.2927, while EfficientNet exhibits the least favorable performance among the three models, recording an accuracy of 0.2318. In terms of loss, Resnet-50 and MobileNet show comparable values of 1.7319 and 1.7333, respectively, while EfficientNet lags slightly behind with a value of 1.9944.

For these models, validation loss and accuracy roughly resemble those of their training counterparts. Similarly, the test accuracy and test loss align with training accuracy and training loss for Resnet-50 and MobileNet. However, for EfficientNet, a notable deviation occurs as it fails to converge. This divergence may stem from potential overfitting to the training data, capturing noise or specific patterns absent in the test dataset. Additionally, as discussed in the context of binary image classification, the inherent complex architecture of Resnet-50 which is favorable only when there is a large number of dataset, might be the cause for the failure in convergence of the output.

Analyzing the results of image classification shows that the model's performance on correctly identifying and classifying images is relatively limited. This suggests potential issues within the dataset possibly related to dataset complexity, insufficient training data, imbalanced data and model complexity. Despite the subpar accuracy in image classification accuracy, object detection is carried out on these dataset. This undertaking holds significance as it provides the exact location of the objects, can handle multiple objects in a single image and provides more fine-grained analysis of the image. Object detection can prove beneficial in identifying and addressing specific challenges associated with image classification. For instance, it aids in finding out which regions of the image are challenging for the model to identify. This additional information can be crucial in applications where a detailed analysis of image is necessary.

## 5.2 Object Detection

After completing the image classification phase and gaining insights into the expected outcomes of the subsequent experiment, the focus shifts to object detec-



**Figure 5.1:** Class Distribution for all the classes except No findings

tion. Three different models were chosen for carrying out object detection. These includes detectron2, MM detection and YOLOv8 models. These models are recognized for its ease of implementation and adaptability for fine-tuning, and is therefore chosen as the state-of-the-art model for object detection. To obtain a comprehensive understanding of the dataset's overall performance, training is conducted using the entire dataset. This method strives to evaluate the model's performance over the entire dataset and provide a comprehensive overview of its object detection abilities.

### 5.2.1 Object Detection for 8 classes

Upon analyzing the experiment, it became evident that the outcomes of object detection across all classes were suboptimal. To gain a more nuanced comprehension of the dataset, a strategic split is implemented, selecting images that contain approximately two thousand five hundred instances of the reports for subsequent defects. This filtering process is applied independently to each class, resulting in a curated dataset consisting of images from eight different classes. The distribution of classes within this refined training dataset is visually presented in Figure 5.2, representing eighty percent of the total dataset post-filtration which totals to fifteen thousand seven hundred and twenty nine images. Subsequently, this tailored dataset undergoes training using the Detectron2 model.

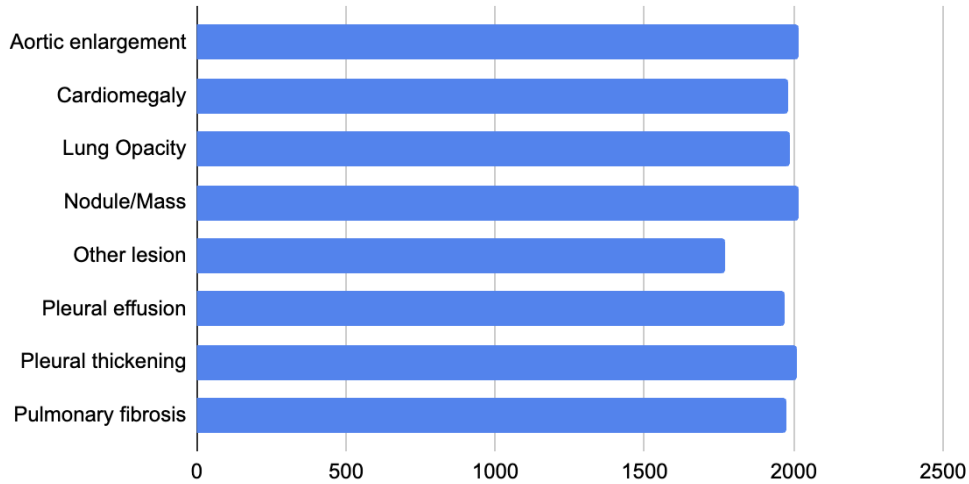


Figure 5.2: Class Distribution for eight classes

Method and Backbone	AP	AP50	AP75
Detectron 2 (MaskRCNN)	0.037	0.097	0.018
MmEngine(RTM Det)	0.176	0.332	0.178
Yolov8(CSP Net)	0.208	0.377	0.210

Table 5.4: Ablation study on different models

Table 5.4 presents the comprehensive findings from an ablation study conducted on three distinct object detection models. The study focused on evaluating the models' performance, specifically looking at their AP values in the context of object detection output. Among the models examined, Yolov8 emerged as the top performer, showcasing a notably superior average precision of 0.208, 0.377 and 0.210 for AP, AP50 and AP75 consecutively. Following closely is MMEngine, utilizing RTM Det as its backbone, with a commendable AP, AP50 and AP75 value of 0.176, 0.332 and 0.178 respectively. In contrast, Detectron2 exhibited the least favorable results among the models, registering a modest AP, AP50 and AP75 values of only 0.037, 0.097 and 0.018 respectively.

This observable performance disparity motivates a more thorough investigation to determine the fundamental causes of the accuracy variance. In order to fully understand the reasons behind the observed mediocre accuracy, one of the models will be chosen for a more thorough examination in later phases of the research. This focused study intends to improve our comprehension of the complex dynamics underlying object detection accuracy and shed light on the particular factors impacting the model's performance.

To obtain this objective, Detectron2 is selected as the preferred model. The initial phase of the analysis involves exploring different output scenarios by making systematic adjustments to the optimizer. This involves evaluating how various



optimizer configurations affect the model's performance in order to obtain an understanding of how these differences affect the object detection process's overall outcomes.

Number of Classes	Optimizer	Accuracy	Total Loss	AP iou = 0.5
8	SGD	0.875	0.9781	0.097
8	Adam	0.8574	1.153	0.091

**Table 5.5:** 8 classes object detection in Detectron 2

The results of training the dataset in Detectron 2 model with different optimizers are depicted in Table 5.5. The experiment involved the comparison of two different optimizers, namely SGD and Adam. Notably, when compared to the Adam optimizer, the SGD optimizer performed better in both of the scenarios. More specifically, SGD achieved an accuracy of 0.875, surpassing Adam's accuracy of 0.8574. Additionally, Detectron 2 models trained using SGD showcased a lower total loss of 0.9781, contrasting with Adam's total loss of 1.153.

Most importantly, in the context of object detection, where average precision is a crucial metric, SGD once again demonstrated a slightly better outcome with a score of 0.097, compared to Adam's score of 0.091. Despite Adam's reputation as a default choice for various applications due to its adaptive learning rate and user-friendly characteristics, the results of disease detection in chest X-rays suggest that SGD outperformed Adam in this specific application.

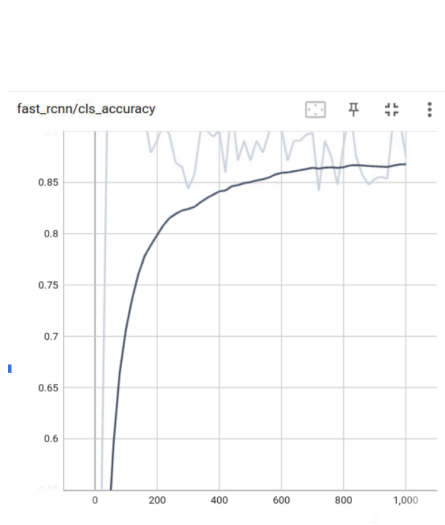
Class	AP for SGD	AP for Adam
Aortic enlargement	10.221	8.757
Cardiomegaly	13.498	14.335
Lung Opacity	1.084	1.029
Nodule/Mass	1.541	1.191
Other lesion	0.051	0.077
Pleural effusion	2.559	2.347
Pleural thickening	0.271	0.232
Pulmonary fibrosis	0.506	0.513

**Table 5.6:** Classwise AP for 8 classes using SGD and Adam Optimizer

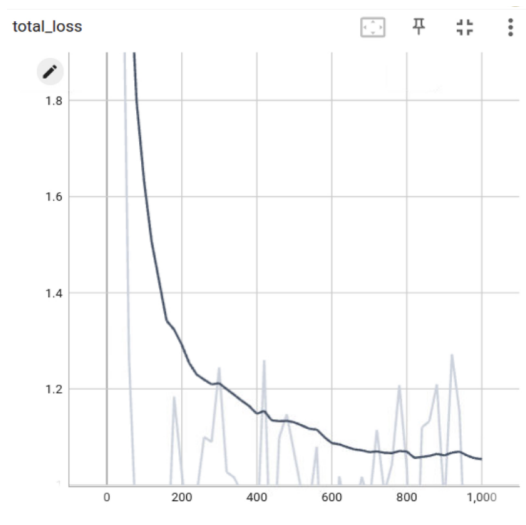
Table 5.6 illustrates the class-wise average precision across various classes. Among the eight classes, cardiomegaly achieved the highest average precision for both SGD and Adam optimizers in Detectron2 model, registering values of 13.498 and 14.335, respectively. Aortic enlargement followed closely with average precision scores of 10.221 and 8.757 for SGD and Adam optimizers, respectively. The average precision then dropped significantly to 2.559 and 2.347 for pleural

effusion, respectively. The other lesions exhibited the lowest average precision, recording values of 0.051 and 0.077 for the two optimizers.

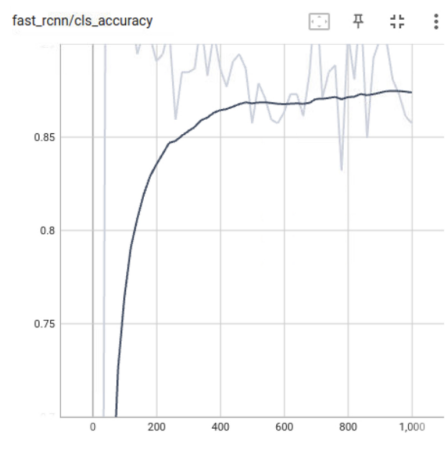
SGD optimizer demonstrated superior performance in six out of the eight classes, whereas Adam optimizer outperformed in only two classes. The most substantial difference in average precision metrics was observed for aortic enlargement, with a margin of 1.464, while the minimum difference was a slight 0.007 for pulmonary fibrosis. Consequently, due to the enhanced performance of SGD over the Adam optimizer, the decision was made to opt for SGD in the subsequent experiments.



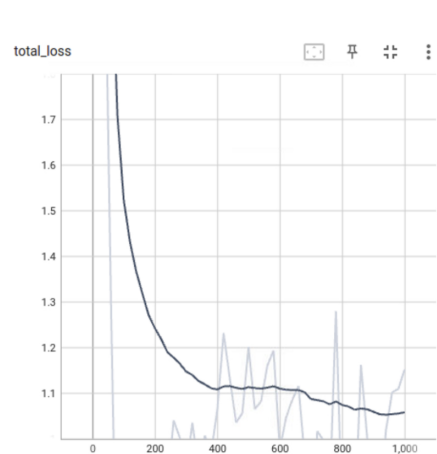
**Figure 5.3:** Class Accuracy for SGD Optimizer



**Figure 5.4:** Total Loss for SGD Optimizer



**Figure 5.5:** Class Accuracy for Adam Optimizer



**Figure 5.6:** Total Loss for Adam Optimizer

The comparison of class accuracy curves for the Detectron 2 model also reveals

a nuanced distinction between the performance of the SGD optimizer and the Adam optimizer in the task of recognizing and classifying objects, as illustrated in Figure 5.3 and 5.5.

Both curves demonstrate rapid increases in class accuracy that plateau around 400 iterations. However, upon closer examination, a subtle yet noteworthy trend can be recognized. Despite marginal improvements in class accuracy for both optimizers beyond the 400th iteration, it is noteworthy that SGD continues to exhibit gradual enhancement until 600 iterations, whereas such progress is very very less in the case of the Adam optimizer. This nuanced difference implies that, in the context of the Detectron 2 model and its object recognition task, the SGD optimizer exhibits a more sustained ability to refine and enhance class accuracy, especially in the later stages of training.

The trajectory of the total loss curve also shows a comparable pattern, characterized by a significant decrease up to the 400th epoch. This tendency is consistent with the behavior in class accuracy that has been seen, indicating a phase of significant model performance improvement. But the behavior disparity that occurs beyond this tipping point becomes really interesting.

Upon reaching the 400th epoch, a distinct contrast emerges between the SGD and Adam optimizers. The SGD optimizer showcases a continued and measured decline in the total loss, suggesting a sustained refinement in the model's predictive capabilities. In contrast, the Adam optimizer demonstrates a plateau in total loss reduction during the subsequent 200 epochs, with only a marginal decrease observed. The subtle differences between the two optimizers are shown by this divergence in the loss curves, which also emphasizes the SGD optimizer's capacity to maintain improvement in total loss even during the later phases of training. Gaining an understanding of these differences in optimizer behavior can help boost training efficiency and improve object detection performance.

To sum up, the analysis of Detectron2's performance in the selected eight classes has shown some interesting differences. Particularly, the average precision for two classes, namely aortic enlargement and cardiomegaly, outshone the remaining classes. Additionally, employing the SGD optimizer demonstrated superior results. Consequently, these two classes will undergo further experimentation to assess how the model's performance evolves when subjected to more focused analysis.

### 5.2.2 Object Detection for 2 and 3 classes

To deepen our comprehension of object detection model behavior in context of finding deformities in chest X-rays, the model is trained specifically on two classes that exhibited superior performance in the prior experiment: Cardiomegaly and Aortic enlargement. Initially, the model undergoes training exclusively on these two classes. After that, an interesting experiment will be discussed to assess the effect of additional data. An additional dataset that has been selected to include just those cases devoid of defects is added to the training dataset. The object is

to determine whether the inclusion of no disease images influences the model's performance positively or if the focused training on specific classes remains the more effective strategy.

Number of Classes	Accuracy	Total loss	AP iou = 0.5
2	0.9355	0.5678	0.519
3	0.9414	0.3773	0.571

**Table 5.7:** Result for 3 classs vs 3 classes

Upon examination of Table 5.7, a noticeable distinction emerges in the class accuracy of the Detectron 2 model when trained exclusively with two classes. In contrast, its performance is compared when the dataset is added with an additional dataset containing no findings, leading to accuracy scores of 0.9355 and 0.9414, respectively. Simultaneously, the total loss undergoes a decrease from 0.5678 to 0.3773. Notably, the average precision over AP iou = 0.5 shows improvement, advancing from 0.519 to 0.571 when incorporating a dataset without findings for the two classes.

This data suggests that, at least for the current analysis, the augmentation with extra data has contributed positively to enhancing the model's performance. However, to get a better understanding of what is happening, a deep study is required.

Class	AP for 2 class training	AP for 3 class training
Aortic Enlargement	21.354	18.108
Cardiomegaly	24.497	21.922
No findings	-	76.739

**Table 5.8:** Classwise AP for 2 and 3 classes

Observing Table 5.8, which presents class-wise Average Precision, reveals that the average precision for Aortic Enlargement and Cardiomegaly is notably higher at 21.354 and 24.497, respectively, in comparison to the model trained with a dataset containing no findings. However, when the model is trained with three classes (Aortic Enlargement, Cardiomegaly, and No Findings), the average precision values change to 18.108, 21.922, and 76.739, respectively.

This finding demonstrates a direct correlation between the higher average precision for images classified as "No Findings" and an increase in the average precision for all classes as a whole. However, in the context of this thesis, the primary emphasis lies in accurately detecting and diagnosing diseases rather than determining the absence of findings. Therefore, considering the specific goal of identifying two particular deformities, prioritizing a dataset exclusively consisting of instances of these deformities proves to be more advantageous than relying on

a model trained with a dataset containing images labeled as having no findings.



## Chapter 6

# Results and Discussion

This chapter is dedicated to examining the results obtained from a series of experiments focused on image classification and object detection applied to chest X-rays, as conducted within the scope of this thesis. Through rigorous investigation and analysis, we aim to uncover valuable insights into the capabilities, strengths, and potential areas of improvement of the models deployed for finding out deformities in chest xrays. The information provided in this chapter will contribute to the collective understanding of how well object detection and image classification techniques perform in real-world settings.

### 6.1 Object Detection for 8 classes

Figures 6.1, 6.3 and 6.5 illustrate the ground truths, that provide an accurate depiction of the actual defects present in the images. On the other hand, Figures 6.2, 6.4 and 6.6 present the outputs generated by the RTMDet model, displaying its predictions based on the given input. It is apparent from looking at Figures 6.2 and 6.4 that the model had difficulty correctly predicting any of the classes, indicating a notable discrepancy in the model's performance. It only accurately predicts one class (pleural thickening) in Figure 6.4. There are instances where it erroneously predicts Pleural effusion, showcasing areas where improvement is needed. Moving on to Figure 6.6, the model correctly predicts Cardiomegaly but incorrectly predicts Aortic Enlargement, a feature which is absent in the ground truth data. In summary, the overall performance of the RTMDet model falls below expectations, emphasising the need for further refinement and enhancement mainly in terms of training dataset to achieve better result.

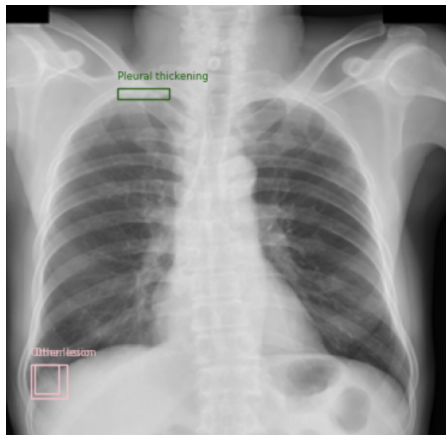


Figure 6.1: Ground truth 1

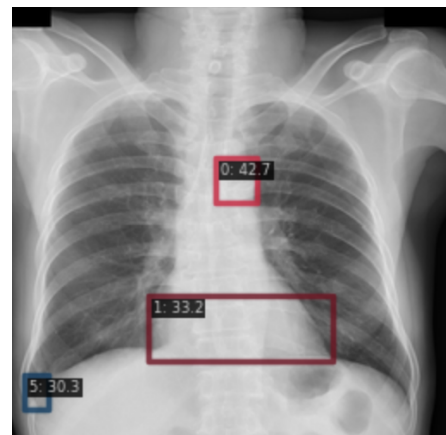


Figure 6.2: RTMDet Output 1

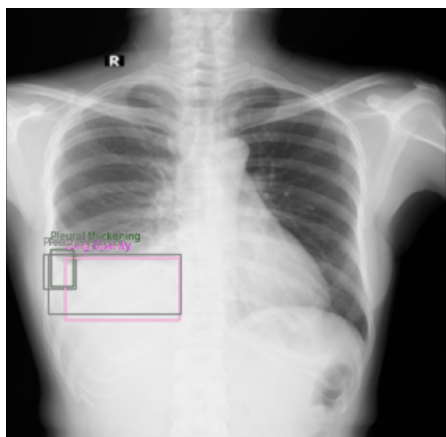


Figure 6.3: Ground truth 2

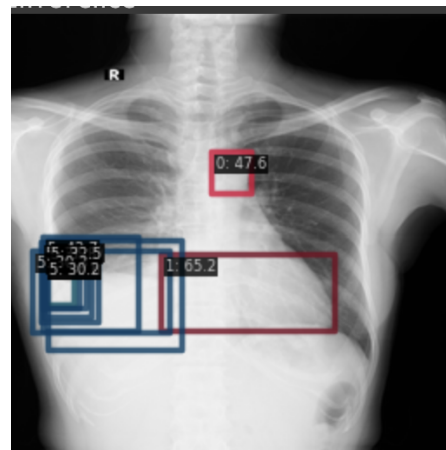


Figure 6.4: RTMDet Output 2

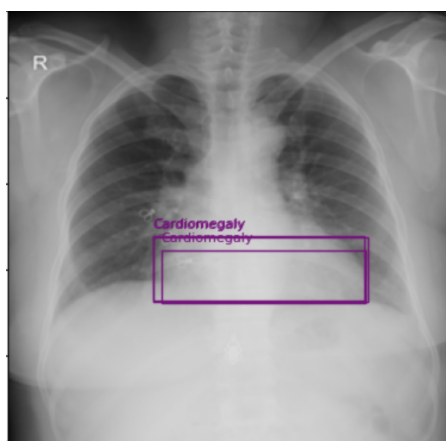


Figure 6.5: Ground truth 3

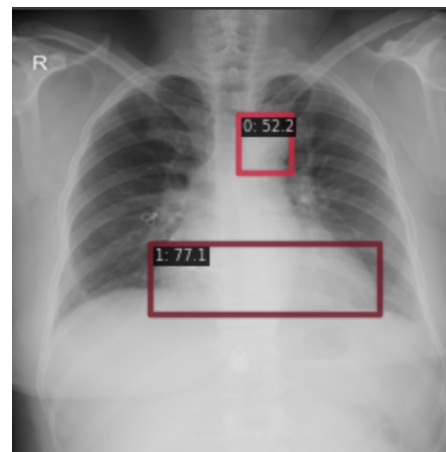


Figure 6.6: RTMDet Output 3



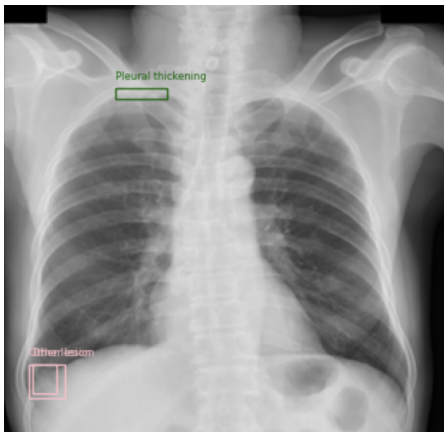


Figure 6.7: Ground truth 4



Figure 6.8: YOLOv8 Output 1

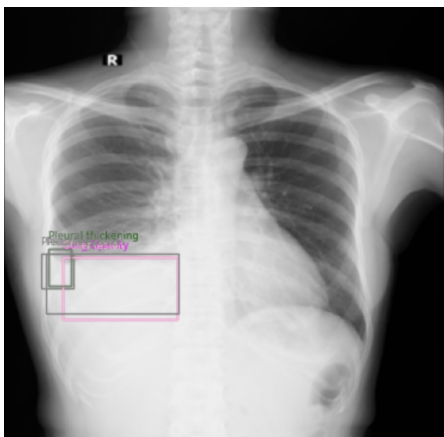


Figure 6.9: Ground truth 5

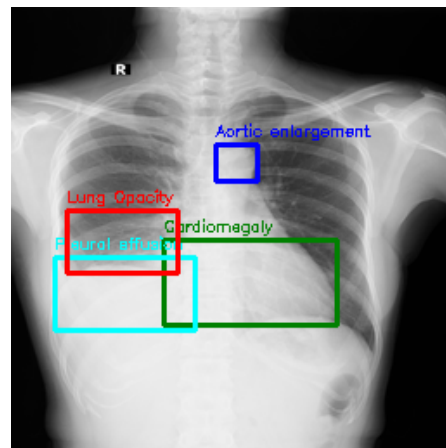


Figure 6.10: YOLOv8 Output 2

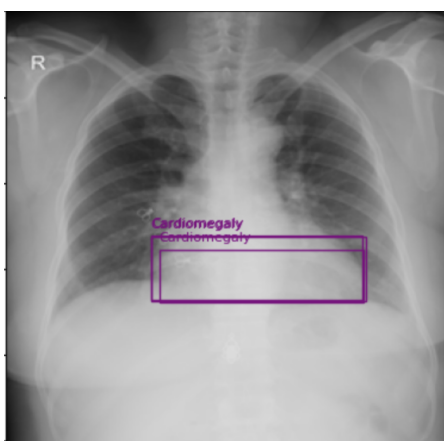


Figure 6.11: Ground truth 6

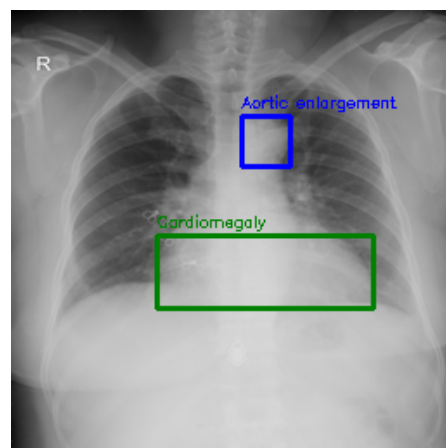


Figure 6.12: YOLOv8 Output 3

The ground truths of the data used during the evaluation of YOLOv8 model are shown in Figures 6.7, 6.9, and 6.11, which consists of the actual defects present in the images. Upon comparing images 6.7 and 6.8, it's evident that the model's detection capabilities fell short, as it failed to detect the two defects present in the images. Figure 6.10 shows correct predictions for lung opacity but inaccurately predicts the other three defects. Finally, in Figure 6.12, it correctly predicted cardiomegaly but also made an erroneous additional prediction of aortic enlargement. Examining these outputs, it becomes apparent that the performance of YOLOv8 is subpar in case of finding deformities in chest x-rays using vindr dataset, indicating the need for further refinement and enhancement to improve model performance.

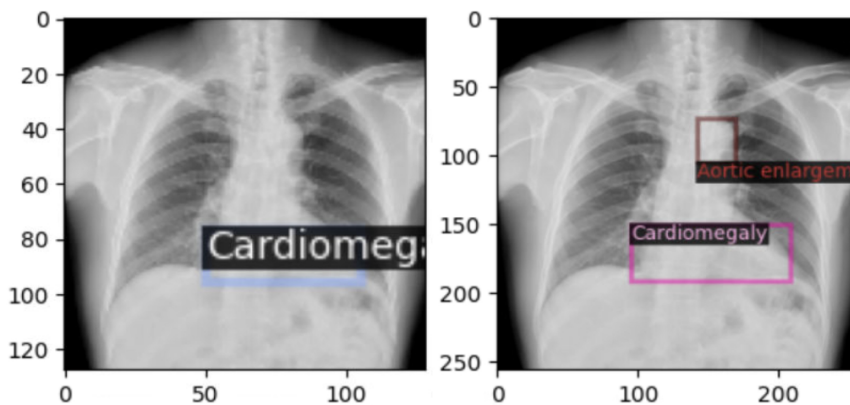


Figure 6.13: SGD optimizer ground truth and output 1

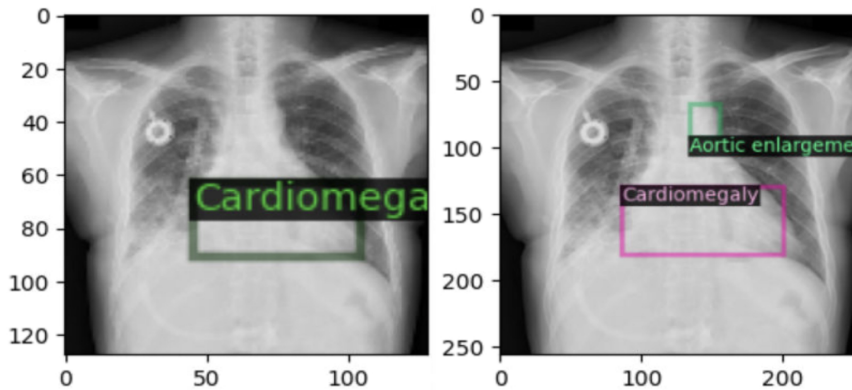


Figure 6.14: SGD optimizer ground truth and output 2

Analysing the SGD optimizer results 6.13 and 6.14, it is clear that the Detectron2 model—which was trained using eight classes—showed a remarkable capacity for cardiomegaly detection, closely matching the expected result for the images. Remarkably, aortic enlargement was detected in both images, even though the ground truth annotations did not provide this information. This disparity could

be caused by a number of things, including possible overfitting to the image data or the existence of a complicated background. Due to these complexity, the model could mistakenly detect patterns in the noise or background that resemble aortic enlargement.

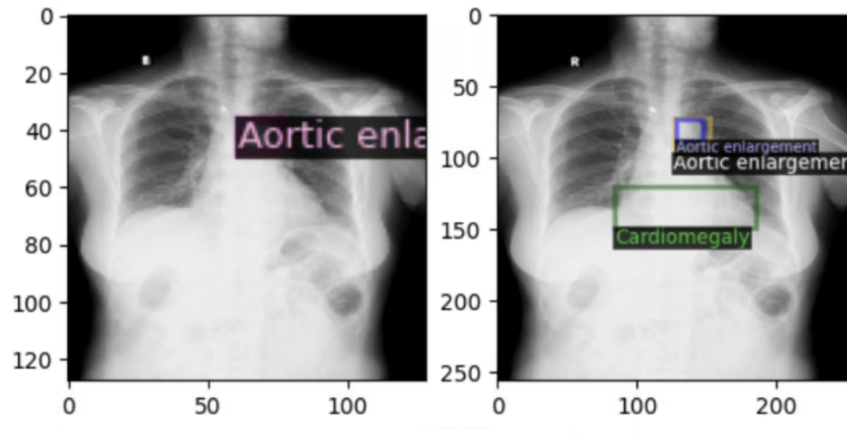


Figure 6.15: Adam optimizer ground truth and output 1

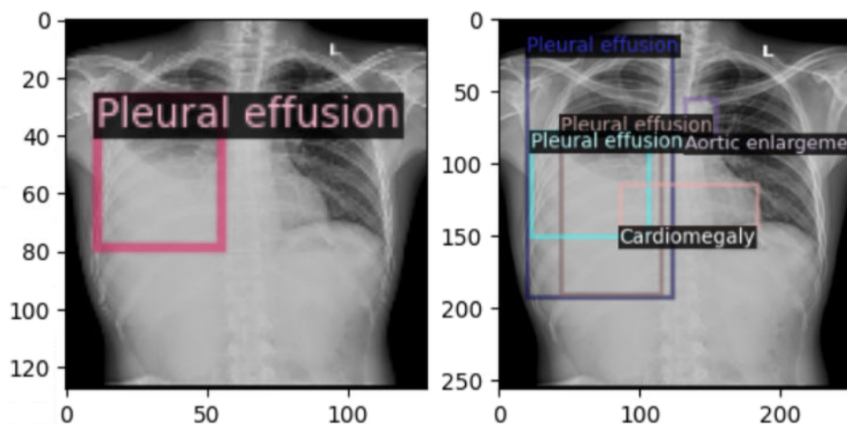


Figure 6.16: Adam optimizer ground truth and output 2

Examining the outcomes achieved with the Adam optimizer 6.15 and 6.16, the consistent trend persists in the detection of diseases not present in the ground truth images. The underlying reasons for this behavior align with those previously discussed. However, a noteworthy observation emerges when examining image 6.16: the model demonstrates the capability to identify the presence of additional diseases, which is pleural effusion.

## 6.2 Object Detection for 2 and 3 classes

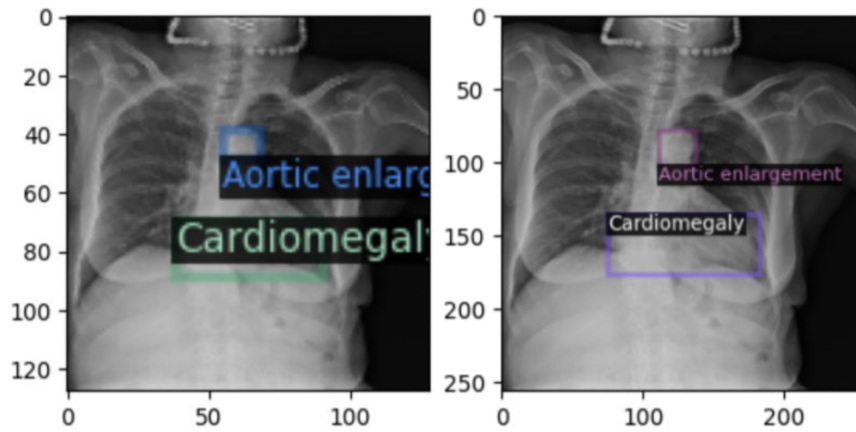


Figure 6.17: 2 classes object detection output 1

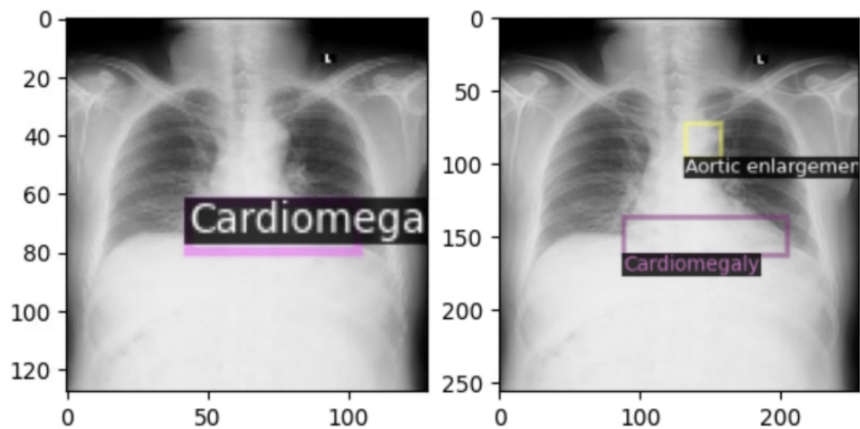


Figure 6.18: 2 classes object detection output 2

When 2-class object detection is carried out, as shown in Figure 6.17, the model successfully recognised both of the defects seen in the picture. On the other hand, Figure 6.18 makes it clear that the model only correctly identified one of the defects. Interestingly, the model detects aortic enlargement in this image even if the ground truth does not support the occurrence of such enlargement.

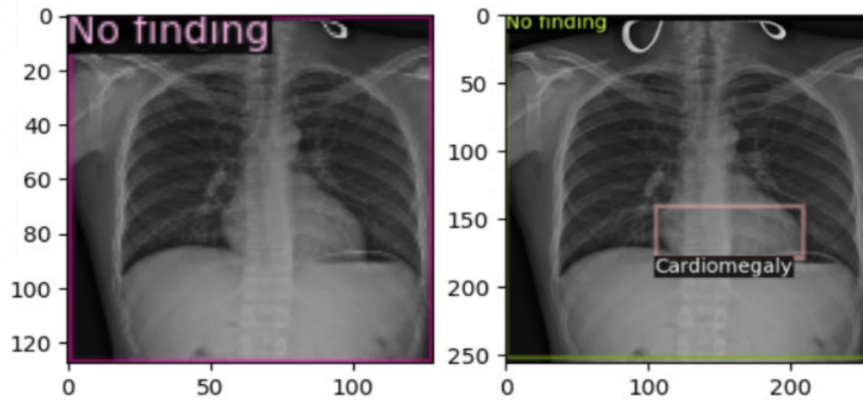


Figure 6.19: 3 classes object detection output 1

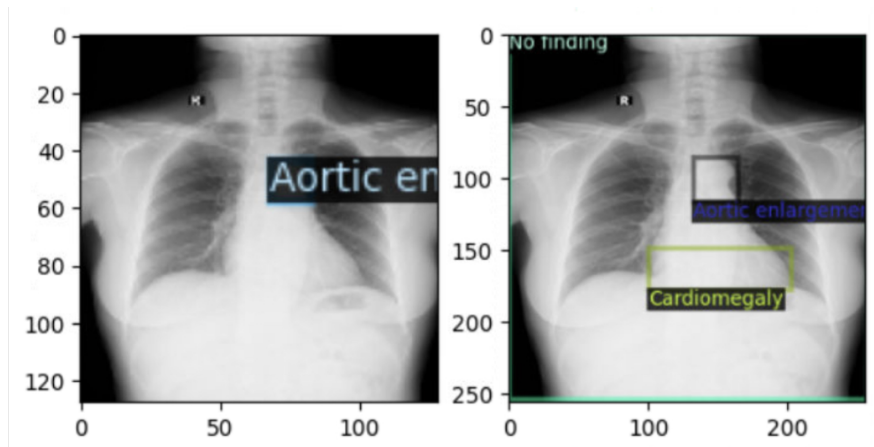


Figure 6.20: 3 classes object detection output 2

As the ground truth indicates that there are no deformities in Figure 6.19, the model unexpectedly predicts the presence of cardiomegaly. On the other hand, in the figure 6.20 when aortic enlargement is the only defects, the model does a very good job of identifying the particular abnormality. In spite of this success, a noteworthy finding is that the model also detects an additional disease, which is not consistent with the ground truth. This suggests that the model may be vulnerable to false positives or over interpretation in some situations. This disparity calls into question the robustness of the model and requires more research into its generalisation potential.

After examining the training models with both two and three classes, it is apparent that there is significant potential for improvement to achieve better results.



## Chapter 7

# Conclusion

In conclusion, this thesis has worked on training and evaluating 3 state-of-the-art object recognition models for abnormality detection in chest x-ray. Various experimentation have been carried out in models-Detectron2, RTMDet and YOLOv3, in order to find deformities in chest x-rays. The evaluation showed that YOLOv8 outperforms all the selected object detection models.

Using Detectron2, object detections with different number of classes i.e. 8,3 and 2 were also carried out to study the impact of altering the class size on the performance of the model. These experimentations revealed notable variations in accuracy where it was found out the average precision of the model improved significantly when the model is trained on a small number of classes.

Beside this, the training and evaluation of 3 different image classification models called Mobilenet v2, Resnet-50 and EfficientNet was also carried out. Resnet-50, which is known for its deep architecture, performed best among all three of the selected models. Analysis of these models not only clarifies the efficiency of medical image classification tasks, but also provides information about how to obtain appropriate balance in between accuracy and model complexity.

The evaluation criteria and visual evaluation of the model outcomes provided valuable insights into the effectiveness of the model. Although all models demonstrated proficiency in some areas, it was also recognized that continuous improvement and refinement was needed to improve the overall performance of the model.

### 7.1 Further Work

Although state-of-the-art models were used in the thesis to investigate ROI detection in chest X-rays, there is still plenty of room for improvement to achieve better results. The existing models were exclusively trained on Vindr Chest-Xray images. To broaden the dataset and potentially enhance model performance, the integration of datasets from diverse sources could be explored during the training process. If ethical constraints prohibit such integration, using transfer learning could be a good substitute to bolster the robustness of the model.

Furthermore, a multi-modal approach of training a machine learning model that uses chest x-rays and medical reports could also be used in order to train a machine learning model and improve the accuracy of the model. Collaborations with healthcare professionals could also be carried out in order to obtain domain-specific knowledge and to guarantee that the model aligns with the practical clinical needs.



# Bibliography

- [1] [https://www.who.int/data/gho/data/indicators/indicator-details/GHO/medical-doctors-\(per-10-000-population\)](https://www.who.int/data/gho/data/indicators/indicator-details/GHO/medical-doctors-(per-10-000-population)), (Accessed on 12/11/2022).
- [2] L. Delrue, R. Gosselin, B. Ilsen, A. Van Landeghem, J. de Mey and P. Duyck, 'Difficulties in the interpretation of chest radiography,' in *Comparative Interpretation of CT and Standard Radiography of the Chest*, E. E. Coche, B. Ghaye, J. de Mey and P. Duyck, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 27–49, ISBN: 978-3-540-79942-9. DOI: 10.1007/978-3-540-79942-9\_2. [Online]. Available: [https://doi.org/10.1007/978-3-540-79942-9\\_2](https://doi.org/10.1007/978-3-540-79942-9_2).
- [3] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille and Y. Zhou, 'Transunet: Transformers make strong encoders for medical image segmentation,' *arXiv preprint arXiv:2102.04306*, 2021.
- [4] J. M. J. Valanarasu, P. Oza, I. Hacıhaliloğlu and V. M. Patel, 'Medical transformer: Gated axial-attention for medical image segmentation,' in *Medical Image Computing and Computer Assisted Intervention—MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part I 24*, Springer, 2021, pp. 36–46.
- [5] J. M. J. Valanarasu and V. M. Patel, 'Unext: Mlp-based rapid medical image segmentation network,' in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2022, pp. 23–33.
- [6] M. H. Hesamian, W. Jia, X. He and P. Kennedy, 'Deep learning techniques for medical image segmentation: Achievements and challenges,' *Journal of Digital Imaging*, vol. 32, no. 4, pp. 582–596, Aug. 2019, ISSN: 1618-727X. DOI: 10.1007/s10278-019-00227-x. [Online]. Available: <https://doi.org/10.1007/s10278-019-00227-x>.
- [7] [https://en.wikipedia.org/wiki/Object\\_detection](https://en.wikipedia.org/wiki/Object_detection).
- [8] <https://cocodataset.org/>.
- [9] H. Bandyopadhyay, <https://www.v7labs.com/blog/image-classification-guide>.
- [10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, *Mobilenetv2: Inverted residuals and linear bottlenecks*, 2019. arXiv: 1801.04381 [cs.CV].

- [11] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto and H. Adam, *Mobilenets: Efficient convolutional neural networks for mobile vision applications*, 2017. arXiv: 1704.04861 [cs.CV].
- [12] K. He, X. Zhang, S. Ren and J. Sun, 'Deep residual learning for image recognition,' *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
- [13] M. Tan and Q. V. Le, 'Efficientnet: Rethinking model scaling for convolutional neural networks,' *International Conference on Machine Learning*, 2019.
- [14] H. Honda, <https://medium.com/@hirotoschwert/digging-into-detectron-2-47b2e794fabd>, (Accessed on 09/12/2023), Jan. 2020.
- [15] C. Lyu, W. Zhang, H. Huang, Y. Zhou, Y. Wang, Y. Liu, S. Zhang and K. Chen, *Rtmdet: An empirical study of designing real-time object detectors*, 2022. arXiv: 2212.07784 [cs.CV].
- [16] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy and D. Lin, *Mmdetection: Open mmlab detection toolbox and benchmark*, 2019. arXiv: 1906.07155 [cs.CV].
- [17] Z. Ge, S. Liu, F. Wang, Z. Li and J. Sun, *Yolox: Exceeding yolo series in 2021*, 2021. arXiv: 2107.08430 [cs.CV].
- [18] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, 'You only look once: Unified, real-time object detection,' in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.91.
- [19] X. Huang, X. Wang, W. Lv, X. Bai, X. Long, K. Deng, Q. Dang, S. Han, Q. Liu, X. Hu, D. Yu, Y. Ma and O. Yoshie, *Pp-yolov2: A practical object detector*, 2021. arXiv: 2104.10419 [cs.CV].
- [20] J. Redmon and A. Farhadi, *Yolov3: An incremental improvement*, 2018. arXiv: 1804.02767 [cs.CV].
- [21] A. Bochkovskiy, C.-Y. Wang and H.-Y. M. Liao, *Yolov4: Optimal speed and accuracy of object detection*, 2020. arXiv: 2004.10934 [cs.CV].
- [22] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, tkianai, A. Hogan, lorenzomamma, yxNONG, AlexWang1900, L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, F. Ingham, Frederik, Guilhen, Hatovix, J. Poznanski, J. Fang, L. Yu, changyu98, M. Wang, N. Gupta, O. Akhtar, PetrDvoracek and P. Rai, *ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements*, version v3.1, Oct. 2020. DOI: 10.5281/zenodo.4154370. [Online]. Available: <https://doi.org/10.5281/zenodo.4154370>.

- [23] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei and X. Wei, *Yolov6: A single-stage object detection framework for industrial applications*, 2022. arXiv: 2209.02976 [cs.CV].
- [24] C.-Y. Wang, A. Bochkovskiy and H.-Y. M. Liao, *Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*, 2022. arXiv: 2207.02696 [cs.CV].
- [25] G. Jocher, A. Chaurasia and J. Qiu, *YOLO by Ultralytics*, version 8.0.0, Jan. 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>.
- [26] N. Sharma, S. Baral, M. Paing and R. Chawuthai, 'Parking time violation tracking using yolov8 and tracking algorithms,' *Sensors*, vol. 23, p. 5843, Jun. 2023. DOI: 10.3390/s23135843.
- [27] H. Lou, X. Duan, J. Guo, H. Liu, J. Gu, L. Bi and H. Chen, 'Dc-yolov8: Small-size object detection algorithm based on camera sensor,' *Electronics*, vol. 12, no. 10, 2023, ISSN: 2079-9292. DOI: 10.3390/electronics12102323. [Online]. Available: <https://www.mdpi.com/2079-9292/12/10/2323>.
- [28] *Intersection over union for object detection*, (Accessed on 10/12/2023), 2023.
- [29] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017. arXiv: 1412.6980 [cs.LG].
- [30] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang and D. Tao, 'A survey on vision transformer,' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 87–110, 2023. DOI: 10.1109/TPAMI.2022.3152247.
- [31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, 'Attention is all you need,' *Advances in neural information processing systems*, vol. 30, 2017.
- [32] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, 'Language models are unsupervised multitask learners,' *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.
- [33] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019. arXiv: 1810.04805 [cs.CL].
- [34] P.-H. Conze, G. Andrade-Miranda, V. K. Singh, V. Jaouen and D. Visvikis, 'Current and emerging trends in medical image segmentation with deep learning,' *IEEE Transactions on Radiation and Plasma Medical Sciences*, vol. 7, no. 6, pp. 545–569, 2023. DOI: 10.1109/TRPMS.2023.3265863.
- [35] X. Liu, L. Song, S. Liu and Y. Zhang, 'A review of deep-learning-based medical image segmentation methods,' *Sustainability*, vol. 13, no. 3, 2021, ISSN: 2071-1050. DOI: 10.3390/su13031224. [Online]. Available: <https://www.mdpi.com/2071-1050/13/3/1224>.

- [36] A. Krizhevsky, I. Sutskever and G. E. Hinton, 'Imagenet classification with deep convolutional neural networks,' *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, ISSN: 0001-0782. DOI: 10.1145/3065386. [Online]. Available: <https://doi.org/10.1145/3065386>.
- [37] K. Simonyan and A. Zisserman, 'Very deep convolutional networks for large-scale image recognition,' *arXiv preprint arXiv:1409.1556*, 2014.
- [38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, 'Going deeper with convolutions,' *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.
- [39] G. Huang, Z. Liu, L. v. d. Maaten and K. Q. Weinberger, 'Densely connected convolutional networks,' *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.
- [40] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy and A. L. Yuille, 'DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018. DOI: 10.1109/TPAMI.2017.2699184.
- [41] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy and A. L. Yuille, 'DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018. DOI: 10.1109/TPAMI.2017.2699184.
- [42] L.-C. Chen, G. Papandreou, F. Schroff and H. Adam, *Rethinking atrous convolution for semantic image segmentation*, 2017. DOI: 10.48550/ARXIV.1706.05587. [Online]. Available: <https://arxiv.org/abs/1706.05587>.
- [43] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff and H. Adam, *Encoder-decoder with atrous separable convolution for semantic image segmentation*, 2018. DOI: 10.48550/ARXIV.1802.02611. [Online]. Available: <https://arxiv.org/abs/1802.02611>.
- [44] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff and H. Adam, 'Mixed atrous convolution for semantic image segmentation,' in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.
- [45] S. Takikawa, D. Acuna, V. Jampani and S. Fidler, 'Rethinking atrous convolution: You only need one,' in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2020.
- [46] B. Liu, W. Chen, B. Zhang, W. Liu, X. Zhou, Y. Ma, X. Hong and P.-A. Heng, 'Metad2net: A deep dense multiscale and multispectral network for remote sensing image scene classification,' *Remote Sensing*, vol. 13, no. 9, p. 1770, 2021.

- [47] O. Ronneberger, P. Fischer and T. Brox, 'U-net: Convolutional networks for biomedical image segmentation,' in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells and A. F. Frangi, Eds., Cham: Springer International Publishing, 2015, pp. 234–241, ISBN: 978-3-319-24574-4.
- [48] D. Pal, P. B. Reddy and S. Roy, 'Attention uw-net: A fully connected model for automatic segmentation and annotation of chest x-ray,' *Computers in Biology and Medicine*, vol. 150, p. 106083, 2022, ISSN: 0010-4825. DOI: <https://doi.org/10.1016/j.combiomed.2022.106083>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0010482522007910>.
- [49] G. Yang, H. Tang, M. Ding, N. Sebe and E. Ricci, *Transformer-based attention networks for continuous pixel-wise prediction*, 2021. arXiv: 2103.12091 [cs.CV].
- [50] Z. Wang, X. Cun, J. Bao, W. Zhou, J. Liu and H. Li, *Uformer: A general u-shaped transformer for image restoration*, 2021. arXiv: 2106.03106 [cs.CV].
- [51] Y. Gao, M. Zhou and D. Metaxas, *Utnet: A hybrid transformer architecture for medical image segmentation*, 2021. arXiv: 2107.00781 [cs.CV].
- [52] K. Islam, *Recent advances in vision transformer: A survey and outlook of recent work*, 2022. arXiv: 2203.01536 [cs.CV].
- [53] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit and N. Houlsby, 'An image is worth 16x16 words: Transformers for image recognition at scale,' in *International Conference on Learning Representations*, 2021. [Online]. Available: <https://openreview.net/forum?id=YicbFdNTTy>.
- [54] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang and S. Lin, 'Swin transformer: Hierarchical vision transformer using shifted windows,' *arXiv preprint arXiv:2103.14030*, 2021.
- [55] H. Touvron, M. Cord, M. Douze, R. D. Hjelm, S. Rendle and J. K. Ben, 'Training data-efficient image transformers & distillation through attention,' *arXiv preprint arXiv:2012.12877*, 2020.
- [56] Q. Hou, Z. Wang, P. Wu, Z.-Q. Hu and C. Wang, 'Cait: Class-attention in image transformers,' *arXiv preprint arXiv:2103.17239*, 2021.
- [57] W. Li, X. Wang, X. Xia, J. Wu, J. Li, X. Xiao, M. Zheng and S. Wen, *Sepvit: Separable vision transformer*, 2023. arXiv: 2203.15380 [cs.CV].
- [58] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić and C. Schmid, 'Vivit: A video vision transformer,' in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 6836–6846.

- [59] X. Dong, J. Bao, D. Chen, W. Zhang, N. Yu, L. Yuan, D. Chen and B. Guo, 'Cswin transformer: A general vision transformer backbone with cross-shaped windows,' in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 12 124–12 134.
- [60] D. Ma, M. R. Hosseinzadeh Taher, J. Pang, N. U. Islam, F. Haghghi, M. B. Gotway and J. Liang, 'Benchmarking and boosting transformers for medical image classification,' in *Domain Adaptation and Representation Transfer*, K. Kamnitsas, L. Koch, M. Islam, Z. Xu, J. Cardoso, Q. Dou, N. Rieke and S. Tsafaris, Eds., Cham: Springer Nature Switzerland, 2022, pp. 12–22, ISBN: 978-3-031-16852-9.
- [61] R. Ghali and M. A. Akhloufi, 'Vision transformers for lung segmentation on cxr images,' *SN Computer Science*, vol. 4, no. 4, p. 414, May 2023, ISSN: 2661-8907. DOI: 10 . 1007 / s42979 - 023 - 01848 - 4. [Online]. Available: <https://doi.org/10.1007/s42979-023-01848-4>.
- [62] R. Ghali and M. A. Akhloufi, 'Arseg: An attention regseg architecture for cxr lung segmentation,' in *2022 IEEE 23rd International Conference on Information Reuse and Integration for Data Science (IRI)*, IEEE, 2022, pp. 291–296.
- [63] S. Xie, R. Girshick, P. Dollár, Z. Tu and K. He, 'Aggregated residual transformations for deep neural networks,' in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.
- [64] S. Ren, K. He, R. Girshick and J. Sun, *Faster r-cnn: Towards real-time object detection with region proposal networks*, 2016. arXiv: 1506.01497 [cs.CV].
- [65] K. He, G. Gkioxari, P. Dollár and R. Girshick, 'Mask r-cnn,' in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.
- [66] R. Girshick, *Fast r-cnn*, 2015. arXiv: 1504.08083 [cs.CV].
- [67] P. K. Das and S. Meher, 'Transfer learning-based automatic detection of acute lymphocytic leukemia,' in *2021 National Conference on Communications (NCC)*, 2021, pp. 1–6. DOI: 10.1109/NCC52529.2021.9530010.

## Appendix A

# Additional Material

### A.1 Code to convert Vindr dataset into COCO dataset Format

```
def get_classId(val):
    for key, value in category_name_to_id.items():
        if str(val)==str(key):
            return value
    return "key_doesn't_exist"

def get_className(val):
    for key, value in category_name_to_id.items():
        if str(val)==str(value):
            return key
    return "key_doesn't_exist"

thing_classes = [
    "Aortic_enlargement",
    "Atelectasis",
    "Calcification",
    "Cardiomegaly",
    "Consolidation",
    "ILD",
    "Infiltration",
    "Lung_opacity",
    "Nodule/Mass",
    "Other_lesion",
    "Pleural_effusion",
    "Pleural_thickening",
    "Pneumothorax",
    "Pulmonary_fibrosis",
    "No_findings"
]
category_name_to_id = {class_name: index for index, class_name in enumerate(thing_classes)}

imageIds=[]
nofindingsImgIds=[]
with open('train.csv', 'r') as csvfile:
    reader = csv.reader(csvfile)
    next(reader)

    annotations=[]
    images=[]
```

```

for row in reader:
    newId = row[9]
    imgId = int(newId)
    classId = int(row[3])

    if(classId == 14 and checkIfImgAvailable(imgId)):
        bbox=[0, 0, IMG_SIZE, IMG_SIZE]
        annotation = {
            "iscrowd": 0,
            "image_id": int(newId),
            "bbox": bbox,
            "category_id": classId,
            "id": imgId
        }
        annotations.append(annotation)
        if imgId not in nofindingsImgIds:
            nofindingsImgIds.append(imgId)

    elif(checkIfImgAvailable(imgId)):

        height=int(row[10])
        width=int(row[11])
        className = row[2]

        xMin=float(row[5])
        yMin=float(row[6])
        xMax=float(row[7])
        yMax=float(row[8])

        xmin = xMin/width*IMG_SIZE
        ymin = yMin/height*IMG_SIZE
        xmax = xMax/width*IMG_SIZE
        ymax = yMax/height*IMG_SIZE

        w = xmax-xmin
        h=ymax-ymin

        bbox=[xmin, ymin, w, h]
        annotation = {
            "iscrowd": 0,
            "image_id": int(newId),
            "bbox": bbox,
            "category_id": classId,
            "id": imgId
        }
        annotations.append(annotation)

        if imgId not in imageIds:
            imageIds.append(imgId)

for id in imageIds:
    img={
        "id": int(id),
        "file_name": str(id)+".png"
    }
    images.append(img)

jsonData = {
    "images": images,

```



```
        "annotations": annotations,  
        "categories": getCategoryes()  
    }  
with open("annotations/instances_train2017.json", "w") as outfile:  
    json.dump(jsonData, outfile)
```

## A.2 Visualization of input images









	Aortic enlargement
	Cardiomegaly
	Lung Opacity
	Nodule-Mass
	Other lesion
	Pleural effusion
	Pleural thickening
	Pulmonary fibrosis

Figure A.1: Respective classes

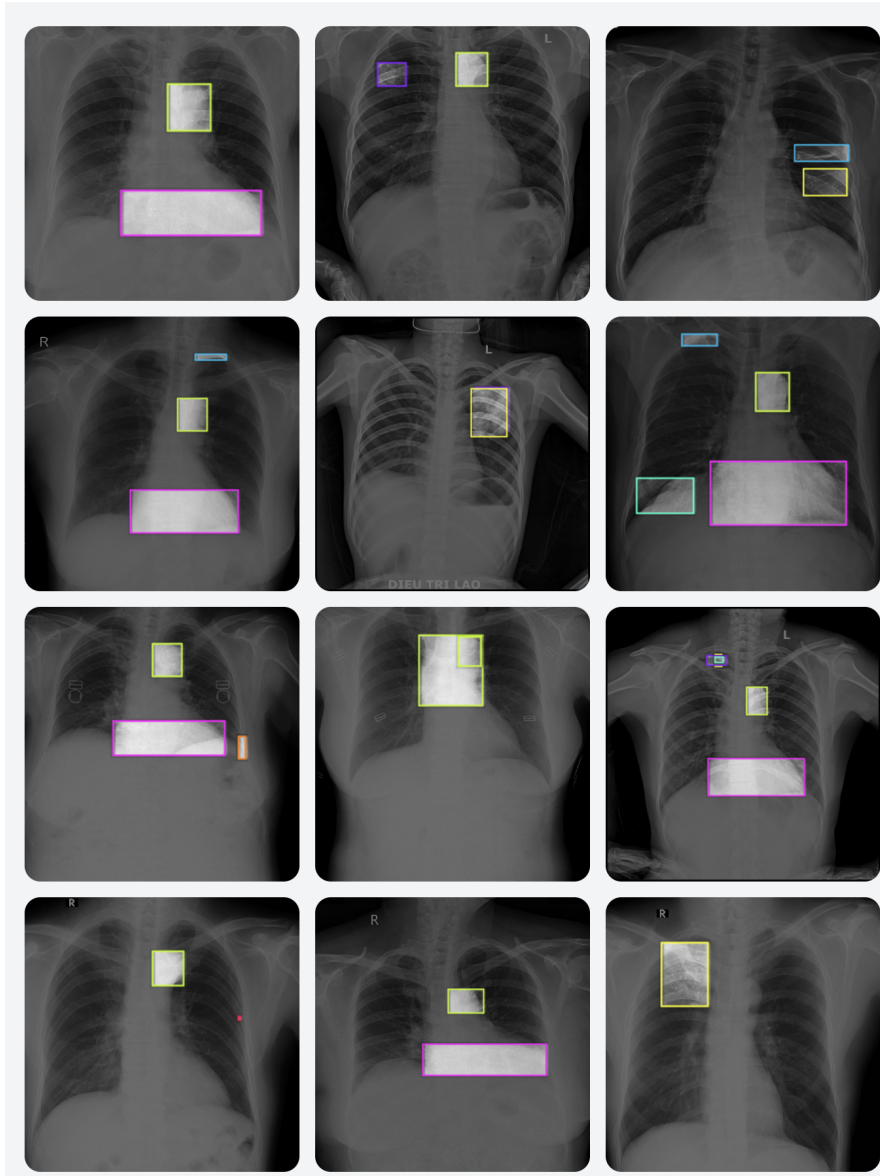


Figure A.2: Input data visualization 1

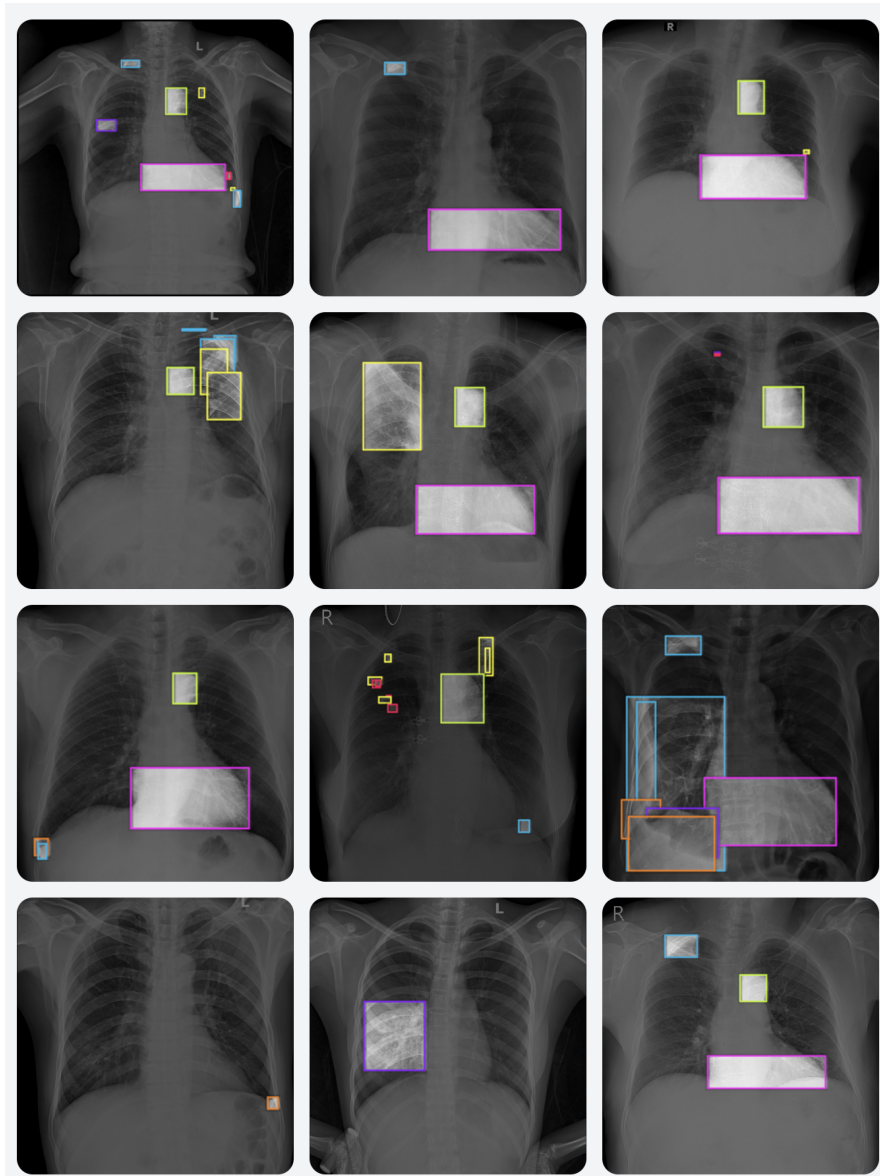


Figure A.3: Input data visualization 2



 **NTNU**

Norwegian University of  
Science and Technology