

Doctor's Thesis  
Academic Year 2024

Efficient Learning of Chest X-ray Imaging by  
Deep Learning for Early Clinical Introduction

Faculty of Information Technology  
Graduate School of Integrative Science and Engineering  
Tokyo City University

Kuniki Imagawa

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Medical Background</b>	<b>4</b>
2.1.	Medical Device Regulation . . . . .	4
<b>3</b>	<b>Computational Background</b>	<b>8</b>
3.1.	Architecture . . . . .	8
3.1.1	Neural Network . . . . .	8
3.1.2	Machine learning . . . . .	10
3.1.3	Convolutional Neural Network . . . . .	15
3.1.4	Vision Transformer . . . . .	21
3.2.	Learning method . . . . .	23
3.2.1	Transfer Learning . . . . .	23
3.2.2	Self-Supervised Learning . . . . .	24
<b>4</b>	<b>Dataset and Pre-processing</b>	<b>26</b>
4.1.	Image type . . . . .	26
4.1.1	Natural image . . . . .	26
4.1.2	Medical image . . . . .	27
4.1.3	DICOM . . . . .	28
4.2.	Pre-processing . . . . .	28
<b>5</b>	<b>Performance Index</b>	<b>31</b>
5.1.	Area under the curve . . . . .	31
5.2.	Confusion Matrix . . . . .	31
5.3.	Confidence Index . . . . .	32

<b>6</b>	<b>Performance change with the number of training data</b>	<b>34</b>
6.1.	Abstract . . . . .	34
6.2.	Introduction . . . . .	35
6.3.	Material . . . . .	38
6.3.1	Datasets . . . . .	38
6.4.	Methodology . . . . .	40
6.4.1	Preprocessing . . . . .	42
6.4.2	Classification . . . . .	42
6.4.3	Evaluation . . . . .	43
6.5.	Experiment and results . . . . .	43
6.6.	Discussion . . . . .	50
6.7.	Conclusion . . . . .	54
<b>7</b>	<b>Performance change with the ratio of training data</b>	<b>56</b>
7.1.	Abstract . . . . .	56
7.2.	Introduction . . . . .	57
7.3.	Theoretical background . . . . .	59
7.3.1	CNNs and fine-tuning . . . . .	59
7.4.	Methodology . . . . .	60
7.5.	Results . . . . .	62
7.6.	Discussion and conclusion . . . . .	63
<b>8</b>	<b>Evaluation of effectiveness of self-supervised learning to reduce annotated images</b>	<b>66</b>
8.1.	Abstract . . . . .	66
8.2.	Introduction . . . . .	67
8.3.	Material and Methods . . . . .	69
8.3.1	Datasets . . . . .	69
8.3.2	Methodology . . . . .	70
8.4.	Results . . . . .	72
8.5.	Discussion . . . . .	74
8.6.	Conclusion . . . . .	78

<b>9</b>	<b>Evaluation of effectiveness of pre-training method</b>	<b>80</b>
9.1.	Introduction . . . . .	80
9.2.	Related work . . . . .	82
9.3.	Methodology . . . . .	83
9.3.1	Vision Transformer . . . . .	83
9.4.	Material and method . . . . .	85
9.4.1	Method and DataSets . . . . .	85
9.4.2	Preprocessing . . . . .	86
9.5.	Experiment and results . . . . .	87
9.6.	Discussion . . . . .	92
9.7.	Conclusion . . . . .	96
<b>10</b>	<b>Conclusion</b>	<b>97</b>
<b>11</b>	<b>Publication</b>	<b>100</b>
	<b>Acknowledgements</b>	<b>102</b>
	<b>References</b>	<b>103</b>



# Chapter 1

## Introduction

Research and development of medical imaging using deep learning (DL), a subset of machine learning (ML), has been actively conducted, and products approved as medical devices have begun to be used in clinical environments. Software, such as medical image analysis intended for diagnosis, prevention and treatment, is applicable to medical devices and is subject to regulation. Regulatory authorities have begun to establish a new regulatory system that aligns with ML characteristics. One of these systems is to establish pre-change control plans before introducing clinical environments. On the other hand, in the medical field, the number of available medical images is limited because of the protection of patient information and expertise involved in annotating processes. Against this background, the purpose of this thesis is to investigate: (1) The factor of performance change during continuous learning in clinical environments, and (2) The efficiency of learning during model development. In this study, a binary classification of chest X-ray (CXR) images for Coronavirus Disease 2019 (COVID-19) and normal cases was conducted. This is because the data formats and imaging methods for CXR images are standardized.

Regarding the factors affecting the performance change: 1) The performance change with respect to the number of training images was verified. Three comparably shallow CNN models and those models fine-tuned on natural images were utilized. Our results show that the performance of all models improved rapidly as the number of training images increased and reached an equilibrium state, whereas models with deeper layers required more training images to reach an equilibrium

state. On the other hand, the models fine-tuned on natural images were more effective, especially when the number of training images was small, and 2) The performance change with respect to the ratio of the labeled training images was verified. The best performance was achieved when the ratio of COVID-19 and normal cases in the training images was the same, and when the ratio of the disease in the training images was biased, the performance decreased, whereas when highlighting the important regions for prediction, the lung field region was captured when the ratio was the same, which is consistent with the COVID-19 characteristics. This supports the classification results.

Regarding the efficiency of learning during model development: 1) The possibility of reducing the number of labeled images using a fine-tuning method with self-supervised learning was verified. The number of labeled images can be significantly reduced by using the same type of medical images as those used in the classification task for self-supervised learning. Furthermore, the computational cost can be reduced significantly by reducing the batch size. However, with a small number of labeled images (tens to hundreds), a large number of pre-training is necessary, and 2) The effectiveness of transfer learning and fine-tuning with natural images or CXR images for CNN and vision transformers (ViT) was verified. It was found that the models fine-tuned on natural images were more effective than those on medical images, and the trend became clearer as the number of pre-training images increased. When highlighting the important regions for prediction, the lung field region was captured.

Our results show that (1) The performance change with respect to continuous learning depends on the type of model, number of training images, pre-training with and without natural images, and ratio of diseases in the training images and (2) The number of labeled images can be significantly reduced by using the same type of medical images as those used in the classification task for self-supervised learning, and the models fine-tuned on natural images are more effective than those of medical images. Our study provides a basic aspect for establishing pre-change control plans and efficiency of learning during model development to establish reasonable and better regulations. However, it is necessary to consider the generalizability of our results since this study is limited to CXR images. The concept and structure of this thesis is summarized in Figure 1.1.

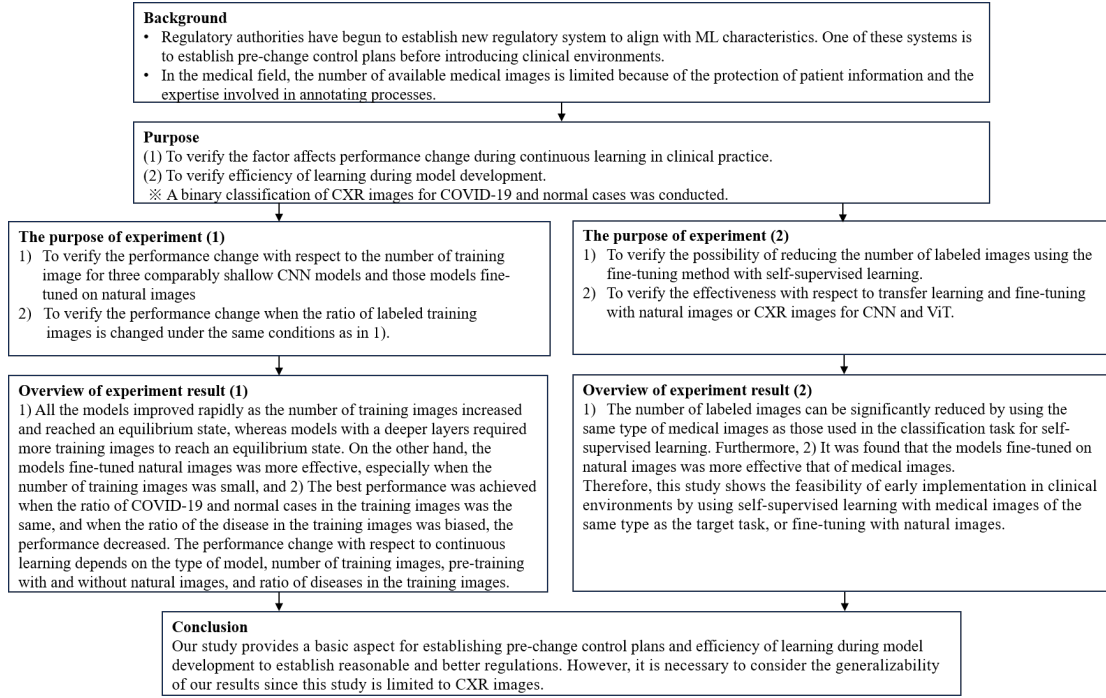


Figure 1.1: The concept and structure of this thesis

# Chapter 2

## Medical Background

### 2.1. Medical Device Regulation

Standalone software intended for diagnosis, treatment, prevention, etc., is treated as a medical device in many nations and regions. Medical device software has a certain patient risk because the output information influences the physician's decision. Recently, extensive interest in AI and ML applications has grown in the medical field. This current boom, called the third AI boom, differs from previous booms because many technologies are being implemented in society, and many ML-based medical devices have been developed and introduced into the clinical environment through regulatory approval. ML-based medical devices are roughly divided into two types: 1) the locked type, which fixes performance prior to marketing and is unable to change performance with use, and 2) the continuous type, which can change performance by continuously training data after market introduction. For the locked type, a fundamental review can be conducted based on the current regulatory framework. However, traditional medical device regulations are not designed for the continuous type, which has the potential to adapt and optimize device performance continuously in real time. In particular, when there is a performance change, it may be necessary to submit a partial change approval or notification to the regulatory authority frequently. With this background, regulatory authorities are trying to develop a total product lifecycle approach that facilitates the nature of ML-based medical devices and allows them to continuously improve while maintaining safety and effectiveness for patients.

In Japan, the Pharmaceutical Affairs Law was revised in 2019 to align with the nature of diversity and constant improvement of medical devices. One of the new approval systems is referred to as improvement design with approval for timely evaluation and notice (IDATEN) [1]. This system was established in response to the characteristics of medical devices, which are frequently modified and improved in clinical environments. Under IDATEN, as shown in Figure 2.1, the Pharmaceutical and Medical Device Agency (PMDA), which is the Japanese regulatory agency, reviews the contents of the pre-determined change plan in advance to verify its safety and effectiveness. After introducing clinical environments, if the changes are confirmed within the scope of the pre-determined change plan, the manufacturer can implement the actual changes 30 days after the submission of only the notification. To establish this system efficiently, it is also important for manufacturers to establish an organizational process to monitor performance changes in clinical environments.

U.S. Food and Drug Administration (FDA) regulates the sale of medical devices in the U.S. and monitors the safety of all regulated medical products. In April 2019, the FDA released a discussion paper on the proposed regulatory framework [2] to address the iterative nature of AI/ML-based medical devices, as shown in Figure 2.2. This paper describes a total product lifecycle approach. As part of this framework, the general principles necessary for this framework are described, including the introduction of good machine learning practice (GMLP). In this paper, it is recommended that a change management plan that includes both the SaMD pre-specifications (SPS) and the algorithm change protocol (ACP) should be developed and used. The SPS specifies planned changes in its performance, inputs, and intended use, while the ACP specifies how to implement, verify, and validate the changes specified in the SPS. Subsequently, in April 2023, the FDA released draft guidance that expanded on these discussion papers [3].

In the meantime, a number of countries have worked together to publish guidance documents. In 2021, the U.S. FDA, Health Canada, and the U.K.’s Medicines and Healthcare products Regulatory Agency (MHRA) collaboratively released on 10 guiding principles [4]: 1) Multi-Disciplinary Expertise Is Leveraged Throughout the Total Product Life Cycle, 2) Good Software Engineering and Security Practices Are Implemented, 3) Clinical Study Participants and Data Sets Are

Representative of the Intended Patient Population, 4) Training Data Sets Are Independent of Test Sets, 5) Selected Reference Datasets Are Based Upon Best Available Methods, 6) Model Design Is Tailored to the Available Data and Reflects the Intended Use of the Device, 7) Focus Is Placed on the Performance of the Human-AI Team, 8) Testing Demonstrates Device Performance during Clinically Relevant Conditions, 9) Users Are Provided Clear, Essential Information and 10) Deployed Models Are Monitored for Performance and Re-training Risks are Managed. These guiding principles will help to promote safe, effective, and high-quality medical devices that use AI and ML. In 2023, they also released five guiding principles for pre-determined change control plans [5]: 1) Focused and Bounded, 2) Risk-based, 3) Evidence-Based, 4) Transparent and 5) Total Product Lifecycle (TPLC) Perspective. These principles are based on the overarching GMLP, particularly principle 10). It states that the performance of the models should be monitored, and the risks of re-training should be managed. On the other hand, the International Medical Device Regulators Forum (IMDRF), who is a voluntary group of medical device regulators from around the world to accelerate international medical device regulatory harmonization and convergence, is developing new guidances for ML-based medical devices for the GMLP and PCCP. These concepts are expected to form the basis for national and regional regulations.

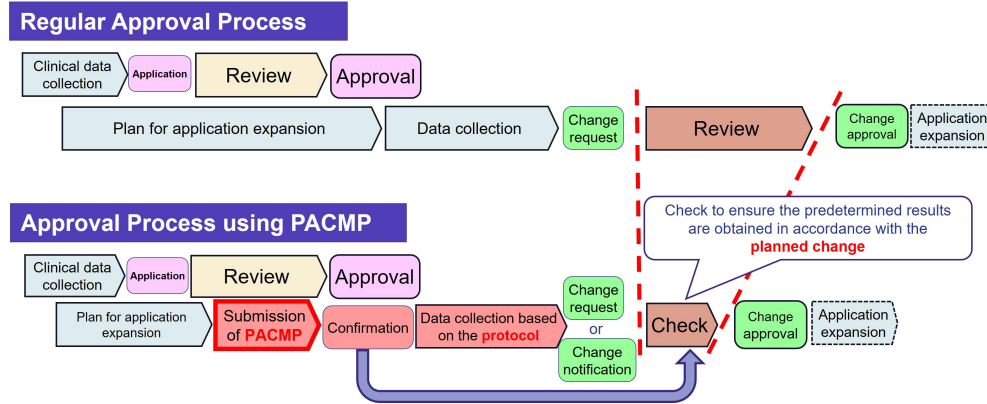


Figure 2.1: Concept of IDATEN in japan

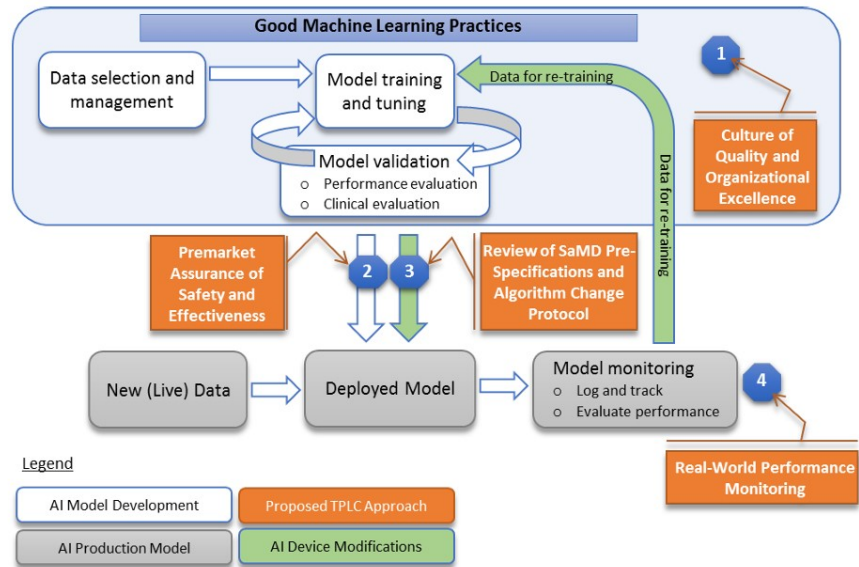


Figure 2.2: The concept of TPLC approach in U.S.

# Chapter 3

## Computational Background

### 3.1. Architecture

#### 3.1.1 Neural Network

Artificial neurons model the structure and function of neurons and nerve cells in the brain. Human cerebrum contains tens of billions of nerves. Neurons have dendrites and axons that extend from their bodies. The connections between neurons are called synapses, in which information is transmitted. The dendrites around the cell body of a neuron receive electrical signals from other neurons via the synapses. If the sum of the input signals is greater than a certain threshold, a pulsed action potential is transmitted from one axon to the other. Signals are transmitted to neurons during many of these actions. The input to a neuron is  $x_i$ , and the weight is  $w_i$ , which corresponds to the synaptic connections that have different strengths and directions; the total input is defined follows:

$$u = \sum_{i=1}^n w_i x_i \quad (3.1)$$

If the activation function is  $f$  and the bias to control the activation is  $b$ , then the output  $y$  of the neuron with respect to the input  $x_i$  is as follows:

$$u = \sum_{i=1}^n w_i x_i + b \quad (3.2)$$



$$y = f\left(\sum_{i=1}^n w_i x_i + b\right) \quad (3.3)$$

An artificial neural network is a network of many artificial neurons that are connected to each other and are represented as a network. A feedforward neural network (FNN) is an artificial neural network characterized by the direction of the information flow from the input, intermediate, and output layers. Neural networks with many intermediate layers are known as deep neural networks. Figure 3.1 shows (a) artificial neural and (b) FNN.

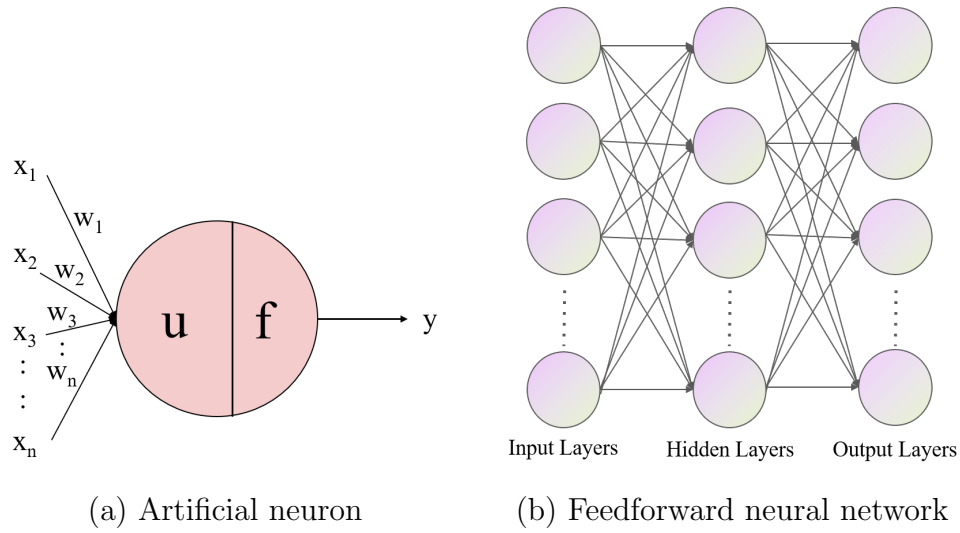


Figure 3.1: Feed forward neural network

The activation function defines a formula associated with a neuron in a neural network that determines the output of the neuron activation value from the inputs to the neuron. The activation function is nonlinear because it does not provide the benefits of multiple layers. Therefore, to take advantage of the multiple layers, it is necessary to use a nonlinear activation function. The sigmoid function in Figure 3.2 (a) is an activation function that has been in use for a long time.

$$f(x) = \frac{1}{1 + \exp(-x)} \quad (3.4)$$

Recently, rectified linear units (ReLU) in Figure 3.2 (b), have been primarily used because of their negligible impact on backpropagation and learning. This is

because the derivative of  $f(x)$  with respect to  $x$  is 1 when  $x$  is greater than 0:

$$f(x) = \begin{cases} x & (x > 0) \\ 0 & (x \leq 0) \end{cases} \quad (3.5)$$

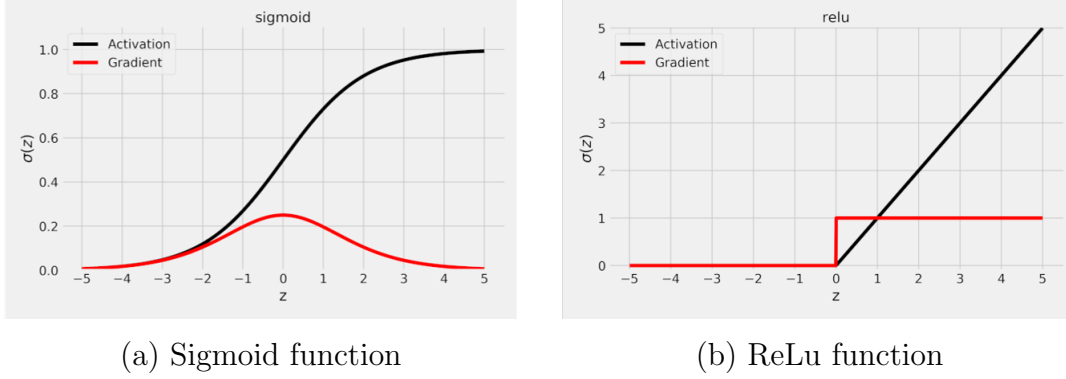


Figure 3.2: Activation function

Neural networks are mainly divided into regression and classification tasks. The regression task is to predict continuous numerical values from input data. The output layer then uses the identity function, which simply outputs the input signal. On the other hands, the classification task is to which class the input data belongs. Softmax functions are usually used for classification tasks and are represented by the following equation:

$$y_k = \frac{\exp(\alpha_k)}{\sum_{i=1}^n \exp(\alpha_i)} \quad (3.6)$$

$\alpha$  is the  $k$ -th output value,  $y$  is the  $k$ -th softmax function value and  $n$  is the number of classification. The output of the softmax function is a real number between 0 and 1, and the sum of its outputs is 1. Thus, the softmax function provides a probabilistic response to the classification problem, in which the intended number of output neurons is determined.

### 3.1.2 Machine learning

ML is a subset of AI that gives computers the ability to learn without explicit programming. DL is a subset of end-to-end machine learning that uses data

to extract knowledge by training neural networks with numerous hidden layers. There are several different types of ML learning methods, including supervised learning, unsupervised or semi-supervised learning. The machine learning data are divided into three datasets: training, validation, and test data. The training data is used to train the machine learning model to optimize the parameters. Validation data can be used to tune the hyperparameters to validate overfitting or data drift. Test data is used to assess the performance of the final trained model. The training and validation datasets are used in the training phase, and hyperparameters are selected to optimize the model parameters.

The loss function quantifies the difference between the predicted output of the machine learning algorithm and the actual target value in supervised learning. Optimization is the process of finding a set of parameters that minimize the loss function when training neural networks. The mean squared error (MSE) or cross-entropy loss is often used as a function. MSE and cross-entropy loss are given by the following equations:

$$E = \frac{1}{2} \sum_k (t_k - y_k)^2 \quad (3.7)$$

$$E = - \sum_k t_k \log y_k \quad (3.8)$$

$y_k$  is the  $k$ -th output of the neural network,  $t_k$  is the  $k$ -th actual annotated value. MSE determines the average of the squared differences between the target and predicted outputs. On the other hand, the cross-entropy error is the output value of the neural network because  $y_k$  is treated as a one-hot vector, thus  $\log y_k = 0$ . The training is performed using randomly chosen training data. If the loss function is sufficiently large, the average value is equal to the approximate value of the loss functions. This is called mini-batch learning. The purpose of training is to minimize the loss function, and the neural network optimizes the weight and bias. The gradient descent method is often used to train deep neural networks because the loss function is very complicated. The gradient descent algorithm repeatedly computes the gradient and then moves it in the direction of the gradient.

$$w^{(t+1)} = w^{(t)} + \Delta w^{(t)} \quad (3.9)$$

$$\Delta w^{(t)} = -\eta \nabla E(w^{(t)}) \quad (3.10)$$

$\eta$  denotes the learning rate. The learning rate is a hyperparameter that controls the extent to which the model learns and updates the neural network parameters with respect to the loss gradient, which continues until all the training data are exhausted. This number is known as an epoch. The loss function value and accuracy between the training data and validation data are confirmed at each epoch to confirm that the trained model is over-fitted to the training data and can not be generalized to general data. The epoch number is also a hyperparameter: when a training epoch is completed, a new training epoch starts at that point. To optimize the parameters, the back propagation algorithm is mainly used.

Meanwhile, there is a local optimization problems in the gradient descent algorithm. The first is the oscillation problem, specifically the loss function has a valley, a plateau, and a precipitation gradient region. In the valley example, the parameter only updates in the direction of gradient and oscillates significantly along the valley, never converging. If  $\eta$  is small, the oscillations will be small. However, these parameters cannot be updated on a normal gradient surface. The second is the existing saddle point. The parameters are rarely updated near the saddle point because the gradient is close to zero. Several methods have been proposed to address this issue. **Momentum** is a method that suppresses oscillations in the gradient method and improves convergence to the minimum value. This is also translated as "inertia" and it prevents fluctuations by delaying the effect of the gradient at the previous time by the following equations:

$$w^{(t+1)} = w^{(t)} + \Delta w^{(t)} \quad (3.11)$$

$$\Delta w^{(t)} = \mu \Delta w^{(t-1)} - (1 - \mu) \eta \nabla E(w^{(t)}) \quad (3.12)$$

$\mu$  is about 0.5 to 0.99. **AdaGrad** focuses on the gradient method. In gradient descent, there was only one learning rate. For example, a loss function with a large gradient in the x-direction and a small gradient in the y-direction updates the parameters rapidly in the x-direction but not at all in the y-direction. AdaGrad solves this problem by dividing  $\eta$  by the square root of the sum of squares of the gradients in each parameter direction. However, AdaGrad has a problem in that its learning rate monotonically decreases. In other words, if a rapid gradient is followed by a gradual gradient, the learning rate is so small that the number of

updates is almost zero for the gradual gradient.

$$\Delta w_i^{(t)} = -\frac{\eta}{\sqrt{\sum_{s=1}^t (\nabla E(w^{(s)})_i)^2}} \nabla E(w_i^{(t)}) \quad (3.13)$$

**RMSprop** is an improved method of AdaGrad. The problem with AdaGrad is that once updates become small, they do not return to a large value. Therefore, the application of an exponential decay factor to the past gradient is a solution to this hyperparameter problem.

$$v_{i,t} = \rho v_{i,t-1} + (1 - \rho) (\nabla E(w^{(t)})_i)^2 \quad (3.14)$$

$$\Delta w_i^{(t)} = -\frac{\eta}{\sqrt{v_{i,t} + \epsilon}} \nabla E(w^{(t)})_i \quad (3.15)$$

The initial value is  $v_{i,0} = 0$ , and  $\epsilon = 10^{-6}$  is used so that the denominator is not zero. **Adam** is a fusion of RMSprop and Momentum methods.

$$m_{i,t} = \rho_1 m_{i,t-1} + (1 - \rho_1) \nabla E(w^{(t)})_i \quad (3.16)$$

$$v_{i,t} = \rho_2 v_{i,t-1} + (1 - \rho_2) (\nabla E(w^{(t)})_i)^2 \quad (3.17)$$

$$\hat{m}_{i,t} = \frac{m_{i,t}}{(1 - (\rho_1)^t)} \quad (3.18)$$

$$\hat{v}_{i,t} = \frac{v_{i,t}}{(1 - (\rho_2)^t)} \quad (3.19)$$

$$\Delta w_i^{(t)} = -\eta \frac{\hat{m}_{i,t}}{\sqrt{\hat{v}_{i,t} + \epsilon}} \quad (3.20)$$

The initial value is  $m_{i,0} = v_{i,0} = 0$ . There are some hyperparameters, and original paper defined as follows:  $\eta = 0.001$ ,  $\rho_1 = 0.9$ ,  $\rho_2 = 0.999$ , and  $\epsilon = 10^{-8}$ . In many cases, various deep learning frameworks also use this value.

When training neural networks, the initial values of the weights are particularly important to solve the gradient vanishing problem. Xavier initialization [6] is used for the linear activation function, the number of neurons in the previous layer is  $n$ , and a Gaussian distribution with a standard deviation of  $\sqrt{n}$  is used as the initial value. This initial value is appropriate for sigmoid and tanh activation functions because they are symmetric and can be approximated as linear functions. He initialization [7] used for the ReLU function, where the number of neurons in

the previous layer is  $n$ , and a Gaussian distribution with a standard deviation of  $\sqrt{2/n}$  is used as the initial value.

In neural network training, the gradient of the loss function with respect to the weight parameters is obtained by numerical differentiation. Numerical differentiation is simple and easy to implement, but time-consuming. Therefore, the backpropagation algorithm is widely used in machine learning to train FNN. Backpropagation computes the gradient of the loss function with respect to the network weights and is efficient compared with those that compute the gradient directly with respect to the individual weights. This makes it easy to train the multilayer neural networks. The weights are updated to minimize the loss function. Figure 3.3 shows an example of each layer with one neuron. Focusing on the connection between the last two neurons, the lost function of a single training defines  $E_0$ .

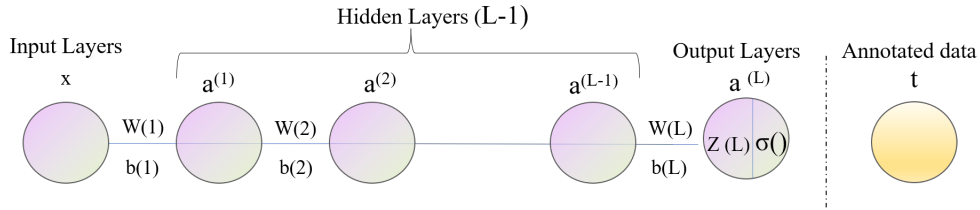


Figure 3.3: Gradient descent algorithm

If  $a$  is the output of the neuron and  $L$  is the layer, then the last layer is  $a^L$ .  $t$  is the actual annotated value, then the equation of  $E_0$  is as follows:

$$E_0 = \frac{1}{2}(a^{(L)} - t)^2 \quad (3.21)$$

Meanwhile,  $z$  is an variable arbitrary of  $w$  as weight and  $b$  as bias, and  $\sigma$  is the activation function; then  $z^L$  and  $a^L$  are as follows:

$$z^{(L)} = w^{(L)}a^{(L-1)} + b^{(L)} \quad (3.22)$$

$$a^{(L)} = \sigma(z^{(L)}) \quad (3.23)$$

The derivative of  $E_0$  with respect to  $w^L$  is called the chain rule. It describes the sensitivity of  $E$  to small changes in  $w^L$ :

$$\frac{\partial E_0}{\partial w^{(L)}} = \frac{\partial z^{(L)}}{\partial w^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial E_0}{\partial a^{(L)}} \quad (3.24)$$

When derivation of above function is calculated;

$$\frac{\partial E_0}{\partial w^{(L)}} = a^{(L-1)} \cdot \sigma'(z^{(L)}) \cdot (a^{(L)} - t) \quad (3.25)$$

$a^{(L-1)}$  is the output of the previous neuron during the feed forward calculation. The derivative with respect to  $w^L$  and  $a^{(L-1)}$  are calculated same method:

$$\frac{\partial E_0}{\partial b} = \sigma'(z^{(L)}) \cdot (a^{(L)} - t) \quad (3.26)$$

$$\frac{\partial E_0}{\partial a^{(L-1)}} = w^{(L)} \cdot \sigma'(z^{(L)}) \cdot (a^{(L)} - t) \quad (3.27)$$

As a result, the backpropagation algorithm continues to iterate this chain rule backward to compute the sensitivity of the loss function to previous weights and biases. This basic idea is applied to deep layers with multiple neurons.

### 3.1.3 Convolutional Neural Network

The image includes the three-dimensional shape, width, height, and channel, and it has important spatial information. Specifically, spatially close pixels have similar values, and there is a close relationship between the RGB channels. However, the fully connected layer ignores the shape and cannot make use of the shape information because all the input images are converted into one-dimensional images. A CNN is designed to automatically and adaptively learn the spatial hierarchies of features through backpropagation using multiple building blocks: convolutional, pooling, and fully connected layers. The convolution layers can maintain the spatial information.

A convolutional layer in Figure 3.6 is a fundamental component of the CNN architecture that performs feature extraction. The two parameters of the convolutional layer are the kernel and the bias. The kernel has a two-dimensional shape, that is, width and height. During the forward pass, each filter strides across the width and height of the input volume, computing the dot products between the filter entries and input at each position. As the filter strides across the width and height of the input volume, a two-dimensional activation map is created that represents the responses of the filter at each spatial location. The distance between two successive filter positions is called the stride, as shown in Figure 3.4.

Zero-padding is used to control the spatial size of the output volume. This pads the input volume with zeros around the border, because the size of the output volume is reduced when only convolutions are performed, as shown in Figure 3.5

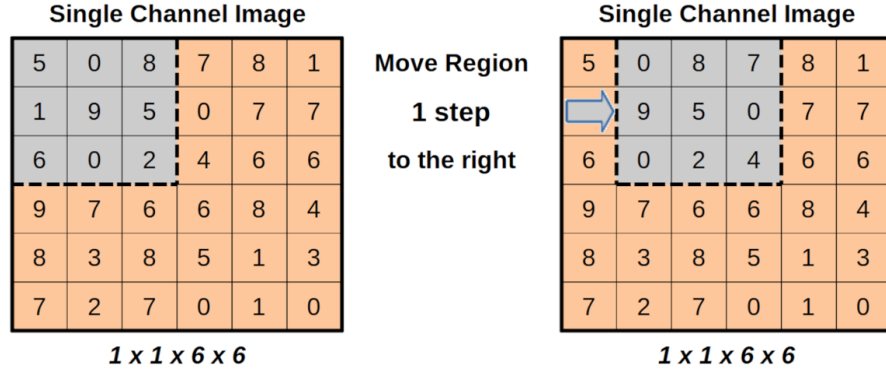


Figure 3.4: Stride

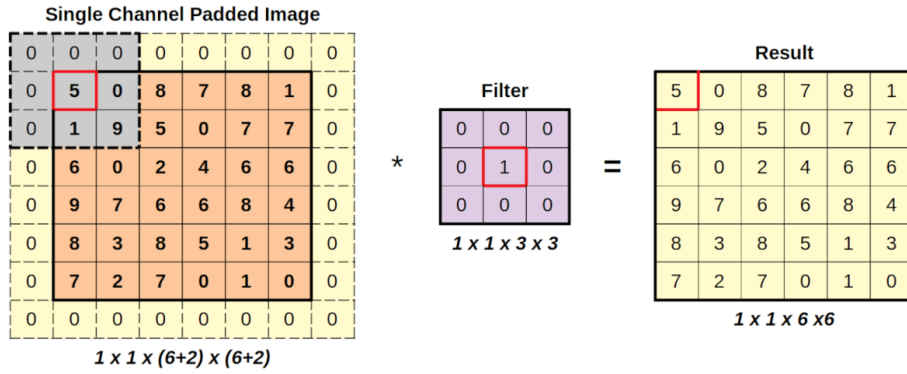


Figure 3.5: Padding

Assume an input size of  $H \times W$ , filter size of  $FH \times FW$ , output size of  $OH \times OW$ , stride of  $S$ , and padding of  $P$ , then the output size follows the equation:

$$OH = \frac{H + 2P - FH}{S} + 1 \quad (3.28)$$

$$OW = \frac{W + 2P - FW}{S} + 1 \quad (3.29)$$



The filter size, stride, and padding are hyperparameters. Therefore, OH and OW values must be divisible. If there are multiple feature maps as channels, convolution of the input data and filters can be performed for each channel, and the results can be added to produce a single output. The number of channels must be the same as that of the input channels.

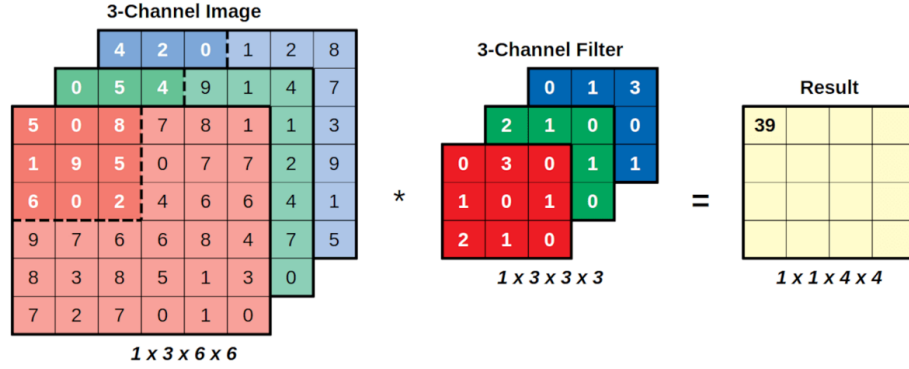


Figure 3.6: Convolution layer

The pooling layer described in Figure 3.7 reduces the spatial size of the feature maps to reduce the number of parameters and introduces translation invariance into small shifts and distortions. There are no learnable parameters, and the number of channels for the input and output data does not change. There are mainly two types of operations: 1) Max pooling, which extracts patches from the input feature maps, outputs the maximum value in each patch, and discards all other values, and 2) Average pooling, which extracts patches from the input feature maps and outputs the average value in a target area.

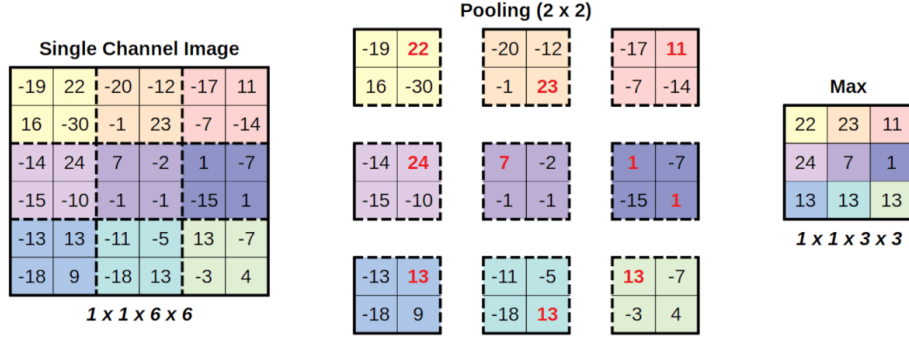


Figure 3.7: Pooling layer

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC), an image recognition competition using ImageNet, was held annually between 2010 and 2017. Various CNN models with excellent object recognition have been developed. In the following, AlexNet, which led to the breakthrough of CNNs; VGG, which is widely used owing to its simple structure; and ResNet, which enables deep models with a large number of layers are described.

- **AlexNet** [8] automatically extracts features from images for object recognition using CNN concept. AlexNet achieved good results at ILSVRC 2012 compared with models previously designed with features from images by human designers. AlexNet comprises three convolutional layers, two pooling layers, and three fully connected layers. Although its architecture is similar to that of LeNet, it contains a huge number of parameters. LeNet has 60k parameters, while AlexNet has 60M parameters. Therefore, it was necessary to train the network while reducing the vanishing/explosive gradient and overfitting. Compared with ReNet, AlexNet is mainly used as follows: 1) The ReLU activation function. 2) Local response normalization (LRN). 3) Dropout and 4) Data augmentation. Replacing the tanh or sigmoid function with a ReLU function prevents the vanishing or exploding gradient, and speeds up learning by increasing the gradient values. LRN is a local normalization method for the response of the middle

layer in image recognition CNNs that improves the local contrast of the feature maps. LRN performs local normalization of the responses of the convolutional and pooling layers of a CNN, and locally normalizes the output values of each layer in the channel direction. This increases the generalization of the CNN. Dropout is a learning method that randomly deletes the neurons. During the training, the neurons to be deleted are randomly selected for each data stream. During testing, the signals of all neurons are transmitted, but the output of each neuron is multiplied by the percentage eliminated during training. On the other hand, multiple GPUs are introduced to reduce the computational cost.

- VGG [9] was designed to demonstrate the performance of image recognition CNNs with even deeper layers, which was the state-of-the-art at the time, AlexNet. The concept of VGG, which enables a deeper layer, is easy and widely used in various fields: 1) using only  $3\times 3$  (or partly  $1\times 1$ ) convolution, 2) reducing the feature map by half by max pooling after stacking several convolutional layers with the same number of output channels, and 3) increasing the number of output channels in the convolutional layer after max pooling by a factor of 2. In this study, six models with different numbers of convolutional layers were confirmed, and VGG16 and VGG19 achieved excellent performance and were the two most commonly used as structures. VGG16 comprises 13 convolutional layers, 4 pooling layers, and 3 fully connected layers. VGG19 comprises 16 convolutional layers, 4 pooling layers, and 3 fully connected layers. The convolutional layer had a kernel size of  $3\times 3$  and stride of 1. The pooling layer has a size and stride of  $2\times 2$ . All connected layers had 4096, 4096, and 1000 neurons.
- **ResNet** [10] was designed to address the degradation problem and vanishing gradient problem for deep layers. There is a problem that the computational cost increases with the size of the model, but theoretically it was thought that more complex problems could be solved by increasing the number of parameters

and building complex deep neural networks. However, even if the model is deepened, the accuracy reaches a plateau at a certain point; further deepening layers cannot maintain this, and the accuracy degrades. Specifically, the 34-layer plain network had an experimentally higher validation error than the 18-layer plain network. This is known as the degradation problem. This implies that when considering a 34-layer network, the later 16 layers cannot train the input data. Therefore, to solve this problem, the direct path between the input and output is added to the original path, implying an identity mapping. The entire module in Figure 3.8 is called a residual module. It also addresses the vanishing gradient problem. The equation shows that the partial derivative of the loss function with respect to the input  $x$  remains  $\frac{\delta F}{\delta H}$ , which means that the gradient remains in the previous layers. Therefore, this residual module is useful in propagation algorithms. ResNet can be trained and outperformed performance at a network depth of 152. Furthermore, one of the characteristics is to use batch normalization. The advantages of this method are the reduction of the learning time, reduction of the dependence on the initial values, and suppression of over-fitting. This is done by normalizing each mini-batch as a unit of mini-batches for the training process. Specifically, it transforms the input data of the mini-batch into data with a mean of 0 and variance of 1. The Batch Norm layer performs a transformation on this normalized data with a unique scale as follows:

$$y_i = \gamma \hat{x}_i + \beta \quad (3.30)$$

The  $\gamma$  and  $\beta$  are parameters. It starts with  $\gamma = 1$ ,  $\beta = 0$  and is adjusted to suitable values through learning.

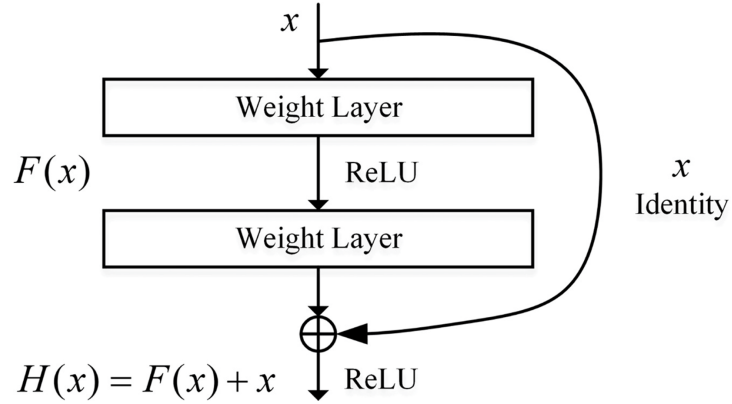


Figure 3.8: Residual Block

$$\frac{\delta L}{\delta x} = \frac{\delta L}{\delta H} \frac{\delta H}{\delta x} = \frac{\delta L}{\delta H} \left( \frac{\delta F}{\delta x} + 1 \right) = \frac{\delta L}{\delta H} \frac{\delta F}{\delta H} + \frac{\delta F}{\delta H} \quad (3.31)$$

### 3.1.4 Vision Transformer

Since the introduction of the transformer in 2017, models based on the Transformer, such as BERT and GPT, have shown high performance on various benchmarks in the natural language processing domain without using previously used RNNs or CNNs, which has generated a great deal of interest. RNNs required sequential processing of the input series data because the hidden state obtained from the previous time was used as input for the processing of the next time. This makes the computation inefficient, because it cannot be parallelized. In contrast, CNNs can be parallelized, but have the problem that it is difficult to capture the relationship between tokens that are far apart. This makes it possible to parallelize the computation while capturing the relationship between distant tokens.

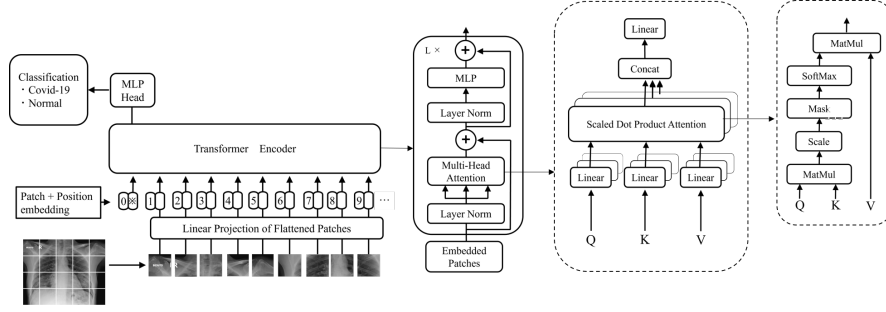


Figure 3.9: Vision transformer

Vision transformer (ViT) [11] is a transformer-based model, and its architecture is illustrated in Figure 3.9. The ViT consists of three main parts: the Input Layer, Encoder, and MLP Head. The standard transformer captures the token embedding as a 1D sequence, so the 2D input image  $X \in \mathbb{R}^{H \times W \times C}$  is reshaped as a flattened 2D patch sequence  $X_p \in \mathbb{R}^{N_p \times (P^2 \cdot C)}$ , where  $(H, W)$  is the input image resolution,  $C$  is the number of channels in the input image, and  $(P, P)$  is the patch image resolution, and  $N$  is the number of each patch image. These flattened patches are mapped to  $D$  dimensions in a trainable linear projection layer, and the  $i$ -th output of this projection  $X_p^i \in \mathbb{R}^{(P^2 \cdot C)}$  is represented as a patch embedding. For the classification task, the class token is prepended to the sequence of patch embeddings, and a positional embedding is appended to the class token and patch embedding to obtain positional information. Finally, the input  $z$  to the transformer encoder can be written as

$$z_0 = [x_{class}; x_p^1 E; x_p^2 E; \dots; x_p^{N_p} E] + E_{pos}, \quad E \in \mathbb{R}^{(P^2 \cdot C) \times D}, E_{pos} \in \mathbb{R}^{(N_p + 1) \times D} \quad (3.32)$$

The transformer encoder consists of several encoder blocks, each comprising Layer Normalization (LN), Multihead Self-Attention (MHSA), and Multi-layer Perceptrons (MLP). Self-attention can learn image features globally by capturing the similarities between all patches. The input  $z$  is projected into Query, Key, and Value, where  $Q = zW^Q$ ,  $K = zW^K$  and  $V = zW^V$  via  $W^q \in \mathbb{R}^{D \times D_h}$ ,  $W^k \in \mathbb{R}^{D \times D_h}$ ,  $W^v \in \mathbb{R}^{D \times D_h}$ . Then, the corresponding attention weight can be written as

$$A = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \quad (3.33)$$

Then, the Self-Attention (SA), which is a product of the  $A$  and  $V$  matrices, is given by

$$SA(z) = AV = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (3.34)$$

The attention weight calculates the inner product of each entire vector and uses a softmax function; therefore, it has only one peak, and small relationships may be lost. Because multiple peaks can improve the expressive power of the network, the MHSA is introduced to obtain multiple attention weights by embedding multiple queries, keys, and values in a single patch. If the number of heads is  $k$  and the self-attention of the  $i$ -th head is  $SA_i(z)$ , the following equation is obtained:

$$MHSA(z) = [SA_1(z); SA_2(z); \dots; SA_k(z)]W^0, MHSA(z) \in \mathbb{R}^{N \times D} \quad (3.35)$$

Finally, the only input to the MLP is the class token. When the number of classifications is  $M$ , the ViT output  $y$  is as follows:

$$z'_l = MHSA(LN(z_{l-1})) + z_{l-1}, \quad l = 1 \dots L \quad (3.36)$$

$$z_l = MLP(LN(z'_l)) + z'_l, \quad l = 1 \dots L \quad (3.37)$$

$$y = LN(Z_L^0)W^y, \quad Z_L^0 \in \mathbb{R}^D, \quad W_L^0 \in \mathbb{R}^D \quad (3.38)$$

## 3.2. Learning method

### 3.2.1 Transfer Learning

Transfer learning utilizes the knowledge of a trained model to learn another set of data. Transfer learning aims to improve learning in the target domain using knowledge from the source domain and learning tasks. This strategy is a common and effective way to train a network on a limited small dataset, such as medical domains, where a network is pretrained on an extremely large dataset, such as ImageNet. Several transfer learning methods have been defined based on the nature of the task and type of data available in the source and target domains. The two main transfer learning scenarios in the medical field are as follows: 1) fixed feature extractor, which removes the last fully connected layer from the pre-trained model while maintaining the remaining network. This is replaced by a

linear classifier, such as a linear support vector machine or a softmax classifier. This approach is uncommon in medical images because of the dissimilarity between medical and natural images. 2) Fine-tuning, which is a process of not only replacing fully connected layers of the pre-trained model with a new set of fully connected layers to retrain on a given dataset but also fine-tuning all or part of the weights in the pre-trained convolutional base by means of backpropagation. All layers can be fine-tuned; alternatively, some earlier layers can be fixed, whereas the rest of the deeper layers can be fine-tuned. This is because the features are more generic in the early layers and more original datasets are specific in the later layers. This fine-tuning method is often applied to medical imaging tasks, especially in radiology research.

### **3.2.2 Self-Supervised Learning**

Self-Supervised Learning (SSL) is a solution to the aforementioned issues and has emerged as one of the most promising techniques that do not necessitate any manual annotations. The SSL training method produces representations using unlabeled images, and is a type of unsupervised learning. SSL is considered a bridge between supervised and unsupervised learning. There are two ways to use SSL: 1) auxiliary pretext tasks, which are auxiliary pretext tasks used to learn representations using pseudo-labels or labels that were created automatically based on the dataset's attributes. 2) Contrastive learning that distinguishes augmented image features. Recent approaches that have proven successful in the medical field use self-supervised contrastive pre-training, followed by supervised fine-tuning. One of the major contrastive learning methods is SimCLR, the SSL method, which learns visual representations as a pretraining process.



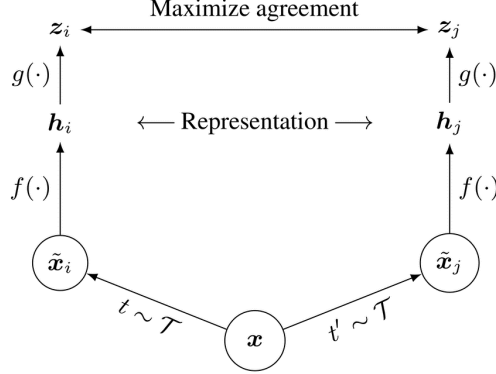


Figure 3.10: A simple framework for contrastive learning of visual representations from Chen et al. [12]

SimCLR [12] is a simple framework that does not require special architecture or a memory bank. Here, an arbitrary image  $x$  is differentially augmented  $\tilde{x}_i$  and  $\tilde{x}_j$  as positive pairs. In addition,  $f(\cdot)$  is an encoder network that generates representations, where  $h_i = f(\tilde{x}_i)$  and  $h_j = f(\tilde{x}_j)$ , and  $g(\cdot)$  is a neural network projection head with one hidden layer used for contrastive loss, where  $z_i = g(h_i) = W^{(2)}\sigma(W^{(1)}h_i)$  and  $z_j = g(h_j) = W^{(2)}\sigma(W^{(1)}h_j)$ .  $N$  is an arbitrary number of batches, with  $2N$  positive pairs in each batch and  $2(N - 1)$  negative pairs. SimCLR maximizes the agreement of the positive pairs and minimizes the agreement of the negative pairs using the contrasting loss, as follows:

$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(z_i, z_k)/\tau)} \quad (3.39)$$

where  $\tau$  is a temperature parameter, and  $\mathbb{1}$  means if  $k = i$  then 0 and  $k \neq i$  then 1. In addition,  $\text{sim}()$  denotes the cosine similarity, and  $N$  is the batch size.

$$\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{2N} [\ell(2k - 1, 2k) + \ell(2k, 2k - 1)] \quad (3.40)$$

The networks  $f$  and  $g$  is trained to minimize  $\mathcal{L}$ .

# Chapter 4

## Dataset and Pre-processing

### 4.1. Image type

#### 4.1.1 Natural image

ImageNet [13] is one of the most popular image recognition datasets, and is often used as a benchmark for image recognition. Some well-known image recognition datasets include MNIST [14], which contains tens of thousands of images with handwritten numbers from 0 to 9, and CIFAR-10 [8], which contains tens of thousands of color images classified into ten classes, including vehicles and animals. ImageNet is a large image database that is designed for object recognition. ImageNet contains above 14 million images, with over 20,000 manually annotated images. Bounding boxes have also been assigned to more than one million images. The ILSVRC, an image recognition competition using ImageNet, was held annually between 2010 and 2017. In the 2012 ILSVRC, a team led by Hinton et al. presented the AlexNet CNN model. Prior to the introduction of AlexNet, it was common for human inputs to be incorporated into machine learning models. AlexNet does not require human input of features and it significantly outperforms the support vector machine (SVM) methods used up to that point. Consequently, AlexNet won first prize at ILSVRC 2012 using the CNN model, and DL has become the focus of much attention.

### 4.1.2 Medical image

Three independent datasets were used for the evaluation. The BrixIA datasets include annotated CXR images for COVID-19 [15], Valencia Region Image Bank (BIMCV) datasets include annotated CXR and CT images for COVID-19 and normal cases [16], and the National Institutes of Health (NIH) dataset includes annotated CXR images for 14 diseases without COVID-19 and normal cases [17]. The first BrixIA dataset comprises 4,707 CXR images of COVID-19 objects taken for both triage and patient monitoring in sub-intensive and intensive care units for one month between March 4 and April 4, 2020, at ASST Spedali Civili di Brescia. The images were retrieved from the facility’s Picture Archiving and Communication Systems (PACS) and disclosed in anonymized digital imaging and communications in medicine (DICOM) formats and annotation files in CSV with additional information (i.e., severity, patient’s age and gender, and modality manufacturer). This study was approved by the Ethical Committee of Brescia (Italy). The second BIMCV dataset contains CXR and computed tomography (CT) images of COVID-19 and non-COVID-19 patients, including DICOM metadata and radiologic reports. COVID-19 patients were identified with at least one positive PCR test or a positive immunological test by querying the Laboratory Information System records from the Health Information Systems in the Comunitat Valenciana. The first iteration of the database included 1,380 CX, 885 DX CXR images, and 163 CT images from 1,311 COVID-19 patients. This HIPAA-compliant retrospective cohort study was approved by the Institutional Review Board (IRB) of Miguel Hernandez University (MHU) approved this HIPAA-compliant retrospective cohort study. The study was approved by the local institutional ethics committee CElm: 12/2020 at the Arnaute Vilanova Hospital in the Valencia Region. The healthcare authorities of Comunitat Valenciana authorized the publication of an open database. The third dataset Chest X-ray 14 dataset comprises 112,120 CXR images of 14 diseases (i.e., atelectasis, cardiomegaly, effusion, infiltration, mass, nodule, pneumonia, pneumonia, pneumothorax, edema, emphysema, fibrosis, pleural, thickening, and hernia), and one for the normal label from 30,805 unique patients. CXR images were extracted from the PACS database through natural language processing (NLP) at the National Institutes of Health (NIH) Clinical Center between 1992 and 2015 and disclosed in portable network graphics

(PNG) formats with additional information (i.e., patient ID, age, gender, and view position). With paramount patient privacy, the dataset was rigorously screened to remove all personally identifiable information before release.

### 4.1.3 DICOM

Digital Imaging and Communications in Medicine (DICOM) is a common standard for medical imaging that defines medical image formats for CT, MRI, and CR, as well as the communication protocols used between medical imaging devices, PACS, and radiology information system (RIS). DICOM is a common medical imaging and communication format that enables the sharing of medical imaging equipment among vendors. DICOM images encapsulate tag information that describes various tag information defined in the standard, such as the format name, data length, unique instance, modality, patient information, audio, and acquisition time. It is a container format in which different types of data can be internalized. Therefore, image data are multiframe, in which multiple images are internalized. The DICOM communication format conforms to the Open Systems Interconnection (OSI) reference model, and data are encapsulated and communicated using the TCP/IP protocol.

## 4.2. Pre-processing

Digital images are represented by numerical values called pixel values. To view this image on a display, pixel values must be converted to brightness (luminance). Windowing converts only a specific density range of an image with a wide range of pixel values, such as a medical image, into a density range of the display system [0 (dark) to 255 (light)]. This process is always performed when the contrast of a digitized image is changed for display. Windowing plays a significant role because medical images often have a density resolution of eight bits or more. This processing has two parameters: the Window Level (WL) and width (WW). Figure 4.1 (a) shows a histogram of pixel values in the original image. The WL and WW determine how the pixel values in the original image are converted to gray levels. Figure 4.1(b) shows the characteristics of only the width of WW in the histogram, where a represents the conversion to gray levels from 0 to 255. In

Figure 4.1, only the width of the WW in the histogram was converted to a gray level between 0 and 255. The values that determine the target area are WL and WW, where WL is the center value of the target area and WW is the width of the target area.

The images stored in DICOM were converted to 8-bit using the WL and WW of the DICOM tag using the following equation:

$$Window = 255 \times \left( \frac{value - min}{max - min} \right), \quad (4.1)$$

$$max = \left( \frac{WL + WW}{2} \right), min = \left( \frac{WL - WW}{2} \right) \quad (4.2)$$

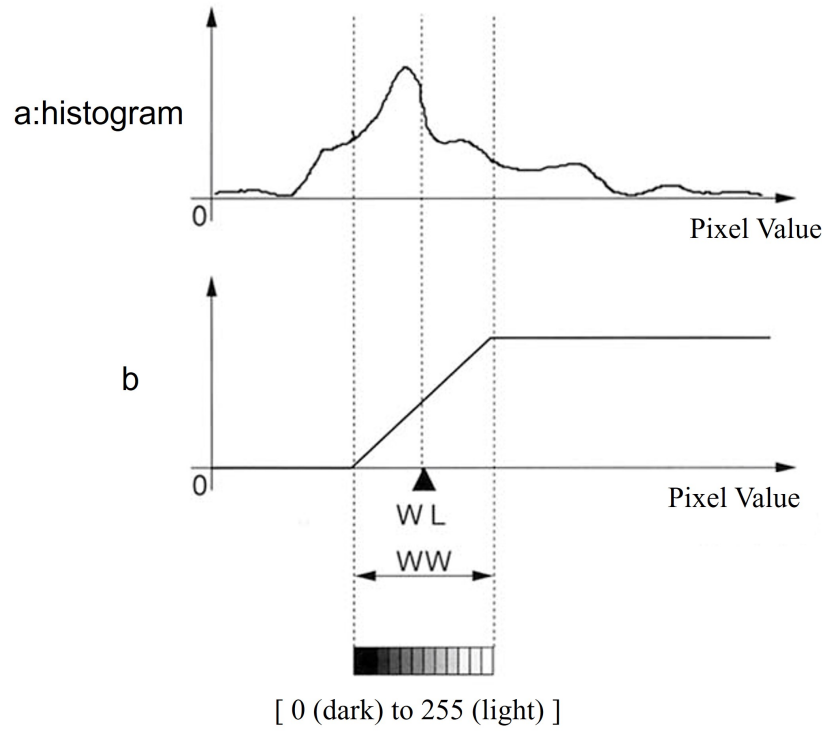


Figure 4.1: Window width and window level in digital imaging.

In this thesis, experiments were conducted using CXR and natural images from

ImageNet, which are widely used in various studies. Therefore, all CXR images were pre-processed for alignment with the ImageNet dataset. The size was resized to  $256 (\times \text{height} \div \text{width}) \times 256$  to fix the aspect ratio, and cropped around the center to  $224 \times 224$ . Converted from a single channel to three channels. Finally, the pixel values of the input images were normalized between 0 and 1.

# Chapter 5

## Performance Index

### 5.1. Area under the curve

The receiver operating characteristic (ROC) curve is a graphical display of the true positive rate (TPR) on the y-axis and the false positive rate (FPR) on the x-axis for various cut-off points. Both axis range from 0 to 1. The TPR is equal to the sensitivity, whereas the FPR is equal to "(1 - specificity)". The area under the curve (AUC) is the definite integral of an ROC curve and is an effective and combined measure of sensitivity and specificity that assesses the inherent validity of a diagnostic test. An AUC closer to 1 indicates better test performance, and the sensitivity and specificity are determined by optimizing the cutoff values. The methods for determining the threshold include minimizing the distance from the point in the upper-left corner using the Yuden index [18] or setting it to 0.5. The Yuden index is often used in the medical field. The Youden Index is the point on the ROC curve that is farthest from the line of equality (diagonal line). The optimal cut-off value at which the determined "sensitivity + specificity - 1" is maximized.

### 5.2. Confusion Matrix

A confusion matrix, also known as an error matrix, is primarily used for statistical classification. This is a specific table layout that allows visualization of the

performance of the algorithm. Each row of the matrix represents an instance in a predicted value, whereas each column represents the actual value. The output matrix has four cells: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). TP means that the actual value and the predicted value are both positive, TN means the actual value is positive but the model predicted value is negative, FP means the actual value is negative but the model predicted value is positive, and, finally, FN means that both the actual and predicted values are negative. Some performances were used in the confusion matrix. The positive predictive value (accuracy) is a measure of the percentage of true positives and true negatives that can be correctly identified for the entire study population.

Table 5.1: Confusion Matrix

		Anotated label	
		Positive	Negative
Predicted Value	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.1)$$

Sensitivity and specificity (precision) indicate the percentage of true detection for each of the positive and negative results, respectively, as annotated labels.

$$Sensitivity(Precision) = \frac{TP}{TP + FP} \quad (5.2)$$

$$Specificity = \frac{TN}{TN + FP} \quad (5.3)$$

### 5.3. Confidence Index

Interval estimation is an estimation of the population (mean or variance) based on a sample drawn from the population. Interval estimation: This method obtain a "confidence interval" which is a range including the population mean and variance with a probability of 95% (or 90%, 99%, etc.). The probability of obtaining a confidence interval is called the "confidence coefficient". A confidence coefficient



of 95% means that if the interval is estimated 100 times by repeated sampling, there is a possibility that the population mean "falls" within the confidence interval 95 times. The lower limit of the confidence interval is called the "lower confidence limit" and the upper limit is called the "upper confidence limit."

The AUC value calculated from the obtained data is denoted by  $A$ , and the standard error of  $A$  is denoted by  $SE(A)$ .  $x\%$  confidence interval  $C(x)$  with respect to  $A$  is expressed as follows:

$$C(x) = A \pm z \left( \frac{1-x}{2} \right) SE(A) \quad (5.4)$$

$z(a)$  is a function that returns the upper  $a\%$  points of a normal distribution. To obtain 95% confidence interval ( $x=0.95$ ), which is often used as a standard in statistics, we use the following formula from the Normal Distribution table:

$$z \left( \frac{1-x}{2} \right) SE(A) = z(0.025) = 1.96 \quad (5.5)$$

As a result, we can show the 95% confidence interval as follows:

$$C(x) = A \pm 1.96SE(A) \quad (5.6)$$

$SE(A)$  in equation 5.4 is calculated by Hanley and McNeil method [19].

$$SE(A) = \sqrt{\frac{A(1-A) + (n_p - 1)(Q_1 - A^2) + (n_N - 1)(Q_2 - A^2)}{n_p n_N}} \quad (5.7)$$

$n_p$  and  $n_N$  represents the number of true positive and true negative. Meanwhile, the study shows that these two probabilities  $Q_1$  and  $Q_2$  are approximated by  $Q_1 = \frac{A}{2-A}$  and  $Q_2 = \frac{2A^2}{1+A}$ .

# Chapter 6

## Performance change with the number of training data

### 6.1. Abstract

One of the features of artificial intelligence/machine learning-based medical devices resides in their ability to learn from real-world data. However, obtaining a large number of training data in the early phase is difficult, and the device performance may change after their first introduction into the market. To introduce the safety and effectiveness of these devices into the market in a timely manner, an appropriate post-market performance change plan must be established at the timing of the premarket approval. In this work, we evaluate the performance change with the variation of the number of training data. Two publicly available datasets are used: one consisting of 4000 images for COVID-19 and another comprising 4000 images for Normal. The dataset was split into 7000 images for training and validation, also 1000 images for test. Furthermore, the training and validation data were selected as different 16 datasets. Two different convolutional neural networks, namely AlexNet and ResNet34, with and without a fine-tuning method were used to classify two image types. The area under the curve, sensitivity, and specificity were evaluated for each dataset. Our result shows that all performances were rapidly improved as the number of training data was increased and reached an equilibrium state. AlexNet outperformed ResNet34 when the number of images was small. The difference tended to decrease as the number of training data

increased, and the fine-tuning method improved all performances. Same trends can be founded to VGG16. In conclusion, the appropriate model and method should be selected considering the intended performance and available number of data.

## 6.2. Introduction

The extensive interest in artificial intelligence (AI) and machine learning (ML) application is growing in the medical field. This is primarily driven by the impressive progress made by deep learning (DL) as a subset of ML because of the increased computational power and an explosion in the availability of large datasets. The number of research papers on the application of DL to medical fields has increased. For the medical image analysis, the number has dramatically increased since 2015. The number of academic papers in major conferences and journals exceeded 300 by the end of 2016 [20]. Furthermore, more than 60 AI/ML-based medical devices have already been approved by the U.S. Food and Drug Administration (FDA) in the United States [21]. There are several types of AI/ML-based medical devices. Muehlematter et al. [22] reported that most AI/ML-based medical devices approved by the FDA are used in radiology, but they span across various medical specialties, such as cardiovascular and neurology. For example, applications in radiology can be categorized into classification, detection, segmentation, etc., and the required performance varies greatly depending on the modality and target diseases.

AI/ML-based medical devices are roughly divided into two types: 1) locked type, which fixes performance prior to marketing and unable to change performance with use; and 2) continuous type, which can change performance by continuously training data after market introduction. To date, several FDA-approved AI/ML-based medical devices are typically locked type, but the FDA announced its marketing authorization for the continuous type on February 2020 [23]. The number of these medical devices is expected to increase in the market in the future. In April 2019, the FDA published a discussion paper for a proposed regulatory framework to account for the iterative nature of AI/ML-based medical devices [2]. The paper describes the total product lifecycle approach. As part of

this framework, the necessity of submitting a predetermined change control plan, in which manufacturers are anticipated to perform modifications to performance, inputs, or intended use prior to marketing, is also described. Other jurisdictions are also preparing papers on regulatory guidance, with a concept similar to that in this discussion paper. Therefore, an appropriate post-market performance change plan must be established at the timing of the premarket approval.

Since the outbreak of the Coronavirus Disease 2019 (COVID-19), many studies have used convolutional neural networks (CNNs) to detect COVID-19 on chest X-ray (CXR) images. For the COVID-19 diagnosis, testing through viral RNA identification in reverse transcriptase polymerase chain reaction (RT-PCR) is currently recommended. However, chest imaging techniques, such as computed tomography (CT) and chest radiography, are considered as part of the diagnostic workup of patients with suspected or probable COVID-19 disease in case RT-PCR is not available or the results are delayed or initially negative in the presence of symptoms suggestive of COVID-19 [24, 25]. Applying ML methods to COVID-19 radiological imaging may improve the diagnosis accuracy compared with the gold-standard RT-PCR while providing a valuable insight into the prognostication of patient outcomes. In particular, chest radiography is widely used, takes less imaging time, and has an accessible diagnostic modality that may be easily brought to the patient’s bed. Arising from the success of the ImageNet Large-scale Visual Recognition Challenge (ILSVRC) in 2012 [26], the CNN is a class of artificial neural networks that has become dominant in object recognition, including radiology [27]. Most of the recent papers on COVID-19 diagnosis were conducted based on existing off-the-shelf models and classified images into two [28–31] or three classes [31–33]: COVID-19 and Normal or COVID-19, non-COVID-19 pneumonia, and Normal. These performances such as accuracy achieves exceed 90% by using data augmentation, transfer learning or combining of publicly available datasets. For example, Nayak et al. [34] evaluated the performance of eight pretrained fundamental CNN models, which achieved excellent results in the ILSVRC, to classify COVID-19 and Normal. They also conducted a comparative analysis by considering hyperparameters, such as batch size, learning rate, and optimizer. ResNet-34 [10] exhibited the best performance, followed by AlexNet [35]. Rahaman et al. [36] evaluated the performance of 15 pretrained

fundamental CNN models to classify COVID-19, non-COVID-19 pneumonia, and Normal. VGG 19 [37] obtained the highest classification accuracy. Tuan [38] performed two- and three-class tasks to classify COVID-19 using three fine-tuned CNN models (i.e., AlexNet, GoogleNet [39], and SqueezeNet [40]) without data augmentation and achieved a high classification performance in terms of accuracy, sensitivity, specificity, precision, F1 score, and area under the curve (AUC). The results suggested that the fine-tuning of network learning parameters is important because it can help avoid the development of more complex models when existing ones can achieve the same or much better performance. On the contrary, many current studies were conducted based on a small number of training data or a combination of training data without demographic statistics (e.g., age and sex distributions) due to the limited publicly available datasets. These estimated performances may probably be optimistic and misleading because of the high-risk bias based on the non-representative selection of control patients and model overfitting [41, 42]. In 2020, Alberto et al. [43] publicly released a large and fully annotated BrixlA dataset of 4703 CXR images related to COVID-19 with additional information on severity, participants' age and sex, and modality manufacturer. A large dataset from the Valencian Region Medical Image Bank containing 3141 CXR and 2239 CT images of patients with COVID-19 along with their radiological findings and locations, pathologies, radiological reports, Digital Imaging and Communications in Medicine (DICOM) metadata, diagnostic antibody tests, etc., was also publicly released [16]. Such large databases are expected to be actively developed and made available to the public in the future. It is hoped that these large datasets with patient demographics will enable the introduction of appropriate AI/ML-based medical devices to the market to support medical decision making through proper regulation.

With the abovementioned background, this study evaluates the performance change as the number of training data increases. Two different CNNs, namely AlexNet and ResNet34, with and without a fine-tuning method are used to classify COVID-19 and Normal. A large BrixlA dataset of CXR images related to COVID-19 are used as the training, validation, and test data. The AUC, sensitivity, and specificity are utilized as the performance evaluation items because they are mainly evaluated through FDA premarket approvals [44]. The major

outcomes of this study are the following: 1) All performances were rapidly improved as the number of training data were increased and reached an equilibrium state. 2) AlexNet outperformed ResNet34 when training data were small, and the difference between the performance of AlexNet and ResNet34 decreased as the training data increased. 3) The fine-tuned CNNs performed better than CNNs trained from scratch in all training datasets; this effect is particularly noticeable in small training data. 4) The change in the performance of the binary classification for COVID-19 and Normal datasets can be generalized as COVID-19 and non-COVID-19 pneumonia datasets.

## 6.3. Material

### 6.3.1 Datasets

Two independent datasets were used for the evaluation. The first dataset is the BrixIA dataset, which comprises 4703 CXR images of COVID-19 objects taken for both triage and patient monitoring in sub-intensive and intensive care units for 1 month between March 4 and April 4, 2020 at ASST Spedali Civili di Brescia. The images were retrieved from the facility’s Picture Archiving and Communication Systems (PACS) and disclosed as DICOM formats with additional information (i.e., severity, patient’s age and gender, and modality manufacturer). The second data set is a Chest X-ray14 dataset comprising 112120 CXR images with 14 diseases and one for the Normal label from 30805 unique patients. The CXR images were extracted from the PACS database through natural language processing (NLP) at the National Institutes of Health Clinical Center between 1992 and 2015 and disclosed as portable network graphics formats with additional information (i.e., patient ID, age, and gender and view position).

The DICOM images in the BrixIA dataset, whose window width (WW) and window center (WC) could not be derived from DICOM Tags, were excluded. The remaining DICOM images were converted from 16-bit into 8-bit through windowing, which changed the picture’s appearance to highlight particular structures using the WW and WC derived from the DICOM Tags. Figure 6.1 depicts an example of the CXR images from both COVID-19 and Normal classes. We ran-

domly selected 4000 COVID-19 images and 4000 CXR images labeled as Normal from the Chest X-ray14 dataset. The combined dataset, which contained 8000 CXR images, was split into two to separate 7000 images for training and validation and 1000 images for testing. The training and validation data were selected as 16 different datasets ( $N = 250, 500, 750, 1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500, 5000, 5500, 6000, 6500, \text{ and } 7000$ ). The ratio of the COVID-19 and Normal classes had the same proportion in all the training, validation, and test data. Table 6.1 and Figure 6.2 depict the detailed patient demographics and age distribution for each dataset.

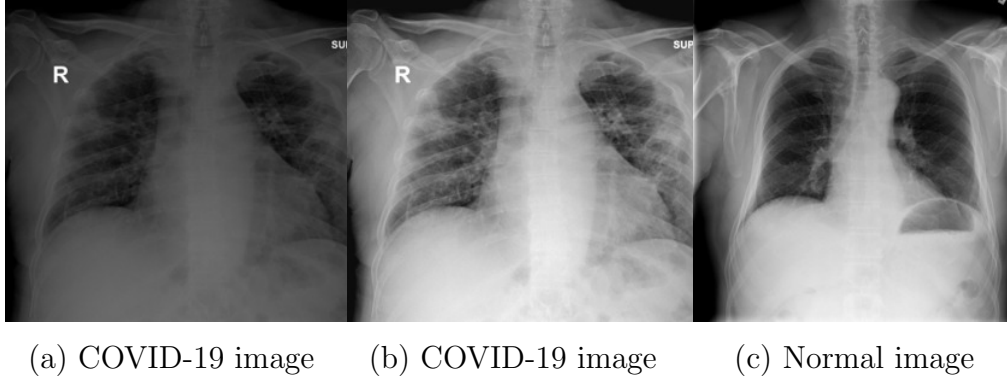


Figure 6.1: Example of CXR images: (a) COVID-19 image without windowing; (b) COVID-19 without windowing derived in the BrixLA dataset; and (c) Normal image from the chest X-ray dataset.

Table 6.1: Data and patient characteristics. AP: anteroposterior and PA: posteroanterior representing the view position. We cannot find AP and PA in the DICOM Tags, but the ratio is reported as AP (87%) and PA (13%) in the BrixLA dataset.

Database Origin	Purpose	Label	Images	Patients	Female	Male	Age	AP	PA
Brixla	Traning and Validation	COVID-19	3500	1804	1060	2440	$59 \pm 14$	-	-
	Test		500	429	130	370	$58 \pm 14$	-	-
Chest X-ray14	Traning and Validation	Normal	3500	3026	1521	1979	$45 \pm 17$	1229	2271
	Test		500	491	223	277	$45 \pm 17$	175	325

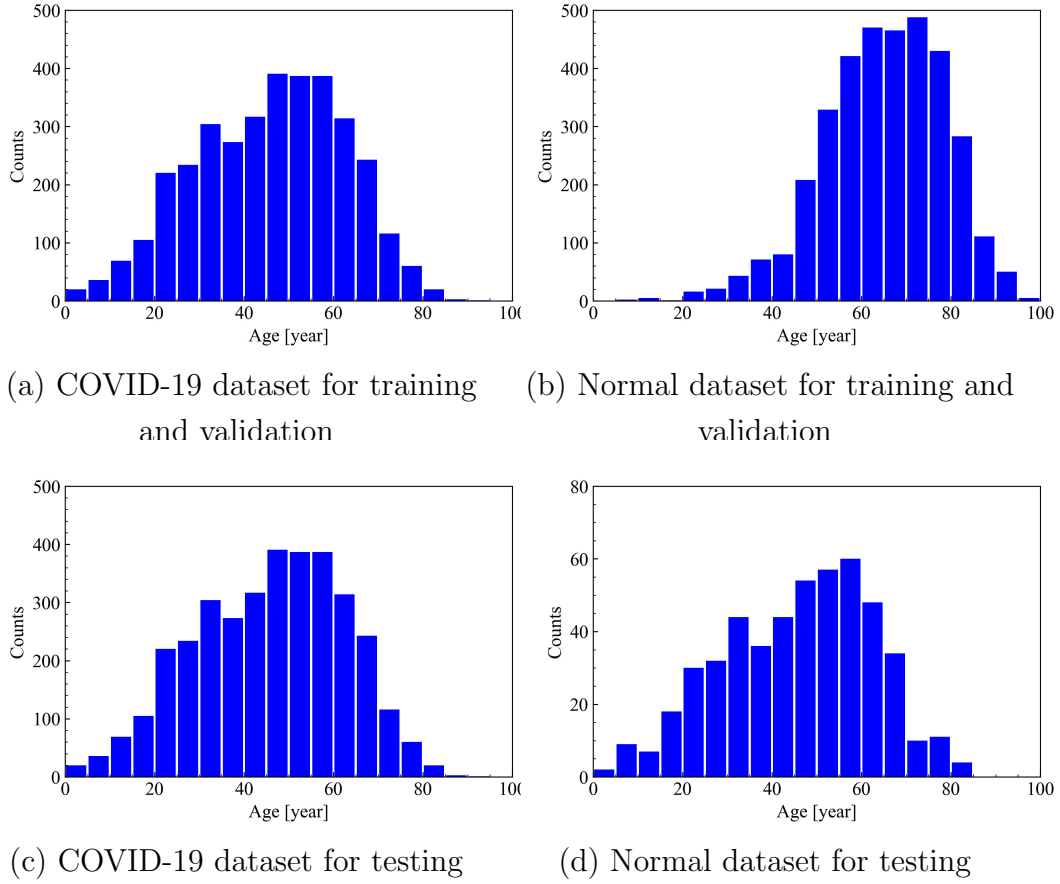


Figure 6.2: Age distribution for each dataset.

## 6.4. Methodology

Figure 6.3 and Figure 6.4 show the proposed method for classifying COVID-19 and Normal, which mainly consisted of preprocessing and classification with a fine-tuned CNN. The detailed preprocessing, classification and evaluation are described in the subsequent sections. The 16 training and validation datasets fed to each CNN model with and without a fine-tuning method were evaluated on the common test dataset.



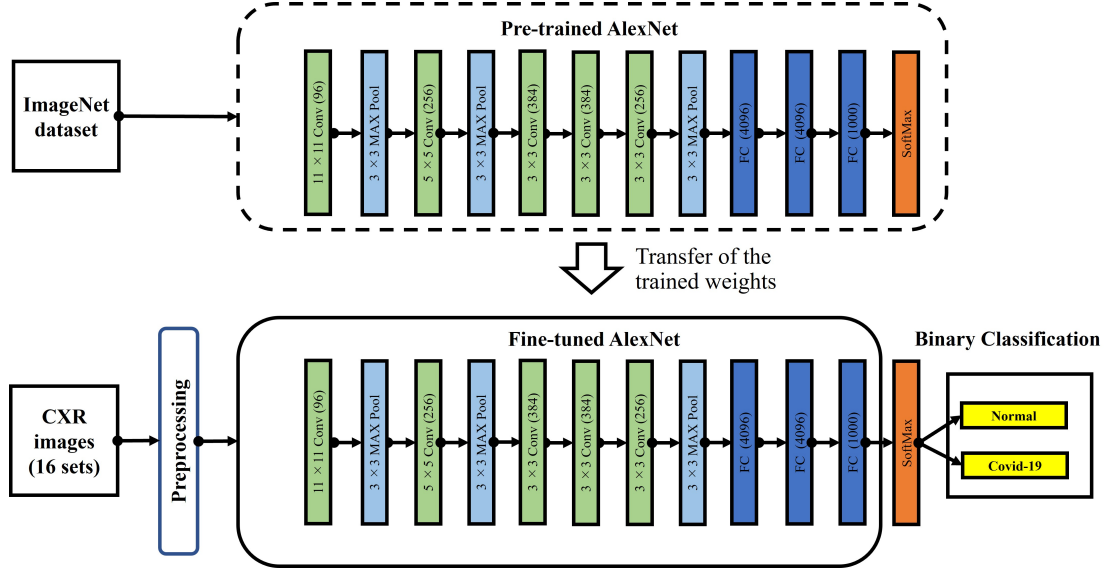


Figure 6.3: Proposed method for classifying COVID-19 and Normal by using a fine-tuning method, that is AlexNet.

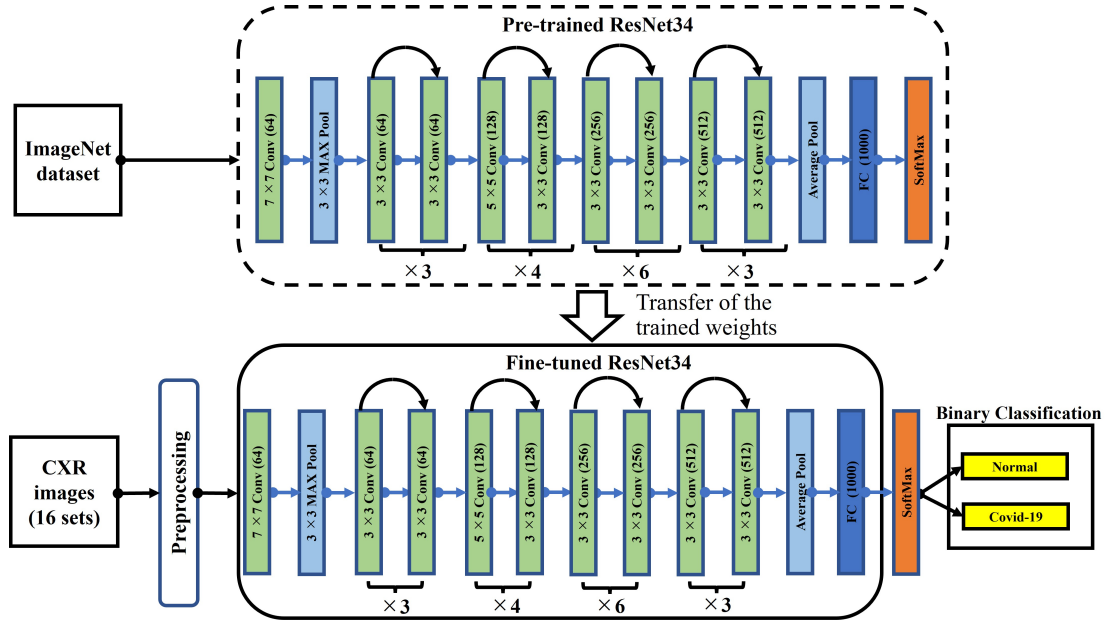


Figure 6.4: Proposed method for classifying COVID-19 and Normal by using a fine-tuning method, that is ResNet34.

### 6.4.1 Preprocessing

Before feeding the CXR images to the system as input, all CXR images were resized to  $256 \times 256$  px and cropped in the center as  $224 \times 224$  px. The grayscale images were converted into a colored format to three channels (RGB: red, green, and blue). The pixel values of the input images were normalized in between ranges 0 and 1 based on the mean and the standard deviation to maintain the numerical stability in the CNN architectures and use a pretrained CNN with ImageNet, which contains 1.4 million images with 1000 classes.

### 6.4.2 Classification

The AlexNet and ResNet34 models were used for the classification. AlexNet was treated as the first breakthrough in the CNN model architecture, serving as the winner of the 2012 ILSVRC. It adopted an eight-layer network structure consisting of five convolutional layers and three fully connected layers. After each convolution in the five convolutional layers, maximum pooling was performed to reduce the amount of data. Data augmentation and dropout were used to reduce overfitting. Rectified linear units were utilized as an activation function instead of a sigmoid or hyperbolic tangent function. AlexNet had eight layers, and the number of parameters was approximately 60 million. Meanwhile, ResNet was the winner of the 2015 ILSVRC, which enables the training of up to hundreds or thousands of layers and inspired many other models. When the network is too deep, the gradients from where the loss function is calculated easily shrink to zero after several chain rule applications. This result on the weights never updates its values; therefore, no learning is performed. ResNet overcomes this degradation problem by introducing residual connections mapping to fit input from a previous layer to the next layer and achieves compelling performance. ResNet34 had 34 layers and 21.8 million parameters.

In addition to the AlexNet and ResNet34 CNN models without pretraining, we utilized a pretrained network, called the fine-tuning method. This method is often applied to radiology studies to replace the fully connected layers of the pretrained model with a new set of fully connected layers to retrain on a given dataset and finetune all the kernels in the pretrained convolutional base by means

of backpropagation. All convolutional base layers can be finetuned. Alternatively, some earlier layers can be fixed while fine-tuning the remaining deeper layers. Our method requires the unfreezing of the entire model originally trained on a large-scale labeled dataset called ImageNet and re-training it on CXR images. Data augmentation was not conducted herein. This work is motivated by the observation that early-layer features appear more generic (e.g., edges applicable to various datasets and tasks), while later features progressively become more specific to a particular dataset or task [45, 46].

### 6.4.3 Evaluation

The AUCs derived from the receiver operating characteristic (ROC) curve, sensitivity, and specificity were used for the evaluation. The ROC curve is graphical display of the true positive rate (TPR) on the y-axis and the false positive rate (FPR) on the x-axis for varying the cut-off points. Both axes are from 0 to 1. The TPR is equal to the sensitivity, while the FPR is equal to "(1 - specificity)". The AUC is the definite integral of an ROC curve and an effective and combined measure of sensitivity and specificity that assesses the inherent validity of a diagnostic test. An AUC closer to 1 indicates a better test performance. Sensitivity and specificity were determined by using the cut-off point defined as the Youden Index [47]. The Youden Index is the point on the ROC curve that is farthest from the line of equality (diagonal line). The optimal cut-off value is that at which the determined "sensitivity + specificity - 1" is maximized. To construct a 95% confidence interval, the standard error was calculated using the Hanley and McNeil method [19] for the AUC and the Wald method [18] for sensitivity and specificity. A nonlinear function,  $y = a - bx^{(-c)}$ , was also fitted to each data. All results will be described herein to show the relation between each performance and the number of training images.

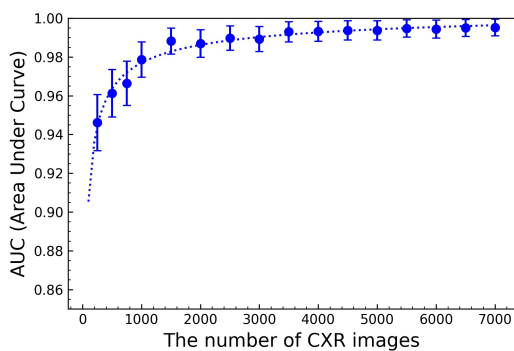
## 6.5. Experiment and results

The 16 training and validation datasets were divided into 32 batches. The training of the AlexNet and ResNet34 CNN models with and without a fine-tuning method was conducted by using an Adam optimizer ( $\beta_1=0.9$ ,  $\beta_2=0.999$ ).

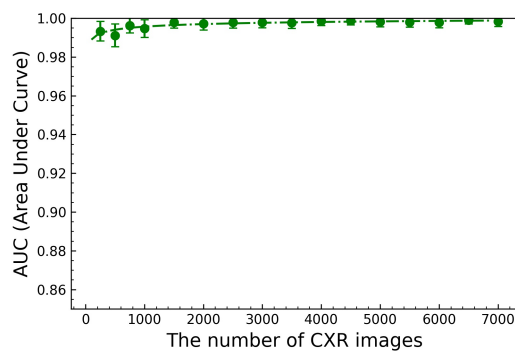
The network was trained 50 epochs. The learning rate was set to  $1 \times 10^{-5}$  for AlexNet and  $1 \times 10^{-6}$  for ResNet34. The hyperparameters were determined based on the comprehensive study of Nayak et al. [34]. To assess the performance of each CNN model, the ROC curve and the AUC on the common test dataset were initially determined through training on the 16 datasets. Figure 6.5 shows the relations between the AUC and the number of training images for all CNN models. The sensitivity and the specificity were determined based on the cut-off values determined by using the Youden Index for each ROC curve. Figure 6.6 presents an example of an ROC curve and the cut-off values obtained using the AlexNet model for the following number of CXR images: 500, 1000, 2000, and 4000 images. The sensitivity and the specificity for each ROC curve were determined. Figure 6.7 and Figure 6.8 depict the relationship between the sensitivity and the number of training images and between the specificity and the number of training images. Figure 6.5, Figure 6.7, and Figure 6.8 show the 95% confidence interval and the fitted nonlinear function, in which parameters a, b, and c were calculated using the open-source Python library used for scientific and technical computing (SciPy; version 1.4.1) with the Levenberg–Marquardt algorithm. Table 6.2 presents parameters a, b, and c.

Table 6.2: Determined parameters of the nonlinear function:  $y = a - bx^{(-c)}$

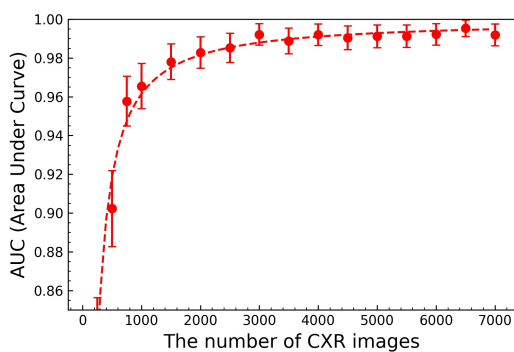
Model	AUC			Sensitivity			Specificity		
	a	b	c	a	b	c	a	b	c
AlexNet	1.0072	1.1554	0.5274	1.0643	0.9958	0.2734	1.0176	0.6486	0.3053
Fine-Tuned AlexNet	1.0033	0.0502	0.2715	72.9690	72.0594	0.0001	1.0043	1.4673	0.5727
ResNet34	0.9994	70.1560	1.0889	115.1554	114.5618	0.0004	0.9670	2336.8376	1.6034
Fine-Tuned ResNet34	0.9991	78.6357	1.3567	0.9885	26.3301	1.0009	1.0035	2.9400	0.6165



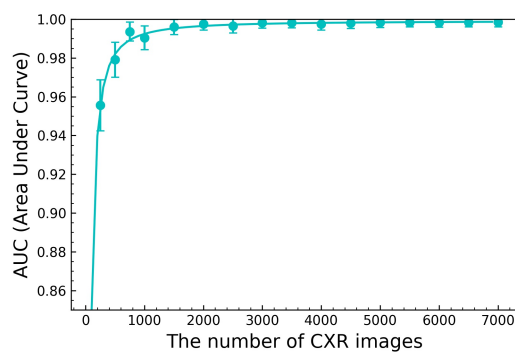
(a) AlexNet



(b) Finetuned AlexNet



(c) ResNet34



(d) Finetuned ResNet34

Figure 6.5: Relation between the AUC and the number of trained CXR images.

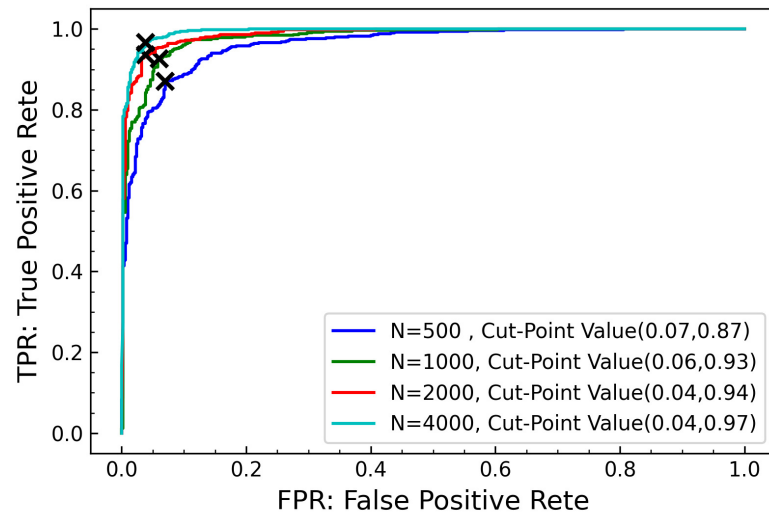
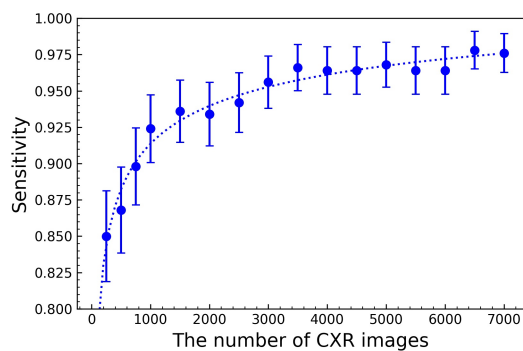
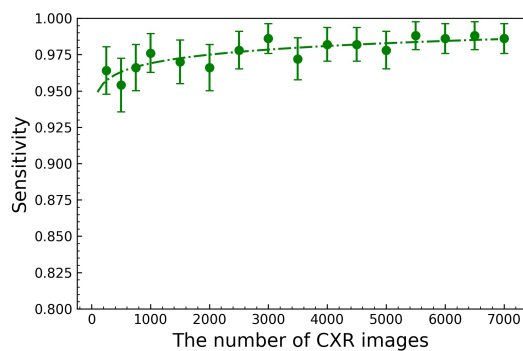


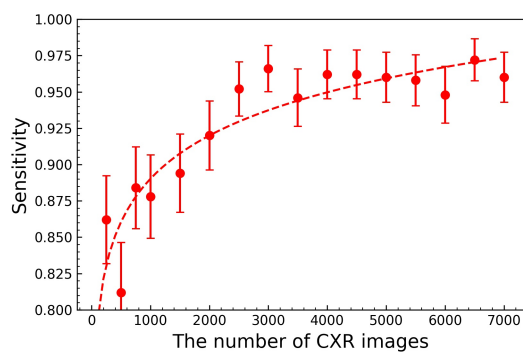
Figure 6.6: Example of the ROC curve and the cut-off value defined as the Youden Index.



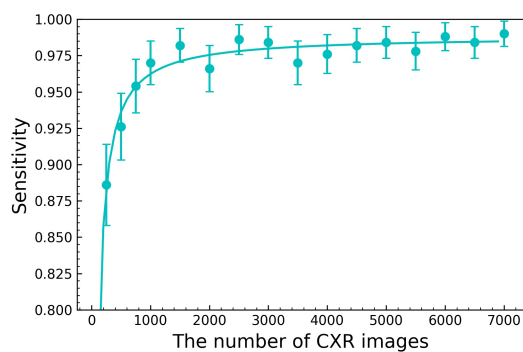
(a) AlexNet



(b) Finetuned AlexNet

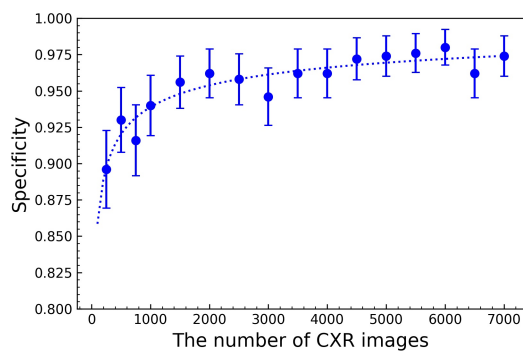


(c) ResNet34

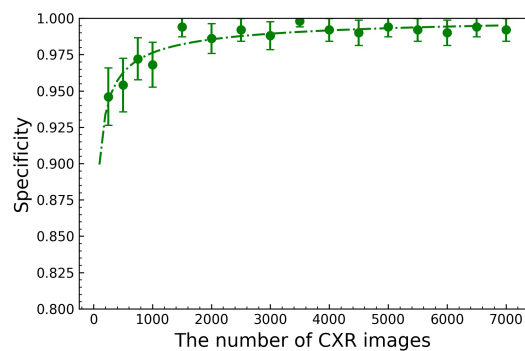


(d) Finetuned ResNet34

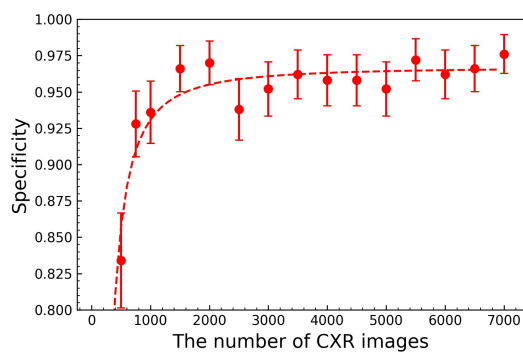
Figure 6.7: Relation between sensitivity and the number of trained CXR images.



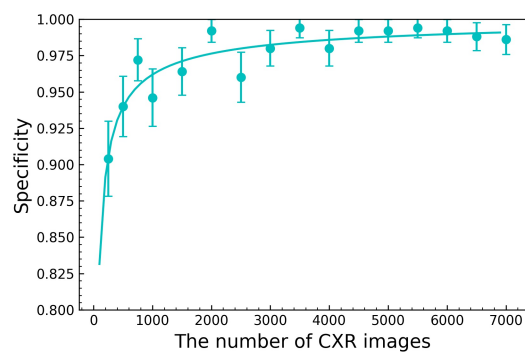
(a) AlexNet



(b) Finetuned AlexNet



(c) ResNet34



(d) Finetuned ResNet34

Figure 6.8: Relation between specificity and the number of trained CXR images.



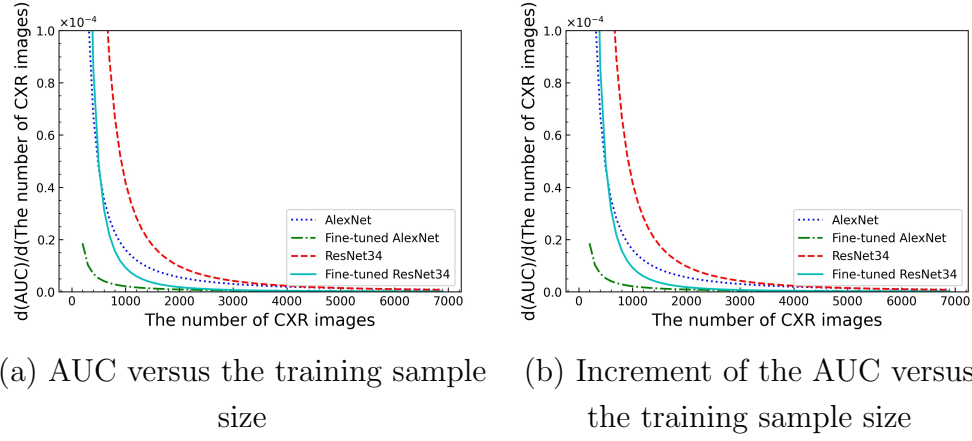


Figure 6.9: AUC versus the training sample size and the increment of the AUC versus the increment of the training sample size.

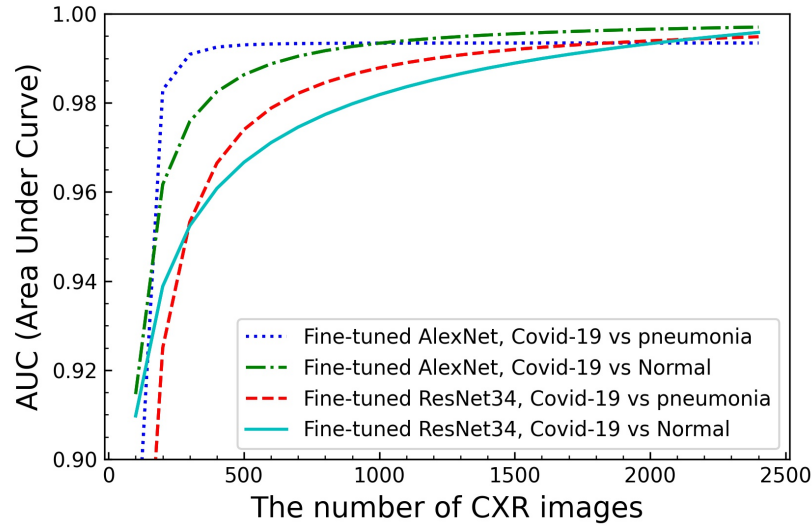


Figure 6.10: AUC versus the training sample size for COVID-19 and Normal and COVID-19 and non-COVID-19 pneumonia.

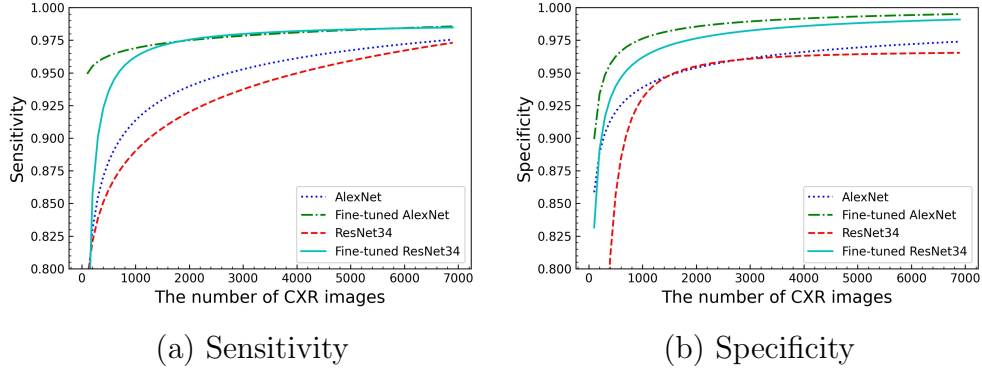


Figure 6.11: Relation between AlexNet and ResNet34 for sensitivity and specificity.

## 6.6. Discussion

One of the features of AI/ML-based medical devices resides in their ability to learn from real-world data. However, obtaining a large number of training data in the early phase is difficult, and the device performance may change after their first market introduction. To introduce the safety and effectiveness of these devices into the market in a timely manner, an appropriate post-market performance change plan should be established at the timing of the premarket approval, and the real-world performance change must be monitored. In this work, we studied how performance changes when the number of training data is changed.

Figure 6.9 shows the relations between the AUC and the training sample size and between the increment of the AUC and that of the training sample size by using the nonlinear function obtained from Table 6.2. All AUCs were rapidly improved as the training data increased and reached an equilibrium state. The 95% confidence interval also decreased as the training data increased. Fang et al. [48] and Samala et al. [49] obtained similar trends for deep learning-based organ auto-segmentation for head-and-neck patients by using CT images and binary classification of malignant and benign masses in digital breast tomosynthesis. Whether the performance of the preapproval stage is in the process of a rapid change or in a steady state must be determined. If the performance is in the process of a rapid change, the manufacturer and regulatory authority should carefully monitor the

real-world performance change after the first market introduction.

Alternatively, the AUCs for fine-tuned CNNs were better than those for CNNs trained from scratch in all training datasets. This effect is particularly noticeable when the training data are small. Tajbakhsh et al. [50] considered four distinct medical imaging applications in three specialties (radiology, cardiology, and gastroenterology) involving classification, detection, and segmentation under three imaging modalities and compared the performance of deep CNNs trained from scratch and fine-tuned pre-trained CNNs using ImageNet. They concluded that deeply fine-tuned CNNs are useful for medical image analysis, performing equally as CNNs trained from scratch and even outperforming them when limited training data are available. Furthermore, they indicated that the performance gap between deeply fine-tuned CNNs and those trained from scratch widened when the size of training sets was reduced. Our result agrees with their findings, which validates the generality of the mentioned case.

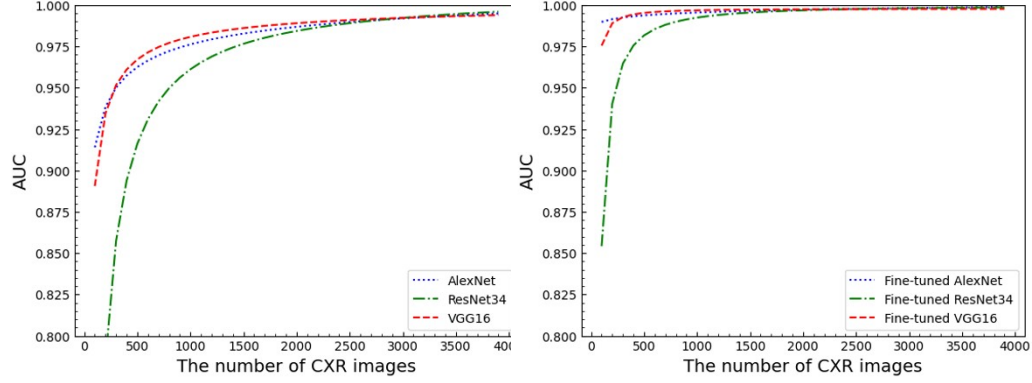
Most of the previous studies used publicly available datasets labeled as COVID-19, which are limited to tens to a few hundreds by using data augmentation, transfer learning or combining of datasets. Ran et al. [51] collected over 10000 CXR images labeled COVID-19 and non-COVID-19 pneumonia from five hospitals and more than 30 clinics by using NLP. In their study, a binary classification was performed by using a modified DenseNet model. The AUC versus the training sample size and the increment of the AUC versus that of the training sample size was confirmed. Their results demonstrated that more than 3000 training samples are needed to achieve an AUC better than 0.90. Moreover, after the training sample size goes beyond 3000, the performance gain with the training sample increase will diminish. Their test dataset was composed of 500 chest radiographs of 500 patients. The COVID-19 and non-COVID-19 pneumonia ratio had the same proportion. A similar trend was observed in our results, where the number of training data for which the gradient disappears was approximately 1600, 400, 2200, and 1100 for AlexNet, Fine-tuned AlexNet, ResNet34, and Fine-tuned ResNet34, respectively (corresponding AUCs: 0.98, 0.99, 0.98, and 0.99, respectively). To generalize our results, additional experiments by using Fine-tuned AlexNet and ResNet34 were conducted to compare the model performance of using the datasets labeled as COVID-19 and Normal with that of using the datasets

labeled as COVID-19 and non-COVID-19 pneumonia. Each dataset comprises 2500 CXR images that are randomly selected in the BrixIA and Chest X-ray 14 dataset. The dataset was split as 2000 images for training and validation, and 500 images for testing. Furthermore, training and validation data were selected as 15 different datasets. Figure 6.10 shows the relation with AUC for COVID-19 and Normal and COVID-19 and non-COVID-19 pneumonia datasets. Similar trends are observed for both results. The AUC in the equilibrium state of the “pneumonia” dataset was almost the same as that of the “Normal” dataset, indicating good performance. Considering Ibrahim et al. [52], who achieved high performance using automatic detection AlexNet to classify CXR images of COVID-19 and non-COVID-19 viral pneumonia and COVID-19 pneumonia and healthy subjects, our results of binary classification for COVID-19 and Normal can be generalized to classify COVID-19 and non-COVID-19 pneumonia. Additional experiments also indicate that the main reason for the higher performance than Ran et al.’s result is not the difference in the target diseases but the collection of their dataset from multiple medical institutes.

Figure 6.11 shows the relation of sensitivity, specificity, and training sample size using the nonlinear function obtained from Table 6.2. The sensitivity and the specificity of AlexNet outperformed ResNet34 with and without the fine-tuning method in the small number of CXR images. In addition, the difference in these performances tended to decrease as the number of training data increased. In other words, if the available data are limited, AlexNet is a more proper model to use compared to ResNet34. D’souza et al. [53] conducted structural analysis and optimization of convolutional neural networks with a small sample size because the number of samples in a dataset can be relatively limited in numerous real-world applications. They trained and tested each structure followed by layer dimension (layer width) optimization using small subsets of these datasets from entirely different data nature (calligraphic, photographic, and microscopic). Their result suggests that “deeper the better” is not always true for CNNs for small datasets, also clearly shows that as the depth increase there is an initial drop in the classification error, but the error soon rises sharply (calligraphic and microscopic). However, this may not always be true, as the microscopic dataset has no clear bias toward deeper or shallower networks. They concluded optimally perform-

ing network is largely determined by the data nature. Our result, which 8-layer AlexNet outperformed the 34-layer ResNet34 with the small training dataset, is the same trend of their result especially for calligraphic (The number of training data is 100, 500, and 1000). In the medical fields, comprehensive research is limited in the literature that uses only small datasets without data augmentation, particularly on the relationship between layer and performance. Therefore, further consideration how the number of the layers affects the performance with small number of training data considering target diseases will be needed as a future work. Sensitivity and specificity required by AI/ML-based medical devices vary depending on their intended use. In general practice, high sensitivity is required if the intended use is screening diagnosis. High specificity is required if it is definitive diagnosis. In the real word, the number of available labeled data is limited. Therefore, the appropriate model and method must be selected, and an appropriate post-market performance change plan must be established by considering the intended use and the available real-world labeled data.

To generalize the results, the experiment on performance changes with respect to the number of training images using VGG16 was conducted. VGG16 is widely used in various fields because of its simple structure. The basic concept of VGG 16 is as follows: 1) use only  $3\times 3$  (or partially  $1\times 1$ ) convolution, 2) reduce the feature map by half by max-pooling after stacking several convolutional layers with the same number of output channels, and 3) increase the number of output channels in the convolutional layer after max-pooling by a factor of 2. The training and validation data were selected from eight different datasets ( $N = 250, 500, 750, 1000, 1500, 2000, 2500, \text{ and } 3000$ ), and other experimental conditions were the same for the AlexNet and ResNet34. The results for AlexNet, VGG16, and ResNet34 are shown in Figure 6.12. Additional results support previous results and conclusions.



(a) CNN models without fine-tuning (b) CNN models with fine-tuning

Figure 6.12: Relation between AlexNet, VGG16 and ResNet34 for AUC.

Our study has some limitations. First, our dataset consists of only COVID-19 and Normal, and the ratio was limited to the same proportion. In the real-world data, the training data are expected to include CXR images with many pathologies not limited to COVID-19 and Normal alone. The ratio of COVID-19 and other pathologies, including Normal, may also vary. Therefore, as a future work, we will conduct a study on the performance change when adding many pathologies to the training data and change the ratio in the training data. Second, our study focused on binary classification to classify COVID-19 and Normal. To achieve a more effective classification, it is desirable to validate multi-class classifications, such as COVID-19, other viral, bacteria, and Normal by using CXR images. Although Cohen et al. [54] [55] made CXR images labeled as detailed Viral, Bacterial, Fungal, etc. available to the public, the number of images is very limited. These data are continuously being collected from public sources and through indirect collection from hospitals and physicians. Therefore, it is hoped that many of these data will become available in the future.

## 6.7. Conclusion

Appropriate performance changes must be predicted to manage the performance of medical devices using AI/ML. In this study, we performed binary classification to classify COVID-19 and Normal by using large datasets. In addition,

we observed the performance changes (i.e., AUC, sensitivity, and specificity) with the change in the number of training data. In the medical field, comprehensive research is limited in the literature that uses large datasets, particularly on the relationship between performance and training data because building large datasets is costly and burdensome for professionals, and there are concerns about ethical and privacy issues. This paper serves as a fundamental insight for regulators, policy makers, researchers, and manufacturers on how to develop appropriate post-market performance change plans.

# Chapter 7

## Performance change with the ratio of training data

### 7.1. Abstract

One of the features of artificial intelligence/machine learning-based medical devices resides in their ability to learn from real-world data. The performance may change after the market introduction. There are many aspects that contribute to the performance change relative to the real-world training data, such as the number and disease ratio. In actual clinical practice, the ratio of obtained training data varies from country to country, from region to region within each country, and from one hospital to another. Therefore, establishing a pre-change control plan at a premarket stage is essential to achieve safety and effectiveness through total product life cycles. In our previous work, we evaluated the performance change on the binary classification of coronavirus disease 2019 (COVID-19) and normal with the number of training data using two publicly large available chest X-ray (CXR) images. However, these results were obtained with the same ratio in the training data. Thus, this study aims to evaluate the performance change with a non-uniform ratio of COVID-19 CXR images based on the results of previous studies. We used the AlexNet and ResNet34 with and without the fine-tuning method as convolutional neural network (CNN) models. A total of 500, 1000, and 2000 CXR images were selected as training and validation datasets. These datasets represent states in which the performance change improves rapidly and



those in which an equilibrium state is reached. Each dataset was divided into seven datasets, and the area under the curve was employed to evaluate the performance change for each dataset through independent 1000 test datasets with the same ratio. Our result shows that all performances indicate that there is an upward convex relationship to the ratio of COVID-19 CXR images, and the vertex is where the ratio is the same. This trend was remarkable for the rapidly improving state and the CNNs without a fine-tuning method. Moreover, the visual explanations technique called Grad-CAM for interpreting classification results of CNN models support these results.

## 7.2. Introduction

There has been broad interest in the application of artificial intelligence (AI) and machine learning (ML) to the medical field. This is primarily driven by the impressive progress made by deep learning as a subset of ML because of the increased computational power and the availability of large datasets generated during all phases of the healthcare process. The number of research papers, especially for medical image analysis, has drastically increased [56, 57]. Furthermore, AI/ML-based medical devices have been approved in various regions or countries and introduced into the healthcare field [22]. Regulatory authorities are making efforts to establish and maintain their regulation tailored to the characteristics of AI/ML-based medical devices. Considering the performance change by continuous learning and iteration after market introduction, it may be effective ways to establish a pre-change control plan (PCCP) at a premarket stage, where manufacturers are anticipated to perform modifications to performance or intended use before marketing [2]. Since the outbreak of the Coronavirus Disease 2019 (COVID-19), many studies have used convolutional neural networks (CNNs), which is a class of artificial neural networks, to detect COVID-19 on chest X-ray (CXR) images. The COVID-19 has been diagnosed by using viral RNA via reverse transcriptase-polymerase chain reaction (RT-PCR) test result, but chest imaging plays an essential role in the early diagnosis and treatment of patients with suspected or probable chest infection caused by COVID-19 [24, 25]. Recently, large databases for chest imaging, including COVID-19 objects, have

been actively developed and made available to the public, and more research from various perspectives will be conducted. Given that the positive rate of COVID-19 varies from country to country and from region to region within each country [58], it is essential to evaluate performance change with the training data ratio, namely the number of COVID-19 and normal in the training data is unbalanced. For example, Reshi, A et al. [59] performed a binary classification to classify COVID-19 and other CXR images using CNNs with two imbalanced training datasets. The CNNs architecture was original with 38 layers, including six convolutional layers. Their result showed the accuracy without the pre-training improved 15% when the ratio of training COVID-19 CXR images was changes from about 75% to 50%. They finally concluded that the data augmentation can be an effective means of addressing these biases. Meanwhile, Shin, H. C. [60] obtained the similar trend to detect thoraco-abdominal lymph node and interstitial lung disease classification by using CT images. They reported that changing the training datasets from “biased” to “equal” slightly improved classification accuracy by 0.001% and 0.03%, respectively. The GoogLeNet [61] with fine-tuning method pre-trained ImageNet was used as CNNs, and test datasets with the balanced ratio was used to assess of its performance fairly. In our previous work [62], we evaluated the performance change on the binary classification of COVID-19 CXR and normal with the number of training data using two publicly large available datasets. The AlexNet [35] and ResNet34 [63] with and without the fine-tuning method were used as CNN models. We found that all performances were rapidly improved as the training data was increased and reached an equilibrium state. However, these results were obtained with a uniform ratio, namely the number of COVID-19 and normal in the training data is same. In these backgrounds, we evaluated the performance change with the ratio of training COVID-19 CXR images, especially for the rapidly improving state and the equilibrium state in our previous study. These datasets were divided into seven datasets with different ratios. We used the area under the curve (AUC) to evaluate the performance change for each dataset through independent test datasets with the same ratio.

## 7.3. Theoretical background

### 7.3.1 CNNs and fine-tuning

CNNs is a type of feed-forward neural network, where information is fed from the input layer to the output layer in one direction for processing data with a grid pattern, such as images, and it is designed to learn spatial information. CNNs mainly comprises three types of layers: convolution, pooling, and fully connected layers (Figure 7.1). The convolutional layer extracts feature from the input images, and the pooling layer is added after the convolutional layer to enhance robustness against slight changes. The fully connected layer maps the extracted features through a flattened layer, which converts the data into a one-dimensional array into the final output. In this paper, we used AlexNet and ResNet34 models with and without a fine-tuning method. The AlexNet consists of eight layers, including five convolutional layers with 60 million parameters, and the ResNet34 has 34 layers with 21.8 million parameters, including 33 convolutional layers. The ResNet overcomes the degradation problem by introducing residual connections mapping to fit input from a previous layer to the next layer and achieves compelling performance. The fine-tuning method, which is the process of fine-tuning the parameters of ML models for a new task by re-training part or all of the pre-trained ML models. Specifically, each model is pre-trained on 1.4 million natural images with 1000 classes called ImageNet [26], and re-training the unfreezing of the entire model on CXR images.

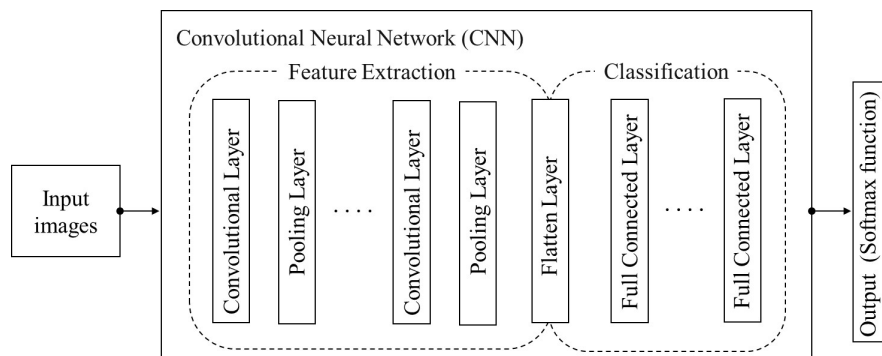


Figure 7.1: Overview of a fundamental CNNs structure

Several attempts have been made to explore a visual explanation to make CNNs more transparent and explainable. Selvaraju et al. [64] introduced a gradient-weighted class activation mapping (Grad-CAM) technique, which uses the gradient information flowing into the last convolutional layer of the CNNs, to produce a localization map highlighting the important regions in the image for predicting the class. The convolution layers retain spatial information, and as the layers get deeper, more complex information is extracted from simple shapes. Therefore, the last convolutional layers are used to compromise between high-level semantics and detailed spatial information.

## 7.4. Methodology

Two independent datasets were used for the evaluation. The first is the BrixLA dataset [15], which comprises 4703 CXR images of COVID-19 objects. The second is a chest X-ray14 dataset [17], which comprises 112120 CXR images with 14 diseases and one normal label. We randomly selected 500, 1000, and 2000 CXR images for training and validation with the seven different ratios of COVID-19 CXR images (Table 7.1). A total of 500 and 1000 CXR images were in a rapidly improving state, and the 2000 CXR images were in the equilibrium state (Figure 7.2). The 1000 CXR images for test with the same ratio of COVID-19 and normal classes were selected independent of training and validation dataset. The reprocessing of CXR images and the hyperparameters are identical to our previous studies [62]. The grayscale images were converted to three-channel color images, and pixel values were normalized from 0 to 1. Additionally, the hyperparameters are shown in Table 7.2. The AUCs derived from the receiver operating characteristic (ROC) through training on each of the seven datasets. The 95% confidence interval was determined using the standard error for the AUC [65].

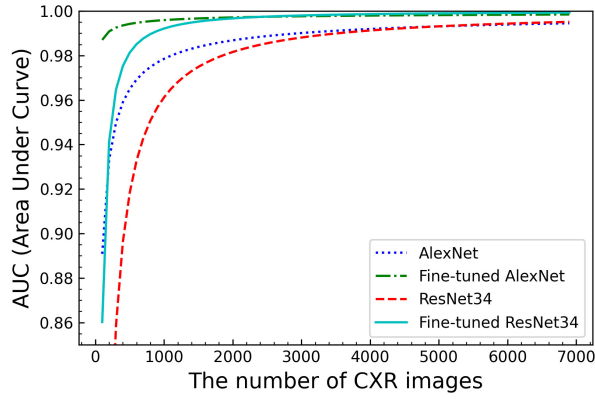


Figure 7.2: Relationships between the AUC and the number of trained CXR images with the same ratio adapted from Imagawa et al [11]

Table 7.1: The training and validation dataset.

The number of CXR images	The ratio of Covid-19 CXR images (%)						
	No.1	No.2	No.3	No.4	No.5	No.6	No.7
500	5	10	25	50	75	90	95
1000	5	12.5	25	50	75	87.5	95
2000	5	12.5	25	50	75	87.5	95

Table 7.2: Hyperparametes.

Hyperparameter	Value
Epochs	50
Batch size	32
Optimization method	Adam
Learning rate	$1 \times 10^{-5}$ (AlexNet) $1 \times 10^{-6}$ (ResNet34)
1st Momentum ( $\beta_1$ )	0.9
2nd Momentum ( $\beta_2$ )	0.999

## 7.5. Results

Figure 7.3 and Figure 7.4 show the relationships between the AUC and the number of training data with each ratio for all AlexNet and ResNet34 with and without a fine-tuning method, respectively. The major trends of our results are summarized as follows. 1) All AUCs indicate that there is an upward convex relationship to the ratio of COVID-19 CXR images, and the vertex is where the ratio is the same. 2) The larger the number of training data, the smaller the AUC change was. 3) The AUC change of the fine-tuned CNNs was smaller than CNNs trained from scratch. 4) The AUC change was extremely large, especially for ResNet34 without fine-tuning method with small training data ( $N = 500, 1000$ ). The lower limit of the y-axis for ResNet34 without fine-tuning method in Figure 7.4 is only different from the other figures.

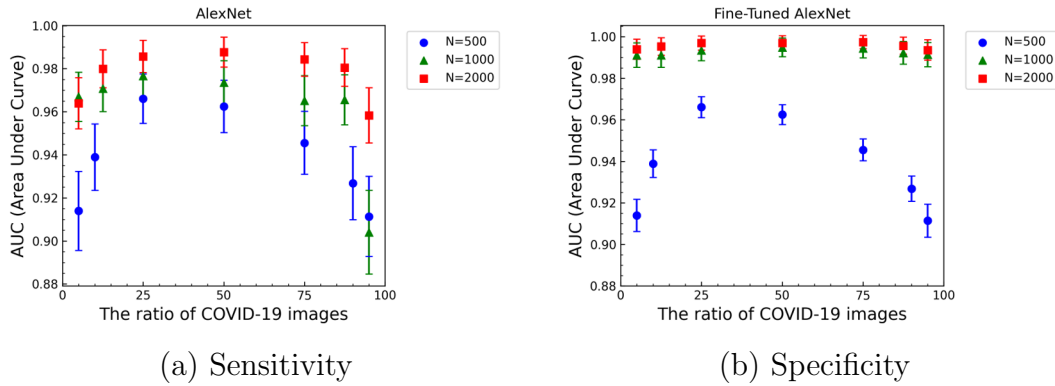


Figure 7.3: Relationships between the AUC and the ratio of COVID-19 CXR images. N is the number of training datasets, and each plot corresponds to the ratio defined in the Table 7.1. (AlexNet)

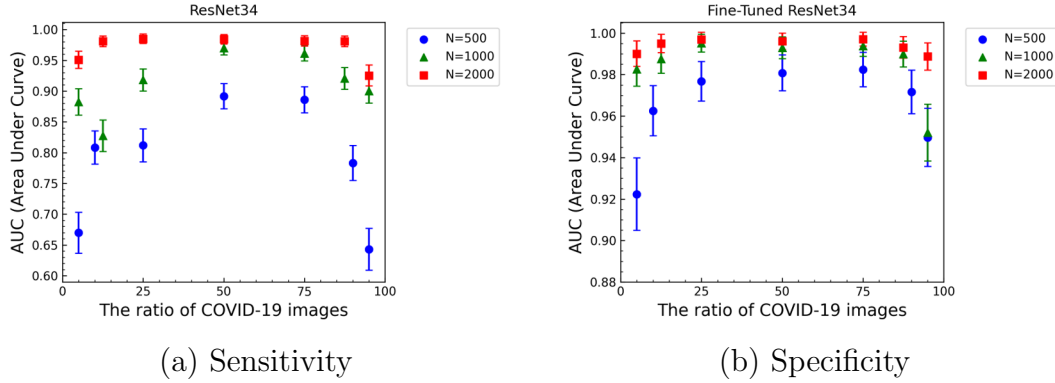


Figure 7.4: Relationships between the AUC and the ratio of COVID-19 CXR images. N is the number of training datasets, and each plot corresponds to the ratio defined in the Table 7.1. (ResNet34)

## 7.6. Discussion and conclusion

Our results are similar to the AUC change as the number of training data increase in Figure 7.2. To achieve stable AUCs change, more training data are needed, and AUCs of fine-tuned CNNs were better than those of CNNs trained from scratch. Furthermore, the results show that 34-layer ResNet34 has the feature of rapid performance change compared to shallow eight-layer AlexNet, especially for the small training data. The CNN models, number, type of images and performance metrics are different from our study, but related work in introduction [8-9] support and generalize our study that the performance is highest when there is no bias, i.e., the ratio of training dataset is same, furthermore, the performance change of the fine-tuned CNNs was smaller than CNNs trained from scratch.

To visualize our results, we apply Grad-CAM to CNN models trained on each dataset. Figure 7.5 shows an example of an activation heatmaps highlighting the important regions for the COVID-19 images (top) and normal images (bottom). The number of CXR images is 1000 with a ratio of 5%, 50%, and 95% of COVID-19 CXR images for AlexNet without a fine-tuning method, and Figure 7.6 shows those of ResNet34. These localization maps for CNN models trained with the same ratio highlight the lung region as class discriminating areas, whereas the other trained CNN models highlight other regions. The CXR features, such as COVID-

19 pneumonia, are present in the lung region. Therefore, this trend supports our result that AUC is highest when there is no bias in training datasets associated with the number of image types, i.e., the normal and COVID-19 CXR images is the same. Additionally, AUC becomes low when there is bias.

In this study, we performed binary classification to classify COVID-19 and normal images with varying ratios of COVID-19 CXR images. Our results show that the performance change depends on the number of training data, CNN model, and fine-tuning method. Given that the number and ratio of COVID-19 vary from one hospital to another, one area to other, our result provides a fundamental insight for COVID-19 detection. Our study has several limitations. First, our test datasets are fixed in the number and ratio of test datasets. In order to confirm more details, the relationship between training and test data should be considered. Second, the CXR images include only COVID-19 and normal. More pathologies, such as pneumonia caused by viral and bacterial pathogens other than COVID-19, should be considered to reflect real-world scenarios. Furthermore, the multi-class classification should also be evaluated. Thus, in our future work, we will study multi-class classification with many pathologies by adding to the relationship between training and test datasets as a fundamental insight for PCCP. Appropriate performance changes must be predicted to manage the performance of medical devices using AI/ML. In this study, we showed the performance change for classification to classify COVID-19 and normal images with varying ratios of training datasets based on the result of our previous study. Although more consideration is needed from different perspectives to reflect real-world scenarios, we hope our results provide to progress this field in the future.



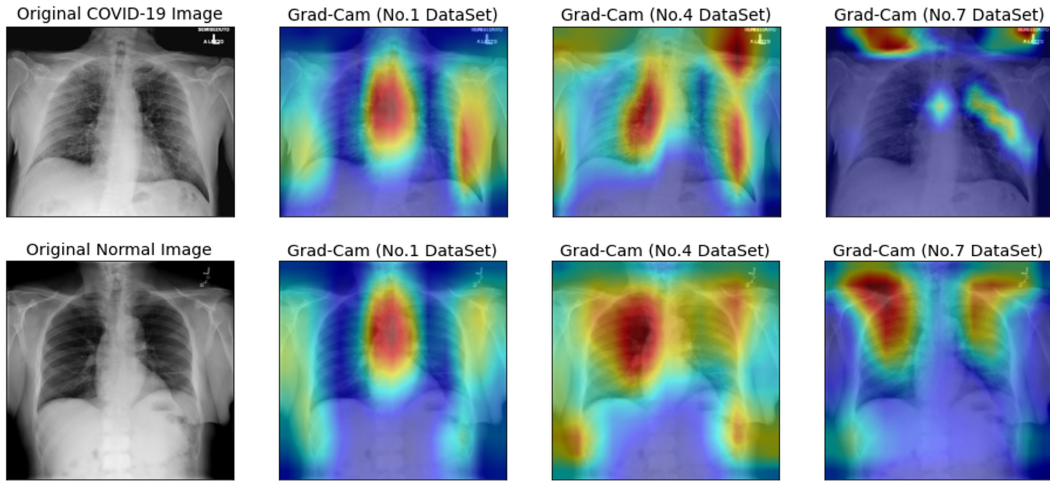


Figure 7.5: Grad-CAM heatmaps of feature importance for CNN model's predictions. This figure is an example of AlexNet without a fine-tuning method trained with 1000 CXR images for each No.1, No.4, and No.7 DataSet defined in the Table 7.1.

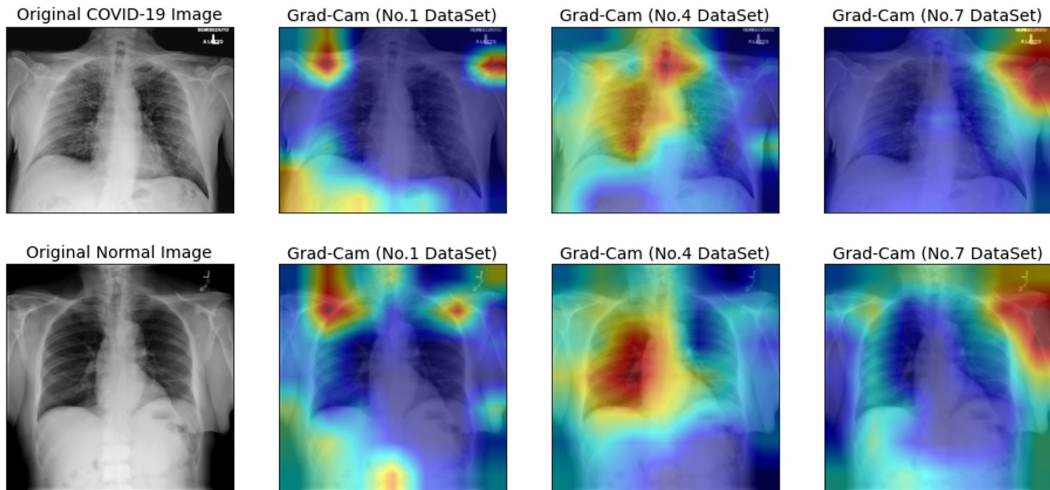


Figure 7.6: Grad-CAM heatmaps of feature importance for CNN model's predictions. This figure is an example of ResNet34 without a fine-tuning method trained with 1000 CXR images for each No.1, No.4, and No.7 DataSet defined in the Table 7.1.

## Chapter 8

# Evaluation of effectiveness of self-supervised learning to reduce annotated images

### 8.1. Abstract

A significant challenge in machine learning-based medical image analysis is the scarcity of medical images. Obtaining a large number of labeled medical images is difficult because annotating medical images is a time-consuming process that requires specialized knowledge. In addition, inappropriate annotation processes can increase model bias. Self-supervised learning (SSL) is a type of unsupervised learning method that extracts image representations. Thus, SSL can be an effective method to reduce the number of labeled images. In this study, we investigated the feasibility of reducing the number of labeled images in a limited set of unlabeled medical images. The unlabeled chest x-ray (CXR) images were pretrained using the SimCLR framework, and then the representations were fine-tuned as supervised learning for the target task. A total of 2,000 task-specific CXR images were used to perform binary classification of coronavirus disease 2019 (COVID-19) and normal cases. The results demonstrate that the performance of pretraining on task-specific unlabeled CXR images can be maintained when the number of labeled CXR images is reduced by approximately 40%. In addition, the performance was significantly better than that obtained without pretraining. In

contrast, a large number of pretrained unlabeled images are required to maintain performance regardless of task specificity among a small number of labeled CXR images. In summary, to reduce the number of labeled images using SimCLR, we must consider both the number of images and the task-specific characteristics of the target images.

## 8.2. Introduction

Machine learning-based medical image analysis has been researched actively and introduced into the medical environment. This use of such technologies is expected to increase in the future due to the success of deep learning, which is a subset of machine learning [20, 21]. However, a common challenge is the scarcity of medical images due to patient privacy concerns. In addition, the scarcity of labeled medical images is also a serious problem because annotating medical images requires specialized knowledge, and an inappropriate annotation process can introduce annotation bias [66, 67]. To overcome the lack of medical images, the most common approach for supervised learning is to pretrain a convolutional neural network (CNN) on a large number of natural images, e.g., ImageNet [13], and then fine-tune the network using labeled medical images for the target medical task. This approach has proven effective in terms of improving performance in some categories [27]. For example, since the outbreak of coronavirus disease 2019 (COVID-19), several X-ray and CT image datasets have been made available to the public, and many COVID-19 classification studies have demonstrated the effectiveness of this approach, especially for limited images. However, there are significant differences in the pretrained and fine-tuned parameters because ImageNet contains approximately 1.4 million natural color images with 22,000 categories and 1,000 labels. Medical applications that utilize CNN models pretrained on ImageNet remain ambiguous and can suffer from model overfitting. In addition, it has been reported that CNN models pretrained on ImageNet can perform worse than models without pretraining, depending on the characteristics of the data [41, 42, 68].

Self-supervised learning (SSL) has been proposed recently as an effective approach to the labeled image scarcity problem. The SSL training method produces

representations using unlabeled images, and it is a type of unsupervised learning. Generally, the pretrained representation is fine-tuned for a downstream task on a few labeled images. This semi-supervised strategy achieves good performance compared to supervised learning. SimCLR [12] is a simple framework for contrastive learning of visual representations, and its success has led to extensive research on contrastive methods. However, SimCLR requires a large batch size to achieve sufficient performance; thus, many revised contrastive learning frameworks, e.g., Momentum Contrast (Moco) [69] and Bootstrap Your Own Latent (BYOL) [70] have emerged to reduce the batch size. In the medical fields, there has been an increase in the number of studies demonstrating the effectiveness of SSL methods [71]. For example, Shih-Cheng et al. [72] reviewed literature published after 2012 for SSL on medical image classification. They demonstrated the potential of SSL to reduce the amount of labeled data and improve performance and transferability. However, these previous studies demonstrated the effectiveness of SSL by pretraining on a large number of unlabeled images.

In a related study, Shekoofeh et al. [73] demonstrated that pretraining on unlabeled ImageNet and chest x-ray (CXR) images with SimCLR outperformed a supervised method pretrained on ImageNet and improved transferability to dermatology images. In addition, Hari et al. [74] demonstrated that pretraining on CXR images with Moco can reduce the number of labeled images and outperform CXR images without a pretraining process. Guang et al. [75] performed pretraining on labeled ImageNet, and then performed pretraining on unlabeled CXR images six SSL methods (Cross, BYOL, SimSiam, SimCLR, PIRL-jigsaw, and PIRL-rotation), which was followed by transfer learning to the target task. They demonstrated the improvement in COVID-19 detection compared to supervised learning and pretraining with SSL methods. These studies also performed pretraining on a large number (in excess of several hundred thousand) of unlabeled CXR images or natural images using SSL.

In real world applications, it would be difficult to prepare such a large number of medical images, even unlabeled images because utilization of medical data must comply with the laws and regulations of each country. Thus, this study was conducted to confirm the effectiveness of SSL for pretraining on a limited number of unlabeled medical images to reduce the amount of labeled data and

improve performance. The methodology employed in this study is similar to that used in various previous studies, where the pretraining representation is fine-tuned for a binary classification. Specifically, we evaluate the number of CXR labeled images with the fine-tuning method (i.e., label fraction) for task-specific and non-task-specific images using supervised learning without pretraining, unsupervised pretraining, and supervised pretraining. Here, a total of 2,000 task-specific images and 90,000 non-task-specific images were used for classification of COVID-19. In this paper, the “task-specific” pretraining dataset means that pretraining dataset matches the test dataset primarily in terms of image type, e.g., medical images and natural images, medical images acquired using medical devices, and the target diseases. The results demonstrate that the performance obtained by pretraining on task-specific unlabeled CXR images can be maintained when the number of labeled CXR images is reduced by approximately 40%. In addition, the performance was significantly better than that obtained without pretraining. In contrast, a large number of pretrained unlabeled images are required to maintain performance regardless of task specificity among a small number of labeled CXR images. In summary, to reduce the number of labeled images and improve performance with SSL, we must consider both the number of images and also the task-specific characteristics of the images.

## 8.3. Material and Methods

### 8.3.1 Datasets

In this study, two publicly available CXR datasets were used, i.e., the National Institutes of Health (NIH) dataset [76] and the BrixLA dataset [43]. The NIH dataset contains 112,120 CXR images with 14 diseased and normal images from 30,805 unique patients, and the BrixLA dataset contains 4,703 CXR images from COVID-19 patients. Here, a total of 2,000 training and validation images for COVID-19 and normal cases were selected randomly at the same ratio because our previous study demonstrated that CNN models, e.g., AlexNet and ResNet, achieve consistent performance when supervised learning is performed on 2,000 training images.

The training and validation datasets were divided into labeled images (N=100, 250, 500, 1,000, 1,500, and 2,000) and unlabeled images (N=1,900, 1,750, 1,500, 1,000, 500, and 0) from the NIH and BrixLA datasets. Note that there was no duplication between labeled and unlabeled images. In addition, the 90,000 unlabeled CXR images in the NIH dataset and the 90,000 unlabeled natural images in the ImageNet dataset were selected randomly for training and validation. A test dataset of 1,000 CXR images, independent of the training datasets, was used for binary classification of COVID-19. Here, all CXR images were resized to  $256 \times 256$  pixels and cropped around the center to  $224 \times 224$  pixels. The grayscale CXR images were converted from 16-bit to 8-bit and converted to three color channels (red, green, and blue). The pixel values of the input images were normalized between ranges 0 and 1.

### 8.3.2 Methodology

In our experiments, we employed SimCLR [12] as the SSL method to learn the visual representations as a pretraining process. Figure 8.1 shows the method used in our experiments. SimCLR is a simple framework that does not require a special architecture or memory banks. Here, image  $x$  is differentially augmented  $\tilde{x}_i$  and  $\tilde{x}_j$  as positive pairs. In addition,  $f(\cdot)$  is an encoder network to generate representations, where  $h_i = f(\tilde{x}_i) = \text{ResNet}(\tilde{x}_i)$ , and  $g(\cdot)$  is a neural network projection head with one hidden layer used for contrastive loss, where  $z_i = g(h_i) = W^{(2)}\sigma(W^{(1)}h_i)$ .  $N$  is an arbitrary number of batches with  $2N$  positive pairs in each batch and  $2(N - 1)$  negative pairs. SimCLR maximizes the agreement of the positive pairs and minimizes the negative pairs using the contrasting loss as follows:

$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(z_i, z_k)/\tau)} \quad (8.1)$$

where  $\tau$  is a temperature parameter, and  $\mathbb{1}$  means if  $k = i$  then 0 and  $k \neq i$  then 1. In addition,  $\text{sim}(\cdot)$  denotes the cosine similarity, and  $N$  is the batch size.

We performed some experiments to select the type of data augmentation, the backbone network, and the hyperparameters such as the number of batches and the number of training epochs, because the original paper [12] reported that these

have a significant impact on performance. For data augmentation technique, two types of transformations were used in this evaluation, i.e., spatial transformations, e.g., cropping, rotation, vertical flipping, and horizontal flipping, and appearance transformations, e.g., Gaussian blur and grayscale. In addition, different numbers of layers ( $N = 18, 34, 50$  and  $101$ ) in the ResNet backbone were also evaluated, and we investigated different numbers of batches ( $N = 32, 64, 128$  and  $256$ ) and training epochs ( $N = 100, 300$  and  $500$ ). Note that other SimCLR hyperparameters were unchanged, and the hyperparameters for supervised learning were based on our previous results [62]. The area under the curve (AUC) was used for evaluation. The sensitivity and specificity depend on the classification threshold, and accuracy is dependent on the proportion of the test dataset. The 95% confidence interval was constructed using the method proposed by Hanley and McNeil method [65]. To evaluate the effectiveness of the SSL, the AUC on the common test dataset was determined as a function of pretraining on unlabeled images and the number of labeled CXR images for the subsequent supervised learning process. The types and number of images used for each training and test are shown in Table 8.1.

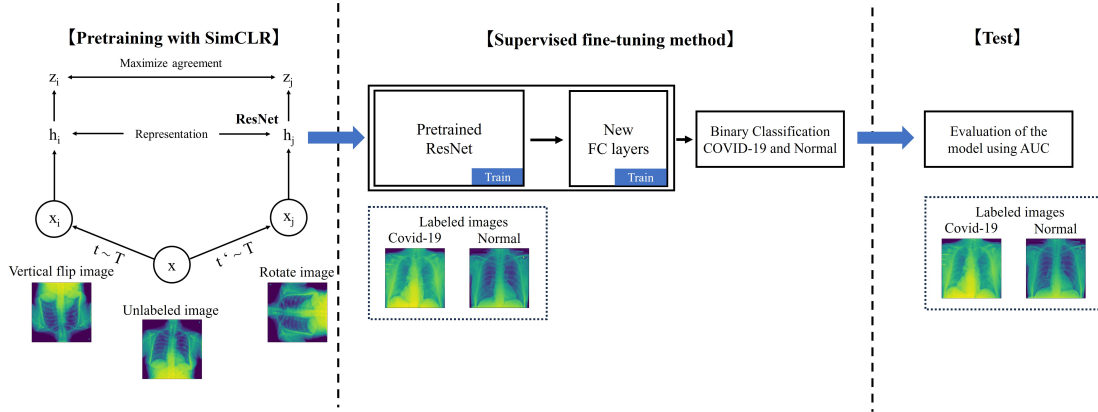


Figure 8.1: Proposed pretraining process with SimCLR and fine-tuning method for binary classification of COVID-19

Table 8.1: Types and number of images used for training and test.

Number of labeled CXR images with supervised fine-tuning method (NIH and BrixIA datasets)			100	250	500	1,000	1,500	2,000
Method	Image type	Dataset	Number of pretraining images					
No pretraining	-	-	0	0	0	0	0	0
Unsupervised pretraining (SimCLR)	CXR images <sup>1</sup>	NIH and BrixIA	1,900	1,750	1,500	1,000	500	0
	CXR images <sup>2</sup>	NIH	90,000	90,000	90,000	90,000	90,000	0
	Natural images	ImageNet	90,000	90,000	90,000	90,000	90,000	0
Supervised pretraining (ResNet34)	Natural images	ImageNet	1,400,000	1,400,000	1,400,000	1,400,000	1,400,000	0
Test	CXR images	NIH and BrixIA	1,000	1,000	1,000	1,000	1,000	1,000

<sup>1</sup> The images are identical to the test images in terms of the dataset and the target disease.

<sup>2</sup> The images are not included target disease.

## 8.4. Results

Figure 8.2 shows the performance obtained using these enhancements when applied in various combinations. We found that the combination of rotation and vertical flipping obtained the highest performance. Figures 8.3 and 8.4 show the effect of batch size and the number of training epochs on performance. As can be seen, the best performance was obtained with a batch size of 128. In addition, higher numbers of training epochs tended to improve performance; however, but 300 epochs were used due to computational resources. We also evaluated the effect of different numbers of ResNet layers. ResNet34 was the most effective deep layer compared to other layers in Figure 8.5. Note that the fundamental model and hyperparameter configurations were the same for all experiments, and the detailed experimental conditions are shown in Figure 8.2, Figure 8.3, Figure 8.4 and Figure 8.5.

The pretrained representations with different types and numbers of images were subsequently fine-tuned on different numbers of labeled CXR images. Figure 8.6 shows AUC versus the number of CXR labeled images with the fine-tuning method for each unsupervised method pretrained on unlabeled images. Note that Figure 8.6 includes several task-specific CXR images extracted from the NIH and BrixIA datasets (red), 90,000 CXR images extracted from only the NIH dataset (blue) and 90,000 natural images extracted from ImageNet (green). The supervised Supervised learning without pretraining was also included as a baseline method (black). We found that the AUCs for pretraining on task-specific CXR



images can be maintained when the number of labeled CXR images is reduced from 2,000 to 500 (i.e., a reduction of approximately 40%). In addition, performance was significantly better than the supervised baseline in this area. In contrast, the effect of SimCLR was reduced drastically when the number of labeled images was reduced to less than 500. When pretraining was performed on 90,000 non-task-specific CXR images and 90,000 natural images, we observed slightly better performance compared to the supervised baseline. In contrast, the improvement in terms of the effectiveness of SimCLR can be observed in the small labeled image region.

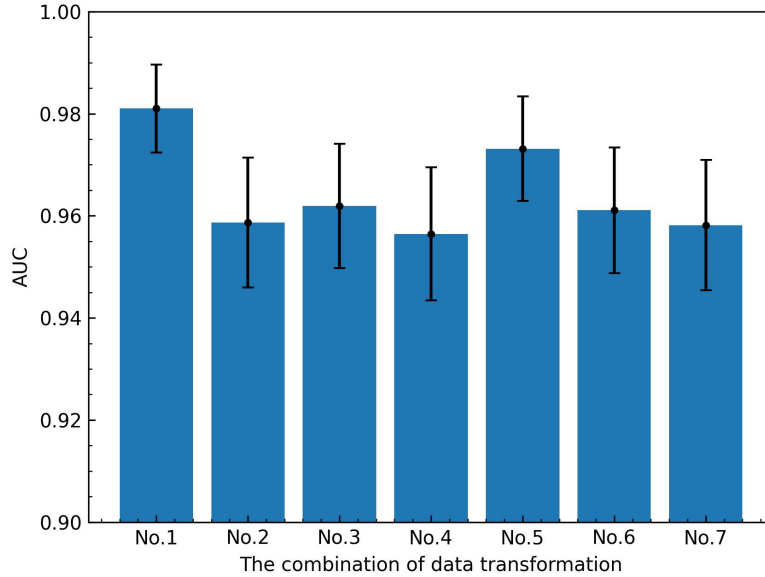


Figure 8.2: AUC versus the combination of transformations. No.1: rotate and vertical; No.2: rotate and horizontal flip; No.3: rotate and crop; No.4: rotate and Gaussian blur; No.5: rotate and grayscale; No.6: crop and vertical flip; No.7: Gaussian blur and grayscale. SimCLR (500 images) and ResNet34 (1,500 images) with batch sizes of 128 and 300 epochs were used in this evaluation.

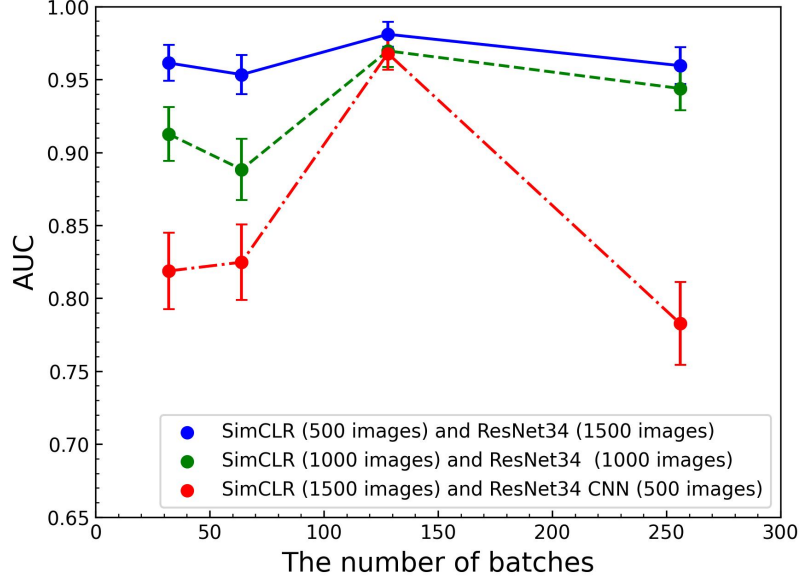


Figure 8.3: AUC versus the number of batches ( $N = 32, 64, 128$  and  $256$ ). Here, a combination of rotation and vertical flip was used over 300 epochs.

## 8.5. Discussion

The implementation of SimCLR on ImageNet requires large batch sizes; thus, this approach consumes significant computational resources. However, our results demonstrate that SimCLR specific to the CXR images did not require a large batch size. We assume that the CXR images have common anatomical structures across the images; thus, so there is no need to create many negative pairs in each batch to reduce the loss function. The results of data augmentation and the backbone network are also specific to the CXR images. As a backbone, we found that ResNet does not require deeper layers, and this trend is similar to the supervised learning results presented by D’souza et al. [77]. They suggested that ”deeper is better” is not always true, especially for small amounts of data, and the optimal CNN network depends on the nature of the training data. This suggests that feature extraction with SimCLR does not necessarily require a deep model because CXR images are simpler than ImageNet and the number of images is very small. In addition, from a computational resource perspective, our results also

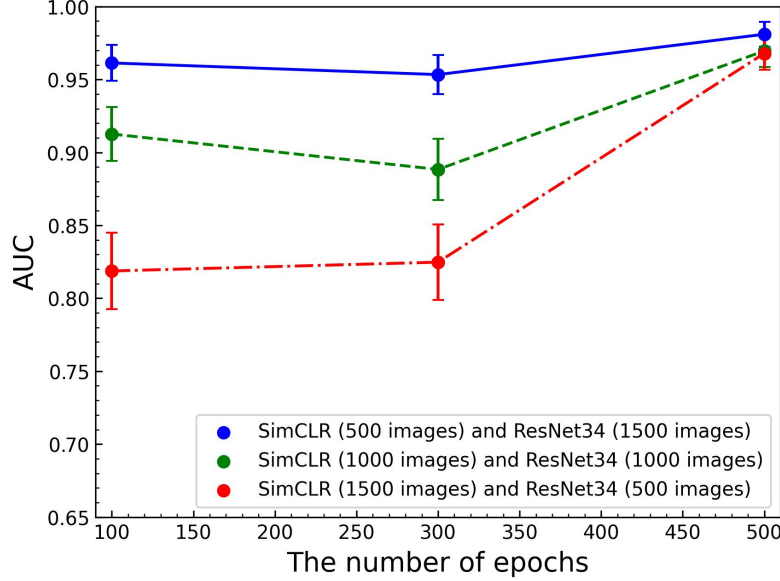


Figure 8.4: AUC versus the number of epochs ( $N = 100, 300$  and  $500$ ). A combination of rotation and vertical flip was used with a batch size of 128.

demonstrate that it is important to use SimCLR with task-specific images rather than ImageNet.

In this study, SimCLR was used specifically for CXR images, and we found that the AUCs of the pretrained task-specific CXR images can be maintained when the number of labeled CXR images is reduced by approximately 40%. Kyungjin et al. [78] also investigated the effectiveness of an SSL method and demonstrated the performance for some CXR datasets, e.g., the CheXpert datasets [79] with fine-tuning method for multiclassification used by pretraining on non-task-specific 4.8 million unlabeled CXR images with Moco. Table 8.2 compares the comparison these results in terms of the reduction of fine-tuned labeled images. Although performing direct quantitative comparisons is difficult due to the differences in the type of SSL method, the number of classifications, and the number of labeled images, the results demonstrate a similar trend. In other words, AUCs can be maintained when the number of labeled CXR images is reduced by approximately half. Considering that the number of unlabeled pretraining images used in the

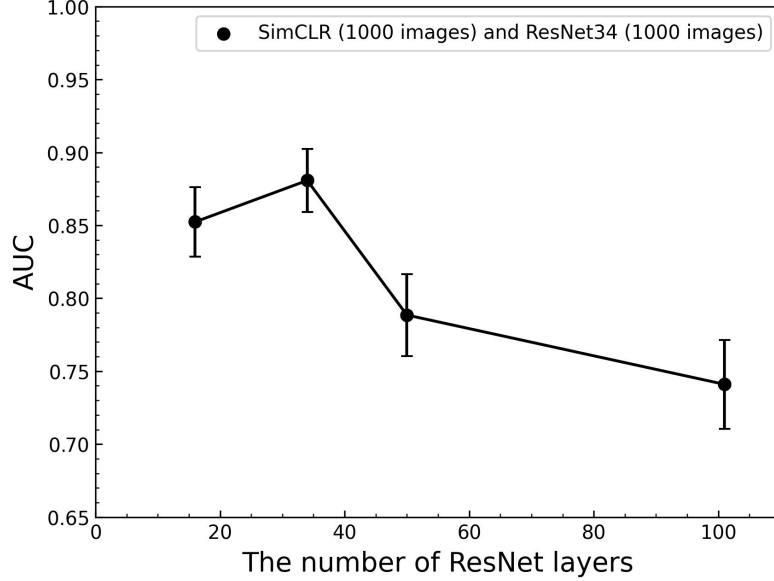


Figure 8.5: AUC versus the number of ResNet layers ( $N = 18, 34, 50$  and  $101$ ). A combination of rotation and vertical flip was used over 300 epochs with a batch size of 128.

current study was extremely small (ranging from hundreds to thousands), the acquisition of task-specific images is an important factor in terms of data efficiency.

In contrast, the AUC performance obtained when pretraining with task-specific images demonstrated a significant reduction compared to the performance obtained with non-task-specific images among a small number of labeled CXR images. Figure 8.7 shows the number of CXR labeled images with the fine-tuning method for both unsupervised and supervised methods pretrained on unlabeled and labeled images. This figure includes task-specific unlabeled CXR images extracted from the NIH and BrixIA datasets (red), and the labeled natural images extracted from ImageNet (purple) as pretraining. Note that the supervised learning without pretraining is also included as a baseline method (black). As shown, pretraining on 1.4 million labeled ImageNet clearly outperforms in small labeled image regions (less than 500 images). A related study by Sellergren, etc. [80] achieved an AUC comparable to state-of-the-art deep supervised learning models

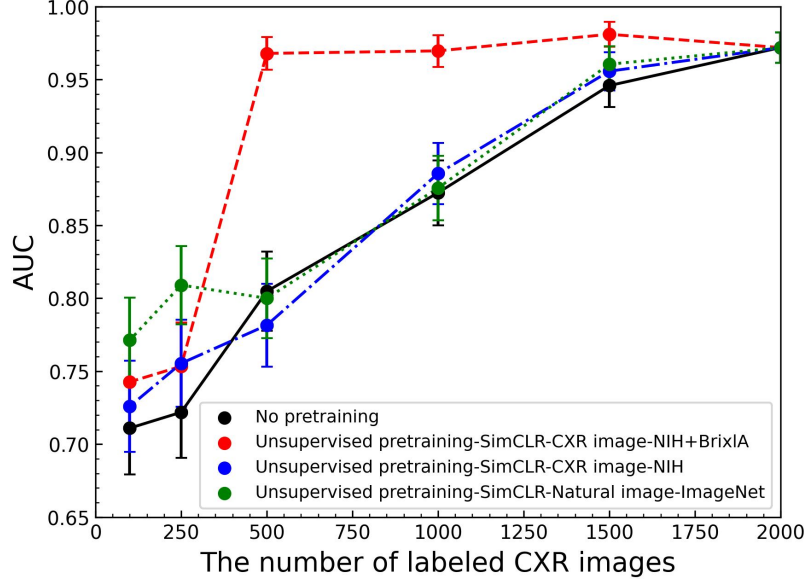


Figure 8.6: AUC versus the number of labeled CXR images with the supervised fine-tuning method used by supervised learning without pretraining and unsupervised pretraining process. The numbers of images for each training and test are shown in the Table 8.1

on tens to hundreds of labeled images by performing pretraining on 821,544 unlabeled CXR images using the SSL method. This suggests that a large number of images is required to maintain sufficient performance regardless of the pretraining method and task specificity in small labeled image regions.

In this study, we found that the performance obtained by pretraining on the task-specific unlabeled images with SimCLR can reduce the number of labeled images and outperform the process that does not employ pretraining. However, this result is limited in that it does not apply to a small number of labeled images. Recently, many studies have reported the effectiveness of SSL; however, to the best of our knowledge, few studies have investigated that use of a small number of labeled and unlabeled images that reflect the real world. Thus, the results of the current study provide fundamental insight into the effectiveness of SSL in the medical field. Note that our study is limited in terms of the generalizability

of our findings. First, the experiment conducted in this study only investigated SimCLR; thus, other improved and refined SSL frameworks should be considered in the future. In addition, more detailed hyperparameters should be considered. Second, a wider variety of diseases should be considered to reflect actual clinical practices. For COVID-19 detection, similar diseases, e.g., viral and bacterial pneumonia, should be investigated. In addition, other medical images should also be considered.

Table 8.2: Fraction of labeled images

Label Fraction (%)	1	5	10	12.5	50	100
Ours	N/A	0.742	N/A	0.753	0.970	0.972
Guang et al. [78]	0.638	N/A	0.746	N/A	0.790	0.807

## 8.6. Conclusion

To the best of our knowledge, this paper represents the first report demonstrating the effectiveness of the SSL method with pretraining on a small number of unlabeled images. We hope that the results of this study will contribute to the ongoing development of machine learning-based medical image analysis.

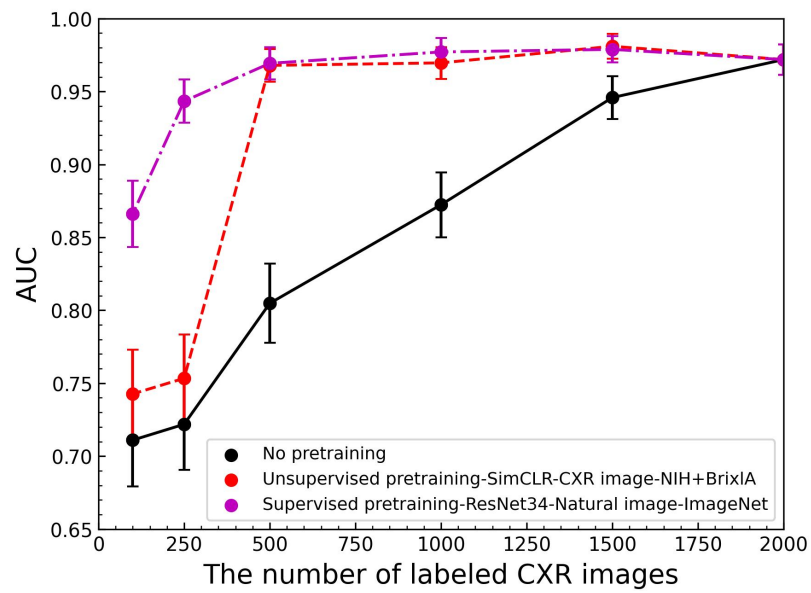


Figure 8.7: AUC versus the number of labeled CXR images with the supervised fine-tuning method used by supervised learning without pretraining, unsupervised pretraining and supervised pretraining. The numbers of images used for each training and test are shown in Table 8.1

# Chapter 9

## Evaluation of effectiveness of pre-training method

### 9.1. Introduction

There is extensive interest in deep learning (DL) as a subset of machine learning (ML) applications in many fields because of the increased computational power and explosion in the availability of large-scale data. The current boom is called the third artificial intelligence (AI) boom, and it differs from previous booms because many technologies are being implemented in society [81]. Deep convolutional networks (CNNs), which enable the learning of data-driven complex features without handcrafted feature extraction, have become dominant in the field of computer vision. This is because of [82] and the successful training of up to hundreds or thousands of layers in ILSVRC 2015 [63]. Medical image analysis is no exception and the number of studies on CNN applications has increased since 2015 [57]. Furthermore, the number of approved AI/ML-enabled medical devices is increasing. Computer-aided diagnostic medical devices using CNNs for medical images in radiology have been introduced in many medical practices in the USA and Europe through FDA approval and CE marks [22].

Transformers, which primarily use the self-attention mechanism [83], has recently shown great potential as a new type of neural network. Transformers have shown high performance in the field of natural language processing with various models, such as BERT [84] and GPT [85]. Although CNN-based models dominate



the computer vision field, the transformer-based Vision Transformer (ViT), which uses self-attention instead of convolution, has achieved state-of-the-art results in various image-classification tasks [11]. The medical imaging community has also seen an exponential growth in the number of transformer-based technologies, and the number of such studies has consistently increased since 2021 [86]. ViT can achieve excellent performance compared to state-of-the-art CNN-based models, but requires large, annotated imaging datasets because of the lack of inductive biases [11]. In the medical field, obtaining a large number of medical images is difficult because annotation by medical professionals is time-consuming, and patient privacy must be considered. Therefore, it is important to consider how to build an effective ViT model by using limited medical images. One way to overcome this data scarcity problem is transfer learning (TL), in which neural networks are pre-trained and then the parameters are utilized for a target task. This method can be categorized into two types. One is the feature extraction method, which freezes the pre-trained network and replaces the fully connected (FC) layers with other machine learning classifiers, such as random forests, support vector machines, and new FC layers. The other method is the fine-tuning (FT) method, which replaces the FC layers with new FC layers and retrain all or part of the pre-trained model. Since the emergence of TL for medical image classification in 2016, the number of publications has grown rapidly over consecutive years.

Chest radiography (CXR) is the most common medical imaging modality and is widely used in various clinical practices. Large datasets of more than 100k labeled images are publicly available such as NIH dataset [76], CheXpert dataset [79] and MIMIC-CXR dataset [87]. These abundant CXR images contribute to the development of deep learning models. Since the outbreak of Coronavirus Disease 2019 (COVID-19), some datasets labeled COVID-19 have been introduced to the public, and many studies have been conducted to develop models to diagnose COVID-19 instead of viral RNA identification using reverse transcriptase polymerase chain reaction (RT-PCR). Applying ML methods to COVID-19 radiological imaging may improve diagnostic accuracy compared with the gold-standard RT-PCR, while providing valuable insight into the prognostication of patient outcomes. However, many studies have been conducted using a small number of training images or a combination of training images. Therefore, these results

are optimistic, and there are concerns regarding risk bias such as model overfitting [41, 42]. Recently, several large datasets with detailed demographic statistics (e.g., patient age and sex, manufacturer, and view position) [16, 43] have become publicly available, and development studies with a low risk of bias are expected before introducing the clinical environment.

With the mentioned above, the purpose of this study is to investigate an effective pre-training method for ViT. Specifically, an evaluation of the binary classification of COVID-19 and normal by using CXR images was conducted as a first step. The major contributions of this study are as follows: 1) pre-training with natural images outperformed that with CXR images using the fine-tuning method; 2) pre-training with natural images captured more global features, especially the shape of lung field regions; and 3) the trends shown in 1) and 2) became clearer as the number of pre-training natural images increased.

## 9.2. Related work

Several studies have evaluated the ViT and CNNs using the TL method. There have been a number of studies on the evaluation of ViT and CNNs using the TL method. For example, Kim et al. [88] conducted a literature review on TL for medical image classification and demonstrated the efficacy of transfer learning, particularly for deep CNN models (e.g., ResNet and Inception). It has also been reported that the FT method is more frequently applied in radiology research [89]. As specific research results, Imagawa et al. [62] evaluated the binary classification of COVID-19 CXR images using the CNNs and concluded that the CNNs in the FT method from natural images such as ImageNet improved the performance, especially for the small amounts of training data. Usman et al. [90] evaluated the multiclass classification of CXR images with 14 observations using pre-trained CNNs and ViT models. They concluded that the ViT in the FT method from natural images improved the results compared with those of CNNs. Christos et al. [91] conducted classification using diabetic retinopathy, dermoscopic, and mammography images, and then evaluated the CNNs and ViT models from scratch, FT from natural images, and the FT method plus self-supervised learning from medical imaging tasks. They concluded that ViT was worse than CNNs with

training from scratch; however, the FT method bridged the performance gap between CNNs and ViT, where the performances were similar. Thus, several results of these studies have been reported, but there are no papers that broadly summarize TL, such as the number of pre-training images and differences in pre-training image characteristics between natural and medical images.

## 9.3. Methodology

### 9.3.1 Vision Transformer

ViT is a transformer-based model, and its architecture is illustrated in Figure 9.1. The ViT consists of three main parts: the Input Layer, Encoder, and MLP Head. The standard transformer captures the token embedding as a 1D sequence, so the 2D input image  $X \in \mathbb{R}^{H \times W \times C}$  is reshaped as a flattened 2D patch sequence  $X_p \in \mathbb{R}^{N_p \times (P^2 \cdot C)}$ , where  $(H, W)$  is the input image resolution,  $C$  is the number of channels in the input image, and  $(P, P)$  is the patch image resolution, and  $N$  is the number of each patch image. These flattened patches are mapped to  $D$  dimensions in a trainable linear projection layer, and the  $i$ -th output of this projection  $X_p^i \in \mathbb{R}^{(P^2 \cdot C)}$  is represented as a patch embedding. For the classification task, the class token is prepended to the sequence of patch embeddings, and a positional embedding is appended to the class token and patch embedding to obtain positional information. Finally, the input  $z$  to the transformer encoder can be written as

$$z_0 = [x_{class}; x_p^1 E; x_p^2 E; \dots; x_p^{N_p} E] + E_{pos}, \quad E \in \mathbb{R}^{(P^2 \cdot C) \times D}, E_{pos} \in \mathbb{R}^{(N_p + 1) \times D} \quad (9.1)$$

The transformer encoder consists of several encoder blocks, each comprising Layer Normalization (LN), Multihead Self-Attention (MHSA), and Multilayer Perceptrons (MLP). Self-attention can learn image features globally by capturing the similarities between all patches. The input  $z$  is projected into Query, Key, and Value, where  $Q = zW^Q$ ,  $K = zW^K$  and  $V = zW^V$  via  $W^q \in \mathbb{R}^{D \times D_h}, W^k \in \mathbb{R}^{D \times D_h}, W^v \in \mathbb{R}^{D \times D_h}$ . Then, the corresponding attention weight can be written as

$$A = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \quad (9.2)$$

Then, the Self-Attention (SA), which is a product of the  $A$  and  $V$  matrices, is given by

$$SA(z) = AV = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (9.3)$$

The attention weight calculates the inner product of each entire vector and uses a softmax function; therefore, it has only one peak, and small relationships may be lost. Because multiple peaks can improve the expressive power of the network, the MHSA is introduced to obtain multiple attention weights by embedding multiple queries, keys, and values in a single patch. If the number of heads is  $k$  and the self-attention of the  $i$ -th head is  $SA_i(z)$ , the following equation is obtained:

$$MHSA(z) = [SA_1(z); SA_2(z); \dots; SA_k(z)]W^0, MHSA(z) \in \mathbb{R}^{N \times D} \quad (9.4)$$

Finally, the only input to the MLP is the class token. When the number of classifications is  $M$ , the ViT output  $y$  is as follows:

$$z'_l = MHSA(LN(z_{l-1})) + z_{l-1}, \quad l = 1 \dots L \quad (9.5)$$

$$z_l = MLP(LN(z'_l)) + z'_l, \quad l = 1 \dots L \quad (9.6)$$

$$y = LN(Z_L^0)W^y, \quad Z_L^0 \in \mathbb{R}^D, \quad W_L^0 \in \mathbb{R}^D \quad (9.7)$$

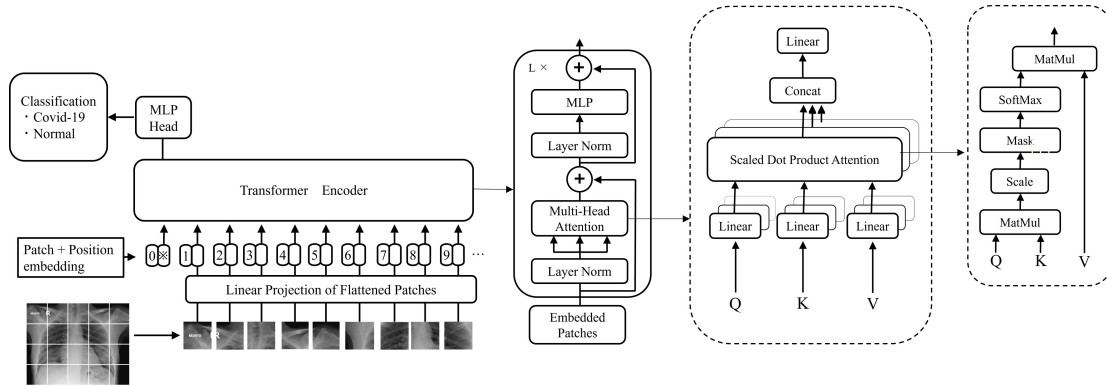


Figure 9.1: Vision Transformer Architecture

## 9.4. Material and method

### 9.4.1 Method and DataSets

The method in this experiment is illustrated in Figure 9.2. Different numbers of natural images or CXR images were pre-trained, and followed by transfer learning on common CXR images for the binary classification of COVID-19 and normal images. Three publicly available datasets were used in this study: NIH Chest X-ray14 [76] , ImageNet [13] and Valencian Region Medical ImageBank (BIMCV) [16]. The NIH dataset comprising 112120 CXR images with 14 diseases and one for the Normal label from 30805 unique patients. The CXR images were extracted from the PACS database through natural language processing (NLP) at the National Institutes of Health Clinical Center between 1992 and 2015 and disclosed as portable network graphics formats with additional information (i.e., patient ID, age, and gender and view position); The ImageNet is maintained by Stanford University and manually annotated over 1.4 million images with 1000 classes; and the BIMCV dataset contains CXR and CT images of COVID-19 and non-COVID-19 including digital imaging and communications in medicine (DICOM) metadata and radiologic reports. The COVID-19 patients were identified with at least one positive PCR test or positive immunological test by querying the Laboratory Information System records from the Health Information Systems in the Comunitat Valenciana. In this experiment, the NIH and ImageNet datasets were used for pre-training, and the number of pre-training images was randomly selected as 30000, 60000 and 90000 images. Meanwhile, 3000 CXR images from the BIMCV dataset were selected for binary classification after the pre-training. It was split into 2000 CXR images for training and validation and 1000 CXR images for testing. The ratio of the test dataset was the same for COVID-19 and normal. The number of images using the pre-training and transfer learning is described in Table 9.1.

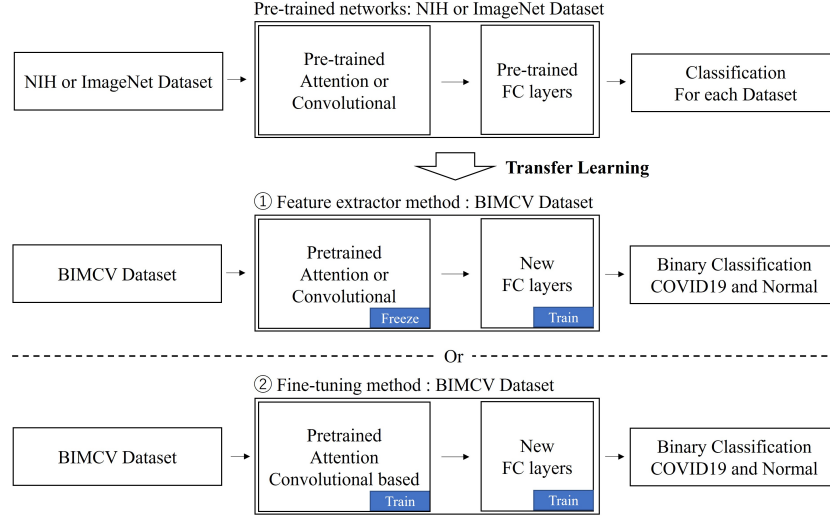


Figure 9.2: The feature extractor method and the fine-tuning method

Table 9.1: The number of images using the pre-training, transfer learning and test

Pre-trained (NIH or ImageNet dataset)	Transfer learning (BIMCV dataset)	Test (BIMCV dataset)
30000	2000	1000
60000	2000	1000
90000	2000	1000
1400000	2000	1000

## 9.4.2 Preprocessing

All CXR images derived from the NIH dataset are in png format, but the BIMCV dataset contains all images with DICOM format. It is a standard that defines the format of medical images and communication protocol between medical imaging devices. A digital image is represented by a pixel value. To view this image on a display, pixel values must be converted into brightness values. Windowing is a process that converts only a specific density range of an image with a wide range of pixel values, such as a medical image, into the density range of the display system. The DICOM images were converted from 16-bit to 8-bit png format using the Window Level (WL) and Window Width (WW) of the DICOM

tag using the following equation:

$$Window = 255 \times \left( \frac{value - min}{max - min} \right), \quad (9.8)$$

$$max = \left( \frac{WL + WW}{2} \right), min = \left( \frac{WL - WW}{2} \right) \quad (9.9)$$

Finally, all CXR images were resized to  $256 \times 256$  px and cropped in the center to  $224 \times 224$  px. The grayscale images were converted to a three-channel color format (RGB: red, green, and blue) to align with the ImageNet dataset. The pixel values of the input images were normalized between 0 and 1 based on the mean and the standard deviation to maintain the numerical stability. Figure 9.3 shows examples of CXR images in the NIH dataset, natural images in the ImageNet dataset and CXR images in the BIMCV dataset.

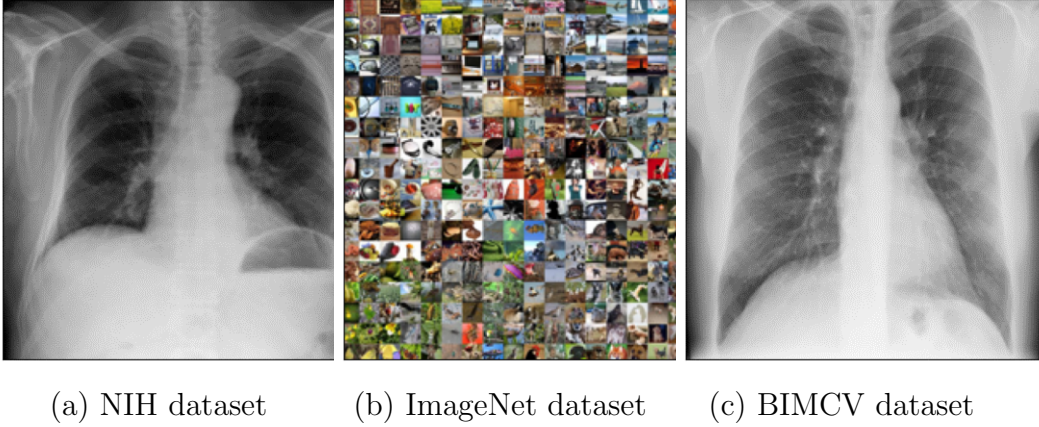


Figure 9.3: The example of a image for each dataset

## 9.5. Experiment and results

In the experiments, we adopted ViT-B/16 with 12 layers, 768 hidden sizes, 3072 MLP sizes, 12 heads, and 86 million parameters. Training was conducted using the Adam optimizer. The network was trained for 50 epochs and the learning rate was set to  $1 \times 10^{-5}$ . The loss and activation functions are the cross-entropy loss and rectified linear unit, respectively. All the performances were evaluated using a common test dataset from the BIMCV dataset. The area under the curve

(AUC) derived from the receiver operating characteristic (ROC) curve was used for the discrimination performance because the sensitivity and specificity depend on the threshold and the accuracy depends on the ratio of the test dataset. The ROC curve is a plot of the false positive rate (FPR) on the horizontal axis and the true positive rate (TPR) on the vertical axis for varying cutoff values for the output of the models through a softmax function. In this experiment, TPR is the percentage of all labeled COVID-19 cases that were correctly interpreted as COVID-19, and FPR is the percentage of all labeled normal cases that were correctly interpreted as normal. TPR is equal to sensitivity and FPR is equal to  $(1 - \text{specificity})$ . Both sensitivity and specificity take values between 0 and 1. AUC is the definite integral of an ROC curve and is an effective and combined measure of sensitivity and specificity that assesses the inherent validity of discrimination performance. An AUC closer to 1 indicates better test performance and 0.5 indicates random classification. To construct a 95% confidence interval, the standard error is calculated using the method of Hanley and McNeil [65].

Table 9.2 shows the results of pre-trained on different numbers of images for the feature extraction and the FT methods for ViT. All AUCs in the FT method were superior to those in the feature extraction method. Table 9.3 summarizes the comparison between the pre-trained CXR images and the natural images obtained using the FT method. This table includes the results of the same experiment using ResNet34 with 34 layers and 21.8 million parameters to compare with the trends with ViT-B/16. All the pre-trained on natural images outperformed the pre-trained CXR images, and the AUCs were better when a larger number of pre-trained images were used. On the other hand, the result of ResNet34 was superior to that of ViT-B/16. Furthermore, the results of pre-training on a large number of natural images are shown in Table 9.4. ResNet34 did not improve the performance; however ViT-B/16 improved the performance when a larger number of pre-trained images were used.

Figure 9.4, Figure 9.5 and Figure 9.6 show an example of the ROC curve. The better the predictive model is, the more the ROC curve bulges to the top left. Figure 9.4 shows The FT method pre-trained on 90000 natural is best the discrimination performance. Figure 9.5 shows 90000 natural images for ResNet34 in the FT method, which exhibits the best discrimination performance. Meanwhile,



Figure 9.6 shows that ResNet34 does not improve the performance, whereas ViT-B/16 improves the performance when a larger number of pre-trained images are used.

Table 9.2: The AUC comparison between the Feature Extractor method and the Fine-Tuning method pre-trained on CXR images and natural images for ViT-B/16.

Pre-trained images	CXR image		Natural image	
	Feature Extractor	Fine-Tuning	Feature Extractor	Fine-Tuning
30000	$0.517 \pm 0.036$	$0.573 \pm 0.035$	$0.562 \pm 0.035$	$0.684 \pm 0.033$
60000	$0.540 \pm 0.036$	$0.597 \pm 0.035$	$0.569 \pm 0.036$	$0.687 \pm 0.033$
90000	$0.551 \pm 0.036$	$0.581 \pm 0.035$	$0.574 \pm 0.035$	$0.707 \pm 0.032$

Table 9.3: The AUC comparison between CXR images and natural images for ViT-B/16 and ResNet34 in the Fine-Tuning method.

Pre-trained images	ViT-B/16		ResNet34	
	CXR image	Natural image	CXR image	Natural image
30000	$0.573 \pm 0.035$	$0.684 \pm 0.033$	$0.675 \pm 0.033$	$0.740 \pm 0.031$
60000	$0.597 \pm 0.035$	$0.687 \pm 0.033$	$0.684 \pm 0.033$	$0.766 \pm 0.029$
90000	$0.581 \pm 0.035$	$0.707 \pm 0.032$	$0.689 \pm 0.032$	$0.760 \pm 0.030$

Table 9.4: The AUC comparison between ViT-B/16 and ResNet34 pre-trained on natural images.

Pre-trained images	Natural image	
	ViT-B/16	ResNet34
0	$0.660 \pm 0.034$	$0.682 \pm 0.033$
90000	$0.707 \pm 0.032$	$0.760 \pm 0.030$
1400000	$0.761 \pm 0.030$	$0.760 \pm 0.030$

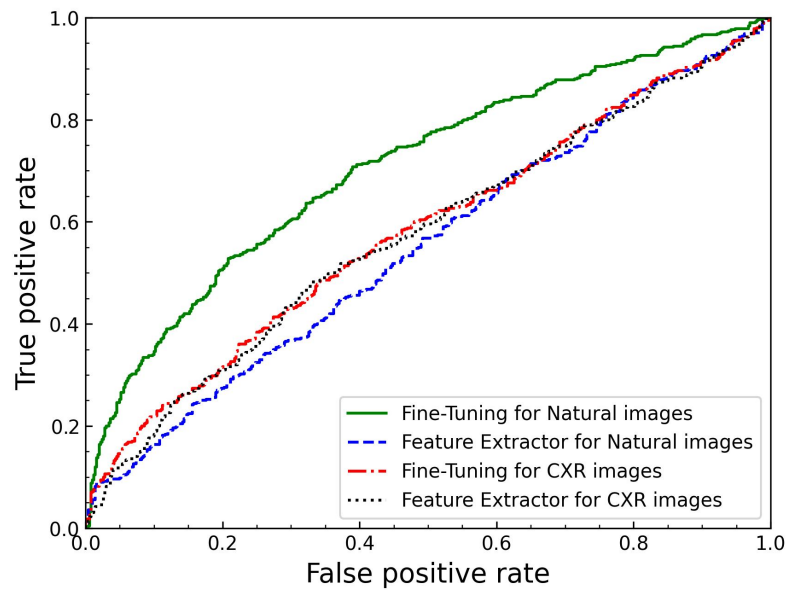


Figure 9.4: The ROC curve for the Feature Extractor method and the Fine-Tuning method pre-trained on 90000 natural and CXR images for ViT-B/16.

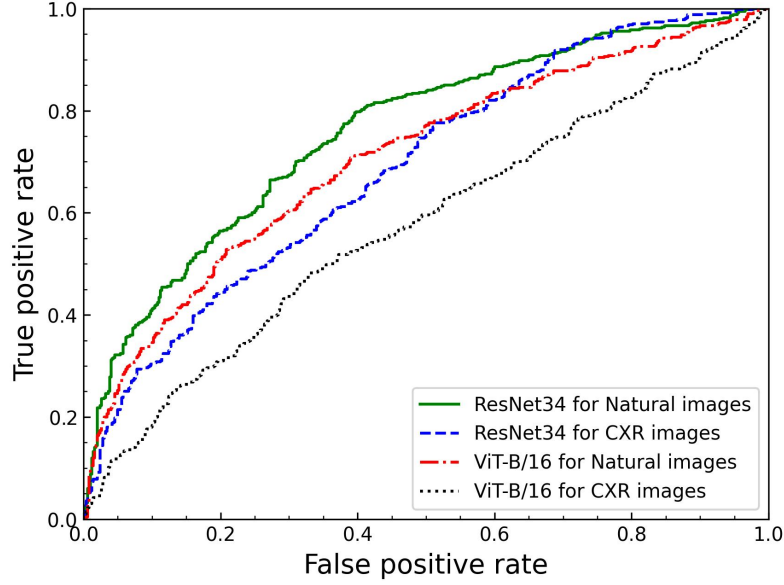
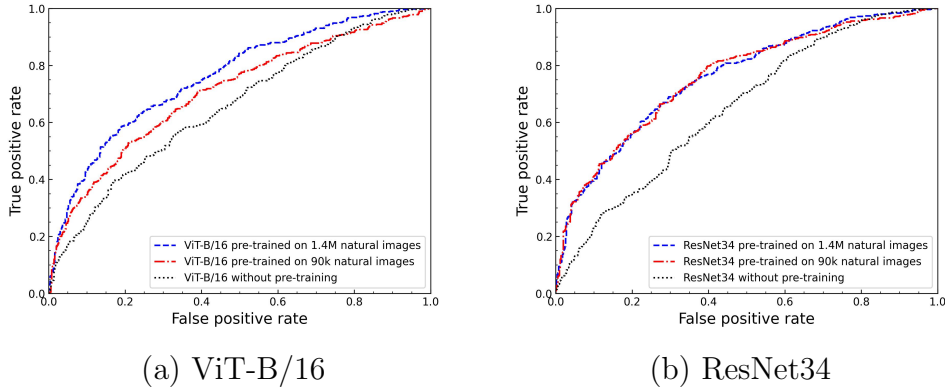


Figure 9.5: The ROC curve for 90000 natural and CXR images for ViT-B/16 and ResNet34 in the Fine-Tuning method



(a) ViT-B/16

(b) ResNet34

Figure 9.6: The ROC curve for the number of 0, 90000, and 1400000 natural and CXR images for ViT-B/16 and ResNet34 in the Fine-Tuning method.

## 9.6. Discussion

The following inferences were drawn from the experimental results: (1) the FT method was more effective than the feature extraction method; (2) pre-trained natural images using the FT method were more effective than task-specific images, namely CXR images; (3) the performance of ViT tended to become more effective as the number of pre-trained natural images increased.

Inference (1) showed that the FT method was also useful in ViT, similar to the CNN in radiology. The results of inference (2) were contrary to our expectations; therefore, further investigation was conducted. Peng et al. [68] obtained a similar trend for CNNs. They concluded that their results show that the medical-to-medical FT method does not perform better than the FT method on natural images because feature diversity is the most crucial quality required for the pre-trained models in the FT method. Our CNN results support their conclusions. Specifically, the number of CXR images in this study was in the tens of thousands, which are grayscale images and have a common anatomical structure across images. In contrast, the natural images such as ImageNet contains approximately 1.4 million natural color images with 22000 categories and 1000 labels.

To confirm the effect of the ViT pre-trained on natural images in more detail, more detailed experiments were conducted. In ViT, a learnable parameter called Position Embedding (PE) is added to each patch to retain position information. The PEs for pre-trained on CXR images and natural images ( $N = 30000, 60000$ , and  $90000$ ) are shown in Figure 9.7. Each model contained 1414 PEs, with each representing a patch location. The colors indicate the cosine similarity results for each patch. Yellow indicates higher similarity and blue indicates lower similarity. Because the cosine similarity is also calculated between own patches, the similarity of the reference own patch position is always 1 (brightest). Figure 9.7 shows that as the number of pre-trained on CXR images and natural images increased, the entirety of each patch became blurred and brighter. This is more remarkable when natural images are used for pre-training. We assume this is also because natural images have greater feature diversity than CXR images, and they can learn more long-range dependencies.

On the other hands, it is possible to indicate which batch of images ViT focused on for classification because of a mechanism for obtaining correspondence

between batches. The attention maps were visualized using the attention rollout method [92] to determine the patches the ViT focused on for class classification, as shown in Figure 9.8. The CXR image is an example of an image labeled COVID-19 in the BIMCV test dataset. The attention rollout first averages the attention weight per head.

$$\hat{A}^b = \frac{1}{H} \sum_{h=1}^H A_h^b \quad (9.10)$$

The unit matrix  $I \in \mathbb{R}^{(N+1) \times (N+1)}$  is then added to  $\hat{A}^b$  and multiplied by the layer.

$$\hat{A}^b = \hat{A}^b + I \quad (9.11)$$

$$\bar{A} = \prod_b \hat{A}^b \quad (9.12)$$

The overall attention weight obtained in the Attention Rollout  $\bar{A}$  is a matrix of  $(N + 1) \times (N + 1)$  patches including the class token. From this matrix, the Attention Weight between the class tokens and each patch token was calculated as an attention map. The position of the decision basis is represented from blue to red; the closer it is to red, the more likely that the image is the decision basis for classification. When the CXR images were used for pre-training, the local image regions were highlighted. When natural images were used, gradually larger image regions were highlighted as the number of pre-trained images increased. Furthermore, the highlight was centered on the contours of the lung fields in a remarkably large number of natural images. There is currently no clear definition of COVID-19 pneumonia. However, like other pneumonias, COVID-19 pneumonia causes the density of the lungs to increase. This may be seen as whitening of the lungs on CXR images [93]. In addition, transformers have been demonstrated to capture the global features of an image, specifically the shape of an object, for image recognition [94]. Based on these backgrounds and our results, we conclude that the fine-tuning of ViT-pre-trained on natural images is more effective than that of medical images especially for COVID-19 in tens of thousands of pre-training images. Finally, inference (3) shows that CNNs have an induced bias and can learn features locally in the images because they learn features with small size kernels, whereas transformers treat all batches equally so that they can learn features globally in the image; therefore, a large amount of data is needed

to learn these relationships. Meanwhile, the AUC for ResNet34 did not improve with a significant increase in the number of pre-trained images, which supports our previous conclusion that the AUC reached a state of stability after a certain amount of training [62].

Our study has some limitations. For example, pulmonary nodules on medical images are defined as lesions smaller than 30 mm. To downsize the image to align it with the ImageNet dataset, the lesion may be lost [78]. Hyunji et al. [95] conducted a comparison between CNN and ViT for classification of alzheimer’s disease (AD) using brain positron emission tomography (PET) images and concluded that it is hard to argue that the ViT model is better at AD classification than the CNN model. Chest radiography, computed tomography (CT), and magnetic resonance imaging (MRI) images are anatomical imaging modalities, while PET image is a functional imaging that picks up and images signals that reflect specific functions of the human body. Given this background, it is necessary to study diseases with local features other than COVID-19 in order to generalize our results. In addition, the characteristics of medical images such as CT, MRI, and PET images other than CXR images should also be considered. Furthermore, it is necessary to validate the results using different models because the ViT and CNN models are limited to ViT-B/16 and ResNet34 in this study. Therefore, in future research, we would like to investigate the feasibility of generalization to our results for various diseases, images, and model types. As another aspect, it is difficult to evaluate the same number of medical images as natural images. This is because the number of available medical images, especially annotated medical images, is limited. Recently, self-supervised learning methods that do not require annotated images have been actively studied. Therefore, we also would like to investigate the effectiveness of applying these learning methods to pre-training in the medical field.

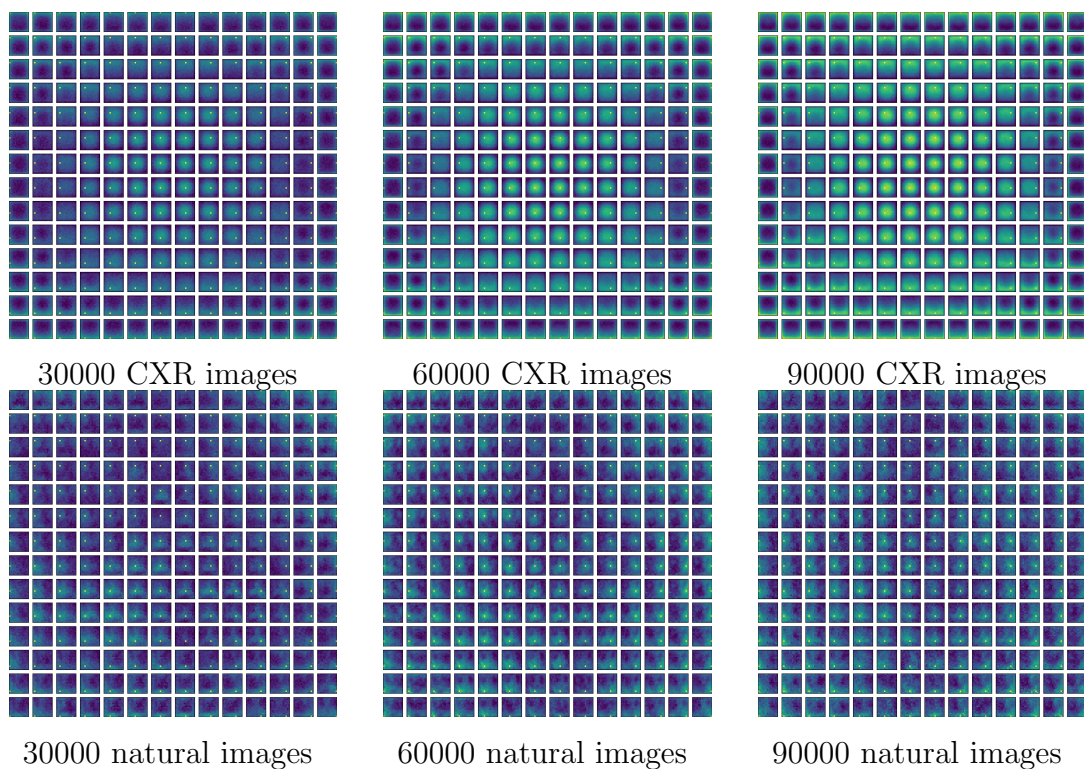


Figure 9.7: The visualization of Position Embeddings for each pre-training image.

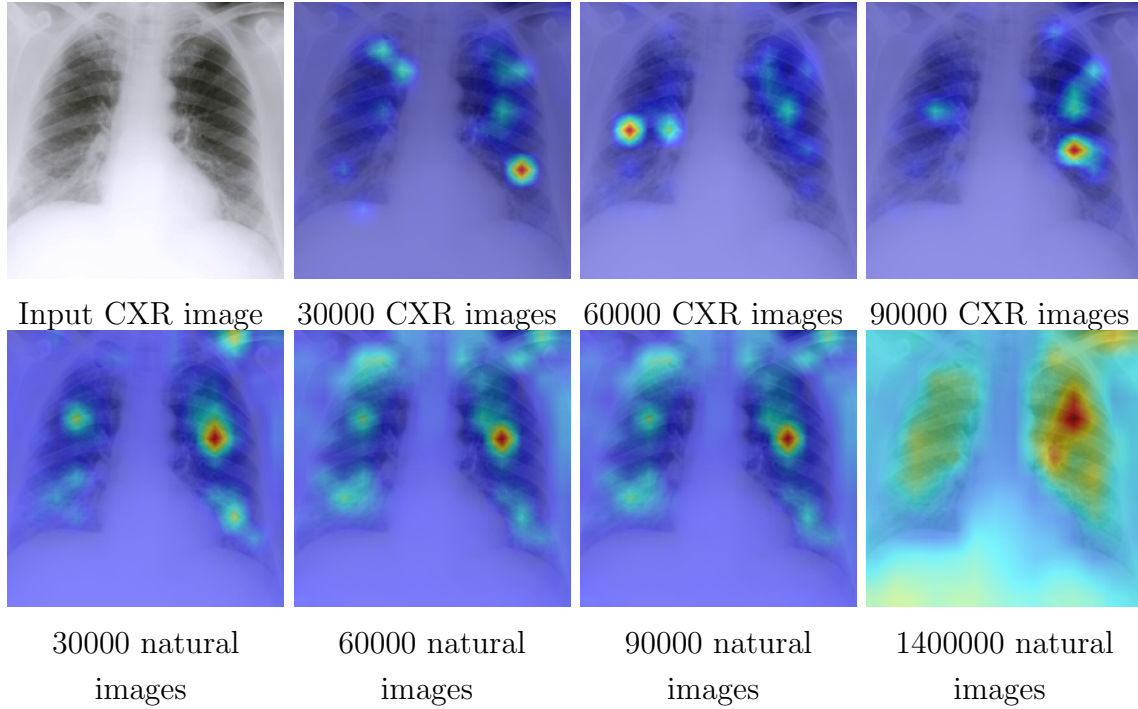


Figure 9.8: The visualization by using Attention Rollout method for each pre-training image.

## 9.7. Conclusion

The efficient use of labeled images is especially important for medical applications. The effectiveness of the pre-training method on chest X-ray s and natural images was evaluated for binary classification of COVID-19 and normal. Our results suggest that fine-tuning with a large amount of ImageNet pre-training data using ViT has the best discrimination performance for this binary classification. To generalize our results, it is necessary to consider the various pathologies, medical images, and models. Although more consideration is needed for generalization, our results provide fundamental insights into the effectiveness of the pre-training method.



# Chapter 10

## Conclusion

Our results show that (1) The performance change with respect to continuous learning depends on the type of model, number of training images, pre-training with and without natural images, and ratio of diseases in the training images. In particular, AlexNet and VGG16 with a few layers are more appropriate when the intended performance is low and the number of available training images is insufficient. Meanwhile, ResNet34 with many layers is appropriate when the intended performance is high and the number of available training images is sufficient. Performance degradation is also expected when only medical images are obtained at a limited number of facilities; therefore, these factors should be considered when establishing a change control plan before introducing into clinical environments, and (2) The number of labeled images can be significantly reduced by using the same type of medical images as those used in the classification task for self-supervised learning, and the models fine-tuned on natural images are more effective than those of medical images. This can be used to demonstrate the potential for early implementation in clinical practice.

Our study has some limitation and future works, in particular: 1) the relationship between accuracy and computational resources, 2) the applicability of CNNs designed for ImageNet to other specific fields and 3) pre-processing including RGB weight. Alfredo et al. [96] analyzed the relational accuracy, memory footprint, parameters, number of operations, inference time, and power consumption of several CNN models to provide practical applications, considering the limited resources in real world deployments. One of their results show that the neural network layer,

parameters, and inference time are not proportional to accuracy, and accuracy and inference time have a hyperbolic relationship: a small increase in accuracy requires a lot of computation time. Therefore, it is necessary to select CNN models not only in terms of accuracy but also in terms of computational cost in the real world when considering development and design. Bressem et al. [97] evaluated the effectiveness of 15 CNNs with five architectures (ResNet, DenseNet, VGG, SqueezeNet, Inception v4, and AlexNet), and the AUC results for the classification and training times are demonstrated. Deeper neural networks do not necessarily outperform shallow networks, and the accurate classification of CXR images can be achieved with comparably shallow networks. They concluded that increasing the complexity and depth of artificial neural networks is not always necessary for achieving state-of-the-art results. In particular, using these networks with limited hardware could be advantageous because they can be trained faster and more efficiently. As mentioned above consideration, the computational cost also image type should also be considered in real-world scenarios when deploying CNN models designed for ImageNet. On the other hand, In this study, the RGB weights for the training data were not considered, and color images were created by stacking grayscale images equally. In a previous study [98], a comparative experiment was conducted for the classification of CXR images using CNN fine-tuned on natural images: (1) The classification performance of CXR grayscale images by pre-training natural images that were converted into gray images using Luma’s formula; that is, RGB weights were added, and (2) The classification performance of CXR color images when the natural images were pre-trained as color images and the CXR images were converted into color images without adding RGB weights. These results showed that (1) grayscale results statistically outperformed (2) color results for 8 of the 14 diseases. Furthermore, the computation time for (1) grayscale results was reduced by 20% compared to (2) color results. RGB weighting during pre-processing is one of the factors that influences the performance change with respect to continuous learning because most medical images are grayscale images. More detailed studies will be the subject of future research.

There are some limitations and future works that need to be considered regarding the generalizability of our results. However, our study provides a basis for establishing pre-change control plans and learning efficiency during model de-

velopment for reasonable and better regulations.

# Chapter 11

## Publication

### Peer-reviewed Journal Papers

1. Imagawa, K., and Shiomoto, K. (2022). Performance change with the number of training data: A case study on the binary classification of COVID-19 chest X-ray by using convolutional neural networks. *Computers in Biology and Medicine*, 142, 105251.
2. Imagawa, K., and Shiomoto, K. (2023, July). Performance Change with the Ratio of Training Data A Case Study on the Binary Classification of COVID-19 Chest X-Ray by using Convolutional Neural Networks. In *2023 IEEE International Symposium on Medical Measurements and Applications (MeMeA)* (pp. 1-5). IEEE.
3. Imagawa, K., and Shiomoto, K. (2024). Evaluation of Effectiveness of Self-Supervised Learning in Chest X-Ray Imaging to Reduce Annotated Images. *Journal of Imaging Informatics in Medicine*, 1-7.

### International Conference Presentation

1. Imagawa, K., and Shiomoto, K. (2023, June). Performance Change with the Ratio of Training Data A Case Study on the Binary Classification of COVID-19 Chest X-Ray by using Convolutional Neural Networks. In *2023 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*. IEEE.(Jeju Island)

## Japanese Conference Presentation

1. Shibuya.Y, Imagawa, K., and Shiimoto, K.(2023, March). Effectiveness of FineTuning with ViT in Medical Imaging. Technical Committee on Electromagnetic Compatibility / Technical Committee on Healthcare and Medical Information Communication Technology, MICT2022-56,vol.122,MICT-447,pp.1-5. (Tokyo)

# Acknowledgements

I would like to thank my supervisor, Prof. Kohei Shiimoto, for his invaluable guidance. I have been given an appropriate roadmap for conducting research in my working life. I would also like to thank my colleagues at the Pharmaceutical and Medical Device Agency for giving me the opportunity to conduct research while working. Finally, I would like to thank my family for their support.

# References

- [1] Labour Ministry of Health and Welfare. Handling of post-approval change management protocol (pacmp) of medical device. notification no.0831-14 of psehb, August 2020.
- [2] U.S. Food and Drug Administration. Proposed regulatory framework for modifications to artificial intelligence/machine learning (ai/ml)-based software as a medical device (samd) - discussion paper and request for feedback., April 2019.
- [3] U.S. Food and Drug Administration. Marketing submission recommendations for a predetermined change control plan for artificial intelligence/machine learning (ai/ml)-enabled device software functions, April 2023.
- [4] The U.S. Food, Health Canada Drug Administration (FDA), the United Kingdom’s Medicines, and Healthcare products Regulatory Agency (MHRA). Good machine learning practice for medical device development: Guiding principles, October 2021.
- [5] The U.S. Food, Health Canada Drug Administration (FDA), the United Kingdom’s Medicines, and Healthcare products Regulatory Agency (MHRA). Predetermined change control plans for machine learning-enabled medical devices: Guiding principles, October 2023.
- [6] Justin Sirignano and Konstantinos Spiliopoulos. Scaling limit of neural networks with the xavier initialization and convergence to a global minimum, 2022.

- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, 2015.
- [8] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- [9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.
- [13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [14] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [15] Alberto Signoroni, Mattia Savardi, Sergio Benini, Nicola Adami, Riccardo Leonardi, Paolo Gibellini, Filippo Vaccher, Marco Ravanelli, Andrea Borghesi, Roberto Maroldi, et al. Bs-net: Learning covid-19 pneumonia severity on a large chest x-ray dataset. *Medical Image Analysis*, 71:102046, 2021.
- [16] Maria de la Iglesia Vayá, Jose Manuel Saborit, Joaquim Angel Montell, Antonio Pertusa, Aurelia Bustos, Miguel Cazorla, Joaquin Galant, Xavier



- Barber, Domingo Orozco-Beltrán, Francisco García-García, et al. Bimcv covid-19+: a large annotated dataset of rx and ct images from covid-19 patients. *arXiv preprint arXiv:2006.01174*, 2020.
- [17] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017.
  - [18] William J Youden. Index for rating diagnostic tests. *Cancer*, 3(1):32–35, 1950.
  - [19] J. A. Hanley and B. J. McNeil. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology*, 148(3):839–843, Sep 1983.
  - [20] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Sánchez. A survey on deep learning in medical image analysis. *Med Image Anal*, 42:60–88, Dec 2017.
  - [21] Ryuji Hamamoto, Kruthi Suvarna, Masayoshi Yamada, Kazuma Kobayashi, Norio Shinkai, Mototaka Miyake, Masamichi Takahashi, Shunichi Jinnai, Ryo Shimoyama, Akira Sakai, Ken Takasawa, Amina Bolatkan, Kanto Shozu, Ai Dozen, Hidenori Machino, Satoshi Takahashi, Ken Asada, Masaaki Komatsu, Jun Sese, and Syuzo Kaneko. Application of artificial intelligence technology in oncology: Towards the establishment of precision medicine. *Cancers*, 12(12), 2020.
  - [22] Urs J Muehlemaier, Paola Danio, and Kerstin N Vokinger. Approval of artificial intelligence and machine learning-based medical devices in the usa and europe (2015–20): a comparative analysis. *The Lancet Digital Health*, 2021.
  - [23] U.S. Food and Drug Administration. Artificial intelligence and machine learning (ai/ml) software as a medical device action plan., January 2021.

- [24] World Health Organization. Use of chest imaging in covid-19: a rapid advice guide, june 2020.
- [25] American College of RadiologyO. Acr recommendations for the use of chest radiography and computed tomography (ct) for suspected covid-19 infection, MARCH 2020.
- [26] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.
- [27] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi. Convolutional neural networks: an overview and application in radiology. *Insights Imaging*, 9(4):611–629, Aug 2018.
- [28] Ezz El-Din Hemdan, Marwa A. Shouman, and Mohamed Esmail Karar. Covidx-net: A framework of deep learning classifiers to diagnose covid-19 in x-ray images. *arXiv:2003.11055*, 2020.
- [29] Ali Narin, Ceren Kaya, and Ziyne Pamuk. Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks. *arXiv:2003.10849*, 2020.
- [30] Prabira Kumar Sethy and Santi Kumari Behera. Detection of coronavirus disease (covid-19) based on deep features. *MDPI AG*, 2020.
- [31] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. Rajendra Acharya. Automated detection of COVID-19 cases using deep neural networks with X-ray images. *Comput Biol Med*, 121:103792, 06 2020.
- [32] Linda Wang and Alexander Wong. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *arXiv:2003.09871*, 2020.
- [33] F. Ucar and D. Korkmaz. COVIDiagnosis-Net: Deep Bayes-SqueezeNet based diagnosis of the coronavirus disease 2019 (COVID-19) from X-ray images. *Med Hypotheses*, 140:109761, Jul 2020.

- [34] S. R. Nayak, D. R. Nayak, U. Sinha, V. Arora, and R. B. Pachori. Application of deep learning techniques for detection of COVID-19 cases using chest X-ray images: A comprehensive study. *Biomed Signal Process Control*, 64:102365, Feb 2021.
- [35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.
- [36] M. M. Rahaman, C. Li, Y. Yao, F. Kulwa, M. A. Rahman, Q. Wang, S. Qi, F. Kong, X. Zhu, and X. Zhao. Identification of COVID-19 samples from chest X-Ray images using deep learning: A comparison of transfer learning approaches. *J Xray Sci Technol*, 28(5):821–839, 2020.
- [37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.
- [38] Tuan D Pham. Classification of covid-19 chest x-rays with deep learning: new models or fine tuning? *Health Information Science and Systems*, 9(1):1–11, 2021.
- [39] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *arXiv:1409.4842*, 2014.
- [40] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and 0.5mb model size. *arXiv:1602.07360*, 2016.
- [41] L. Wynants, B. Van Calster, G. S. Collins, R. D. Riley, G. Heinze, E. Schuit, M. M. J. Bonten, D. L. Dahly, J. A. A. Damen, T. P. A. Debray, V. M. T. de Jong, M. De Vos, P. Dhiman, M. C. Haller, M. O. Harhay, L. Henckaerts, P. Heus, M. Kammer, N. Kreuzberger, A. Lohmann, K. Luijken, J. Ma, G. P. Martin, D. J. McLernon, C. L. Andaur Navarro, J. B. Reitsma, J. C. Sergeant, C. Shi, N. Skoetz, L. J. M. Smits, K. I. E. Snell, M. Sperrin, R. Spijker, E. W. Steyerberg, T. Takada, I. Tzoulaki, S. M. J.

- van Kuijk, B. van Bussel, I. C. C. van der Horst, F. S. van Royen, J. Y. Verbakel, C. Wallisch, J. Wilkinson, R. Wolff, L. Hooft, K. G. M. Moons, and M. van Smeden. Prediction models for diagnosis and prognosis of covid-19: systematic review and critical appraisal. *BMJ*, 369:m1328, 04 2020.
- [42] Michael Roberts, Derek Driggs, Matthew Thorpe, Julian Gilbey, Michael Yeung, Stephan Ursprung, Angelica I Aviles-Rivero, Christian Etmann, Cathal McCague, Lucian Beer, et al. Common pitfalls and recommendations for using machine learning to detect and prognosticate for covid-19 using chest radiographs and ct scans. *Nature Machine Intelligence*, 3(3):199–217, 2021.
- [43] A. Signoroni, M. Savardi, S. Benini, N. Adami, R. Leonardi, P. Gibellini, F. Vaccher, M. Ravanelli, A. Borghesi, R. Maroldi, and D. Farina. BS-Net: Learning COVID-19 pneumonia severity on a large chest X-ray dataset. *Med Image Anal*, 71:102046, 07 2021.
- [44] Lu Wang, Hao Wang, Chen Xia, Yao Wang, Qiaohong Tang, Jiage Li, and Xiao-Hua Zhou. Toward standardized premarket evaluation of computer aided diagnosis/detection products: insights from fda-approved products. *Expert review of medical devices*, 17(9):899—918, September 2020.
- [45] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.
- [46] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *arXiv:1411.1792*, 2014.
- [47] Rajeev Kumar and Abhaya Indrayan. Receiver operating characteristic (roc) curve for medical researchers. *Indian pediatrics*, 48(4):277–287, 2011.
- [48] Yingtao Fang, Jiazhou Wang, Xiaomin Ou, Hongmei Ying, Chaosu Hu, Zhen Zhang, and Weigang Hu. The impact of training sample size on deep learning-based organ auto-segmentation for head-and-neck patients. *Physics in Medicine & Biology*, 66(18):185012, 2021.

- [49] Ravi K Samala, Heang-Ping Chan, Lubomir Hadjiiski, Mark A Helvie, Caleb D Richter, and Kenny H Cha. Breast cancer diagnosis in digital breast tomosynthesis: effects of training sample size on multi-stage transfer learning using deep neural nets. *IEEE transactions on medical imaging*, 38(3):686–696, 2018.
- [50] Nima Tajbakhsh, Jae Y Shin, Suryakanth R Gurudu, R Todd Hurst, Christopher B Kendall, Michael B Gotway, and Jianming Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE transactions on medical imaging*, 35(5):1299–1312, 2016.
- [51] Ran Zhang, Xin Tie, Zhihua Qi, Nicholas B Bevins, Chengzhu Zhang, Dalton Griner, Thomas K Song, Jeffrey D Nadig, Mark L Schiebler, John W Garrett, et al. Diagnosis of coronavirus disease 2019 pneumonia by using chest radiography: Value of artificial intelligence. *Radiology*, 298(2):E88–E97, 2021.
- [52] Abdullahi Umar Ibrahim, Mehmet Ozsoz, Sertan Serte, Fadi Al-Turjman, and Polycarp Shizawaliyi Yakoi. Pneumonia classification using deep learning from chest x-ray images during covid-19. *Cognitive Computation*, pages 1–13, 2021.
- [53] R. N. D’souza, P. Y. Huang, and F. C. Yeh. Structural Analysis and Optimization of Convolutional Neural Networks with a Small Sample Size. *Sci Rep*, 10(1):834, 01 2020.
- [54] Joseph Paul Cohen, Paul Morrison, Lan Dao, Karsten Roth, Tim Q Duong, and Marzyeh Ghassemi. Covid-19 image data collection: Prospective predictions are the future. *arXiv 2006.11988*, 2020.
- [55] Joseph Paul Cohen, Paul Morrison, and Lan Dao. Covid-19 image data collection. *arXiv 2003.11597*, 2020.
- [56] Ryuji Hamamoto, Kruthi Suvarna, Masayoshi Yamada, Kazuma Kobayashi, Norio Shinkai, Mototaka Miyake, Masamichi Takahashi, Shunichi Jinnai, Ryo Shimoyama, Akira Sakai, et al. Application of artificial intelligence

- technology in oncology: Towards the establishment of precision medicine. *Cancers*, 12(12):3532, 2020.
- [57] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.
  - [58] [7] The Center for Systems Science and Engineering at Johns Hopkins University. Covid-19 dashboard” coronavirus covid-19 (2019-ncov), June 2022.
  - [59] Aijaz Ahmad Reshi, Furqan Rustam, Arif Mehmood, Abdulaziz Alhossan, Ziyad Alrabiah, Ajaz Ahmad, Hessa Alsuwailem, and Gyu Sang Choi. An efficient cnn model for covid-19 disease detection based on x-ray image classification. *Complexity*, 2021:1–12, 2021.
  - [60] Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016.
  - [61] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.
  - [62] Kuniki Imagawa and Kohei Shiimoto. Performance change with the number of training data: A case study on the binary classification of covid-19 chest x-ray by using convolutional neural networks. *Computers in Biology and Medicine*, 142:105251, 2022.
  - [63] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

- [64] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.
- [65] James A Hanley and Barbara J McNeil. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology*, 148(3):839–843, 1983.
- [66] Ibrahim Tolga Öztürk, Rostislav Nedelchev, Christian Heumann, Esteban Garces Arias, Marius Roger, Bernd Bischl, and Matthias Aßenmacher. How different is stereotypical bias across languages? *arXiv preprint arXiv:2307.07331*, 2023.
- [67] Bogdan A Bercean, Andreea Birhala, Paula G Ardelean, Ioana Barbulescu, Marius M Benta, Cristina D Rasadean, Dan Costachescu, Cristian Avramescu, Andrei Tenescu, Stefan Iarca, et al. Evidence of a cognitive bias in the quantification of covid-19 with ct: an artificial intelligence randomised clinical trial. *Scientific Reports*, 13(1):4887, 2023.
- [68] Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. *Advances in neural information processing systems*, 32, 2019.
- [69] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.
- [70] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhao-han Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent—a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.
- [71] Saeed Shurrab and Rehab Duwairi. Self-supervised learning methods and

- applications in medical imaging analysis: a survey. *PeerJ. Computer science*, 8:e1045, 2022.
- [72] Shih-Cheng Huang, Anuj Pareek, Malte Jensen, Matthew P Lungren, Serena Yeung, and Akshay S Chaudhari. Self-supervised learning for medical image classification: a systematic review and implementation guidelines. *NPJ Digital Medicine*, 6(1):74, 2023.
- [73] Shekoofeh Azizi, Basil Mustafa, Fiona Ryan, Zachary Beaver, Jan Freyberg, Jonathan Deaton, Aaron Loh, Alan Karthikesalingam, Simon Kornblith, Ting Chen, et al. Big self-supervised models advance medical image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3478–3488, 2021.
- [74] Hari Sowrirajan, Jingbo Yang, Andrew Y Ng, and Pranav Rajpurkar. Moco pretraining improves representation and transferability of chest x-ray models. In *Medical Imaging with Deep Learning*, pages 728–744. PMLR, 2021.
- [75] Guang Li, Ren Togo, Takahiro Ogawa, and Miki Haseyama. Covid-19 detection based on self-supervised transfer learning using chest x-ray images. *International Journal of Computer Assisted Radiology and Surgery*, 18(4):715–722, 2023.
- [76] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, M Bagheri, and R Summers. Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *IEEE CVPR*, volume 7, page 46. sn, 2017.
- [77] Rhett N D’souza, Po-Yao Huang, and Fang-Cheng Yeh. Structural analysis and optimization of convolutional neural networks with a small sample size. *Scientific reports*, 10(1):834, 2020.
- [78] Kyungjin Cho, Ki Duk Kim, Yujin Nam, Jiheon Jeong, Jeeyoung Kim, Changyong Choi, Soyoung Lee, Jun Soo Lee, Seoyeon Woo, Gil-Sun Hong, et al. Chess: Chest x-ray pre-trained model via self-supervised contrastive learning. *Journal of Digital Imaging*, pages 1–9, 2023.



- [79] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silvana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 590–597, 2019.
- [80] Andrew B. Sellergren, Christina Chen, Zaid Nabulsi, Yuanzhen Li, Aaron Maschinot, Aaron Sarna, Jenny Huang, Charles Lau, Sreenivasa Raju Kalidindi, Mozziyar Etemadi, Florencia Garcia-Vicente, David Melnick, Yun Liu, Krish Eswaran, Daniel Tse, Neeral Beladia, Dilip Krishnan, and Shravya Shetty. Simplified transfer learning for chest radiography models using less data. *Radiology*, 305(2):454–465, November 2022. Funding Information: Supported by Google. The authors thank the members of the Google Health Radiology and labeling software teams for software infrastructure support, logistical support, and assistance in data labeling. For the ChestX-ray14 data set, we thank the NIH Clinical Center for making it publicly available. Sincere appreciation also goes to the radiologists who enabled this work with their image interpretation and annotation efforts throughout the study, Jonny Wong, BA, for coordinating the imaging annotation work, and Akinori Mitani, MD, and Craig H. Mermel, MD, PhD, for providing feedback on the manuscript. Publisher Copyright: © RSNA, 2022.
- [81] Hiroshi Fujita. Ai-based computer-aided diagnosis (ai-cad): the latest review to read first. *Radiological physics and technology*, 13(1):6–19, 2020.
- [82] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.
- [83] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [84] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

- [85] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- [86] Fahad Shamshad, Salman Khan, Syed Waqas Zamir, Muhammad Haris Khan, Munawar Hayat, Fahad Shahbaz Khan, and Huazhu Fu. Transformers in medical imaging: A survey. *Medical Image Analysis*, page 102802, 2023.
- [87] Alistair EW Johnson, Tom J Pollard, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Yifan Peng, Zhiyong Lu, Roger G Mark, Seth J Berkowitz, and Steven Horng. Mimic-cxr-jpg, a large publicly available database of labeled chest radiographs. *arXiv preprint arXiv:1901.07042*, 2019.
- [88] Hee E Kim, Alejandro Cosa-Linan, Nandhini Santhanam, Mahboubeh Janesari, Mate E Maros, and Thomas Ganslandt. Transfer learning for medical image classification: a literature review. *BMC Med. Imaging*, 22(1):69, April 2022.
- [89] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9:611–629, 2018.
- [90] Mohammad Usman, Tehseen Zia, and Ali Tariq. Analyzing transfer learning of vision transformers for interpreting chest radiography. *Journal of digital imaging*, 35(6):1445–1462, 2022.
- [91] Christos Matsoukas, Johan Fredin Haslum, Magnus Söderberg, and Kevin Smith. Is it time to replace cnns with transformers for medical images? *arXiv preprint arXiv:2108.09038*, 2021.
- [92] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*, 2020.
- [93] Joanne Cleverley, James Piper, and Melvyn M Jones. The role of chest radiography in confirming covid-19 pneumonia. *bmj*, 370, 2020.

- [94] Shikhar Tuli, Ishita Dasgupta, Erin Grant, and Thomas L Griffiths. Are convolutional neural networks or transformers more like human vision? *arXiv preprint arXiv:2105.07197*, 2021.
- [95] Hyunji Shin, Soomin Jeon, Youngsoo Seol, Sangjin Kim, and Doyoung Kang. Vision transformer approach for classification of alzheimer’s disease using 18f-florbetaben brain images. *Applied Sciences*, 13(6):3453, 2023.
- [96] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. An analysis of deep neural network models for practical applications, 2017.
- [97] Keno K Bressem, Lisa C Adams, Christoph Erxleben, Bernd Hamm, Stefan M Niehues, and Janis L Vahldiek. Comparing different deep learning architectures for classification of chest radiographs. *Scientific reports*, 10(1):13590, 2020.
- [98] Yiting Xie and David Richmond. Pre-training on grayscale imagenet improves medical image classification. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018.
- [99] U.S. Food and Drug Administration. Fda guidance for industry and food and drug administration staff computer-assisted detection devices applied to radiology images and radiology device data - premarket notification [510(k)] submissions., October 2019.
- [100] George C Linderman, Manas Rachh, Jeremy G Hoskins, Stefan Steinberger, and Yuval Kluger. Efficient algorithms for t-distributed stochastic neighborhood embedding. *arXiv preprint arXiv:1712.09005*, 2017.
- [101] Xiaolong Pei, Yu hong Zhao, Liwen Chen, Qingwei Guo, Zhiqiang Duan, Yue Pan, and Hua Hou. Robustness of machine learning to color, size change, normalization, and image enhancement on micrograph datasets with large sample differences. *Materials & Design*, 232:112086, 2023.
- [102] Gerosh Shibu George, Pratyush Raj Mishra, Panav Sinha, and Manas Ranjan Prusty. Covid-19 detection on chest x-ray images using homomorphic transformation and vgg inspired deep convolutional neural network. *Biocybernetics and Biomedical Engineering*, 43(1):1–16, 2023.