

YAMAGUCHI UNIVERSITY

DOCTORAL THESIS

**Study on Deep Learning with
Pseudo-labeling Mechanism for Chest
X-ray Image Diagnosis**

(胸部X線画像診断のための疑似ラベリング機構付き深層学習に関する研究)

March, 2024

Author:

Gerdprasert THANAWIT

Supervisor:

Dr. Shingo MABU

*A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Engineering*

in the

**Biomedical Information Systems Engineering Laboratory
Graduate School of Sciences and Technology for Innovation**

Declaration of Authorship

I, Gerdprasert THANAWIT, declare that this thesis titled, “Study on Deep Learning with Pseudo-labeling Mechanism for Chest X-ray Image Diagnosis” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

GERDPRASERT THANAWIT

Abstract

Deep learning, a sophisticated subset of machine learning, has significantly impacted various sectors, notably within computer vision. This paradigm involves algorithms adept at autonomously deciphering patterns from vast datasets. In computer vision, these algorithms process visual information, extracting invaluable insights. The healthcare domain stands as a prime beneficiary, heralding the era of Computer-Aided Diagnosis (CAD).

Equipped with deep learning capabilities, CAD systems are increasingly assisting medical professionals. By analyzing medical imagery—ranging from X-rays to MRIs—these tools detect anomalies, offer diagnostic recommendations, or foresee certain diseases. While not designed to supersede medical expertise, CAD systems amplify it. Acting as a secondary diagnostic layer, they bolster accuracy and timeliness, fostering enhanced patient care.

Yet, the efficiency of deep learning models in CAD hinges on access to comprehensive labeled datasets. These entail image annotation (labeling) with relevant details, like disease presence or tumor location. Given the precision required, medical experts play a central role in this labeling, rendering the process resource-intensive. Consequently, the problem of a paucity of adequately labeled medical imagery occurs. Ethical concerns related to patient data further complicate data availability for research.

In light of this data challenge, the spotlight has shifted to semi-supervised learning. Semi-supervised learning leverages both labeled and unlabeled data. As one of the semi-supervised learning methods, pseudo-labeling emerges as a standout method in this domain. Initially, a model is trained on available labeled data. Following this, it predicts labels for the unlabeled data. The predicted pseudo-labels then augment the original training data and refine the model iteratively.

However, pseudo-labeling in medical imaging presents challenges. The accuracy

of pseudo-labels is paramount—if the predictions of the initial model skew, it risks embedding biases or inaccuracies. Additionally, how to set appropriate confidence levels for pseudo-labels, how to mitigate biases from dataset imbalances, and how to prevent the model from overfitting are areas of concern.

This dissertation deeply delves into the pseudo-labeling techniques within the medical imaging context. Through meticulous research, it aims to enhance pseudo-labeling mechanisms, confront its inherent challenges, and assess its benefits in CAD. By bridging deep learning and medical diagnosis, this study aspires to advance healthcare technology, championing more precise and efficient diagnostic methodologies.

Acknowledgements

I would like to extend my deepest gratitude to a number of people without whom this dissertation would not have been possible.

First of all, I would like to express my gratitude to Professor Shingo Mabu, who gave generous advice and suggestions and assisted me throughout the research. I am deeply indebted to his wisdom and encouragement; it has left an indelible mark on this work.

Secondly, I would like to thank the professor, seniors, and peers in the laboratory who provided much advice, comments, and support throughout the research, and my friends in the department of software who gave me suggestions, comments, and solutions to many problems that occurred during the research.

Thirdly, I would like to thank the JST scholarship under Training Doctoral Students through Interdisciplinary Research Practices Programs for the financial support and funding of this dissertation.

Thirdly, I would like to thank the Department of Information Science and Engineering for the working environment and many facilities for me to conduct my thesis.

Last but certainly not least, my heartfelt thanks go to my family. Especially to my late grandmother, who was born in a rough environment, led to the demand that education be one of the most important individual qualities that push me to this day.

Thank you all from the bottom of my heart.

Contents

Declaration of Authorship	i
Abstract	ii
Acknowledgements	iv
1 Introduction	1
1.1 Research background and motivation	1
1.2 Research objective	3
1.3 Semi-supervised learning	4
1.4 Advancement of artificial intelligence in medical image analysis . .	5
1.5 Objective and structure of this dissertation	7
2 Object detection for chest X-ray image diagnosis using deep learning with pseudo labeling	9
2.1 Chapter introduction	9
2.2 Object detection of deep learning	10
2.3 Proposed method	13
2.4 Experiments	14
2.4.1 Dataset	14
2.4.2 Evaluation criteria	16
2.4.3 Comparison between object detection architectures	18
2.4.4 Comparison between different numbers of labeled samples .	19
2.4.5 Evaluation of pseudo labeling approach	20

2.5	Limitation and Future work	22
2.6	Chapter summary	25
3	Pseudo-labeling with contrastive perturbation using CNN & ViT for chest X-ray classification	26
3.1	Chapter introduction	26
3.2	Related work	27
3.2.1	Comparison between CNN and ViT	28
3.2.2	Consistency regularization	29
3.3	Proposed method	32
3.3.1	Pseudo-labeling with contrastive perturbation using CNN & ViT for chest X-ray classification	32
3.3.2	Dataset	33
3.4	Experimental results	34
3.4.1	Comparison between difference deep learning architectures on COVID-19 classification task	35
3.4.2	Evaluation of various augmentation combinations and correctness of pseudo-labeling	36
3.4.3	Evaluation of pseudo-labeling with contrastive perturbation	37
3.4.4	Limitation	39
3.5	Chapter summary	41
4	Ensemble learning of pseudo-labeling framework for chest X-ray image diagnosis	43
4.1	Chapter introduction	43
4.2	Ensemble learning frameworks	44
4.2.1	Techniques of classification ensemble	44
4.2.2	Ensemble technique for object detection	46
4.3	Proposed method	47

4.3.1	Ensemble learning for improving pseudo-labeling for object detection	47
4.3.2	Ensemble learning for improving pseudo-labeling for classification	49
4.4	Experimental results	51
4.4.1	Evaluation of ensemble learning for disease area detection	51
4.4.2	Evaluation of ensemble learning for classification	53
4.5	Chapter summary	54
5	Conclusions	56
	Bibliography	58

List of Figures

2.1	Overview of two-stage object detection framework	11
2.2	Overview concept of the sliding windows in one-stage object detection	12
2.3	Structure of Feature Pyramid Network backbone architecture	12
2.4	Flowchart of pseudo-labeling process.	14
2.5	Different traits of X-ray lung opacity. Image (a) is white lung, image (b) is other masses and modules opacity, image (c) is enlarged heart opacity, and image (d) is consolidation type opacity	17
2.6	The calculation formula for the Intersection over union. The dotted box represents the detected bounding box, and the solid box represents the ground truth.	18
2.7	Confusion Matrix in object detection	18
2.8	Example of different detection results. The dotted line represents the model's prediction, and the solid line represents the ground truth. Image (a) is true positive, image (b) is false positive, and image (c) is false negative	18
2.9	Example Plotting to calculate Average Precision	19
2.10	The overall datasets have been split into labeled, unlabeled, and test portions for the experiment.	20
2.11	X-rays images with abnormal postures	23
2.12	X-ray images with black background noises	24
3.1	The structure of ConvNext from PyTorch library	30
3.2	The structure of Swin Transformer from PyTorch library	30

3.3	Difference between Swin Transformer and Visual Transformer	31
3.4	The overall flowchart of Pseudo-labeling with Contrastive Perturbation.	32
3.5	Class distribution in RSNA COVID-19 Challenge.	35
3.6	The example of augmented images used for training	38
3.7	The ratio of the size of the disease areas to the whole image.	41
4.1	Comparison of the bounding box ensemble process between WBF and NMS	49
4.2	The Ensemble Learning for Pseudo-labeling with Contrastive Pertur- bation flowchart.	51
4.3	Example of the ensemble result using RetinaNet and YoloV5	53

List of Tables

2.1	Comparison of mAP between object detection architectures	19
2.2	Comparison of mAP between different numbers of labeled samples obtained by RetinaNet	21
2.3	mAP obtained from different confidence thresholds	21
2.4	Investigation on the amount of pseudo-labeling in each iteration . . .	21
2.5	The object detection performance based on different amounts of pseudo- labeled data	22
3.1	Characteristics of each class	34
3.2	Comparison of precision, recall, F1 score, and accuracy obtained by ResNet, ViT, Swin Transformer, and ConvNext [%]	36
3.3	Comparison of pseudo-labeled sample accuracy from various aug- mentation combinations	37
3.4	Evaluation of Pseudo-labeling with Contrastive Perturbation [%] . . .	39
4.1	Comparison between pseudo-labeling methods with/without ensem- ble method. Parenthesis in the table shows the difference between baseline and ensemble method	52
4.2	Evaluation of Ensemble Learning for Pseudo-labeling with Contrastive Perturbation [%]	54

List of Abbreviations

WBF	Weight Box Fusion
CNN	Convolutional Neural Network
ViT	Visual Transformer
NMS	Non Maximum Suppression
CAD	Computer Aid Diagnosis
MRI	Magnetic Resonance Imaging
PNG	Portable Network Graphic
CT	Computed Tomography

Chapter 1

Introduction

1.1 Research background and motivation

Deep learning has provided superior solutions or assistance for many ongoing problems in the current era. One of the well-known fields where deep learning succeeded is computer vision [1]; this is due to its ability to learn the features of any given dataset. This strength leads to many active types of research in the deep learning field and also creates influence on other fields. For example, medical field[2],[3] has been influenced by the advancement of deep learning[4],[5], leading to more usage of computer-aided diagnosis (CAD)[6]. Even though CAD performance might not be equally matched to the real experts in the field, the second opinion that CAD provides helps lessen the burden of the experts. Nowadays, we can see more and more CAD that is integrated into our disease diagnosis, and the quality of healthcare has been significantly improved. However, for a deep learning model to be effective, one of the major requirements is a sufficient amount of training data, which can be challenging to obtain in some fields, like the medical field. The main problem for data acquisition in the medical field is the requirement for labeling data (annotation) using precious human resources. Therefore, many kinds of research that try to alleviate this problem have been conducted[7]; a well-known one is semi-supervised learning. Unlike supervised learning, the goal of semi-supervised learning is to develop a

model from not only the labeled samples (data) but also unlabeled samples. Semi-supervised learning techniques are commonly applied when there is only a handful of labeled samples but an abundant amount of unlabeled samples. This leads to various techniques that aim to make the most use of the labeled samples[8] or try to learn or extract the essential features from unlabeled samples for realizing objectives[9].

One of the well-known semi-supervised learning techniques is pseudo-labeling. The aim of pseudo-labeling is to perform the labeling task on unlabeled data instead of relying only on field experts. In detail, a model is trained with a limited number of labeled data to create an expert model or a teacher model. Then, the expert model gives "pseudo" labels to unlabeled data. Finally, pseudo-labeled data are added to the original labeled dataset, and the model is trained again with the newly created dataset. Since there is an assumption that deep learning performs better as the number of data increases, the model trained with the combination of the pseudo-labeled and the original labeled data should perform better than a model trained with only inadequate labeled data.

Pseudo-labeling is a simplistic yet effective semi-supervised learning method. However, there are also many challenges regarding pseudo-labeling, such as finding the optimal threshold for accepting annotation given for the unlabeled data and the expert model's inability to annotate unlabeled data properly due to the lack of labeled data. Another challenge is to address the bias that occurs from the pseudo-labeling process, which arises when specific types of traits are in the dataset, and they cause the training bias of the generated expert model. This problem can be caused by various factors, such as an imbalanced dataset, incorrect annotation of pseudo-labeled data, and specific prominent traits when training an expert model.

In this dissertation, I focused on designing and enhancing the pseudo-labeling mechanism in the medical imaging field. I believe that applying pseudo-labeling to building a CAD model of medical image diagnosis can improve the overall performance

by making better use of the unlabeled data that are usually discarded since the labeling cost is too expensive. In addition, I also use Chest X-ray images as the target datasets because X-ray images are the commonly used medical images, for example, annual medical checkups.

1.2 Research objective

This dissertation aims to explore the profound impact of pseudo-labeling in computer vision with deep learning, emphasizing its transformative role in medical diagnostics. It delved into the challenges of semi-supervised learning and critically examined the pseudo-labeling techniques, especially their application and enhancement in medical imaging. The objective of this dissertation is as follows:

- Introducing and exploring the concept of pseudo-labeling and deep learning.
- Applying deep learning with pseudo-labeling to two medical diagnostic problems.
 - Object (disease area) detection using deep learning with pseudo-labeling
 - Classification using Deep Learning with Pseudo-labeling
- Enhancing the proposed pseudo-labeling framework.
 - Improvement in object detection framework using the iterative process for increasing model robustness
 - Improvement in Classification framework by integrating consistency regularization.
- Summarizing the findings and future extension of the proposed methods

1.3 Semi-supervised learning

Semi-supervised learning has been a topic of interest in addressing the lack of labeled training samples. Semi-supervised learning aims to train a model using a limited amount of given training samples along with the abundance of unlabeled samples[10].

Given the broad scope of the problem, there are many approaches to the semi-supervised learning methods[11], such as the aim of using the limited training data as sufficient as possible while trying to yield the best possible results, e.g., bootstrapping[12], self training[13], or using both labeled, unlabeled samples to make the best use of the information contained in those samples. Furthermore, semi-supervised learning also reflects many real-world problems in which there are many unlabeled samples, but the labeling cost is expensive.

Pseudo-labeling

One of the common approaches to semi-supervised learning is pseudo-labeling. As proposed by Lee et al. [14], pseudo-labeling can be categorized as a self-learning technique that lets the model learn and improve by itself. The process of pseudo-labeling is to generate the pseudo-labels for the unlabeled samples using the model trained on the limited amount of labeled samples. The purpose of pseudo-labeling is to train the model, usually named the expert model or teacher model, using the available labeled samples, then let this model perform the labeling task for the unlabeled samples. Lastly, the model is then trained using the original limited labeled samples along with the newly labeled samples generated by the expert model.

Overall, pseudo-labeling is a straightforward yet effective method for utilizing unlabeled samples, which can be applied to many situations. Moreover, thanks to its simplicity, pseudo-labeling can be incorporated into other machine learning techniques, including deep learning.

However, the commonly known problem in pseudo-labeling is the teacher bias during the labeling process[15],[16]. The teacher bias problem occurs when ineffective unlabeled data is incorporated into the training process. Since the number of unlabeled samples usually overwhelms the labeled ones, the model generalization will be shifted toward the unlabeled samples instead, resulting in the model performing worse.

Many researchers have tried to solve the teacher problem in recent studies using various approaches. One of the successful examples is incorporating the consistency regularization [17],[18] concept during model training. The model performs significantly better by applying augmentation or perturbation to the pseudo-labeled samples and making the model try to learn from the perturbed and original samples. However, the main limitation of the consistency regularization technique is that it relies on various augmentation techniques, which may even lead to a bias toward the augmented data.

1.4 Advancement of artificial intelligence in medical image analysis

The concept of medical image analysis was introduced in the 1990s, with the origin of X-ray images dating back to 1895, which let us view the inside of the human body without the need for a surgical process. Following the X-ray, both MRI (Magnetic Resonance Imaging) and CT (Computed Tomography) images were introduced, each with its own strength and use. CT lets the radiologist capture the body's cross-sectional image and a clear view of the soft tissue. On the other hand, MRI uses magnetic fields and radio waves to produce detailed images of the inside of the body. Medical image analysis in this era was done mainly manually by radiologists or clinicians; however, with the introduction of digital image processing techniques, such as thresholding, edge detection, or morphological operations, the efficiency of medical

image analysis has increased tremendously. This is when the Computer Aided Diagnosis or CAD system was developed. CAD refers to the use of computer algorithms to assist radiologists and medical professionals in interpreting medical images. Its development has been closely linked with advancements in medical imaging, computational power, and artificial intelligence.

It was not until the 2000s that machine learning started to integrate into medical image analysis successfully. While there are many machine learning techniques that have become popular, the one that stands out the most is the Support Vector Machines (SVM). SVM is a very powerful machine learning framework that can handle high-dimensional data and have clear margins of decision boundary, making it suitable for lesion detection or tissue classification. In addition, there were many public datasets that were created to be used for benchmarking the machine learning performance.

Starting from the 2010s, the AI boom has become prevalent with the rise of deep learning, mainly convolutional neural networks (CNN). A major advantage of CNN is its ability to automatically learn hierarchical feature representations from data, which reduces the need for hand-engineered features. Thanks to the mentioned reason, CNN has outperformed traditional methods in many tasks, including image classification, segmentation, and object detection. Deep learning also benefits heavily from other fields with the usage of data augmentation or transfer learning, which is the technique of learning from common datasets and fine-tuning for specific medical tasks. Nowadays, deep learning has been integrated into CAD to help the expert in the field with widespread tasks such as cancer diagnosis. As technology advances, it is likely that CAD systems will become even more integrated into regular clinical workflows, offering more advanced features and better performance.

However, as mentioned above, the most important problem in deep learning for medical image diagnosis is to collect a sufficient amount of labeled data. Therefore, in this dissertation, I focused on pseudo-labeling and its extension using ensemble learning to further enhance the applicability of deep learning to CAD, especially in the Chest

X-ray disease diagnosis task. The benefit of X-ray images is the accessibility of a large amount of sample numbers, thanks to the fact that X-ray check-ups occur daily and are cost-efficient. However, the X-ray diagnosis proves to be a challenging task due to the image's opacity being hard to discern, even from the field expert's perspective. Thus, the possibility of developing a CAD system for X-ray diagnosis might prove to be difficult due to how complicated the X-ray image is, but it will definitely improve the ability of the disease diagnosis for certain.

1.5 Objective and structure of this dissertation

The structure and the objective of each chapter are summarized in this section. Each section will guide readers through the theoretical background, the proposed methodology, experimental findings, and conclusion of findings.

In Chapter 1, I have provided an overview of the research background and introduced the core concept that is commonly considered in this dissertation, that is, general knowledge of pseudo-labeling and medical images.

In Chapter 2, the first proposed method, "Object detection using deep learning with pseudo-labeling on X-ray pneumonia disease area detection," is presented. In this chapter, the applicability of pseudo-labeling for disease area detection is evaluated. In detail, the created model aims to locate the disease area and the disease class simultaneously. First, I briefly introduce the fundamentals regarding object detection, such as the well-known architectures and evaluation criteria. Then, I introduce an iterative pseudo-labeling mechanism to improve the stability of the detection performance. In detail, the pseudo-labeled data are gradually generated, and the detection model is also gradually trained with the combination of labeled and pseudo-labeled data. If we train the detection model only once after obtaining all the pseudo-labeled data, the incorrectly labeled pseudo-labeled data may deteriorate the model; thus, the implementation of an iterative process (gradual training) is adopted.

In Chapter 3, "Disease classification using deep learning with pseudo-labeling," I deal with the topic of classification instead of detection. In this chapter, I propose a pseudo-labeling for classification framework that uses two different deep learning architectures, namely convolutional neural network (CNN) and vision transformer (ViT), and also incorporates another semi-supervised learning technique called consistency regularization. The core concept of consistency regularization is that in order for pseudo-labeled data to be reliable to include in the training dataset, the prediction results obtained by both CNN and ViT must be the same. In addition, the robustness of the pseudo-labeling mechanism is realized by applying data augmentation. A detailed explanation of each architecture and the concept of consistency regularization is also provided in this chapter, as well as the experimental results implemented on COVID-19 chest X-ray images.

In Chapter 4, I propose an enhanced pseudo-labeling framework using ensemble learning. While ensemble has been a common technique applied to any machine learning techniques, in pseudo-labeling, I show the effectiveness of ensemble in the pseudo-labeling mechanism. Ensemble learning can solve the problem of models trained with pseudo-labeled data tending to be biased toward specific latent features in pseudo-labeled data. In addition, I design ensemble methods for solving object detection tasks and classification tasks, respectively, considering the characteristics of each task.

I finished this dissertation with conclusions in Chapter 5. I conclude and summarize the findings, the significance of the studies, and the limitations. I also suggest further research and potential that could be achieved by the pseudo-labeling in the medical field.

Chapter 2

Object detection for chest X-ray image diagnosis using deep learning with pseudo labeling

2.1 Chapter introduction

In this chapter, I would like to implement pseudo-labeling in object detection tasks of deep-learning for chest X-ray disease area detection. As I explained the concept of pseudo-labeling in the previous chapter, since pseudo-labeling has the possibility to improve the performance of any deep learning framework, I would like to try the simple yet effective implementation of object detection tasks with the additional enhancement of the pseudo-labeling framework. The proposed method introduces the iterative pseudo-labeling process, which focuses on improving the pseudo-labeling by reducing the bias that occurs during the training of the pseudo-labeling model, where the model is trained on the combination of pseudo-labeled samples and limited labeled samples. The goal of the proposed method is to alleviate the problem when there are specific and common characteristics or features in pseudo-labeled samples that could cause the deterioration in the model performance. By implementing the iterative process, the model that performs a pseudo-labeling task is calibrated using

the labeled dataset in every step. For evaluating the proposed method, the Chest X-ray image for Pneumonia Disease area detection is chosen as the target problem. The specific implementation of the proposed method will be described in later sections.

This chapter is organized as follows: I will start the chapter by briefly introducing the concept of object detection in deep learning and the candidate frameworks that I used for implementation, which are YoloV5, Faster RCNN, and RetinaNet. In addition, the information on evaluation criteria, which is named mean average precision or mAP, is described, including a calculation method. In the next section, the proposed method that implements the iterative process of pseudo-labeling and its application to the object detection framework are explained. Next, the experimental results obtained by the proposed method are shown, which includes the performance comparison between several object detection models. In the last section, I summarize the findings and describe the discussion.

2.2 Object detection of deep learning

Deep learning has revolutionized the next level of computer vision, including object detection. Instead of annotating the main features of each individual image in the datasets, deep learning can learn the image features by itself using the loss function during the training process. With this unique strength, deep learning has outperformed most machine learning methods in the object detection field. Since object detection has been widespread in deep learning, many architectures have advantages and specialties, but the architectures are usually categorized into two types: two-stage object detection framework and one-stage object detection framework.

The first well-known object detection framework started with the region-based architecture. The region-based architecture is a two-stage object detection framework that uses two neural networks: a region proposal network or backbone network and

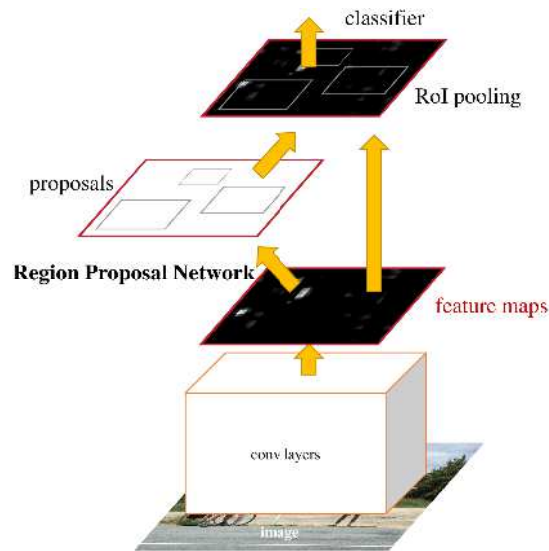


FIGURE 2.1: Overview of two-stage object detection framework

a box classification network or a head network. The first network proposes the regions of interest, and the regions are passed to the second network that performs a box regression and object classification. While this architecture allows the two networks to be trained separately for better performance, the training process is time-consuming. The two-stage object detection framework that I implemented in this research is Faster Region Convolution Neural Network (Faster R-CNN)[19], which improved the detection speed by removing the bottleneck that occurs in the process of the region proposal network. The model structure of the Faster RCNN can be seen in Fig. 2.1. From the figure, it can be seen the input image was changed to a feature map and then passed to two networks: one for the box regression task and the other for the classification task.

The following successful framework is You Only Look Once (Yolo)[20], which is one-stage object detection. Instead of using a region proposal network, Yolo uses sliding windows that are bounding boxes with predetermined sizes and different anchors. The windows are slid through the whole image. Each bounding box is used as an input to the neural network to perform the object detection task. Figure 2.2 can be used as an example of how the sliding windows work; each grid will be used for finding the bounding box class probability together to find the final detection result.

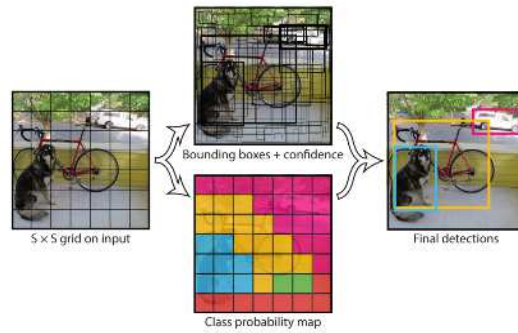


FIGURE 2.2: Overview concept of the sliding windows in one-stage object detection

Furthermore, by eliminating the region proposal network, Yolo can perform a detection task in a real-time scenario and thus receive much support from the research community. The Yolo version implemented in this research is YoloV5[21], which introduces many features compared to the original Yolo. For example, the architecture is tremendously changed by using Cross Stage Partial Network[22] as the backbone for feature extraction instead of the original Darknet[23] and also using Feature Pyramid Network (FPN)[24] for generating bounding boxes. The structure of FPN can be seen in Fig. 2.3. FPN is the well-known backbone of the CNN for the feature extraction task. The strength of FPN is the hierarchy learning ability of bottom-up and top-down architecture that lets the information be recycled and used throughout the network.

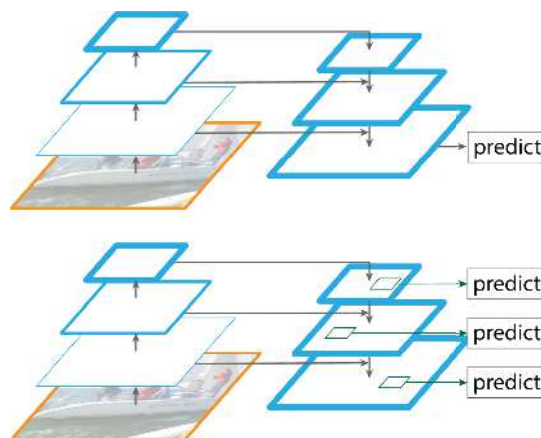


FIGURE 2.3: Structure of Feature Pyramid Network backbone architecture

In addition, the significant improvement is the usage of various augmentation and auto-learning anchors that can adapt to many types of problems.

While the speed of one-stage detection is the main advantage, the main problem is that the number of negative samples (areas) that come from the background of the image during the region proposal overwhelms the number of positive samples. RetinaNet[25] addresses this problem by introducing a new loss function called Focal loss, which introduces the weight parameter applied to the cross-entropy loss function. Focal loss tremendously penalizes the background loss, leading to the model performing significantly better. RetinaNet also utilizes FPN for the region proposal task.

2.3 Proposed method

Object detection using deep learning with pseudo labeling

In order to utilize the unlabeled samples to improve the performance of object detection, I propose a pseudo-labeling method. The diagram and pseudo code of the proposed method are shown in Fig. 2.4 and Algorithm 1, respectively. As a preparation, I split the dataset into a labeled training dataset, an unlabeled training dataset, and a test dataset to replicate a semi-supervised learning environment. In addition, since the pseudo-labeling technique tends to have a high bias, I split all of the unlabeled datasets into smaller batches instead of using them all at once.

First, the base models (YoloV5, Faster R-CNN, and RetinaNet) are trained using all the accessible labeled data. After the initial training phase is finished, the trained model can be used as the expert model to perform the labeling task for us. Next, the expert model performs the labeling task on the unlabeled samples. Then, if an unlabeled sample is classified with high confidence by the expert model ($T > \alpha$), it is considered an acceptable pseudo-labeled sample; else, the sample ($T \leq \alpha$) is discarded.

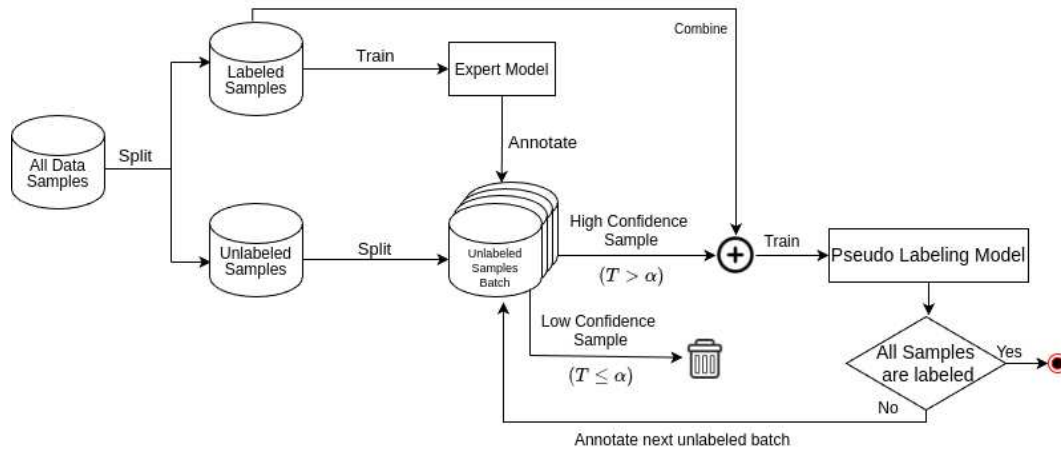


FIGURE 2.4: Flowchart of pseudo-labeling process.

After the pseudo-labeling task for one batch of unlabeled samples is finished, the new model is trained with new pseudo-labeled samples combined with the original labeled samples. Then, the newly trained model performs the annotation to the next batch of the unlabeled samples and repeats the process until all of the unlabeled samples are processed. When the labeling phase is completed, the final expert model is trained using all the pseudo-labeled samples and the originally labeled samples.

The main reason I perform pseudo-labeling batch by batch is not to overwhelm the less influenced by the pseudo-labeled samples compared with the labeled samples. If the pseudo-labels are given to all the unlabeled samples at the same time, the weight of the pseudo-labeled samples on the training loss becomes suddenly large, which results in worse outcomes. Therefore, the training of the expert model should be gradually implemented.

2.4 Experiments

2.4.1 Dataset

The open dataset of chest X-ray images provided by the Radiology Society of North America¹ is used for the experiments in this chapter. Originally, the dataset was

¹<https://www.kaggle.com/c/rsna-pneumonia-detection-challenge>

Algorithm 1: Pseudo-Labeling for Object Detection

Result: Trained pseudo-labeling model

initialization;

for *all available batches of unlabeled data* **do**
 use the previously trained expert model to perform class labeling to
 unlabeled batch;

for *every single label in the batch, given the confidence of T* **do**
if $T > \alpha$ **then**

| add the data to the pseudo-labeled list;

else

| skip the sample;

end
end

 retrain the expert model again using the original samples and samples in a
 pseudo-labeled data list ;

evaluate the retrained expert model;

end

used for the competition of the chest X-ray Pneumonia disease area detection challenge, but I used it for another problem, that is, the evaluation of the pseudo-labeling method. While a chest X-ray image is difficult to classify due to its many traits and characteristics, it is the most common diagnostic method due to its simplicity and inexpensiveness, resulting in abundant unlabeled samples. Therefore, semi-supervised learning with a pseudo-labeling approach is very effective for building a CAD for chest X-ray images.

Pneumonia disease detection aims to detect abnormal opacity in the X-ray images. However, there are many difficulties in pneumonia detection. While pneumonia is commonly found in the lung area, the actual disease area can be varied by the patient's posture during the X-ray scan, making it difficult to locate or crop the lung area in data preprocessing. In addition, pneumonia disease also has many characteristics that can also be associated with different body traits, and examples can be seen in Fig. 2.5. Image a) is an example of white lung or hemithorax opacity where there is high opacity in the lung area, which is caused by the fluid from pneumonia. Image b) is an example of high opacity that came from masses and nodules but not pneumonia. Image c) is an example of a normal patient with an enlarged heart, which can be seen

as high opacity. Lastly, Image d) is an example of consolidation-type pneumonia. These variations of lung features make abnormal area detection difficult compared to more medical image types, such as CT images, where the patient posture is always fixed.

The dataset includes 6,011 images, where 30,228 pneumonia locations are recorded; that is, a single image can contain multiple pneumonia locations. I used only the positive images and filtered out the images with a bounding box (positive area) of fewer than 8 pixels for training. This process eventually led to the training dataset of 4,400 images and 6,027 pneumonia locations (bounding boxes). The size of each sample image was 1024×1024 , with a bounding box size according to its size. The test set contained randomly chosen positive and negative samples with its bounding box. The bounding box is attached only when the image is a positive case. The image size is 1024×1024 , the same as the training set.

2.4.2 Evaluation criteria

In the experiments, mean average precision (mAP) is used. The mAP was proposed as a standard for the PascalVOC challenge[26], but currently, mAP is commonly used as an evaluation metric in object detection problems[25], [20], [19]. mAP can be calculated by using true positive (TP), false positive (FP), and false negative (FN). However, unlike the classification problem, these criteria for the detection problem are calculated based on the Intersection over Union (IoU).

The IoU is calculated by comparing the bounding boxes predicted by the model and the ground truth (GT). The figure of the IoU calculation can be seen in Fig. 2.6, where A and B represent the bounding boxes from the ground truth and detection result, respectively. A detection result is regarded as TP when the bounding boxes of GT and prediction are overlapped and IoU is higher than the predetermined threshold. A detection result is regarded as FP when IoU is less than the threshold or the model generates duplicate bounding boxes at the same location as the GT bounding

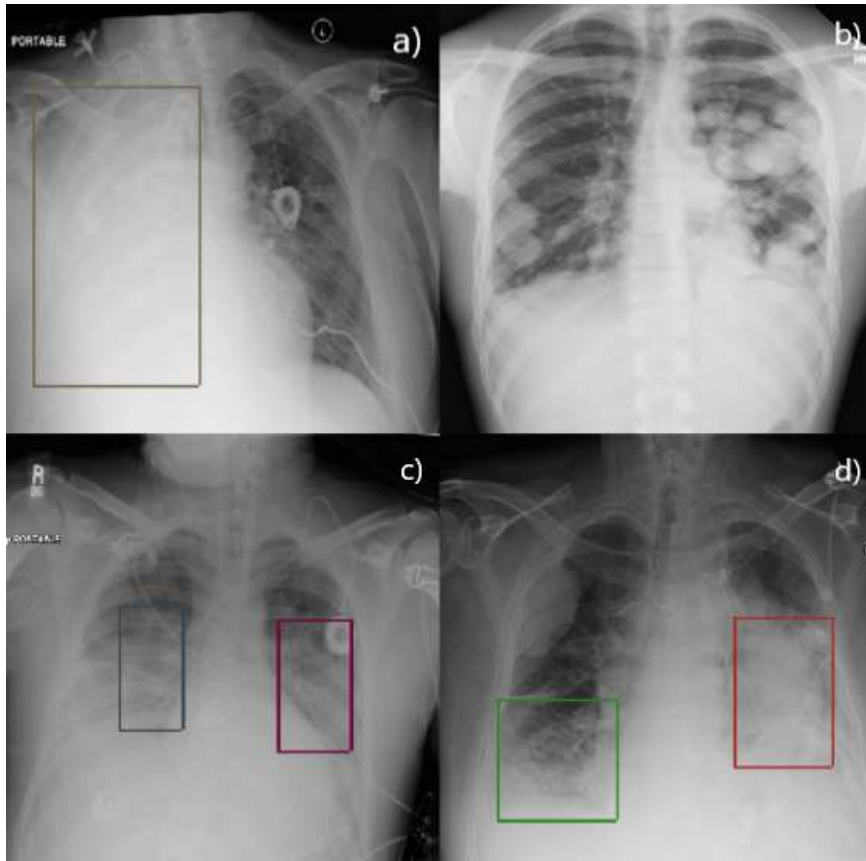


FIGURE 2.5: Different traits of X-ray lung opacity. Image (a) is white lung, image (b) is other masses and modules opacity, image (c) is enlarged heart opacity, and image (d) is consolidation type opacity

box. Lastly, FN can occur when there is no predicted bounding box when the GT bounding box is presented or the predicted class of the bounding box is incorrect. The confusion matrix for the object detection task can be seen in Fig. 2.7, showing how the box is categorized. In addition, examples of each type of detection result in the X-ray image format can be viewed in Fig. 2.8.

After calculating TP, FP, and FN, we can calculate the precision and recall from detection results, which will be used for plotting the Precision-Recall Curve. Finally, the mAP is calculated from the area under the curve of the Precision-Recall Curve as this formula Average Precision = $AP = \int_0^1 P(R) dR$. The example of the mAP calculation can be seen in Fig. 2.9.

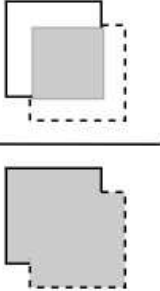
$$\text{Intersection over union (IoU)} = \frac{A \cap B}{A \cup B} = \frac{\text{Intersection Area}}{\text{Union Area}}$$


FIGURE 2.6: The calculation formula for the Intersection over union. The dotted box represents the detected bounding box, and the solid box represents the ground truth.




	Ground Truth is in the predicted area	Ground Truth is not in the predicted area
Prediction Found	True Positive ($\text{IoU}_{\text{pred}} > \text{IoU}_{\text{threshold}}$) 	False Positive ($\text{IoU}_{\text{pred}} < \text{IoU}_{\text{threshold}}$) 
Prediction not found	False Negative 	True Negative

FIGURE 2.7: Confusion Matrix in object detection

2.4.3 Comparison between object detection architectures

In this experiment, YoloV5, Faster R-CNN, and RetinaNet are compared in terms of mAP to find the best model to carry on the pseudo-labeling process in further experiments. All the models are trained for 100 epochs with various augmentation techniques. The applied augmentation techniques are the following: brightness adjustment from 10 to -10, 50% chance of horizontal flipping, and 10 degrees rotation

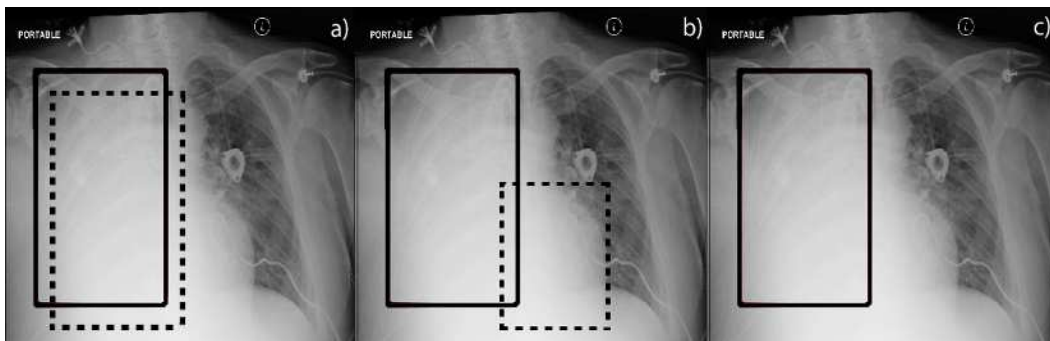


FIGURE 2.8: Example of different detection results. The dotted line represents the model's prediction, and the solid line represents the ground truth. Image (a) is true positive, image (b) is false positive, and image (c) is false negative

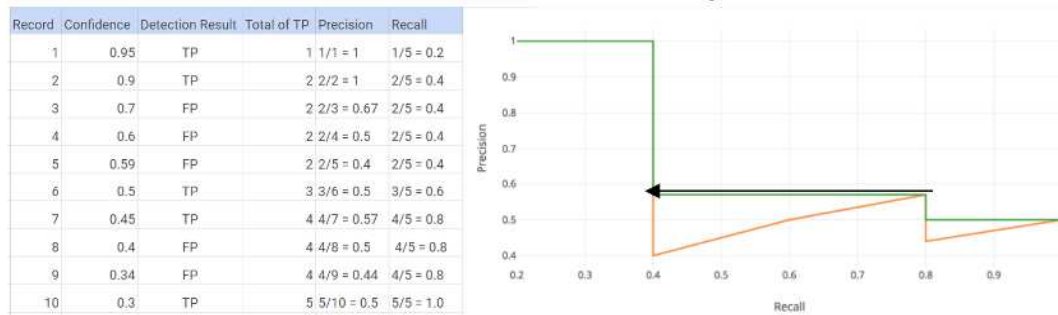


FIGURE 2.9: Example Plotting to calculate Average Precision

TABLE 2.1: Comparison of mAP between object detection architectures

Architecture	mAP for difference IoU thresholds					
	0.1	0.2	0.3	0.4	0.5	0.6
Faster RCNN	50.89	49.98	45.47	34.31	7.560	1.550
YoloV5	60.27	60.13	59.09	55.38	48.39	37.26
RetinaNet	85.12	85.12	85.12	85.12	84.96	84.28

either clockwise or counterclockwise. The experimental result is shown in Table 2.1, where it can be seen that RetinaNet performs the best, followed by YoloV5. Thus, I adopted RetinaNet as the main model for the pseudo-labeling task and YoloV5 as the supporting model for the ensemble method proposed in Chapter 4.

2.4.4 Comparison between different numbers of labeled samples

We simulated a semi-supervised learning problem by splitting the whole dataset into a labeled dataset, an unlabeled dataset, and a test dataset. I fixed the test dataset as 10% of the whole dataset, and this experiment was conducted by changing the ratios of labeled and unlabeled samples. Table 2.2 shows the result obtained by RetinaNet where it can be seen that the model trained using labeled data of 10% of the whole dataset (called 10% model) shows very low mAP and it is not suitable as the expert model because the pseudo-labeled samples will contain many incorrect labels. In addition, since the difference between the 40% model and 50% model is small, I chose only 30% and 50% models as the candidate for combining them with the pseudo-labeling method.

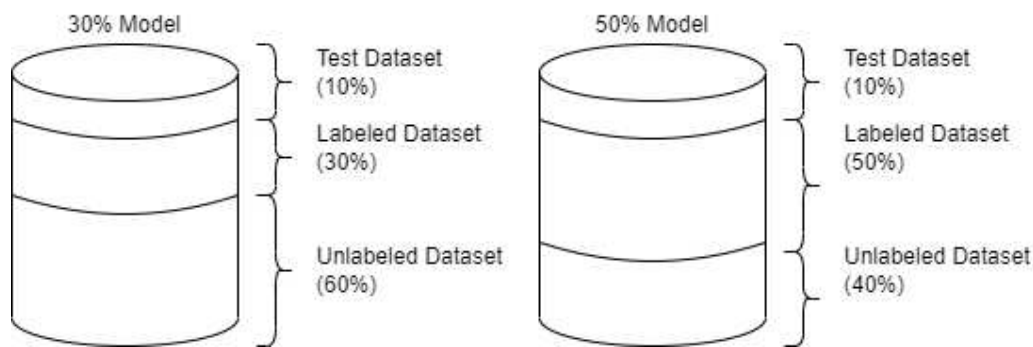


FIGURE 2.10: The overall datasets have been split into labeled, unlabeled, and test portions for the experiment.

2.4.5 Evaluation of pseudo labeling approach

In this experiment, the pseudo-labeling is carried out on the datasets with 30% and 50% labeled samples used in the previous experiment. The images of how I split the training data can be seen in Fig. 2.10. The models trained in the previous experiment were used to label the unlabeled data. Instead of labeling the unlabeled samples at once, the labeling process is split into some smaller subprocesses with 20% (i.e., 880 images) for each labeling process because I aimed to make the model converge over a part of samples in one-time training so that the bias of the unlabeled samples is not overwhelming the original labeled samples. In the pseudo-labeling process, samples with confidence scores more than a predetermined threshold were selected as samples with pseudo labels, and the rest were discarded.

We applied the pseudo-labeling to the 50% model and compared several confidence thresholds, that is, 0.7, 0.9, and 0.95. In addition, the numbers of pseudo-labeled samples that the model generated are changed (20% and 40%) to find the best amount.

From Table 2.3, it can be seen that mAP obtained by confidence threshold 0.95 is the best for all the IoU thresholds. Since the pseudo labels need to be the genuine labels as close as possible, and the low confidence can lead to a model performing worse with the pseudo-labeling [15], the threshold of 0.95 was selected. After the labeling process, the original ground truth samples were combined with the newly labeled samples. A new model with the same architecture and hyperparameters was trained

TABLE 2.2: Comparison of mAP between different numbers of labeled samples obtained by RetinaNet

No. of labeled samples	mAP for different IoU thresholds					
	0.1	0.2	0.3	0.4	0.5	0.6
10% (440 images)	33.57	24.82	18.00	10.79	8.860	8.350
30% (1320 images)	63.89	63.09	60.92	57.29	49.29	41.10
40% (1760 images)	70.87	70.61	68.93	64.13	56.35	44.72
50% (2200 images)	74.54	72.51	71.16	68.31	62.81	57.39

TABLE 2.3: mAP obtained from different confidence thresholds

Ratios of samples (original labels + pseudo labels)	Confidence threshold	mAP for different IoU threshold					
		0.1	0.2	0.3	0.4	0.5	0.6
50% (Baseline)	-	72.51	72.32	71.16	68.31	62.81	57.39
50% + 40%	0.70	33.57	24.82	18.00	10.79	8.960	8.350
50% + 40%	0.90	62.88	62.28	59.65	51.88	40.27	25.37
50% + 40%	0.95	75.37	75.37	75.15	73.41	71.48	68.78

as its original base model. The process repeated until all of the unlabeled samples were labeled.

To guarantee that the pseudo-labeling process successfully increases the number of pseudo-labeled samples, an investigation experiment is performed to find the pseudo-labeled samples in each iteration from both proposed models. The experimental result can be seen in Table.2.4, where the amount of each pseudo-labeled sample increases as the iteration increases. In addition, the 30% models also see more increase in the accepted samples compared to the 50% model counterpart.

The mAP obtained by 30% and 50% models when the numbers of pseudo-labeled

TABLE 2.4: Investigation on the amount of pseudo-labeling in each iteration

Model Name	Unlabeled Samples (#)	Accepted samples (#)	Total Samples
30% + 20%	20% (1,760 images)	482	482
30% + 40%		689	1171
30% + 60%		778	1949
50% + 20%		602	602
50% + 40%		642	1242

TABLE 2.5: The object detection performance based on different amounts of pseudo-labeled data

Ratio of Samples (original labels + pseudo labels)	mAP for difference IoU thresholds with performance difference					
	0.1	0.2	0.3	0.4	0.5	0.6
30% (Baseline)	63.89	63.09	60.92	57.29	49.29	41.10
30+20%	67.16 (+3.27)	67.16 (+4.07)	65.04 (+4.12)	61.28 (+3.99)	52.95 (+3.66)	44.03 (+2.93)
30+40%	67.93 (+4.04)	67.93 (+4.84)	65.94 (+5.02)	62.40 (+5.11)	53.90 (+4.61)	44.52 (+3.42)
30+60%	71.28 (+7.39)	71.10 (+8.01)	69.17 (+8.25)	65.09 (+7.80)	56.96 (+7.67)	46.42 (+5.32)
50% (Baseline)	72.51	72.32	71.16	68.31	62.81	57.39
50%+20%	74.54 (+2.03)	74.54 (+2.22)	74.47 (+3.31)	72.56 (+4.25)	70.01 (+7.20)	66.05 (+8.66)
50%+40%	75.37 (+2.86)	75.37 (+3.05)	75.15 (+3.99)	73.41 (+5.10)	71.48 (+8.67)	68.78 (+11.39)

samples are gradually increased can be seen in Table 2.5. Table 2.5 shows the improvement of mAP when the number of labeled samples increases with a comparison of the performance between pseudo-labeling and the baseline model shown in the parenthesis. The model trained on the 30% labeled samples combined with 60% pseudo labeled samples (30%+60% model) sees an increase in mAP by 7.39% at 0.1 IoU and 5.32% at 0.6 IoU. Additionally, the model trained on the 50% labeled samples combined with 40% pseudo labeled samples sees an increase in mAP by 2.86% at 0.1 IoU and 11.39% at 0.6 IoU. It proves that, without letting non-labeled samples be wasted, I can use the trained model to perform a labeling process on behalf of the experts to improve the model performance.

2.5 Limitation and Future work

While there are many challenges in machine learning, there are always limitations to some extent, depending on given datasets and environments. Some limitation of pseudo-labeling has been addressed in this dissertation, such as finding optimal confidence threshold and reducing model bias. There are still some difficult problems to be solved in the future. For example, there is a problem in the way of evaluation. How to set evaluation method or benchmark is difficult because the performance of any pseudo-labeling models depends on the accessible amount of labeled samples, making it difficult to find a common experimental setup that can be used as a benchmark, especially in a field like medicine where the dataset is already scarce enough.

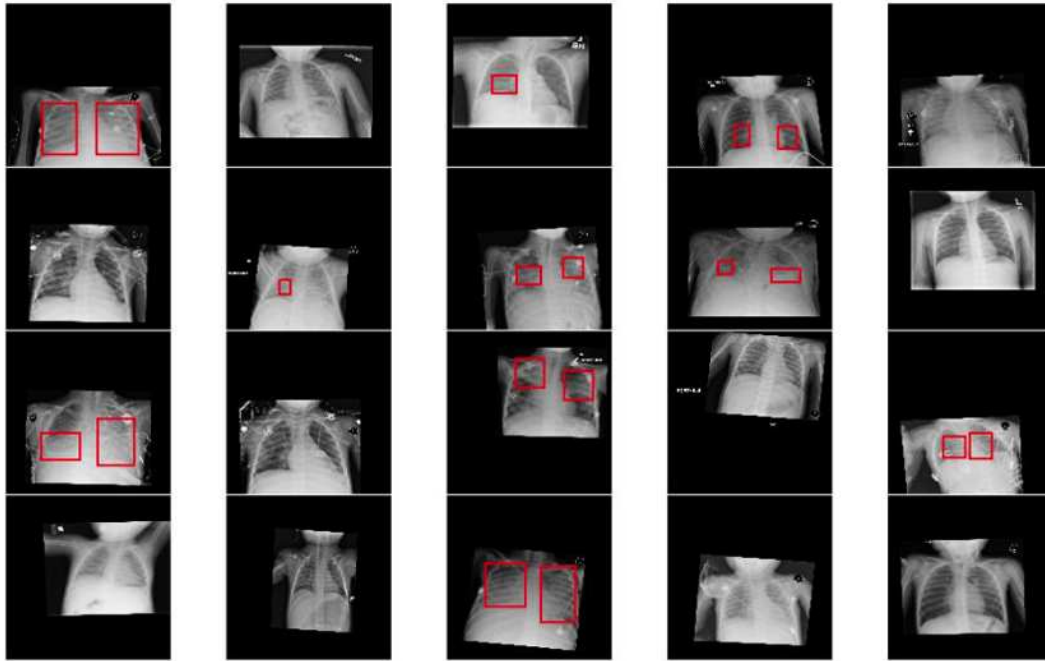


FIGURE 2.11: X-rays images with abnormal postures

Therefore, the advantages and disadvantages of each pseudo-labeling model should be carefully considered when it is applied to the target dataset.

Even though the aim of this dissertation is to find the correct classes for chest X-ray images since chest X-ray is the most common and realistic way to acquire a large number of unlabeled samples, There is also a problem in the nature of X-ray images. For example, since the diseases are located in the lung field, it is usually easy to roughly determine the target area to be analyzed when the postures of the patients are correctly adjusted, which is beneficial for removing the outliers. However, suppose the postures of the patients during the examination change the location of the diseases largely (Fig. 2.12). In that case, the difficulty of the prediction made by machine learning becomes increase. In addition, some images contain black background noise (Fig. 2.11) due to many reasons, such as differences in patients' body sizes. Also, many datasets of X-ray images are usually very imbalanced, for example, many normal images and few abnormal images.

During the pseudo-labeling process, there are many samples that are discarded due

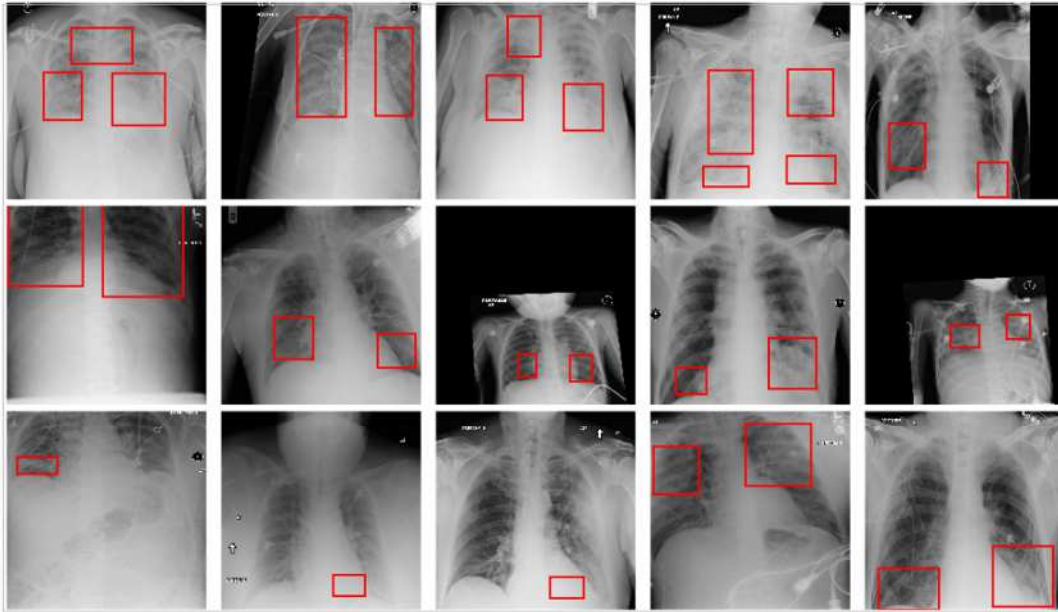


FIGURE 2.12: X-ray images with black background noises

to low confidence scores that do not surpass a threshold (the threshold in the experiments was set at 0.95 for all the proposed frameworks). Since the threshold should be relatively high to maintain the reliability of the pseudo labels, many correct labels, but having low confidence, are also discarded in the process. I believe that discarding all the pseudo-labeled samples in this manner is wasteful. It would be better if there could be a way to make use of the disease images or the regions of interest that contain lower confidence scores, for example, between 0.6 - 0.94. Actually, this could be done in many ways, such as implementing another model for these samples or using those samples anyway but giving penalties or weights to make sure that the model would not collapse.

As for ensemble learning, while the proposed method is applicable for the combination of multiple deep learning models, only two models were used for the ensemble currently. Therefore, it would be better to combine various models that perform well and eliminate the models that underperform compared to the rest of the models. This method is especially suitable for ensemble learning with contrastive perturbation, which relies heavily on the strength and uniqueness of each model.

There are still many traditional ways to improve the overall framework, such as transfer learning[27] or Test Time Augmentation[28]. In addition, it would be beneficial to introduce shared weight parameters obtained by the pseudo-labeling process or to design a better loss function that could prioritize the labeled samples with large importance or give different importance to pseudo-labeled samples.

2.6 Chapter summary

In this chapter, I introduced the implementation of the pseudo-labeling framework for the object detection task. The pseudo-labeling method successfully improved the disease area detection task by letting the model make use of unlabeled samples. In addition, the proposed framework with iterative pseudo-labeling methods further enhanced the robustness and achieved better performance. In the 30% model, 30% of the data were labeled for training by the radiologists, and in the 50% model, 50% of the data were labeled. The experimental results from both models showed performance improvement when more pseudo-labeled samples were used in the training process. Furthermore, I also carried out the experiment with the different object detection frameworks when the pseudo-labeling technique was applied and investigated the effects of the confidence threshold for the pseudo-labeling process.

This chapter has achieved the objective of illustrating that pseudo-labeling can be used in object detection tasks of the chest X-ray image dataset, which is a sensitive dataset but directly related to real-world application usage. In the next chapter, I would like to further design a pseudo-labeling technique for classification, where a new method is proposed to improve the quality of pseudo-labels under the same chest X-ray dataset used in this chapter.

Chapter 3

Pseudo-labeling with contrastive perturbation using CNN & ViT for chest X-ray classification

3.1 Chapter introduction

In the previous chapter, the pseudo-labeling concept in the object (pneumonia) detection problem of chest X-ray images was explored. This chapter will change the object detection task into the classification task instead. Image classification has always been a staple challenge in computer vision, where the goal is to assign labels or classes to the images. Since deep learning has the ability to learn the feature hierarchy and the representation by itself, it outperformed other traditional feature extraction techniques [29].

In this chapter, I propose a framework for enhancing pseudo-labeling for image classification tasks. The proposed method makes use of the other semi-supervised learning technique called Consistency Regularization, which focuses on the consistency of the image even if the perturbation or augmentation is applied to the original image. The proposed method for this chapter is called Pseudo-labeling with Contrastive

Perturbation. By introducing perturbation to the unlabeled samples, the classification model will have better generalization ability because the model has to discern between the original and perturbed version of that data. To strengthen the effects of consistency regularization, two types of deep learning architectures, convolutional neural network (CNN) and vision transformer (ViT)[30], are used for the pseudo-labeling process. In addition, since both architectures have their own strengths and uniqueness, I integrated a contrast augmentation regime for each neural network architecture to provide another layer of generalization. Lastly, to evaluate our proposed method, The SIIM RSNA Covid-19 Chest X-ray classification task[31] is used and the performance of the proposed framework is evaluated.

This chapter is organized as follows: first, the core difference between the traditional convolutional neural network and the visual transformer will be introduced and discussed. In addition, I will also briefly introduce the concept of Consistency Regularization in semi-supervised learning, which is the main strength of the proposed method. The next section is a detail of the implementation of the proposed method, called pseudo-labeling with contrastive perturbation, followed by the experimental results. In the last section of this chapter, I will provide a summary of the chapter and set up the discussion for the next chapter.

3.2 Related work

In this section, I would like to introduce further concepts that will be used in the proposed method; these are the introduction and comparison of two main deep learning architectures that are implemented in this chapter. In addition, since the assumption of the proposed method is based on the concept of consistency regularization, this section will briefly introduce the concept and how we apply the concept to the proposed framework.

3.2.1 Comparison between CNN and ViT

The world of computer vision has witnessed a series of evolutionary strides in recent years. A significant part of this evolution involves the architectural design of neural networks tailored for visual tasks. Historically, CNNs have been the main candidate in this domain. However, a newer architecture, ViTs has arisen in recent years. Adapted from the success of the original concept of transformer in natural language processing, ViT has been challenging the competitor to CNNs.

The foundation of CNNs lies in their unique ability to process images using convolutional layers. These CNN layers apply a series of filters to input images, allowing the network to capture and recognize local patterns. Simple patterns, such as edges and textures, are identified in the earlier layers, while deeper layers discern more abstract and complex features, such as shapes and objects. This hierarchical, spatial processing ensures that CNNs understand images in a manner that is intuitive to human perception without requiring any human knowledge or feature engineering.

On the other hand, ViT utilizes an entirely different paradigm. They borrow the concept from the transformer architecture, originally crafted for handling sequences in natural language tasks. Instead of relying on spatial convolutions, ViT dissects an image into fixed-sized patches, transforms these patches into flat vectors, and then processes them as a sequence. The unique concept of the transformer, the attention mechanism, allows the model to weigh the importance of different parts of an image, even if these parts are distant from each other. This long-range dependency handling is a stark departure from the local processing seen in CNNs. The way that ViT handles the classification problem is by understanding the spatial structure of an image using positional embeddings. Unlike CNNs, which inherently recognize spatial hierarchies, transformers are agnostic to the positional arrangement of the sequence. Therefore, to account for spatial information, ViTs incorporate positional embeddings alongside the patch embeddings, ensuring that the model remains sensitive to the layout of the image.

Data has always been the deciding factor of deep learning methods; this also means that the data requirements for these two architectures vary considerably. CNNs, affected by inductive biases from their convolutional layers, often exhibit better performance even with moderate-sized datasets. Their design ensures that it studies visual hierarchies in a manner consistent with the whole image using the reception field concept. ViTs, however, have a different story. Training a ViT from scratch usually requires a tremendous number of data. However, a strategy of pretraining ViTs on colossal datasets, like ImageNet, and fine-tuning of ViTs on more specific, smaller datasets is proven to be one of the stable practices. Depending on the type of datasets, it can make one of the architectures perform better compared to one another.

As of the current time of writing, there are already many implementations of the mentioned architecture that further enhance the concept to new heights. On the CNN part, there is a ConvNext[32] that borrows the transformer architecture concept and applies it to CNN for better hierarchy learning. The model structure in the code format written in PyTorch can be seen in Fig.3.1. On the ViT side, the fixed-sized patches that are used for the attention mechanism have been addressed and improved upon in SwinTransformer[33] by introducing the Shifted Window to combine the patches in various aspect ratios. The difference between the common ViT and Swin Transformer can be seen in Fig. 3.3, on the right is the normal ViT, which has a fixed patch size of 16 by 16. However, Swin Transformer will divide patches into various sizes and have its own policy of combining the patches together. The model structure in the code format written in PyTorch can be seen in Fig.3.2.

3.2.2 Consistency regularization

In recent years, many kinds of research focused on addressing this vulnerable point to improve pseudo-labeling performance, such as introducing a new loss function that penalizes incorrect samples[16] or developing a better expert model[34]. One of the techniques that has started to become more popular in recent years is consistency

Layer (type:depth-idx)	Output Shape	Param #
ConvNextForImageClassification	[1, 1000]	--
└ConvNextModel: 1-1	[1, 768]	--
└┬ConvNextEmbeddings: 2-1	[1, 96, 56, 56]	--
└└┬Conv2d: 3-1	[1, 96, 56, 56]	4,704
└└└ConvNextLayerNorm: 3-2	[1, 96, 56, 56]	192
└└┬ConvNextEncoder: 2-2	[1, 768, 7, 7]	--
└└└ModuleList: 3-3	--	27,813,696
└└└LayerNorm: 2-3	[1, 768]	1,536
└Linear: 1-2	[1, 1000]	769,000
Total params: 28,589,128		
Trainable params: 28,589,128		
Non-trainable params: 0		
Total mult-adds (M): 395.23		
Input size (MB): 0.60		
Forward/backward pass size (MB): 131.27		
Params size (MB): 114.33		
Estimated Total Size (MB): 246.21		

FIGURE 3.1: The structure of ConvNext from PyTorch library

Layer (type:depth-idx)	Output Shape	Param #
SwinForImageClassification	[1, 1000]	--
└SwinModel: 1-1	[1, 768]	--
└┬SwinEmbeddings: 2-1	[1, 3136, 96]	--
└└┬SwinPatchEmbeddings: 3-1	[1, 3136, 96]	4,704
└└└LayerNorm: 3-2	[1, 3136, 96]	192
└└└Dropout: 3-3	[1, 3136, 96]	--
└└┬SwinEncoder: 2-2	[1, 49, 768]	--
└└└ModuleList: 3-4	--	27,512,922
└└└LayerNorm: 2-3	[1, 49, 768]	1,536
└└└AdaptiveAvgPool1d: 2-4	[1, 768, 1]	--
└Linear: 1-2	[1, 1000]	769,000
Total params: 28,288,354		
Trainable params: 28,288,354		
Non-trainable params: 0		
Total mult-adds (M): 62.80		
Input size (MB): 0.60		
Forward/backward pass size (MB): 137.29		
Params size (MB): 113.06		
Estimated Total Size (MB): 250.95		

FIGURE 3.2: The structure of Swin Transformer from PyTorch library

regularization[35, 7]. Consistency regularization emerged as one of the candidates and can be applied along with any semi-supervised learning[36, 37]. Consistency regularization is a technique used in machine learning, especially in tasks where we teach computers to recognize patterns or objects. The main idea is to ensure that the predictions or decisions given by computers remain stable or consistent, even when there are small changes to the input data.

In more technical terms, this means if we have two slightly different versions of the same input (like the original and slightly changed cat picture), the output (or prediction) of the algorithm should be close or the same for both. If the prediction results are wildly different for similar inputs, it might mean the computer is too sensitive and might not work well in situations where the data can be a bit messy.

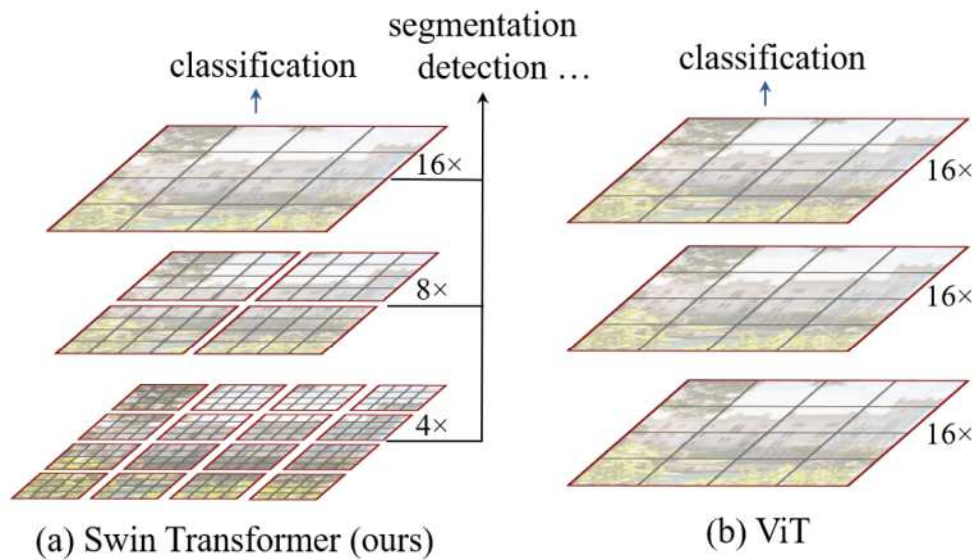


FIGURE 3.3: Difference between Swin Transformer and Visual Transformer

In short, the goal of consistency regularization is to help make sure that the decisions are stable and reliable. Consistency regularization can be realized by penalizing inconsistent decisions, ensuring that similar inputs lead to similar outputs. This technique can make the deep learning models more robust and better at handling data with various characteristics or traits.

For the pseudo-labeling cases, I would like to implement consistency regularization to the annotation task and show that the concept of giving annotation by the consensus of the two networks can make the pseudo-labeled samples be reliable and accepted.

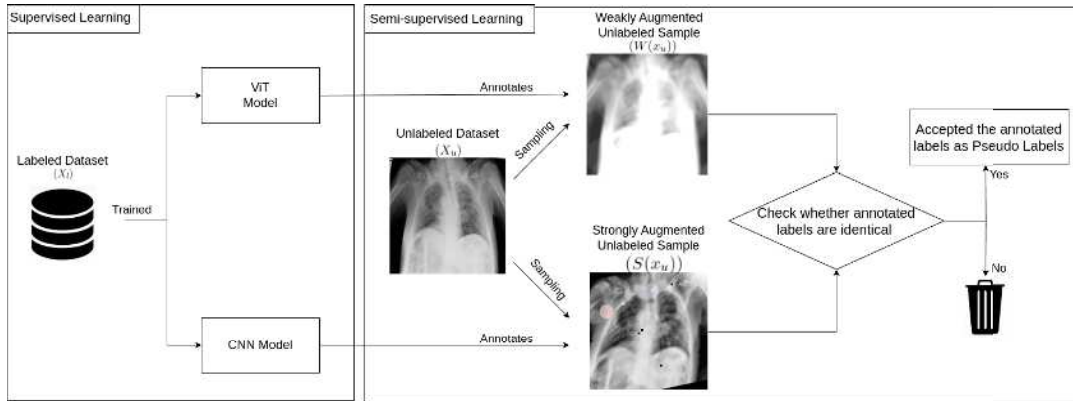


FIGURE 3.4: The overall flowchart of Pseudo-labeling with Contrastive Perturbation.

3.3 Proposed method

3.3.1 Pseudo-labeling with contrastive perturbation using CNN & ViT for chest X-ray classification

Inspired by many kinds of research aiming to improve deep learning performance by using consistency regularization, I design pseudo-labeling with contrastive perturbation, which aims to tackle the semi-supervised learning problem by incorporating pseudo-labeling with labeling ensembling. The aim of the proposed method is to increase the performance of pseudo-labeling by integrating two different deep-learning architectures with their unique strength, along with applying different perturbations onto the unlabeled samples, respectively, to each architecture. The flowchart of the proposed method is shown in Fig. 3.4, and the algorithm is in Alg.2.

First, the classification models, ViT and CNN, are trained in a supervised learning fashion using all the available labeled samples that can be accessed to. Next, weak augmentation $W(x_u)$ and strong augmentation $S(x_u)$ are applied to every unlabeled sample (x_u) in an unlabeled dataset (X_u) . The type of augmentation techniques will be further shown in the experiment section.

I let the trained CNN perform an annotation task on the unlabeled samples with strong augmentation $S(x_u)$, and ViT performs an annotation task on those with

Algorithm 2: Annotation Process in Pseudo-Labeling with Contrastive Perturbation

Result: Apply pseudo-labeling onto unlabeled samples

initialization;

for $X_l := \text{all available labeled data}$ **do**

 | trained the $model_{cnn}$ and $model_{vit}$ using x_l ;

end

for $X_u := \text{all available unlabeled data}$ **do**

 | $S(x_u) := \text{apply strong augmentation to } x_u$; // (1)

 | $F(S(x_u)) := \text{class when } model_{cnn} \text{ annotates } S(x_u)$;

 | $W(x_u) := \text{apply weak augmentation to } x_u$; // (2)

 | $G(W(x_u)) := \text{class when } model_{vit} \text{ annotates } W(x_u)$;

 | **if** $F(S(x_u)) == G(W(x_u))$ **then**

 | gives pseudo-labeled onto x_u ;

 | **else**

 | discards the sample;

 | **end**

end

weak augmentation $W(x_u)$. This process can alleviate the generalization bias during pseudo-labeling since the pseudo-labeled samples are produced from different neural networks with different generalization abilities.

After the pseudo-labels are given to each unlabeled sample by CNN and ViT, if both of the labeling results are the same class, the given class label is accepted; otherwise, the sample is discarded since it could cause deterioration of the classification performance. After finishing the pseudo-labeling process, a classification model (ResNet, ViT[30], Swin Transformer[33], and ConvNext[32] in this paper) are trained using the labeled and pseudo-labeled samples to evaluate the performance of the proposed method.

3.3.2 Dataset

RSNA COVID-19 Challenge was held for both classification tasks and detection tasks, but I focused on classification tasks because the pseudo-labeling with contrastive perturbation are designed to enhance the classification ability of deep learning. The task is to identify the abrupt changes in lung opacity that could contain

TABLE 3.1: Characteristics of each class

Radiographic Classification	CXR Finding	Suggested Reporting Language
Typical appearance	Multifocal bilateral, peripheral opacities Opacities with rounded morphology Lower lung-predominant distribution	Findings typical of COVID-19 pneumonia are present. However, these can overlap with other infections, drug reactions, and other causes of acute lung injury
Indeterminate appearance	Absence of typical findings AND Unilateral, central or upper lung predominant distribution	Findings indeterminate for COVID-19 pneumonia and which can occur with a variety of infections and noninfectious conditions
Atypical appearance	Pneumothorax or pleural effusion Pulmonary edema Lobar consolidation Solitary lung nodule or mass Diffuse tiny nodules	Findings atypical or uncommonly reported for COVID-19 pneumonia. Consider alternative diagnoses
Negative for pneumonia	No lung opacities	No findings of pneumonia. However, chest radiographic findings can be absent early in the course of COVID-19 pneumonia

diseases. The dataset contains four types of COVID-19 viewing classes: Atypical, Typical, Indeterminate, and Negative. Table 3.1 shows the detailed characteristics of each class. The overall dataset consists of 6,334 images, and the ratio of each class is shown in Fig. 3.5. From Fig. 3.5, we can see that the ratio of four classes is different; that is, this dataset contains class imbalance. Originally, the images were in DICOM format with an image size of 2,330 by 2,783 pixels. As for data preprocessing, each image was converted from DICOM to PNG and resized to 224 by 224 pixels; the imbalance in height and width was handled using a center crop. Three hundred thirty-four images with the smallest disease area were removed from the Typical class to reduce the class imbalance to some extent and make it easier for evaluation using cross-validation.

3.4 Experimental results

In this section, first, the experimental procedure conducted is explained. In the first experiment, I aim to find the best image classification architecture among ResNet, ViT, Swin Transformer, and ConvNext, which will be used for the next experiment. In the second experiment, the preliminary experiment is conducted to find the best

Class Distribution in the Dataset

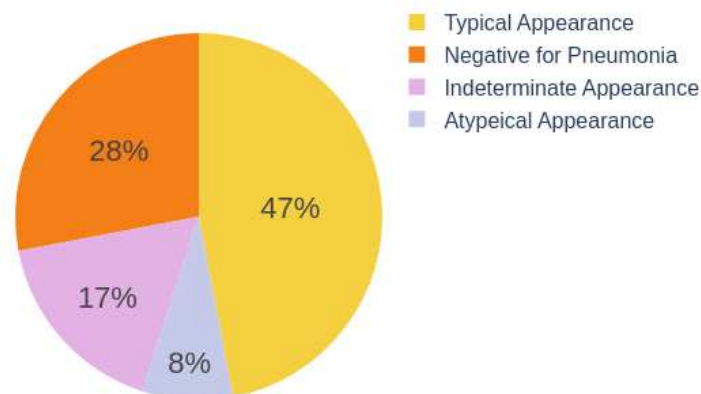


FIGURE 3.5: Class distribution in RSNA COVID-19 Challenge.

augmentation combination to be used by for the pseudo-labeling process of the proposed method. In the last experiment, the performance of the proposed method is evaluated and compared with the model without pseudo-labeling.

3.4.1 Comparison between difference deep learning architectures on COVID-19 classification task

To evaluate the performance of the proposed method, it is necessary to find the best network architecture as a classifier for chest CT image classification. The four models described before were trained with all the data with 90% training and 10% testing split. All the models were trained for 100 epochs with batch sizes of 32. The precision, recall, F1 score, and accuracy for the testing data are shown in Table 3.2. According to the result, ConvNext shows the best performance for all the evaluation metrics. In addition, the SwinTransformer also shows very competitive results with only difference in average of 2% performance difference compared to ConvNext, confirming that it is very suitable to perform conservative perturbation techniques using these two frameworks as deep learning network candidates. ConvNext was selected as the final classification model used in the next experiment when training the last pseudo-labeled model. The best performance shown by ConvNext indicates

TABLE 3.2: Comparison of precision, recall, F1 score, and accuracy obtained by ResNet, ViT, Swin Transformer, and ConvNext [%]

Model Architecture	Precision	Recall	F1 Score	Accuracy
ResNet	33.4	42.6	37.4	62.7
ViT	90.9	91.5	91.2	90.6
SwinTransformer	92.1	91.2	91.5	92.4
ConvNext	93.4	94.0	93.6	94.1

that the CNN structure enhanced by the combination with the ViT concept would contribute to other medical image diagnoses, e.g., CT, MRI, and so on.

3.4.2 Evaluation of various augmentation combinations and correctness of pseudo-labeling

This experiment is the preliminary experiment to find the best augmentation techniques to used for the pseudo-labeling process. I included all of the possible combinations of the augmentation methods: 1) Weak augmentation unlabeled samples for both CNN and ViT, 2) Strong augmentation unlabeled samples for both CNN and ViT, 3) Weak augmentation on CNN and Strong augmentation on ViT, 4) Strong augmentation on CNN and Weak augmentation on ViT. The experiment is performed using the 30% models to find the performance of the pseudo-labeling under different augmentation policies. Since we have access to all of the original true labels of the pseudo-labeled datasets, I can evaluate the accuracy of the pseudo-labeled samples under different augmentation combinations. The preliminary experiment results can be seen in Table 3.3, where the number of the unlabeled samples and that of the accepted pseudo-labeled samples are shown, where accepted means that the expert models of both CNN and ViT agree on the classification. The table also shows that the accuracy of the accepted pseudo-labeled samples given by various augmentation combinations is shown.

From the results, it can be seen that not all of the proposed augmentation methods have the same accuracy. The scenarios where CNN handled the strong augmentation,

TABLE 3.3: Comparison of pseudo-labeled sample accuracy from various augmentation combinations

Augmentation Method		Total # of	Accepted # of	# of correctly labeled	Accuracy
CNN	ViT	unlabeled samples	pseudo-labeled samples	pseudo-labeled samples	
Strong	Strong	3600	2165	1735	80.1
Strong	Weak	3600	2496	2242	89.8
Weak	Strong	3600	2463	2068	84.0
Weak	Weak	3600	2970	2581	87.0

and ViT handled the weak augmentation performed best, and the situation where both strong augmentations were applied was the worst. From these experimental results, the pseudo-labeling process is handled by using CNN with a strong augmentation method and ViT with a weak augmentation method.

3.4.3 Evaluation of pseudo-labeling with contrastive perturbation

To properly simulate the real-life scenarios, the dataset was split into labeled, unlabeled, and test sets. Here, two variations of the data split were considered: 30% labeled dataset with 60% unlabeled dataset (called 30% model) and 50% labeled dataset with 40% unlabeled dataset (called 50% model). The remaining 10% data is used as a testing dataset. With this setting, we can analyze and compare both scenarios where the number of labeled samples is larger and smaller than that of unlabeled ones. For the augmentation technique, the following augmentation techniques were used as weak augmentation: shear by 15 degrees, horizontal flip, rotation by 10 degrees, image scaling by 0.85 to 1.10, and Gaussian blur with a kernel size of three to seven. Here, these augmentations were applied with the possibility of 50%. The reason we chose these augmentations as weak augmentation is that they could possibly occur or be found in the X-ray image diagnosis under various circumstances, and also, they still retain the nature of the image. The weakly augmented data were used for pseudo-labeling by the ViT model. The strong augmentations were GridDropout, Contrast Limited Adaptive Histogram Equalization (CLAHE)[38], affine transformation, random sunflare, random crop, and random cutout; these augmentations change

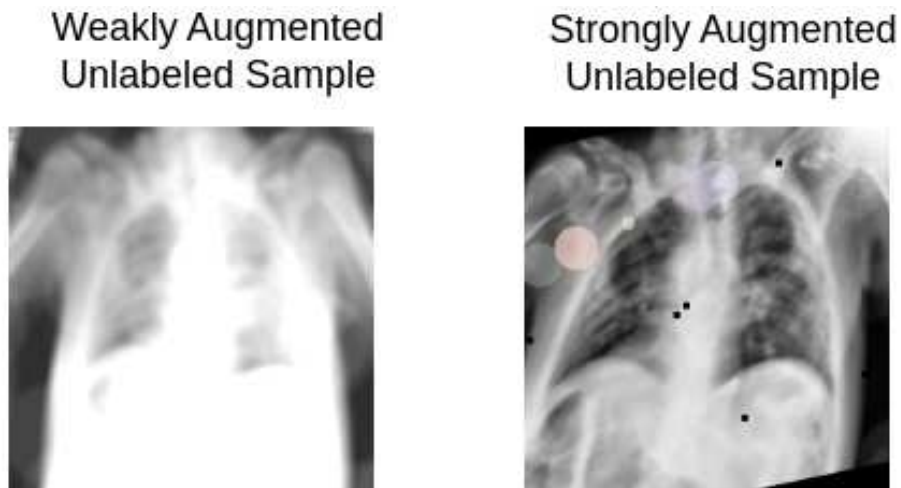


FIGURE 3.6: The example of augmented images used for training

the image heavily; thus, the possibility of applying these techniques is 10%. These augmentations drastically change the image to the point that it could not be possible in a real-life scenario, but if the model can correctly predict the pseudo-labeled of that sample, then that sample should produce less bias for the pseudo-labeled model. The heavily augmented data were used for pseudo-labeling by the CNN model. The example of the augmented samples can be seen in Fig. 3.6. In this research, Swin-Transformer was used as the ViT model, and ConvNext was used as the CNN model. Note that ConvNext, selected in the previous section, is used to build a classifier after the pseudo-labeling is completed by the combination of CNN and ViT models. The criteria are precision, recall, and accuracy. In addition, all the models were evaluated using ten-fold cross-validation. The experimental results can be seen in Table.3.4.

We can see from the table that performance improvement can be realized across all of the criteria when incorporating pseudo-labeled samples. For the 50% model, the precision, recall, and accuracy are increased by 4.5%, 3.2%, and 1.6%, respectively. As for the 30% model, the precision, recall, and accuracy are increased by 5.6%, 3.1%, and 3.5%, respectively. Another finding from the results is that most of the performance increases are from the classes with a smaller number of data, that is, Indeterminate and Atypical classes that occupy the whole data with only 17% and 8%, respectively. Especially, the precision of indeterminate and atypical classes can

TABLE 3.4: Evaluation of Pseudo-labeling with Contrastive Perturbation [%]

Labeled Sample	Unlabeled Sample	Class Name	Precision	Recall	Accuracy
50% (3,000 images)	None	Typical	71.7 ± 2.1	73.2 ± 2.3	
		Negative	64.3 ± 2.4	74.4 ± 2.7	
		Indeterminate	26.4 ± 4.1	22.8 ± 4.8	
		Atypical	30.4 ± 7.3	18.4 ± 4.7	
		Model's Average	48.2 ± 3.9	47.2 ± 8.8	63.6 ± 1.0
40% (2400 images)	None	Typical	73.1 ± 1.7	78.1 ± 2.8	
		Negative	65.9 ± 2.8	81.1 ± 2.0	
		Indeterminate	32.7 ± 5.8	20.6 ± 4.1	
		Atypical	38.9 ± 7.2	21.7 ± 3.8	
		Model's Average	52.7 ± 4.4	50.4 ± 3.2	65.2 ± 1.0
30% (1,800 images)	None	Typical	70.9 ± 1.6	75.6 ± 1.8	
		Negative	63.8 ± 2.9	74.2 ± 3.6	
		Indeterminate	24.1 ± 4.3	17.6 ± 4.0	
		Atypical	25.2 ± 7.2	15.3 ± 4.8	
		Model's Average	46.0 ± 4.0	45.7 ± 3.5	60.5 ± 1.6
60% (3,600 images)	None	Typical	71.9 ± 1.4	80.8 ± 1.9	
		Negative	64.9 ± 2.1	80.6 ± 2.7	
		Indeterminate	31.9 ± 5.0	16.2 ± 2.8	
		Atypical	37.5 ± 6.8	17.4 ± 4.4	
		Model's Average	51.6 ± 3.8	48.8 ± 2.9	64.0 ± 1.3

be improved more than typical and negative classes. From these results, it can be said that the proposed method improves the pseudo-labeling performance by decreasing the bias toward the majority class.

3.4.4 Limitation

In this chapter, pseudo-labeling with contrastive perturbation was proposed, but there are some remaining problems to be solved in the future. The aim of this method is to find the correct classes for the COVID-19 chest X-ray images. However, it is challenging due to many factors. First, the nature of chest X-ray images can vary due to many circumstances, such as the patient's posture, age, or noise during the examination. In addition, the datasets are also very imbalanced, as shown in Fig. 3.5, toward the Typical classes that contain almost 47% data compared to 8% data in the Atypical class, which plays a massive role in the pseudo-labeling environment. Therefore, the evaluation of different datasets should be implemented to show the

effectiveness of the proposed method.

The RSNA COVID-19 diagnosis task for Chest X-ray images is also a challenge because most of the disease areas are minuscule and come in many variances in size. The size of disease areas is investigated and summarized in Fig. 3.7. The horizontal axis of Fig. 3.7 shows the size of the disease area, and the vertical axis shows the number of images, where we can see that there are many minuscule disease areas. Since some disease areas are tiny when resizing of images is implemented, it could lead to a struggle to perform a classification task for some architecture; for instance, an original ViT architecture has a predetermined reception area determined by the patchify process, which is very sensitive to the image size changes. In the future, we could incorporate various model architectures or increase the number of neural networks used for the pseudo-labeling process to enhance the performance.

While I have shown that most of the pseudo-labeling frameworks improve the detection or classification performance, the reason behind the improvement is difficult to interpret since the structure of deep learning is well-known for being a black box[39]. However, from most of the experimental results, it can be seen that the performance of the proposed models was drastically increased, especially the improvement of the 30% models, which is larger compared to the 50% models in most of the results. This was also especially true for the case where ensemble learning is applied to the pseudo-labeling. The improvement of the 30% model realized by ensemble learning can be interpreted as enhancing the pseudo-labeling stability by complementing the reliability of the pseudo-labeled samples. In contrast, the performance of most of the 50% models only increased slightly when incorporating ensemble learning, which can be interpreted as the 50% model was already stable enough and was not impacted by ensemble learning.

The last point to be addressed is the pseudo-labeling process itself. Currently, the classification model is trained from scratch after obtaining pseudo-labeled samples. It would be more beneficial to introduce shared weight parameters obtained by the

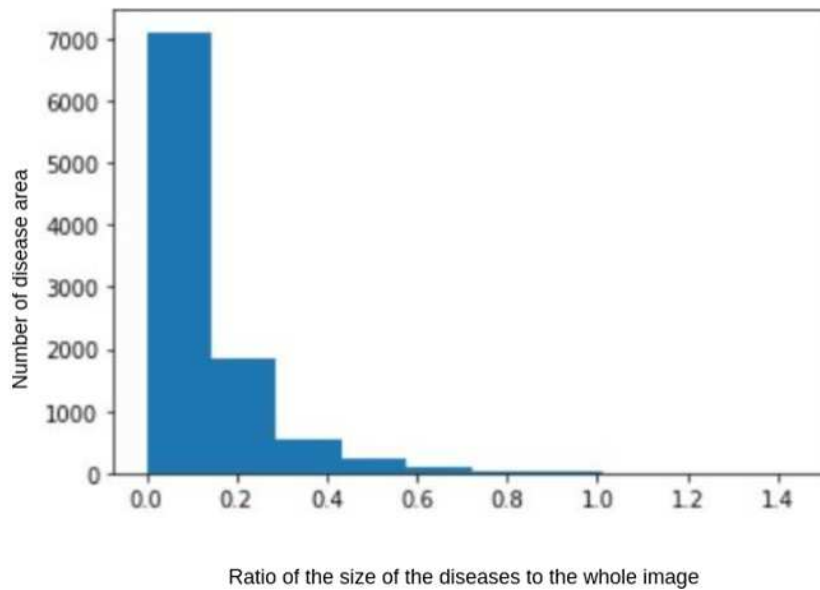


FIGURE 3.7: The ratio of the size of the disease areas to the whole image.

pseudo-labeling process or to design a better loss function that could prioritize the labeled samples with large importance or give different importance to pseudo-labeled samples.

3.5 Chapter summary

In this chapter, I illustrated and implemented pseudo-labeling with contrastive perturbation for chest X-ray COVID-19 classification. The proposed model improved the pseudo-labeling annotation process by applying consistency regularization, which required the annotation results for perturbed unlabeled samples given by two deep learning architectures to be identical to be accepted as pseudo-labeled samples. In addition, the model is further strengthened by using totally different deep learning architectures, which are ConvNext and SwinTransformer, accompanied by the contrastive augmentation regime. ConvNext, which is more tolerant to augmentation, is required to annotate samples under strong augmentation, and SwinTransformer, which is weaker to change in images, handles the sample with weak augmentation applied. The proposed constrastive perturbation framework successfully improved

the robustness of the pseudo-labeling framework and produced better experimental results compared to the traditional pseudo-labeling method.

In the next chapter, I would like to further improve the pseudo-labeling methods proposed in this chapter and the previous chapter to be more effective by introducing ensemble learning as the solution.

Chapter 4

Ensemble learning of pseudo-labeling framework for chest X-ray image diagnosis

4.1 Chapter introduction

In the previous chapters, I have proposed two frameworks based on pseudo-labeling for detection and classification problems with the improvement to address the bias. In Chapter 1, I addressed the bias that occurred by the models trained with the pseudo-labeled samples, while Chapter 2 mainly focused on the bias during the process of pseudo-labeling. That is, the bias problem addressed in Chapter 1 has not been solved yet. Therefore, in this chapter, I would like to propose a simple yet effective method to improve both previously introduced pseudo-labeling frameworks. The proposed method is based on the concept of Ensemble Learning, which is one of the most well-known methods in machine learning that can be easily applied to many situations[40]. When applying Ensemble learning to pseudo-labeling, I aim to improve performance stability [41] by reducing the bias caused by pseudo-labeled samples by using various detection or classification architectures.

In this chapter, I will illustrate how ensemble learning is implemented in the previously introduced frameworks. Since the tasks of each framework (object detection and classification) are different, I propose two methods for one of the frameworks.

This chapter is organized as follows: The first section introduces the common and traditional methods of ensemble learning, starting with the object detection ensemble and classification ensemble. Then, the implementation of the proposed ensemble learning in the pseudo-labeling is shown. The following section shows the experiment results, and the wrap-up section for this chapter is in the last section.

4.2 Ensemble learning frameworks

The ensemble has been one of the most practical machine learning algorithms[42, 43] for integrating multiple models or predictions to result in better performance in overall results. Since the ensemble is a very effective method, there are many researchers that implemented this concept and further improved upon it[44, 45, 40]. There are many factors that contributed to the success of the ensemble, which can be varied from the policy for training multiple machine learning models to the method behind how the models are combined together.

In this section, I will briefly introduce the common method of ensemble learning for machine learning tasks and give more context to the reason why the selected technique is adopted for the proposed method.

4.2.1 Techniques of classification ensemble

The main goal of ensemble techniques for classification is to improve the predictive performance of a model by combining the predictions from multiple classifiers. It simply involves using multiple learning algorithms or regimes with different parameters to obtain better predictive performance than that could be obtained from any of the individual learning algorithm alone. This approach is based on the wisdom of

collective decision-making, where collective decision-making often leads to better outcomes than isolated judgments. Common ensemble techniques include Bagging, Boosting, and Voting.

Bagging, originally an abbreviation of bootstrap aggregating, is the concept of training multiple models in parallel while using a different subset of a dataset that splits from the whole dataset. The goal is that since the models are trained using the subset of the data, the final decision made by majority voting for classification tasks should provide more reliability and result in better performance than using a single model. Random Forest[46] is an example of the bagging method, which is based on the voting of various decision trees for the final decision.

As for Boosting, it trains models sequentially, with each model focusing on the errors made by its predecessors. This is usually done by using a weight system that penalizes or prioritizes the wrong prediction samples. When the model is trained again in the next step, the weight system aims to correct the mistake that was made previously, and the final prediction is combined through the weight sum approach in the last prediction. Examples of popular algorithms of boosting are mostly related to a training regime, which is AdaBoost[47], Gradient Boosting[48], and XGBoost[49], where each subsequent model refines the overall classification performance.

The last method that I would like to introduce is voting. In the voting methods, multiple models are trained, and their predictions are combined to make a final decision. Similar to bagging, the fundamental idea is to improve predictive performance by considering the collective opinions of various models rather than relying on a single model. There are two main types of voting methods: hard voting and soft voting. In hard voting, each model in the ensemble votes for a class, and the class that gets the majority of the votes is chosen as the final prediction. This method is like a democratic election where each model has one vote, and the candidate (class) with the most votes wins. Soft voting, on the other hand, takes into account the probability or confidence scores assigned by each model to the potential classes. Instead

of a simple majority voting, the predictions are weighted based on these confidence scores.

4.2.2 Ensemble technique for object detection

Ensemble techniques, which have proven to enhance the performance of classification tasks, are increasingly being used for object detection as well. Instead of the normal classification framework, the object detection framework not only outputs the class of the image, but the model has to successfully locate the position of the bounding box where the target object exist. This led to many ways to ensemble the bounding boxes and combine the predictions from multiple models.

For the bounding box ensemble techniques, the most well-known technique is Non-Maximum Suppression[50] or NMS in short. NMS is a common method in object detection ensemble tasks to prune multiple bounding boxes predicting the same object down to the single most likely box. After the object detection model predicts bounding boxes, NMS first selects the box with the highest confidence score. NMS then compares the selected box with all the other boxes and if the Intersection over Union (IoU) of a box with the selected box exceeds a certain threshold (usually set between 0.3 and 0.7), that box is removed because that box is overlapped with the selected box. The process is repeated until all the boxes are processed. While the NMS method is very simple and effective in many scenarios, the problem arises when there are multiple boxes that overlap each other; that is, only the single box is used as the prediction results, and the overlapped boxes are discarded. To address this problem, a Soft Non-Maximum Suppression (Soft NMS)[51] ensemble was introduced to handle the cases where there are multiple overlapped bounding boxes. Instead of discarding the overlapping boxes like NMS Soft NMS reduces their confidence scores depending on their IoU with the selected box. The reduction in confidence is usually proportional to the IoU instead of outright removal. Boxes are then re-ranked based on these updated scores. The process continues until all boxes have been processed.

4.3 Proposed method

4.3.1 Ensemble learning for improving pseudo-labeling for object detection

Since pseudo-labeling is commonly known for having teacher bias and weak generalization ability [15],[34],[52], which occurs when the model learns from the incorrect pseudo-labeled samples or tends to trust only a few characteristics of the samples. To solve this problem, two models are combined to provide more reliable pseudo-labeled samples instead of using a single object detection model. After the pseudo-labeling process of the two models is finished, the pseudo-labeled samples from each model are assembled together before retraining the models.

For the ensemble method, I applied the object detection ensemble technique called Weight Box Fusion (WBF)[53]. Commonly, the box ensemble algorithm tends to choose the most suitable box among the obtained boxes. Instead, the weight box fusion ensembles the boxes using all the detected boxes and averages their location prioritized by their confidence score. The WBF algorithm can be summarized as follows.

Let b_i be the i^{th} bounding box and B be a set of bounding boxes, i.e., $b \in B$, and $b_i = (x_i, y_i, w_i, h_i, c_i)$ where x and y are the coordinates of a center of the bounding box, w is a width, h is a height of the bounding box from the center, and c is confidence score. Let b_j be a bounding box after applying WBF and B_{wbf} be a set of b_j , i.e., $b_j \in B_{wbf}$.

Since the detection performance can vary due to the characteristics of each architecture, the weight factor to prioritize the models is designed. The weights of model a and b are calculated by Eq. 4.1 and Eq. 4.2, respectively. W_a is weight for model a , W_b is that for model b , and mAP_a and mAP_b are mean average precision (mAP) of model a and b , respectively. The mAP has been explained in section 2.4.2.

$$W_a = \frac{mAP_a}{mAP_a + mAP_b} \quad (4.1)$$

$$W_b = \frac{mAP_b}{mAP_a + mAP_b} \quad (4.2)$$

Then b_j is obtained by Eqs.4.3 through 4.7.

$$x_j = W_a \frac{\sum_{i_a=1}^{T_a} (x_{i_a} c_{i_a})}{\sum_{i_a=1}^{T_a} (c_{i_a})} + W_b \frac{\sum_{i_b=1}^{T_b} (x_{i_b} c_{i_b})}{\sum_{i_b=1}^{T_b} (c_{i_b})} \quad (4.3)$$

$$y_j = W_a \frac{\sum_{i_a=1}^{T_a} (y_{i_a} c_{i_a})}{\sum_{i_a=1}^{T_a} (c_{i_a})} + W_b \frac{\sum_{i_b=1}^{T_b} (y_{i_b} c_{i_b})}{\sum_{i_b=1}^{T_b} (c_{i_b})} \quad (4.4)$$

$$w_j = W_a \frac{\sum_{i_a=1}^{T_a} (w_{i_a} c_{i_a})}{\sum_{i_a=1}^{T_a} (c_{i_a})} + W_b \frac{\sum_{i_b=1}^{T_b} (w_{i_b} c_{i_b})}{\sum_{i_b=1}^{T_b} (c_{i_b})} \quad (4.5)$$

$$h_j = W_a \frac{\sum_{i_a=1}^{T_a} (h_{i_a} c_{i_a})}{\sum_{i_a=1}^{T_a} (c_{i_a})} + W_b \frac{\sum_{i_b=1}^{T_b} (h_{i_b} c_{i_b})}{\sum_{i_b=1}^{T_b} (c_{i_b})} \quad (4.6)$$

$$c_j = \frac{W_a}{T_a} \sum_{i_a=1}^{T_a} c_{i_a} + \frac{W_b}{T_b} \sum_{i_b=1}^{T_b} c_{i_b} \quad (4.7)$$

where i_a and i_b are the bounding box number of model a and b , respectively, T_a and T_b are the total numbers of bounding boxes obtained by model a and b , respectively, $c_{i_a}^a$ is a confidence score of bounding box i_a obtained by model a , $c_{i_b}^b$ is that obtained by model b , and $x_{i_a}, y_{i_a}, w_{i_a}, h_{i_a}, c_{i_a}$ and $x_{i_b}, y_{i_b}, w_{i_b}, h_{i_b}, c_{i_b}$ determine the characteristics of bounding box number i_a and i_b , respectively.

The difference between the introduced NMS bounding box ensemble method and WBF bounding box ensemble methods can be seen in Fig. 4.1. In this figure, the red box represents class A, and the blue boxes represent class B. The red box is located

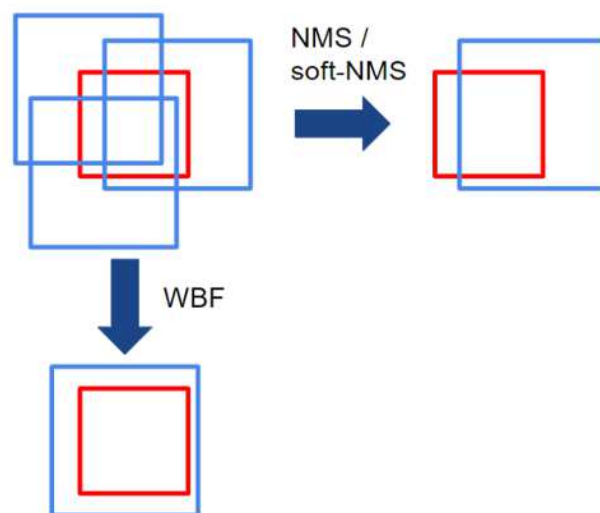


FIGURE 4.1: Comparison of the bounding box ensemble process between WBF and NMS

just to show the fixed position in the figure, that is, to show the changes in the position of blue boxes after the ensembling process. Therefore, the blue boxes are important to understand this figure. For NMS and soft NMS, since the blue boxes all have the same class, this ensemble method will return only the best box of the overlapped class. On the other hand, the WBF ensemble method will find and average all the boxes and result in a singular box that is constructed from all of the boxes.

By introducing the ensemble process, pseudo-labeled samples can be generated by integrating information from several aspects, leading to better performance.

4.3.2 Ensemble learning for improving pseudo-labeling for classification

To further strengthen the performance of the pseudo-labeling with a contrastive perturbation network proposed in chapter 3, ensemble learning is implemented to improve prediction stability and make the final classification models. The Ensemble Learning method's objective is to use various combinations of augmentation techniques for the pseudo-labeling instead of using only one pair of augmentation methods. We introduced four combinations of augmentation techniques to the framework

that perform the annotation task onto them: these are 1) Weak augmentation unlabeled samples for both CNN and ViT, 2) Strong augmentation unlabeled samples for both CNN and ViT, 3) Weak augmentation on CNN and Strong augmentation on ViT, 4) Strong augmentation on CNN and Weak augmentation on ViT.

Utilizing these four variations of augmentation will result in four pseudo-labeled datasets that have different variances and characteristics. In the next step, the models are trained using these four pseudo-labeled samples, resulting in four different final models trained on different pseudo-labeled datasets. All the final models perform the classification task to the test datasets and ensemble together using the traditional majority voting ensemble method, where the results will be accepted when three of the resulting classes are identical. The overall framework of the proposed method can be seen in Fig. 4.2. By combining different final predictions of various pseudo-labeling models, we aimed to remove the bias of any prominent characteristics or traits that may come from pseudo-labeled samples that would lead to deteriorating the final prediction performance.

After training all four pseudo-labeling models, the final classification result will be obtained by majority vote ensemble method using Eq. 4.8, where M is the total number of models, and the function $F(C_j(\mathbf{x}) = i)$ outputs 1 when the classification result for input x obtained by model j is class i , and it outputs 0 when the result is not class i .

$$\hat{y} = \arg \max_i \sum_{j=1}^M w_j F(C_j(x) = i) \quad (4.8)$$

The proposed method combines different types of classification models, that is, CNN and ViT, trained on different training environments, and also combines the outputs by weighted majority vote; thus, the overall model can generate final classification results flexibly depending on the characteristics of the given dataset. In addition, the combination of the outputs removes the bias of any prominent characteristics or traits that may come from pseudo-labeled samples that lead to deteriorating the final

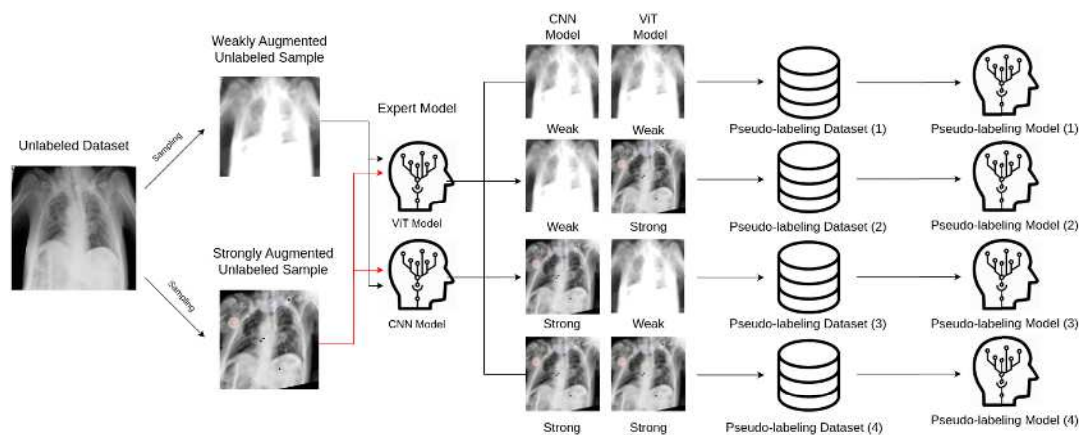


FIGURE 4.2: The Ensemble Learning for Pseudo-labeling with Contrastive Perturbation flowchart.

prediction performance.

4.4 Experimental results

4.4.1 Evaluation of ensemble learning for disease area detection

The experiment carried out in this section is a continuation of the experiments in chapter 2. Therefore, I aim to clarify whether or not the proposed ensemble method improves the performance obtained by the previous experiment.

Since the last experiment, the confidence threshold to accept the pseudo-labeled samples was set at 0.95 to prevent including incorrectly labeled samples. However, the problem with this setting is that the model tends to grow biased toward certain characteristics of the images and leads to poor generalization. Therefore, a weight box fusion ensemble is implemented to reduce the bias caused by pseudo-labeling, where the pseudo-labeled samples generated by two object detection models, RetinaNet and YoloV5, are combined.

The comparison between pseudo-labeling methods with and without ensemble labeling is shown in Table 4.1. The baseline results were obtained by RetinaNet, and the ensemble results were obtained by the combination of RetinaNet and YoloV5. The ratio of labeled and unlabeled samples used for the training were set at (30% and

TABLE 4.1: Comparison between pseudo-labeling methods with/without ensemble method. Parenthesis in the table shows the difference between baseline and ensemble method

Model Architecture	Labeled Samples	Unlabeled Samples	mAP for difference IoU thresholds					
			0.1	0.2	0.3	0.4	0.5	0.6
RetinaNet (Baseline)	30%	60%	71.28	71.10	69.17	65.09	56.96	46.42
Ensemble (RetinaNet + YoloV5)	30%	60%	73.47 (+2.19)	73.47 (+2.37)	73.20 (+4.03)	72.61 (+7.52)	70.17 (+13.21)	65.53 (+19.11)
RetinaNet (Baseline)	50%	40%	75.37	75.37	75.15	73.41	71.48	68.78
Ensemble (RetinaNet + YoloV5)	50%	40%	79.29 (+3.92)	79.29 (+3.92)	79.29 (+4.14)	77.11 (+3.70)	74.86 (+3.38)	71.62 (+2.84)

60%) or (50% and 40%). The remaining 10% samples were used for testing. IoU thresholds determine the strictness of the evaluation; thus, in the same way as chapter 2, they were set at from 0.1 to 0.6. From the result, we can see the performance improvement across all of the experimental settings when applying the ensemble to the pseudo-labeling. The most significant improvement is obtained by the 30% model when integrating 60% unlabeled samples. The mAP increases by 2.19% and 19.11% for the 30% models at 0.1 and 0.6 IoU, respectively. The 50% model also sees the performance improvement by 3.92% and 2.84% for 0.1 and 0.6 IoU, respectively. It can also be seen that the mAP increased only slightly at the low IoU threshold, but the performance became significantly better at higher IoU for the 30% model. However, the 50% model does not see an increase as significant as the 30% model. This result indicates that more performance improvement can be achieved when the number of labeled data is smaller. On the other hand, when the number of labeled data originally given to the model is larger, like the 50% model, the effect of ensemble learning becomes small. However, considering the whole results, we can conclude that introducing the ensemble can help alleviate the generalization problem from pseudo-labeling, leading to performance improvement and model stability. An example of the detection result obtained by the ensemble method is shown in Fig. 4.3. The top-left image is the detection result obtained by RetinaNet, the bottom-left is by YoloV5, the center image is the result of an ensemble, and the right image is the ground truth. The ensemble result looks very slightly different from the result of RetinaNet, but the mAP is improved, which is important for medical image diagnosis that requires even a little improvement.

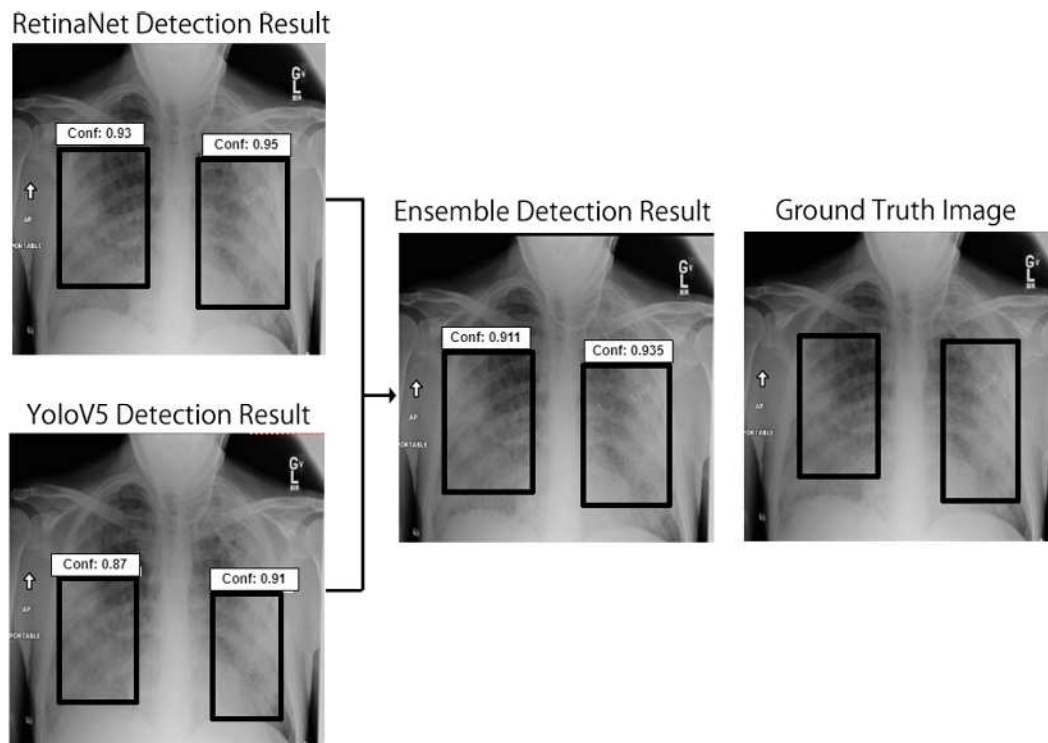


FIGURE 4.3: Example of the ensemble result using RetinaNet and YoloV5

4.4.2 Evaluation of ensemble learning for classification

Since the proposed method aims to improve the performance obtained in chapter 3, the results of chapter 3 are used as the baseline for comparison. The evaluation of the proposed pseudo-labeling model using the Majority Voting Ensemble method can be seen in Table. 4.2.

From the results, we can see overall performance improvement on average for all criteria. However, if we take a close look at the performance of each class, it can be seen that in the Typical class, the majority of the samples performed slightly worse compared to the model without using Ensemble Learning.

On the contrary, the minority of classes that suffer from the imbalance datasets problem saw a huge improvement. It contributed to the overall performance of the final prediction results, which surpassed the method without ensemble learning.

TABLE 4.2: Evaluation of Ensemble Learning for Pseudo-labeling with Contrastive Perturbation [%]

Model Name	Labeled Sample	Unlabeled Sample	Class Name	Precision	Recall
Baseline Model	50% (3,000 images)	None	Typical	71.7	73.2
			Negative	64.3	74.4
			Indeterminate	26.4	22.8
			Atypical	30.4	18.4
			Model's Average	48.2	47.2
Contrastive Perturbation Pseudo-labeling	50% (3,000 images)	40% (2,400 images)	Typical	73.1	78.1
			Negative	65.9	81.1
			Indeterminate	32.7	20.6
			Atypical	38.9	21.7
			Model's Average	52.7	50.4
Ensemble Contrastive Perturbation Pseudo-labeling	50% (3,000 images)	40% (2,400 images)	Typical	70.1	76.4
			Negative	65.3	82.3
			Indeterminate	37.4	25.5
			Atypical	42.8	20.4
			Model's Average	53.9	51.2
Baseline Model	30% (1,800 images)	None	Typical	70.9	75.6
			Negative	63.8	74.2
			Indeterminate	24.1	17.6
			Atypical	25.2	15.3
			Model's Average	46.0	45.7
Contrastive Perturbation Pseudo-labeling	30% (1,800 images)	60% (3,600 images)	Typical	71.9	80.8
			Negative	64.9	80.6
			Indeterminate	31.9	16.2
			Atypical	37.5	17.4
			Model's Average	51.6	48.8
Ensemble Contrastive Perturbation Pseudo-labeling	30% (1,800 images)	60% (3,600 images)	Typical	70.6	79.3
			Negative	69.6	83.2
			Indeterminate	34.3	20.7
			Atypical	41.6	18.4
			Model's Average	54.0	50.4

From these results, we concluded that introducing Ensemble Learning for the contrastive perturbation framework improved the model stability and decreased bias toward the majority classes, especially the problem of dealing with class imbalance.

4.5 Chapter summary

In this chapter, I explored the concept of ensemble learning and how it can be implemented to improve the pseudo-labeling framework in chest-X ray diagnosis tasks. The chapter also aims to increase the model performance and robustness of the previously introduced pseudo-labeling framework. The object detection with pseudo labeling was enhanced by weight box fusion that introduces weight parameters that prioritize and stress the output obtained by each model. As for the classification framework, the voting mechanism was adopted as an ensemble, which is simple but yielded effective results. The success of the classification contributed to the further

development of contrastive perturbation, which originally had a kind of ensemble learning concept. In the next chapter, I would like to conclude the research and discuss the potential for any possible future work.

Chapter 5

Conclusions

In this dissertation, I proposed the pseudo-labeling framework and how it can be integrated into deep learning and improve the performance of disease diagnosis. At the beginning of the dissertation, I briefly introduced the overall semi-supervised learning, the candidate methods, including pseudo-labeling, and the connection between the current machine learning trends and medical image analysis, and established the objective of this dissertation.

In Chapter 2, I introduced the pseudo-labeling framework for pneumonia area detection in chest X-ray images to alleviate the lack of labeled data. The proposed method aims to utilize unlabeled samples to improve performance through the implementation of an iterative pseudo-labeling process. The iterative process strengthens the pseudo-labeling by controlling the model stability and making the model more stable to the bias that comes from using the whole pseudo-labeled samples for training in one go. After various experiments, it was found that the suitable architecture, confidence threshold, the amount of labeled samples for the pseudo-labeling, and the performance improvement.

In Chapter 3, I introduced the pseudo-labeling framework for classification and applied it to the COVID-19 disease classification task. The proposed method makes use of the two deep-learning architectures to perform the pseudo-labeling task on the unlabeled samples with different degrees of perturbation. In the experiments, I

designed two scenarios to replicate real-world scenarios. The first scenario, where 30% of labeled samples were used with 60% of unlabeled samples, saw an increase in average accuracy by 1.6% . In the second scenario, where 50% labeled samples and 40% of unlabeled samples were used, the average accuracy increased by 3.5%.

In Chapter 4, first, I reinforced the detection framework proposed in Chapter 2 by utilizing an ensemble technique to reduce bias and increase the reliability of pseudo-labeling. As a result, the detection performance is further improved. The proposed method improves the mean average precision up to 5.32, compared with the method without the pseudo-labeling. Additionally, the mean average precision further increases by up to 19.11 when applying ensemble learning. Then, I also implemented further enhancements to the classification framework proposed in chapter 3 by introducing Voting ensemble methods. From the experimental results, it can be clarified that the proposed method improved the classification performance further.

In conclusion, I successfully implemented a pseudo-labeling framework to handle and improve the performance of the chest X-ray diagnosis task, including both disease area detection and classification tasks. Ensemble learning was also introduced to both architectures to further increase performance by reducing teacher bias from the pseudo-labeling process. I believe that in a situation where there are many unlabeled samples lying around, it is possible to increase the deep learning performance by applying the pseudo-labeling process rather than waiting for the expert labeling, and it is usually better to apply the ensemble to improve the performance.

Bibliography

- [1] Kaiming He et al. “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”. In: *2015 IEEE International Conference on Computer Vision (ICCV)* (2015), pp. 1026–1034.
- [2] Justin Ker et al. “Deep Learning Applications in Medical Image Analysis”. In: *IEEE Access* 6 (2018), pp. 9375–9389. doi: [10 . 1109 / ACCESS . 2017 . 2788044](https://doi.org/10.1109/ACCESS.2017.2788044).
- [3] Geert Litjens et al. “A survey on deep learning in medical image analysis”. In: *Medical Image Analysis* 42 (2017), pp. 60–88. issn: 1361-8415.
- [4] Jeremy Irvin. “CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison”. In: *AAAI* 33.01 (July 2019), pp. 590–597.
- [5] Tulin Ozturk et al. “Automated detection of COVID-19 cases using deep neural networks with X-ray images”. In: *Computers in Biology and Medicine* 121 (2020), p. 103792. issn: 0010-4825.
- [6] Heang-Ping Chan, Lubomir M. Hadjiiski, and Ravi K. Samala. “Computer-aided diagnosis in the era of deep learning”. In: *Medical Physics* 47.5 (2020), e218–e227.
- [7] Ting Chen et al. “A simple framework for contrastive learning of visual representations”. In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607.
- [8] Takeru Miyato et al. “Virtual adversarial training: a regularization method for supervised and semi-supervised learning”. In: *IEEE transactions on pattern analysis and machine intelligence* 41.8 (2018), pp. 1979–1993.

-
- [9] Antti Tarvainen and Harri Valpola. “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results”. In: *Advances in neural information processing systems* 30 (2017).
- [10] Jesper E. van Engelen and Holger H. Hoos. “A survey on semi-supervised learning”. In: *Machine Learning* 109.2 (Feb. 2020), pp. 373–440.
- [11] Avital Oliver et al. “Realistic evaluation of deep semi-supervised learning algorithms”. In: *Advances in neural information processing systems* 31 (2018).
- [12] B. Efron. “Bootstrap Methods: Another Look at the Jackknife”. In: *The Annals of Statistics* 7.1 (1979), pp. 1–26. doi: [10 . 1214 / aos / 1176344552](https://doi.org/10.1214/aos/1176344552). URL: <https://doi.org/10.1214/aos/1176344552>.
- [13] Qizhe Xie et al. “Self-Training With Noisy Student Improves ImageNet Classification”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 10684–10695.
- [14] Dong-hyun Lee. *Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks*. ICML, Workshop on challenges in representation learning. 2013.
- [15] Eric Arazo et al. “Pseudo-Labeling and Confirmation Bias in Deep Semi-Supervised Learning”. In: *2020 International Joint Conference on Neural Networks (IJCNN)* (2020), pp. 1–8.
- [16] Mamshad Nayeem Rizve et al. “In Defense of Pseudo-Labeling: An Uncertainty-Aware Pseudo-label Selection Framework for Semi-Supervised Learning”. In: *CoRR* abs/2101.06329 (2021).
- [17] Kihyuk Sohn et al. “Fixmatch: Simplifying semi-supervised learning with consistency and confidence”. In: *Advances in neural information processing systems* 33 (2020), pp. 596–608.
- [18] Qizhe Xie et al. “Unsupervised data augmentation for consistency training”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6256–6268.

-
- [19] Shaoqing Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. NIPS’15. Montreal, Canada: MIT Press, 2015, pp. 91–99.
- [20] Joseph Redmon and Ali Farhadi. “YOLOv3: An Incremental Improvement”. In: *CoRR* abs/1804.02767 (2018).
- [21] Glenn Jocher et al. “ultralytics/yolov5: Initial Release”. In: (June 2020). doi: [10.5281/zenodo.3908560](https://doi.org/10.5281/zenodo.3908560).
- [22] Chien-Yao Wang et al. “CSPNet: A New Backbone that can Enhance Learning Capability of CNN”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2020), pp. 1571–1580.
- [23] Joseph Redmon. *Darknet: Open Source Neural Networks in C*. <http://pjreddie.com/darknet/>. 2013.
- [24] Tsung-Yi Lin et al. “Feature Pyramid Networks for Object Detection”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 936–944.
- [25] Tsung-Yi Lin et al. “Focal Loss for Dense Object Detection”. In: *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 2999–3007.
- [26] M. Everingham et al. *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results*.
- [27] Fuzhen Zhuang et al. “A comprehensive survey on transfer learning”. In: *Proceedings of the IEEE* 109.1 (2020), pp. 43–76.
- [28] Divya Shanmugam et al. “Better aggregation in test-time augmentation”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 1214–1223.
- [29] David G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *Int. J. Comput. Vision* 60.2 (Nov. 2004), pp. 91–110. ISSN: 0920-5691.
- [30] Alexey Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020).

-
- [31] Paras Lakhani et al. *The 2021 SIIM-FISABIO-RSNA Machine Learning COVID-19 Challenge: Annotation and Standard Exam Classification of COVID-19 Chest Radiographs*. Oct. 2021. doi: [10.31219/osf.io/532ek](https://doi.org/10.31219/osf.io/532ek).
- [32] Zhuang Liu et al. “A convnet for the 2020s”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 11976–11986.
- [33] Ze Liu et al. “Swin transformer: Hierarchical vision transformer using shifted windows”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 10012–10022.
- [34] Xudong Wang et al. “Debiased Learning From Naturally Imbalanced Pseudo-Labels”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 14647–14657.
- [35] Yves Grandvalet and Yoshua Bengio. “Semi-supervised Learning by Entropy Minimization”. In: *Advances in Neural Information Processing Systems*. Ed. by L. Saul, Y. Weiss, and L. Bottou. Vol. 17. MIT Press, 2004, pp. 529–536.
- [36] David Berthelot et al. “ReMixMatch: Semi-Supervised Learning with Distribution Matching and Augmentation Anchoring”. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. 2020.
- [37] Hong-Yu Zhou et al. “SSMD: Semi-Supervised medical image detection with adaptive consistency and heterogeneous perturbation”. In: *Medical Image Analysis* 72 (2021), p. 102117. ISSN: 1361-8415.
- [38] Garima Yadav, Saurabh Maheshwari, and Anjali Agarwal. “Contrast limited adaptive histogram equalization based enhancement for real time video system”. In: *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. 2014, pp. 2392–2397. doi: [10.1109/ICACCI.2014.6968381](https://doi.org/10.1109/ICACCI.2014.6968381).

-
- [39] Vanessa Buhrmester, David Münch, and Michael Arens. “Analysis of explainers of black box deep neural networks for computer vision: A survey”. In: *Machine Learning and Knowledge Extraction* 3.4 (2021), pp. 966–989.
- [40] Zhi-Hua Zhou. “When Semi-supervised Learning Meets Ensemble Learning”. In: vol. 6. Jan. 2009, pp. 529–538. ISBN: 978-3-642-02325-5. DOI: [10.1007/s11460-011-0126-2](https://doi.org/10.1007/s11460-011-0126-2).
- [41] Samuli Laine and Timo Aila. “Temporal Ensembling for Semi-Supervised Learning”. In: *International Conference on Learning Representations*. 2017.
- [42] Yassine Ouali, Céline Hudelot, and Myriam Tami. “An overview of deep semi-supervised learning”. In: *arXiv preprint arXiv:2006.05278* (2020).
- [43] M.A. Ganaie et al. “Ensemble deep learning: A review”. In: *Engineering Applications of Artificial Intelligence* 115 (2022), p. 105151. ISSN: 0952-1976. DOI: <https://doi.org/10.1016/j.engappai.2022.105151>. URL: <https://www.sciencedirect.com/science/article/pii/S095219762200269X>.
- [44] Racheal S. Akinbo and Oladunni A. Daramola. “Ensemble Machine Learning Algorithms for Prediction and Classification of Medical Images”. In: *Machine Learning*. Ed. by Jaydip Sen. Rijeka: IntechOpen, 2021. Chap. 4. DOI: [10.5772/intechopen.100602](https://doi.org/10.5772/intechopen.100602). URL: <https://doi.org/10.5772/intechopen.100602>.
- [45] Dominik Müller, Iñaki Soto-Rey, and Frank Kramer. “An analysis on ensemble learning optimized medical image classification with deep convolutional neural networks”. In: *Ieee Access* 10 (2022), pp. 66467–66480.
- [46] Leo Breiman. “Random Forests”. In: 45.1 (Oct. 2001), pp. 5–32. ISSN: 0885-6125. DOI: [10.1023/A:1010933404324](https://doi.org/10.1023/A:1010933404324).
- [47] Robert E Schapire. “Explaining adaboost”. In: *Empirical inference*. Springer, 2013, pp. 37–52.
- [48] Jerome H Friedman. “Greedy function approximation: a gradient boosting machine”. In: *Annals of statistics* (2001), pp. 1189–1232.

-
- [49] Tianqi Chen and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’16. San Francisco, California, USA: ACM, 2016, pp. 785–794. ISBN: 978-1-4503-4232-2. DOI: [10 . 1145 / 2939672 . 2939785](https://doi.org/10.1145/2939672.2939785). URL: <http://doi.acm.org/10.1145/2939672.2939785>.
- [50] Alexander Neubeck and Luc Van Gool. “Efficient non-maximum suppression”. In: *18th international conference on pattern recognition (ICPR’06)*. Vol. 3. IEEE, 2006, pp. 850–855.
- [51] Navaneeth Bodla et al. “Soft-NMS—improving object detection with one line of code”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 5561–5569.
- [52] Ruifei He, Jihan Yang, and Xiaojuan Qi. “Re-Distributing Biased Pseudo Labels for Semi-Supervised Semantic Segmentation: A Baseline Investigation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 6930–6940.
- [53] Roman Solovyev, Weimin Wang, and Tatiana Gabruseva. “Weighted boxes fusion: Ensembling boxes from different object detection models”. In: *Image and Vision Computing* 107 (2021), p. 104117.