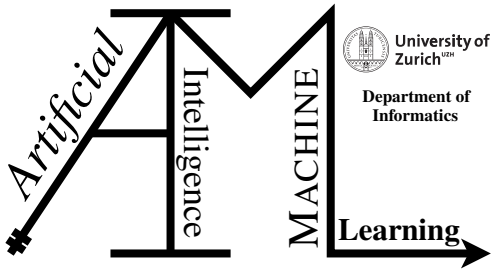
Explainable Classification of COVID-19 in Chest X-ray Images

Master Thesis

Yingying Chen
19-759-588

Submitted on
October 03 2022

Thesis Supervisor
Prof. Dr. Manuel Günther
Prof. Dr. Carlo Menon
Dr. Mohamed Elgendi



Master Thesis

Author: Yingying Chen, yingying.chen@uzh.ch

Project period: March 21 2022 - October 03 2022

Artificial Intelligence and Machine Learning Group
Department of Informatics, University of Zurich

Acknowledgements

I would like to express my deepest appreciation to my supervisor Dr. Mohamed Elgendi for his guidance and support throughout the thesis. I am extremely grateful to Prof. Dr. Manuel Günther for his valuable advice and patience. I also would like to thank Artificial Intelligence and Machine Learning Group (AIML) for providing the resources to work on this project. In the past six months, I have learned a lot from them. Without their support, I could not have completed this thesis. Before my thesis, my knowledge of machine learning came from coursework. Not only did they teach me about machine learning, but they also gave me a chance of doing interpretable machine learning research. At the beginning of this project, I had never been exposed to research related to the combination of medical image processing and machine learning. Sincere thanks to them for opening the door for me to understand and decode machine learning and image processing.

I would also like to extend my sincere thanks to Prof. Dr. Carlo Menon and the entire Biomedical and Mobile Health Technology Lab (BMHT) at ETH for their help. I am grateful to them for providing the environment and the opportunity to work on this project.

Besides, I would like to thank my parents, family, boyfriend and friends for their unconditional support and encouragement. Every time I am stuck in this project, I can always feel their love and support, which encourages me to move on at every step of my journey.

The Pytorch community is very active and helpful. I gain a lot of practical experience from it. My appreciation extends to them and all the developers in the Pytorch community.

Abstract

Nowadays, developing an automated diagnosis system that can detect COVID-19 or other pneumonia from CXR images without human intervention still meets with great challenges when the system lacks interpretability of the deep learning model. We need not only the high accuracy of the model but also the interpretability of the model. In this work, we focus on the explainable classification of chest X-ray images. We pay attention not only to COVID-19 but other kinds of lung infections. First, we propose three new and simple visualization methods to improve the interpretability of the deep learning model. The proposed methods are based on the class activation map (CAM) (Zhou et al., 2016) framework. Second, we propose a quantitative metric, acceptable mask ratio, to evaluate the interpretability of the deep learning model so that we can assess different methods intuitively.

Through experiments, we find that better performance of the model does not necessarily correspond to better interpretability. With the help of acceptable masking rates, we can contribute to a certain extent to the selection of models with high accuracy and good interpretability for automated diagnostic systems. Furthermore, the proposed visualization methods can be used to interpret the classification results of deep learning models and help clinicians to build more credible diagnostic models.

Zusammenfassung

Heute ist es noch immer eine große Herausforderung, ein automatisiertes Diagnosesystem zu entwickeln, das COVID-19 oder andere Lungenentzündungen aus CXR-Bildern ohne menschliche Intervention erkennen kann, wenn das System die Interpretierbarkeit des tiefen Lernensystems nicht hat. Wir brauchen nicht nur die hohe Genauigkeit des Modells, sondern auch die Interpretierbarkeit des Modells. In dieser Arbeit konzentrieren wir uns auf die erklärbare Klassifizierung von Röntgenaufnahmen der Brust. Wir achten nicht nur auf COVID-19, sondern auch auf andere Arten von Lungenerkrankungen. Zuerst schlagen wir drei neue und einfache Visualisierungsmethoden vor, um die Interpretierbarkeit des Deep-Learning-Modells zu verbessern. Die vorgeschlagenen Methoden basieren auf dem Class Activation Map (CAM) (Zhou et al., 2016) Framework. Zweitens schlagen wir eine quantitative Metrik, Acceptable Mask Ratio, vor, um die Interpretierbarkeit des Deep-Learning-Modells zu bewerten, damit wir verschiedene Methoden intuitiv bewerten können.

Durch Experimente finden wir heraus, dass eine bessere Leistung des Modells nicht unbedingt mit einer besseren Interpretierbarkeit korrespondiert. Mit Hilfe der akzeptablen Maskierungsrate können wir zu einem gewissen Grad zur Auswahl von Modellen mit hoher Genauigkeit und guter Interpretierbarkeit für automatisierte Diagnosesysteme beitragen. Darüber hinaus können die vorgeschlagenen Visualisierungsmethoden zur Interpretation der Klassifizierungsergebnisse von Deep-Learning-Modellen verwendet werden und helfen Ärzten, bessere klinische Diagnosemodelle zu erstellen.

Contents

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Overview . . . 3
- 1.3 Scope . . . 4

2 Related Work **5**

3 Background **7**
- 3.1 Chest X-ray Images . . . 7
- 3.2 Transfer Learning . . . 7
- 3.3 CNN Architecture of Pre-trained Models . . . 8
- 3.4 Global Pooling Methods . . . 8
  - 3.4.1 Global Average Pooling . . . 9
  - 3.4.2 Global Max Pooling . . . 10
  - 3.4.3 Global Exponential Pooling . . . 10
  - 3.4.4 Global Linear Pooling . . . 10
  - 3.4.5 Global Log-sum-exp Pooling . . . 11
- 3.5 Model Interpretability . . . 11
  - 3.5.1 Class Activation Mapping . . . 11
  - 3.5.2 Gradient-weighted Class Activation Mapping . . . 11

4 Methodology **13**
- 4.1 Network Architecture . . . 13
  - 4.1.1 Training and Testing . . . 13
  - 4.1.2 Visualization . . . 13
- 4.2 Lesion localization . . . 14
  - 4.2.1 Exponential CAM . . . 15
  - 4.2.2 Proportional CAM . . . 15
  - 4.2.3 Probabilistic CAM . . . 15
- 4.3 Evaluation Metric of Lesion Localization . . . 16
  - 4.3.1 Acceptable Mask Ratio . . . 16

5 Experiments and results **19**
- 5.1 Experiment Setup . . . 19
  - 5.1.1 Dataset . . . 19
  - 5.1.2 Preprocessing . . . 20
  - 5.1.3 Network construction and Training . . . 21

5.1.4 Configuration of Experiments 21
5.2 Quantitative Analysis 24
5.2.1 Evaluation Metrics for Classification 24
5.2.2 Performance Evaluation for Classification Models 27
5.2.3 Performance Evaluation for Visualization Methods 27
5.2.4 Summary of Lesion Localization Results 38

6 Discussion and Future Work **45**

7 Conclusion **49**

A Attachments **51**

Chapter 1

Introduction

1.1 Motivation

Coronavirus disease 2019 (COVID-19) is a new contagious disease caused by the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). First originated in Wuhan, China, in December 2019, it has spread to over 200 countries with major impacts on USA and Europe due to its deadly effects. Since the COVID-19 cases substantially escalated within three months, World Health Organization (WHO) assessed and declared that COVID-19 is a worldwide pandemic. Until 8th August 2022, more than 580,000,000 cases were infected by COVID-19, and 6,410,961 cumulative cases were dead, which are confirmed by WHO [1].

Covid 19 belongs to the SARS-CoV family (Lam et al., 2020) which attacks the lungs and respiratory system of the human body. With the increase in cases of COVID-19 around the world, the observed symptoms are fever, cough, and pneumonia, followed by respiratory problems (Lai et al., 2020). Furthermore, at least one-third of people are suffering from COVID-19 but are found to be asymptomatic (Oran and Topol, 2021; Lin et al., 2020). However, patients are contagious during the incubation period. Even though they do not have noticeable symptoms, they can already infect others. To prevent rapid spread, the governments urge the public to be tested and screened for COVID-19 through the rapid diagnostic test (RDT). RDT is a type of antibody detection test that can produce results quickly within 15 minutes. However, the effectiveness of the RDT test depends on the sample quality and the onset time. It can yield false positive and false negative results (Drain, 2022). The false positive results generate because it does not distinguish COVID-19 from other viral infections (Alghamdi et al., 2021). People without noticeable symptoms have a higher chance of getting a false negative result than people with symptoms. It means that these people who have already been infected by COVID-19 but are asymptomatic would probably be missed. Therefore, the availability of RDT test kits is limited. Another commonly used viral test is Reverse Transcription Polymerase Chain Reaction (RTPCR) which is more accurate than the RDT. It is considered the gold-standard tool for the first-line screening choice (Lai et al., 2020) since it checks for the genetic material of the virus, which can be found both when a person is infected and after acute illness. However, the RT-PCR test requires it to be conducted in a laboratory setting so that it cannot produce results as quickly as rapid tests. Since this test needs to be repeated to confirm (Dixit et al., 2021), it is more time-consuming and has a lower detection rate than RDT for the same amount of time.

As RDT requires repeated testing to confirm asymptomatic patients and RPCR is sensitive to testing conditions, we need more information to help with the rapid diagnosis. Chest X-ray (CXR) imaging is a standard and important screening tool for suspected cases of COVID-19 since the target of COVID-19 is the respiratory system. Chest X-rays have been widely used in most clinical

[1] https://covid19.who.int/

settings. In severely affected or resource-limited areas, CXR imaging is preferable for its availability, low cost and rapid results. Less time is spent on patient preparation and immediate diagnosis. Therefore, CXR can be used for patient triage, prioritization of patient treatment, and utilization of medical resources. However, due to the rapidly spreading nature of COVID-19, CXR images are insufficient for the efficiency of pandemic control and prevention. The surge in COVID-19 cases has led to a shortage of medical resources, including medical staff, equipment, and related supplies. Therefore, it is necessary to develop an automatic COVID-19 detection system to reduce the workload of first-line healthcare workers. COVID-19 pneumonia has been identified to cause characteristic findings on computed tomography that can be extended to chest X-rays, albeit to a less specific extent. Thus, the assessment of CXR can help triage patients into those with findings suspicious of COVID-19 or other virus infections. But it still needs manual intervention and experienced radiologists to assess CXR and provide a diagnosis of various thoracic lesions. To fight against this pandemic, doctors need to diagnose the infected patients quickly and put them under treatment. In this scenario, AI-based diagnostic models are applied to aid the front-line healthcare workers in screening accurate and rapid results.

In the medical imaging domain, deep learning techniques have been used to improve the performance of image analysis significantly (Altaf et al., 2019). The deep learning models are trained on the CXR images and can classify various pneumonia and normal situation of lungs with high accuracy. However, current automatic diagnosis systems based on deep learning algorithms cannot be independently applied to clinical decision-making. On the one hand, if we interpret the deep learning model by visualizing its internal representations, we find that some predictions were made outside the thoracic cavity, such as the sternum, clavicle, heart, or the background which are not related to the lung area. As we can see in Figure 1.1, significant areas which were detected by the DarkNet-53 model located in the clavicle and predictions from the rest of the models happened in the background to varying degrees. On the other hand, it is hard to trust ML-based systems for healthcare because of their inherent lack of transparency, although their results seem convincing in accuracy and reliability. For some decision situations in healthcare, empirically accurate or reliable results are sufficient. In contrast, other clinical decisions demand comprehensive insights into machine learning-generated outcomes due to their inherent normative implications (Funer, 2022). Therefore, Interpretability is needed to reduce the opacity of the machine learning model. Interpretability refers to the degree to which a human can understand the cause of a decision (Ribeiro et al., 2016). It is a crucial factor in the development of AI-based medical diagnosis systems.

A deep learning model learns by looking for patterns among massive amounts of data by convolutional neural networks (CNNs). To gain better interpretability, we can visualize the internal representations of the deep learning model to understand the decision-making process of the model. In this thesis, we utilize class activation maps to observe the significant regions where the model predicts with. Since we work on chest X-ray images with lung infections, our goal is to highlight the discriminative regions more precisely located within the lungs rather than other areas. Based on this goal, we propose three novel and simple extensions to the class activation map-based approach for lesion localization on chest X-rays with image-level supervision. They are exponential class activation map (exponential CAM), proportional class activation map (proportional CAM), and probabilistic class activation map (probabilistic CAM) respectively. Exponential CAM assigns exponential weights to the class activation map to emphasize the importance of the lesion. Proportional CAM assigns weights to the class activation map based on the proportion of the lesion. Probabilistic CAM incorporates non-dominant class activation maps to generate a probability map for the lesion.

To observe the generalizability of the visualization approaches, we focus not only on COVID-19 but also consider other bacterial and viral lung infections. Therefore, we extend our task to 5 categories of classification tasks, including normal, bacteria, other viruses (excluding COVID-

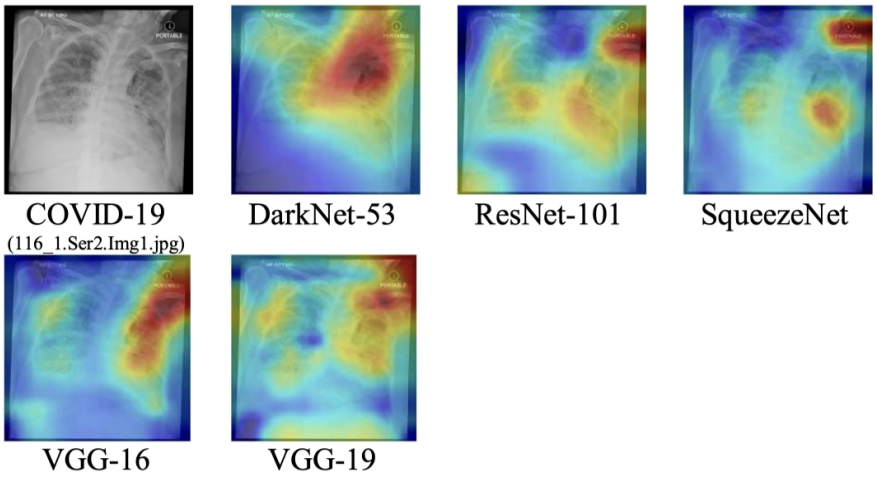


Figure 1.1: Partial results from (Tang et al., 2021). Dark red regions highlight the discriminative area used for image classification.

19 and SARS), COVID-19, and SARS. We test our approaches among various settings of deep learning models. Furthermore, to intuitively compare the performance of various visualization methods, we come up with a quantitative metric, namely acceptable mask ratio, to evaluate the performance among the proposed methods, class activation mapping (Zhou et al., 2016) and Gradient-weighted Class Activation Mapping (Grad-CAM) (Selvaraju et al., 2017).

1.2 Overview

This thesis is divided into the following chapters:

1. **Introduction** In this part, we first describe the motivation of this thesis. Then, the basic terminologies, notations, and concepts that are used across the thesis are introduced. In addition, we present the scope of this work in this chapter.

2. **Related Work** In this part, we provide a detailed review of deep learning applications for COVID-19 CXR image analysis. We review the existing visualization methods for deep learning models and how these methods apply to the COVID-19 CXR image analysis.

3. **Background** We briefly describe the background in this chapter. Firstly, we introduce the image classification task and related machine learning knowledge. Then, the visualization techniques are introduced for model interpretability.

4. **Methodology** The methodologies of this work are described in detail. We start from the dataset collection. And then, we present the model architecture for classification tasks. In addition, we propose our visualization approaches for model interpretability. Finally, we describe associated quantitative metrics that evaluate the performance of models and various visualization approaches.

5. **Experiments and results** In this part, we describe the design, detailed setting, and implementation of experiments. Afterward, we summarize the results for the classification and

visualization of the experiments. Finally, the statistical analysis based on the results is followed.

6. **Discussion** We discuss the observations from the experiments.
7. **Conclusions and Future Work** In the end, we conclude the main contributions of this work and discuss potential future works.

1.3 Scope

This thesis contains mainly two parts. The first part is to build up the image multi-class classification models. The other part is to decipher deep learning models' internals via visualization techniques. In this thesis,

1. We mainly follow and fine-tune the existing architecture of transfer learning models for the first part.
2. We primarily focus on visualization techniques. We propose Exponential CAM and Probabilistic CAM to interpret the models from different aspects.
3. We also introduce the associated metric named Acceptable Mask Ratio to verify the performance of models' interpretability quantitatively.

We try to answer the following questions in this thesis:

1. To what extent, the classification results of machine learning models on CXR images are reasonable?
2. What are the factors that affect the interpretability of machine learning models on CXR images? To what extent do different architectures of CNN affect the interpretability of machine learning models? To what extent do different global pooling layers affect the visualization of machine learning models?

Chapter 2

Related Work

In response to developing an automated diagnosis system, artificial intelligence technologies such as deep learning are promising options for automated diagnosis, as they achieve state-of-the-art performance in visual information and analysis of a wide range of medical images. In this chapter, we review the recent advances in the application of deep learning to COVID-19 detection and localization in CXR images. We first discuss the classification of COVID-19 cases based on CXR images and then focus on the localization of COVID-19 lesions in CXR images.

The COVID-19 detection tasks mainly classify the CXR images into 2–3 classes. For binary classification, it labeled the CXR images to "normal" and "COVID-19" (Tang et al., 2021). For three-class results, they contain "normal", "pneumonia" and "COVID-19" like (Hasan et al., 2021), (Ratul et al., 2020) etc,. Transfer learning has been widely applied in medical imaging applications to detect COVID-19. There are four ways of utilizing transfer learning:

A CNN pre-trained model based on Imagenet dataset (Russakovsky et al., 2015) was applied to initialize the weights of a new network that would be trained on the target CXR data (Narin et al., 2003; Ratul et al., 2020; Minaee et al., 2020).

Multiple pre-trained models were ensembled to improve the performance of the COVID-19 detection system (Hasan et al., 2021; Karim et al., 2020).

(Basu et al., 2020; Narin et al., 2003) and some studies have frozen part of the early layers of pre-trained models, where their weights are kept constant, while the final layer is fine-tuned to the radiographic data set.

Instead of leveraging pre-trained models from ImageNet, (Afshar et al., 2020) chose to pre-train a model in a similar target domain, which trained a model on a radiological dataset of pneumonia patients and non-pneumonia patients. And then, this model was further trained on COVID-19 CXR images.

For explainable localization of deep learning models, Figure 2.1 illustrates the number of studies we viewed that utilized explanatory visualization techniques. The most commonly used method was Gradient-weighted Class Activation Mapping (Grad-CAM), followed by the Class Activation Mapping (CAM) method.

During the cases using Grad-CAM and CAM, they visualized the heatmap of the last convolutional layer to have a look at the localization. However, researchers only focus on the application of Grad-CAM and CAM rather than make improvements to the visualization methods themselves.

(Karim et al., 2020) combined and analyzed the results of gradient-guided class activation maps (Grad-CAM++) (Chattopadhay et al., 2018) and layer-wise relevance propagation (LRP). The LRP propagates the output back to the input layer using the activation created by the network weights and forward propagation. As a result, the pixels that contribute to the output can be visualized (Montavon et al., 2019). Karim et al. found that the LRP method was more precise than Grad-CAM++ in terms of the localization of COVID-19 lesions. However, it failed to provide

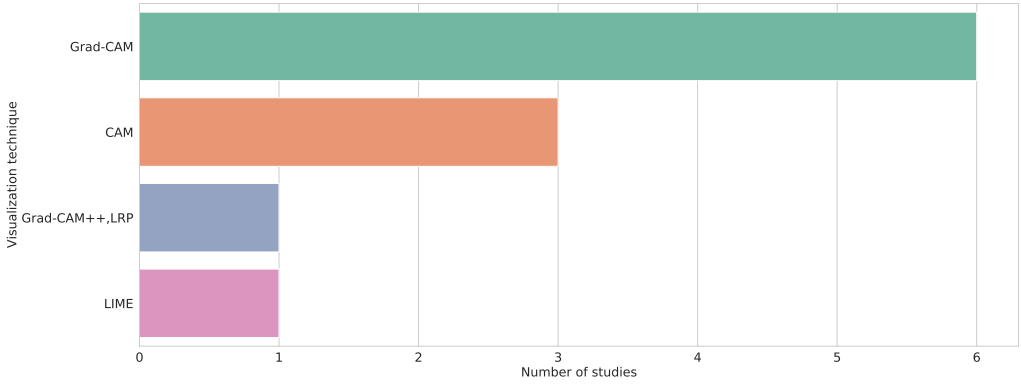


Figure 2.1: Number of Studies Utilizing Explanatory Techniques for COVID-19 Detection in CXR Images among literature Reviewed

attention to critical regions.

(Dixit et al., 2021) used Local Interpretable Model-Agnostic Explanations (LIME) to explain the predictability of models. LIME is a local explanation method that can explain the classification decisions of a black-box classifier. However, the LIME method is not suitable for the COVID-19 detection task for two reasons. First, the COVID-19 detection model is a multi-class classification model, and the LIME method is only suitable for binary classification models. In addition, the LIME method is a local explanation method. Instead of training a global surrogate model, LIME focuses on training local surrogate models to explain individual predictions. It cannot explain the global decision of the model to some extent.

Chapter 3

Background

This chapter introduces the relevant background knowledge evolving in this thesis.

3.1 Chest X-ray Images

Chest X-ray images are a standard diagnostic tool for physicians. They are used to diagnose various conditions, including pneumonia, tuberculosis, and lung cancer. The images are taken by placing the patient in front of an X-ray machine, which emits a beam of X-rays through the patient's chest. The X-rays are absorbed by the patient's body, and the resulting image is a projection of the patient's internal organs. The image is then processed to remove the background and enhance the contrast between the organs. The resulting image is a grayscale image, where the brightness of each pixel represents the amount of X-ray absorption at that location. The image is then analyzed by a physician to diagnose the patient. To cope with a pandemic, the physician may use a computer-aided diagnosis (CAD) system to assist in the diagnosis process. The CAD system can be used to detect abnormalities in the image, such as pneumonia, and to provide diagnosis advice with the help of machine learning and computer vision.

3.2 Transfer Learning

Deep learning algorithms, including convolutional neural networks (CNN) require a large amount of data for training under the assumption that the data is representative of the problem at hand. In the case of medical imaging, i.e., chest X-ray images, the data is limited due to the high privacy of acquiring the data and the radiation exposure to the patient. And in particular, the limited size of medical cohorts and the cost of expert-annotated datasets are some well-known challenges (Kim et al., 2022). Many research efforts have attempted to overcome this problem through transfer learning. These aim to achieve high performance on the target task by leveraging the knowledge learned from the source task.

Transfer learning is a machine learning method where a model trained on one task is exploited as the starting point for a model on a second task. It is usually done for tasks where your dataset has too little data to train a full-scale model from scratch. In the field of medical image analysis, transfer learning is a commonly utilized technique when developing medical imaging models. It has made a major contribution to medical imaging since it overcomes the data scarcity problem and saves time and hardware resources (Kim et al., 2022). One of the first ideas to use transfer learning is to apply pre-trained models of the ImageNet dataset (Russakovsky et al., 2015). The ImageNet dataset is a large-scale dataset of 1.2 million images with 1000 classes. This approach

is effective since the pre-trained models have been trained on a large corpus of photos and made predictions on a wide range of categories. It indicates the models efficiently learn to extract features from photos so that they can perform well on the problem. Although chest X-ray images are different from natural images, features are more generic (e.g., edges), in early layers whereas more original-dataset-specific in later layers (Yosinski et al., 2014). Therefore, the pre-trained models can be used as a starting point for the target task. There are two main approaches to transfer learning: feature extraction and fine-tuning. Feature extraction is the process of using the representations learned by a previous network to extract features from new samples. The features are then run through a new classifier, which is trained from scratch. Fine-tuning is the process of unfreezing a few of the top layers of a frozen model base and jointly training both the newly added part of the model and these top layers. The goal of fine-tuning is to adjust more abstract representations of the model in order to make them more relevant to the problem at hand. In this work, we use the fine-tuning approach to transfer the knowledge from the pre-trained models to the target task.

3.3 CNN Architecture of Pre-trained Models

ResNet-50 is a convolutional neural network architecture that contains 50 layers. It is a variant of the ResNet architecture, which was introduced by (He et al., 2016). The ResNet architecture is a residual network, which is a neural network that uses skip connections to prevent vanishing gradients. The skip connections allow the network to learn an identity function, which maps the input to the output directly, which helps a lot during the backpropagation method. The architecture of ResNet-50 consists of 16 residual blocks, and the output of the last convolutional layer is passed through a global average pooling layer and a fully connected layer.

VGG19 Batch Normalization contains 16 convolutional layers and three fully connected layers. It is a variant of the VGG architecture, which was introduced by (Simonyan and Zisserman, 2015). VGG is based on an analysis of how to increase the depth of such a network. The network utilizes a stack of convolutional layers with a small kernel size (apart from 3×3, 1×1, and 5×5 are also used) and a large number of filters. It is known for its simplicity, where the only other components are a pooling layer and a fully connected layer. The architecture of VGG19 Batch Normalization consists of 16 convolutional layers, three fully connected layers, and one output layer. The convolutional layers are followed by a max pooling layer and a batch normalization layer. The fully connected layers are followed by a dropout layer.

MobileNetV2 introduce the inverted residual with linear bottlenecks as a novel layer module. This module first expands input to high dimensions and filters features using lightweight depth-wise separable convolutions (Sandler et al., 2018). The features are then projected back to a low-dimensional representation with linear convolutions. This structure is effective in that it tradeoff accuracy while reducing the number of parameters and computations. The architecture of MobileNetV2 consists of 22 convolutional layers and one fully connected layer. The convolutional layers are followed by a batch normalization layer and a ReLU activation layer. The fully connected layer is followed by a dropout layer.

3.4 Global Pooling Methods

Convolutional neural networks summarize the presence of features in the input image. For the output feature maps generated by CNN, they are sensitive to the location of features in the input image. One way to address this sensitivity is to downsample the feature maps. Meanwhile, further operations performed to summarize features can help enhance the robustness of the model.

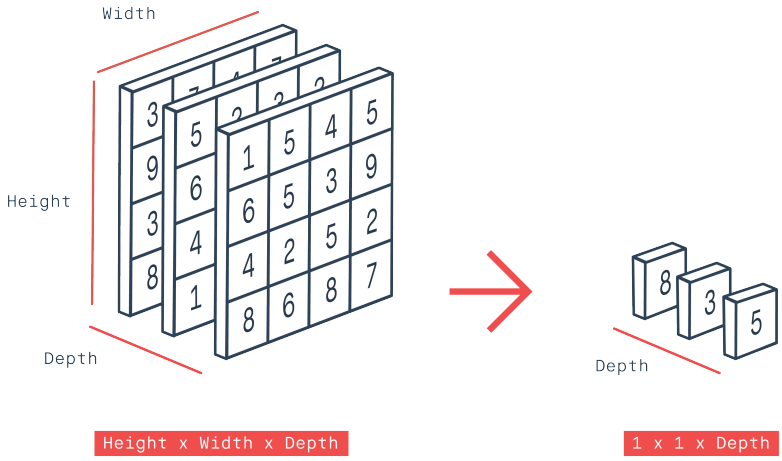


Figure 3.1: Structure of Global Pooling [1]

Pooling layers provide a way to downsample feature maps by summarizing the presence of features in feature patches. Besides, the pooling layer can reduce the dimension of the feature map. Therefore, it also reduces the number of hyperparameters to learn and the amount of computation performed in the network.

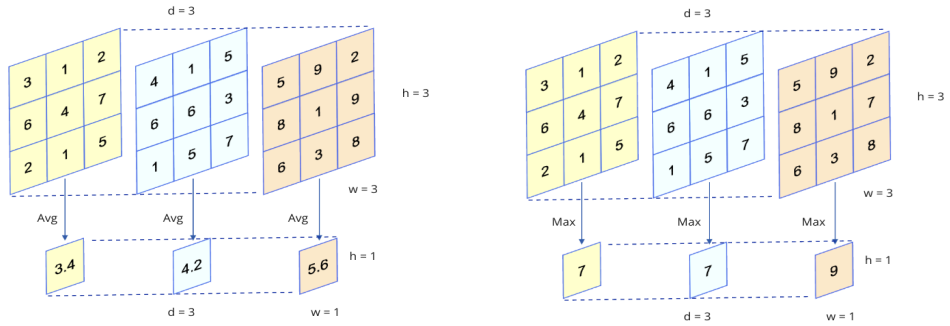
A global pooling layer is a kind of pooling layer that is designed to replace traditional fully connected layers in classic CNN architecture. It is often used in the backend of convolutional neural networks to obtain shapes suitable for dense layers. Normally, a convolutional layer with 1×1 kernel size can be used to reduce the depth to the number of classes. Therefore, flattening does not have to be applied. It considers a feature map of each channel to be each corresponding class of the classification task in the last convolutional layer (Lin et al., 2014). Instead of adding a fully connected layer on top of the feature maps, it takes one value per feature map and feeds the resulting vector directly into the softmax layer. Compared to fully connected layers, the global pooling layer is more native to convolutional structures by enforcing the correspondence between feature maps and classes (Lin et al., 2014). Therefore, feature maps can be easily interpreted as class confidence maps. Besides, there are no parameters to optimize in the global pooling, which can avoid overfitting at this layer.

Figure 3.1 shows how the global pooling method reduces the dimensionality from 3D to 1D. Global pooling outputs one response for every feature map. This response can be the average value, maximum value, or any other value depending on the pooling operation applied. Different approaches to global pooling layers emerge from it.

3.4.1 Global Average Pooling

Global average pooling is used to summarize the average presence of features in the input image. In Equation (3.1), we define p_k as the one response value of channel k. Let $f_k(x, y)$ represent the feature map of channel k in the last convolutional layer at spatial location (x, y). p_k equals to the average of the sum of the feature map $f_k(x, y)$. W and H indicate the width and height of

[1] https://peltarion.com/knowledge-center/modeling-view/build-an-ai-model/blocks/global-average-pooling-2d



(a) Global Average Pooling

(b) Global Max Pooling

Figure 3.2: DIFFERENT GLOBAL POOLING EXAMPLES. (a) Pipeline of Global Average Pooling; (b) Pipeline of Global Max Pooling

the feature map, respectively. According to Figure 3.2(a), global average pooling calculates the average value for each feature map channel by channel. Then, the values computed from all the channels generate the feature embedding with the length of channels.

$$p_k = \sum_{x,y} \frac{1}{W \cdot H} f_k(x, y) \tag{3.1}$$

3.4.2 Global Max Pooling

Global max pooling is used to summarize the most activated presence of features in the input image. In Equation (3.2), p_k is the maximum value of $f_k(x, y)$. Similar to global average pooling, global max pooling finds the maximum value for each feature map channel by channel. Then, the values computed from all the channels generate the resulting vector with the length of channels (Figure 3.2(b)).

$$p_k = \max_{x,y} f_k(x, y) \tag{3.2}$$

3.4.3 Global Exponential Pooling

Global exponential pooling applies the exponential softmax function (Salamon et al., 2017) (Equation (3.3)) to assign a weight of $exp(f_k(x, y))$ to the element-level probability of feature maps (Wang et al., 2019). It is used to summarize the presence of features in the input image.

$$p_k = \sum_{x,y} \frac{\exp(f_k(x, y))}{\sum_{x,y} \exp(f_k(x, y))} f_k(x, y) \tag{3.3}$$

3.4.4 Global Linear Pooling

Global linear pooling leverages the linear softmax pooling function (Wang et al., 2019) (Equation (3.4)) to assign weights equal to the element-level probability of feature maps themselves. As a result, the larger $f_k(x, y)$ will be boosted, while the smaller one will be suppressed.

$$p_k = \sum_{x,y} \frac{f_k(x,y)}{\sum_{x,y} f_k(x,y)} f_k(x,y) \tag{3.4}$$

3.4.5 Global Log-sum-exp Pooling

Global log-sum-exp pooling uses the log-sum-exp function to summarize the presence of features. The log-sum-exp function is a smooth version and convex approximation of the max function (Pinheiro and Collobert, 2015). It is defined as the logarithm of the sum of the exponentials of the arguments. In Equation (3.5), p_k is the log-sum-exp value of $f_k(x,y)$. The hyperparameter γ controls how smooth the approximation is: a high value of γ means an effect similar to the maximum value, while a very low value will have an effect similar to the average of the scores. The advantage of this aggregation is that pixels with similar scores will have similar weights during training, γ controls the concept of similarity (Pinheiro and Collobert, 2015).

$$p_k = \frac{1}{\gamma} \log(\frac{1}{x \cdot y} \sum_{x,y} \exp(\gamma f_k(x,y))) \tag{3.5}$$

3.5 Model Interpretability

3.5.1 Class Activation Mapping

Class Activation Map (CAM) is a technique to visualize the activation of a CNN's last convolutional layer, allowing us to see the network is interested in which part of the image. The CAM for a particular category indicates the discriminative image regions used by CNN to identify that category (Zhou et al., 2016). According to Equation (3.6), M_c indicates the class activation map for class c and w_k^c is the weight corresponding to class c for channels k from the last convolutional layer with 1×1 kernel of the model which uses to reduce the channel number to class number. CAM is generated by multiplying each depth from the feature maps of the final convolutional layer of the backbone CNN network with the corresponding weight connected to the predicted class and summing them up. The global pooling layer summarizes the activation of the last convolutional layer. The weights of the global pooling layer are learned during the training process. Therefore, the CAM can be used to understand the decision made by the model. The resulting heatmap is then resized to the size of the input image, visualizing the discriminative image regions used by CNN to identify that category.

$$M_c(x,y) = \sum_k w_k^c f_k(x,y) \tag{3.6}$$

3.5.2 Gradient-weighted Class Activation Mapping

Gradient-weighted Class Activation Mapping (Grad-CAM) uses the gradient information of the target category flowing into the last convolutional layer to generate a coarse localization map (Selvaraju et al., 2017). This localization map is used to highlight the important regions of the input image for the classification task. The Grad-CAM method is an advancement of the CAM method (Selvaraju et al., 2017). Compared with CAM limiting to using after global pooling layer, Grad-CAM is applicable to a wide variety of CNN architecture. Therefore, it is widely used in the field of medical image analysis.

The Grad-CAM method is based on Equation (3.8). In order to obtain the class-discriminative localization map Grad-CAM (L_c), it first computes the gradient of the output score y_c (before

softmax function) for class c with respect to the feature map activations $f_k(x, y)$ of a convolutional layer, i.e. $\frac{\partial y_c}{\partial f_k(x,y)}$. These gradients flowing back are global-pooled over the spatial locations (x and y respectively) to obtain the neuron importance weights α_k^c (Equation (3.7)). The weights α_k^c apply to the feature map activations $f_k(x, y)$ for each channel. Then the results are linearly combined and followed by a ReLU to obtain L_c. It applies ReLU since it is only interested in the features that positively influence the class of interest, i.e., pixels whose intensity should be increased to increase y_c. Negative pixels likely belong to other categories in the image (Selvaraju et al., 2017).

$$\alpha_k^c = \frac{1}{W * H} \sum_{x,y} \frac{\partial y_c}{\partial f_k(x, y)} \tag{3.7}$$

$$L_c = ReLU(\sum_k \alpha_k^c f_k(x, y)) \tag{3.8}$$

Chapter 4

Methodology

4.1 Network Architecture

In this section, we first describe the network architecture of our multi-class classification models for the training and testing phases. Then, we describe the pipeline of feature visualization for the models.

4.1.1 Training and Testing

In this study, we apply three state-of-art CNN models: ResNet-50, MobileNet V2, and VGG-19 as the backbone CNN network. Since these CNN models have been pre-trained on the 1000-class Imagenet dataset (Russakovsky et al., 2015) for classification, we use finetuning transfer learning to build up our models.

In our method, we first train our models on our CXR images with the architecture shown in Figure 4.1. We initialize the network with a pretrained network as the backbone CNN network and remove its last fully connected layer. After that, we add the global pooling layer to reduce the dimension of the input feature map to a single dimension. For the global pooling layer, we apply five types of methods: global average pooling, global max pooling, global exponential pooling, global linear pooling, and global LSE pooling. We then add a convolutional layer with a 1×1 size of the kernel to reduce the depth to five classes, namely normal, bacteria, virus, COVID-19, and SARS, in our CXR dataset for multiclass classification. Finally, we apply the softmax function to the output of the convolutional layer to get the probability of each class.

We split the dataset into a training set, a validation set, and a test set. We will describe the details of them in the next chapter. In the training phase, we retrain our models on the training set along with validation after each training epoch. After finishing training, we test our models on the test set to get the classification result for each image.

4.1.2 Visualization

To interpret and understand our models, we visualize the internal representations learned by CNNs. The visualization has been proposed after the training process completes and all parameters are fixed. It is a heat mapping process that is applied in an already trained neural network. We mainly have two kinds of pipelines for feature visualization of the models.

The first kind of pipeline is visualization using Grad-CAM. As Figure 4.2 shows, we first map the input image to the activations of the last convolutional layer as well as the output predictions. Then, we compute the gradient of the top predicted class for our input image concerning the

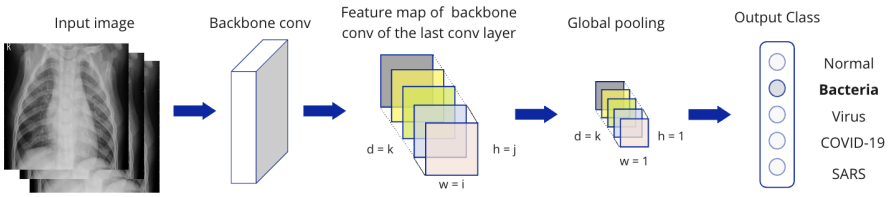


Figure 4.1: The Architecture of Classification Model

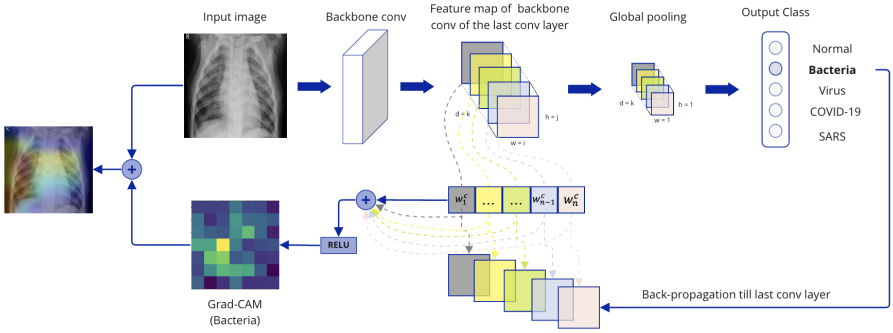


Figure 4.2: Pipeline of Visualization Using Grad-CAM

activations of the last convolutional layer. We weight each channel in the feature map by corresponding gradients and sum all the channels. Then ReLU activation function is applied on top to obtain the heatmap class activation.

The second kind of pipeline structure is applied to visualization using CAM, exponential CAM, proportional CAM, and probabilistic CAM. As Figure 4.3 illustrates, we first evaluate the input image to obtain the top predicted class. Then, we weight the top predicted class activation by the input image to obtain the class activation maps. During this process, we apply different operations on the weighted sum to implement exponential CAM. For the exponential CAM, we apply the exponential function to the weighted sum. For the probabilistic CAM (Figure 4.4), we consider not only the top predicted class but also the other non-dominant categories. We then weight the probabilities by various class activation maps to compute the probabilistic class activation map.

4.2 Lesion localization

In this section, we describe the definition of our proposed methods. The visual discrimination areas vary greatly as models for the existing visualization methods CAM and Grad-CAM. Meanwhile, they highlight irrelevant areas outside the lungs. Based on the above issues, we would like to propose new methods that can be more stable among various model architectures and localize the lesions more precisely. We have two ways to achieve this goal. The first way is to enhance the difference between discriminative and non-discriminative areas. We can highlight the significant regions by filtering out the edge values. This enhancement is applied to the top predicted class classified by the model. Exponential CAM and proportional CAM belong to this kind of idea. The second way is to consider the top predicted class and the other non-dominant categories at

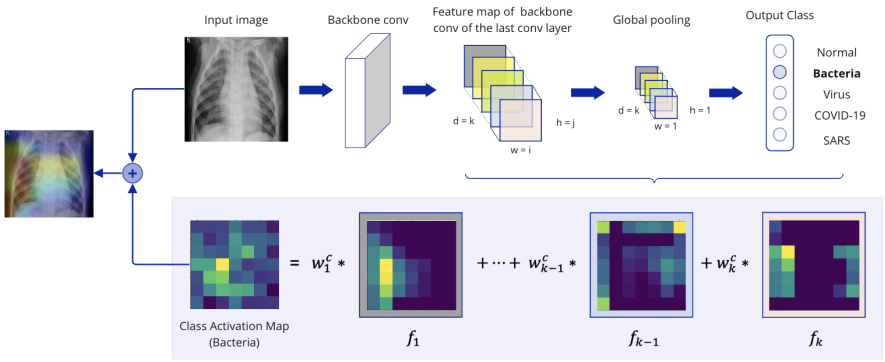


Figure 4.3: Pipeline of Visualization Using CAM

the same time. It aims to incorporating more information from similar categories, especially for minority classes. Probabilistic CAM belongs to this kind of idea.

4.2.1 Exponential CAM

Exponential CAM is a new method that we propose to visualize the internal representations of the models. The idea of exponential CAM is to emphasize the regions that are more important for the classification. Based on CAM, we apply the exponential function for each channel to the weighted sum of the top predicted class activation by the input image to obtain the exponential class activation maps (Equation (4.1)).

$$M_c(x, y) = \sum_k \exp\left(w_k^c f_k(x, y)\right) \tag{4.1}$$

4.2.2 Proportional CAM

Proportional CAM applies linear weighting to the top predicted class activation by the input image to obtain the proportional class activation maps. Equation (4.2) leverages the proportion of each element to the total number in the feature map to reflect the overall structure to compute the final class activation map. We applied the operation to each channel and then sum all the channels to obtain the final class activation map.

$$M_c(x, y) = \sum_k \frac{w_k^c * f_k(x, y)}{|\sum_{x,y} f_k(x, y)|} \tag{4.2}$$

4.2.3 Probabilistic CAM

Compared with CAM and Grad-CAM only considering the top predicted class, probabilistic CAM incorporates the top predicted class and the other non-dominant categories, which observe the contribution of edge information from non-dominant classes. In Equation (4.5), P_c indicates the probability of class c generated by the softmax function (Equation (4.4)) in output neuron (Equation (4.3)). We compute the class activation maps from each category. Then, we weigh the probabilities by various class activation maps to calculate the probabilistic class activation map.

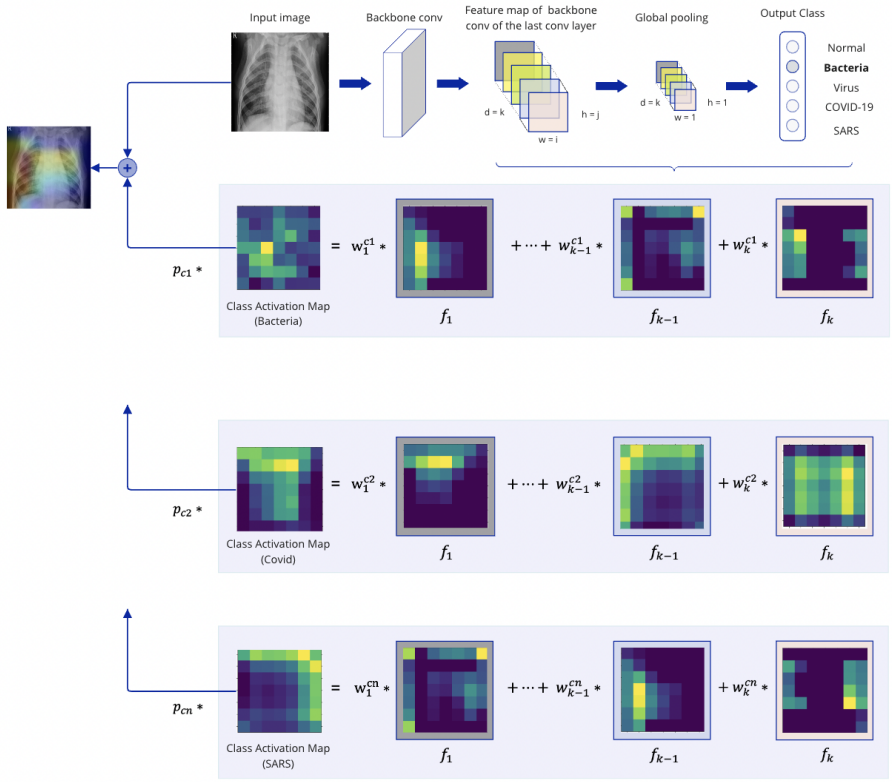


Figure 4.4: Pipeline of Visualization Using Probabilistic CAM

$$P_c = \frac{\exp(S_c)}{\sum_c \exp(S_c)} \tag{4.3}$$

$$S_c = \sum_{x,y} \sum_k w_k^c f_k(x,y) \tag{4.4}$$

$$M(x,y) = \sum_c \sum_k P_c \cdot w_k^c f_k(x,y) \tag{4.5}$$

4.3 Evaluation Metric of Lesion Localization

In this thesis, we proposed a quantitative evaluation metric named acceptable mask ratio to assess lesion localization accuracy.

4.3.1 Acceptable Mask Ratio

The acceptable mask ratio is based on the idea of how many significant lesions detected by the model are located within the acceptable mask region. We define the acceptable mask region as the region of the lung in CXR images. And the significant lesion is defined as the class-specific

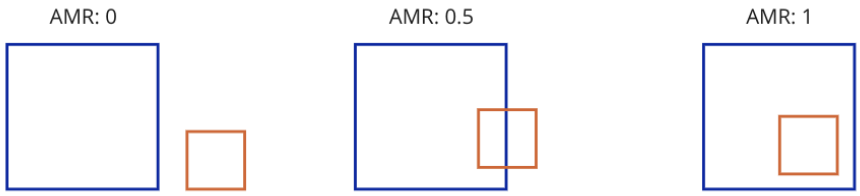


Figure 4.5: Examples of AMRs. The blue box is a masked area. The red box is the discriminative area.

discriminative regions highlighted by the model. For the sake of brevity, in the following description, we will name the acceptable mask region as the masked area and designate the significant lesion as the discriminative area. The acceptable mask ratio aims to compute the percentage of discriminative area positioned within the masked region. As (Equation (4.6)) shows, it calculates the ratio of the intersection of the discriminative area and the masked area over the discriminative area itself.

$$\text{Acceptable Mask Ratio} = \frac{\#(\text{Discriminative area} \cap \text{Masked area})}{\#\text{Discriminative area}} \tag{4.6}$$

In our image classification task, we calculate the pixels of images as the area. For a given image, we compute the acceptable mask ratio (AMR) as Equation (4.7):

$$0 \leq \text{AMR} = \frac{\sum_i^j (D_i * M_i)}{\sum_i^k D_i} \leq 1 \tag{4.7}$$

- We define a filter mask that is applied to the image to mask the unwanted regions. Within the lung area, the value of the masked filter equals 1; otherwise, it is 0.
- Let D_i indicates the pixel of the discriminative area, and M_i indicates the item of the filter mask.
- We count the number of pixels within the area of overlap between the discriminative area and the masked area. This is the area of intersection area.
- The number of pixels in the discriminative area is called the area of the discriminative area.
- The acceptable mask ratio is computed as the intersection area divided by the area of the discriminative area.
- The acceptable mask ratio should be within the range of [0, 1].

In Figure 4.5, we include examples of various values of AMR. In our cases, we consider discriminative locations within the masked region as justified decisions. If the AMR is close to 0, it means few predictions have been made within the lung region. We consider the results to be less reasonable. In contrast, if AMR equals 1, it indicates the model making decisions for classification within the lung region. We consider the results justified. AMR closer to 1 means better lesion localization accuracy.

Chapter 5

Experiments and results

5.1 Experiment Setup

5.1.1 Dataset

We acquired the Chest X-Rays (CXR) images from several public domains, including various views or projections of the chest and lung. They contain the posteroanterior (PA), anteroposterior (AP), normal lateral, and cross-sectional views. In this study, we select the PA and AP Chest X-ray images for multi-class classification. Figure 5.1 shows some images from different patients with normal lungs, infected by bacteria, and diagnosed with COVID-19.

We combine the CXR images from the following public datasets:

- (i.) CoronaHack Chest X-ray dataset [1] that contains publicly available chest X-rays of healthy and various pneumonia-affected patients. This dataset collected 5910 images from different open resources and consisted of seven classes: Bacteria (2772 images), Normal (1576 images), Virus (1493 images), Covid-19 (58 images), Streptococcus (5 images), SARS [2] (4 images), ARDS [3] (2 images).
- (ii.) Figure 1 COVID-19 Chest X-Ray Dataset [4] have 55 chest X-rays images from Covid-19 patients.
- (iii.) We also obtain Covid-19 (423 images) and SARS (134 images) from COVID-19, SARS, MERS X-ray Images Dataset [5].

For this study, we removed Streptococcus and ARDS classes due to a very small number of images. Therefore, we worked with the following five classes: normal, bacteria, virus (excluding the viral infections of COVID-19 and SARS), COVID-19, and SARS. Table 5.1 shows the number of images in each class.

Split the dataset for training

Chest X-Ray images are split into training and testing sets. Among the training set, we split 10% from it as the validation set. The training set is used to train the network. The validation set is

[1] https://www.kaggle.com/datasets/praveengovi/coronahack-chest-xraydataset
[2] Severe Acute Respiratory Syndrome
[3] Acute Respiratory Distress Syndrome
[4] https://github.com/agchung/Figure1-COVID-chestxray-dataset
[5] https://www.kaggle.com/datasets/057e1b6dc41d9691e59dded4445fa8cc2f0b4b5cbcb49aef9583d95233799d5a

(a) Normal (b) Bacteria (c) COVID-19

Figure 5.1: CXR IMAGES EXAMPLES. Chest X-Ray images for (a) a normal lung, (b) a lung infected by bacteria, and (c) a patient diagnosed with COVID-19 from CoronaHack Chest X-ray dataset.

Normal	Bacteria	Virus	COVID-19	SARS
1576	2772	1493	536	138

Table 5.1: Dataset Summary

used to evaluate the model performance during training. And the testing set is used to evaluate the network after finishing the training. The detailed numbers of each dataset for each class are shown in Table 5.2. Each category contains more than one CXR image from the same patient. Although the dataset does not contain the private information of patients, it uses person numbers to distinguish the origin of the images. We make sure that the images in the training set and the testing set are from different patients in each class.

	Training	Validation	Testing
Normal	1208	134	234
Bacteria	2277	253	242
Virus	1211	134	148
Covid-19	438	48	50
SARS	111	12	15
In total	5245	581	689

Table 5.2: The Number of Images in Each Set

5.1.2 Preprocessing

Before we feed our gathered image data into our model architecture, we need to preprocess the data. It is to ensure that the images are compatible with the pre-trained model. The preprocessing steps are as follows:

1. **Resize**: Before using the pre-trained models, we usually resize the images with the right resolution. It is recommended to rescale images to the same size as the pre-trained models demand. In our cases, we resize all images in the dataset to 224×224 pixels.

2. **Interpolation**: Interpolation works by using known data to estimate values at unknown points. In our scenario, there are a number of images that are less than 224×224 in size. Thus, we need to increase the resolution of images and retain quality efficiently at the same time. For our preprocessing, we use bilinear interpolation to resize the images. Bilinear interpolation is a type of interpolation that uses linear interpolation to produce a smooth transition between pixels. It calculates a weighted average of the values of the four corresponding pixels based on distances and applies it to approximate the output pixel (Khosravi and Samadi, 2021).

3. **Data Augmentation**: To improve and generalize the ability of deep learning models, we perform data augmentation on the training data. This is done by randomly flipping the image horizontally. Affine transformations are also used while keeping the center invariant. We apply a shear parallel to the x-axis in the degree range $(-10, +10)$, and also set a scaling factor interval. Then we randomly enlarge or shrink the images by a scale factor from the range $[0.8, 1.2]$ while keeping the original scale. Besides, we also randomly rotate the images by a degree from the range $(-10, +10)$.

4. **Normalization**: Normalization is a technique that normalizes the data to a range of values between 0 and 1. It can reduce skewness which helps learn faster and better. Although the CXR images are in gray level, they have three RGB channels, and all three channels are identical. Besides, it is good to have similar preprocessing as the pre-trained models. Thus, we use the mean and standard deviation of ImageNet (Russakovsky et al., 2015) to normalize the images as the pre-trained models applied.

5.1.3 Network construction and Training

Figure 4.1 shows the overview structure of our network. We apply three widely used pre-trained models for the backbone convolutional neural network: ResNet-50, VGG19 Batch Normalization, and MobileNet V2. In the part of global pooling layers, we provide five operations: average pooling, max pooling, exponential pooling, linear pooling, and log-sum-exp pooling. With different settings, we train 15 models in total. For lesion localization, since we use the weights summarized by the global pooling layer to generate the class activation maps, we implement various pooling operations to observe how these pooling operations affect the performance in localization. Besides, we also want to evaluate if our new visualization approaches are able to perform among models in general.

5.1.4 Configuration of Experiments

Sample Weights

There is an inherent assumption that machine learning algorithms are based on balanced data, which indicates the data is equally distributed across all its categories. However, in practice, class imbalance problem is frequently observed in various deep learning classification tasks. It refers to the highly imbalanced frequency of the target class, i.e., one or more classes appear very frequently compared to the other classes. When training a model on an imbalanced dataset, there is a bias or skewness towards the majority of classes present in the target. As the number of samples available to learn increases, the model learns to perform well in the majority class (Wang and Japkowicz, 2010). While lacking enough samples, the model fails to learn meaningful patterns for the minority class effectively. As shown in Figure 5.2, this training dataset is imbalanced.

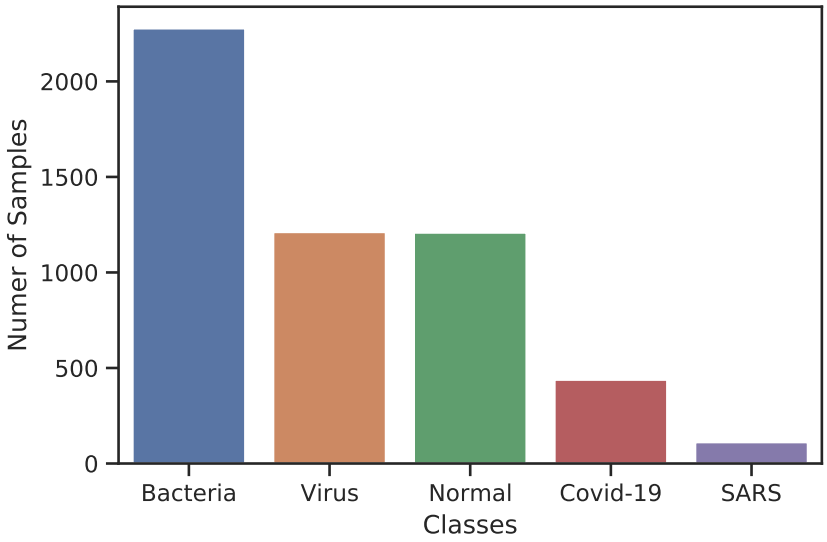


Figure 5.2: Distribution of Training Dataset

Bacteria, virus, and normal are majority classes, while Covid-19 and SARS can be considered minority classes.

For the problem of class imbalance, the common approach is to undersample the majority class or oversample the minority class. That is to delete a certain number of samples related to the majority class or repeat the samples associated with the minority class (Kim et al., 2015). Although either of these two strategies can balance the dataset, it does not directly address this problem. Furthermore, it raises a risk of introducing new issues. On the one hand, undersampling may cause the model to miss learning critical patterns that might have been learned from the removed samples. On the other hand, since oversampling introduces duplicate samples, it can quickly slow down training and cause the model to overfit.

We introduce sample weights into the loss function to overcome the above problems. The sample weights in the loss function independently weight the loss computed by the sample according to whether the sample belongs to the majority class or the minority class. Essentially, it assigns higher weights to the losses encountered by samples correlated with minority classes. It means that different weights are applied to the losses calculated for different samples according to the categories.

In this study, we use the Inverse of Number of Samples (INS) as the weighting scheme. INS weights the samples as the inverse of the class frequency. It define as follows (5.1). For an imbalanced dataset, let λ_t indicate the sample weights per target class and N_t is the number of samples in class t.

$$\lambda_t = \frac{1}{N_t} \tag{5.1}$$

In our experiments, we define the sample weights for each class which would be inversely proportional to the number of samples for each class and normalizes them over classes for training data. For validation and test data, we do not need to balance a batch.

Early Stopping

In machine learning, too many epochs can lead to overfitting the training dataset, while too few can lead to underfitting the model. Early stopping is a form of regularization used to avoid overfitting when training a model with an iterative method. It allows specifying an arbitrarily large number of training epochs and stops training when the model performance stops improving on a holdout validation dataset. During training, early stopping compares the performance or loss of the model on the validation set after each epoch. If the performance, i.e., accuracy of the model on the validation dataset, starts to decrease, then the training process is stopped immediately or after a certain number of epochs. Or validation loss increases for several epochs in a row, and then the training process is stopped. In our experiment, we use the validation loss as our trigger. There are the following processes to implement early stopping.

First, we monitor the model performance. The performance of the model needs to be monitored during training. We use the validation set to monitor the model's performance during training or the loss of the validation dataset. The evaluation of the validation set is executed at the end of each epoch.

Then, we set up a trigger to stop the training process. Triggers use the monitored performance metrics to decide when to stop training. In general, training stops as soon as the performance of the validation dataset degrades, i.e., loss increases, compared to the performance of the validation dataset in the previous training epoch. In practice, more elaborate triggers are introduced since the training of neural networks can be stochastic and potentially noisy. The performance on the validation dataset can fluctuate many times. It is sound to consider some delays or patience before stopping. On average, slower criteria that stop later than others lead to improved generalization ability compared to faster ones (Prechelt, 1998). Therefore, when setting triggers, patience can be used to observe performance degradation for a given number of epochs.

Finally, we save the model with consideration. During training, it needs to consider which model to save. Every time the performance on the validation set improves, a copy of the model parameters can be stored (Goodfellow et al., 2016). When the training terminates, these parameters are returned rather than the latest parameters (Goodfellow et al., 2016). If the trigger observes performance degradation for a fixed number of epochs, then the model is preferred at the beginning of the trigger period.

In our experiments, we compute validation loss after each epoch. If the validation loss increases for five epochs in a row, then the training process is stopped. We save the model with the lowest validation loss.

Hyper-parameter Setting

In order to control experiments, we proposed the same setting of hyper-parameters for various classification models. Besides, the INS weight initialization technique has been used to tackle the class imbalance problem. For the model training, we apply pre-trained deep learning models previously trained with ImageNet Russakovsky et al. (2015) dataset. The other hyper-parameters are included in Table 5.3.

Hardware and Framework

The classification model's training, validation and testing were performed on the Debian Linux server, which has 8 GeForce RTX™ 2080 Ti Turbo 11G GPUs designed for deep learning, 128 CPU cores and 512 GB of memory. The implementations were written in python programming language (Van Rossum and Drake, 2009) with Pytorch (Paszke et al., 2019) which is an open source machine learning framework. And OpenCV (Bradski, 2000) is used to highlight the lung area by adding bounding boxes.

Parameter	Value
Learning rate	1e-3
Batch size	32
Optimizer	SGD (with $momentum = 0.9$)
Criterion	Cross-entropy Loss
Maximum epoch	80
Early stopping patience	5

Table 5.3: Summary of Defined Parameter of Proposed Models

5.2 Quantitative Analysis

In this section, we present the quantitative analysis of the classification model and the proposed visualization methods. We first show the performance of multiclass classification models. Then, we visualize the models with our proposed methods, CAM and Grad-CAM, respectively. We present and compare the results of the proposed methods with CAM and Grad-CAM. Finally, we discuss the results and analyze the performance of the proposed methods.

5.2.1 Evaluation Metrics for Classification

In this part, we use the following evaluation metrics to measure the quality of the multi-class classification models.

Confusion Matrix

In the field of machine learning, a confusion matrix, also known as an error matrix (Stehman, 1997), is a specific table that uses to visualize and summarizes the performance of a classification algorithm, typically a supervised learning one. An example of a confusion matrix for binary classification is shown in Figure 5.3. There are four basic characteristics of a confusion matrix that are defined to assess the classifier.

- **True Positive (TP)** represents the number of positive examples that are indeed positive and are properly classified.
- **True Negative (TN)** represents the number of negative examples classified accurately.
- **False Positive (FP)** represents the number of actual negative examples classified as positive.
- **False Negative (FN)** represents the number of actual positive examples classified as negative.

Evaluation metrics to perform the classification, like accuracy, precision, recall, and F1-score, are calculated based on the above-stated TP, TN, FP, and FN.

F1-score

F1-score is useful for imbalanced classification since it takes into account how data is distributed. The F1-score considers the precision and recall of a classifier into a single metric by taking their harmonic mean.

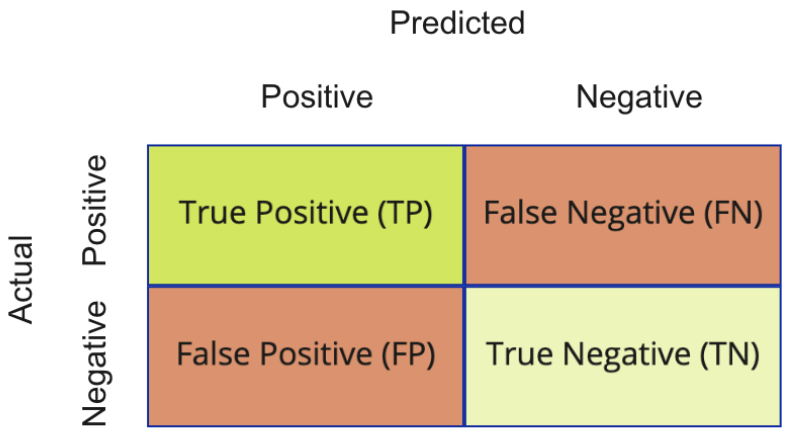


Figure 5.3: Confusion matrix for Binary Classification

- **Precision** quantifies the ratio of true positives among all the positives predicted by the model (Equation (5.2)), including those not identified correctly. It measures the accuracy of a model in classifying a sample as positive. It is useful for the skewed and imbalanced dataset. The more False positives the model predicts, the lower the precision.
- **Recall** (also known as True Positive Rate or sensitivity) summarizes the proportion of actual positives that are correctly classified (Equation (5.3)). It measures the ability of a model to detect positive samples when the actual outcome is positive. The more false negatives the model predicts, the lower the recall.
- **F1-score** is the harmonic mean of the precision and recall (Equation (5.4)). It weights precision and recall equally, which is the most commonly used variant when learning from imbalanced data (Haibo and Yunqian, 2013). F1-score is within the range of [0, 1]. If it is closer to 1, it indicates good precision and recall. If it is 0, precision or recall is 0.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5.2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5.3}$$

$$\text{F1} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} = \frac{TP}{TP + \frac{1}{2} \cdot (FP + FN)} \tag{5.4}$$

Balanced Accuracy for Multi-class Classification

Since our dataset is imbalanced, we use balanced accuracy to evaluate the performance of the models. Compared to accuracy which is a useful measure of a similar balance in the dataset, balanced accuracy is a more appropriate measure for imbalanced datasets. The balanced accuracy is defined as the average of the recall (Equation (5.3)) obtained on each class. The recall value for each category assesses the ability of individuals in that category to be correctly classified.

Thus, balanced accuracy provides an average measure of this concept across different classes. In Equation (5.5), C is the number of classes. The balanced accuracy is calculated as follows:

$$\text{Balanced Accuracy} = \frac{1}{C}\sum_{i=1}^{C} Recall_i \tag{5.5}$$

ROC Curve and AUC

A Receiver Operating Characteristic curve (ROC curve) is a diagnostic graph to show the performance of a classification model. It visualizes the trade-off between the True Positive Rate (TPR or Recall) (Equation (5.3)) and False Positive Rate(FPR) (Equation (5.6)) for a set of predictions at different decision thresholds. False Positive Rate corresponds to the proportion of all negatives that are mistaken for positives. It is also called the false alarm rate since it summarizes how often a negative class is predicted to be positive. The threshold is to decide whether a prediction is labeled true or false and control the tradeoff between TPR and FPR. As Figure 5.4(a) shows, each threshold is a point on the plot that connects to form a curve. A no-skill classifier or random guessing classifier, i.e., predicting the majority class under all thresholds, will be represented by a diagonal line from the bottom left to the top right. Points below this diagonal line are worse than no skill. Any models with a ROC curve below this line can be completely rejected. In contrast, a model with a ROC curve above this line can be considered a better model. A perfect skill model with a high value of TPR and a low value of FPR will be a point at the top left of the plot.

Area Under the Curve (AUC) is used as a summary of the ROC curve which represents the degree or measure of separability between the positive and negative classes. It is the area under the ROC curve. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.

In summary, the ROC curve of various models can compare straightforwardly at different thresholds. And the AUC can be used as a summary of the model skill, which essentially averages diagnostic accuracy across the spectrum of test values Thus, the ROC Curve and AUC are helpful diagnostic tools for the model.

$$\text{False Positive Rate} = \frac{FP}{FP + TN} \tag{5.6}$$

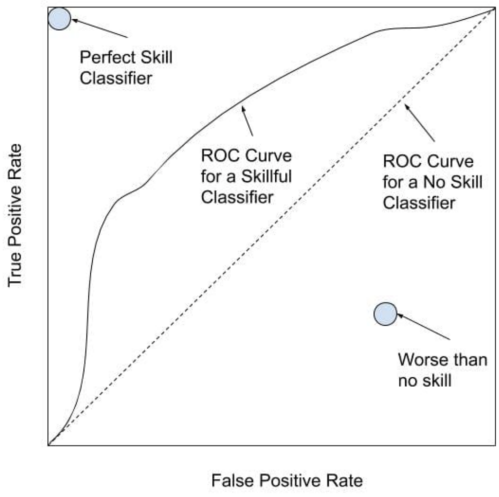
Equal Error Rate

The equal Error Rate (EER) threshold is the threshold optimized for the False Positive Rate and the True Positive Rate (Recall). This value indicates that the proportion of incorrect acceptances is equal to the proportion of incorrect rejections. The lower the iso-error rate value, the more accurate the classifier is.

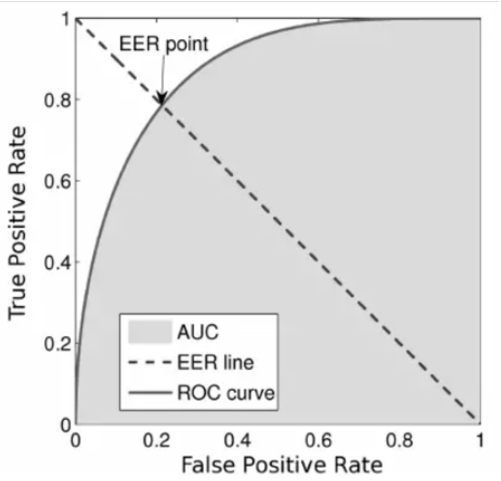
$$\text{False Negative Rate} = \frac{FN}{FN + TP} \tag{5.7}$$

We look into Figure 5.4(a) and Figure 5.4(b) at the same time. The EER point refers to the cut-off point of the ROC curve. The EER threshold is the threshold on the conditions of this EER point.

[6] https://dipranjan.github.io/dsinterviewqns/intro.html



(a) ROC Curve Plot [6]



(b) EER point on ROC Curve Plot (Tronci et al., 2009)

Figure 5.4: ILLUSTRATION OF ROC CURVE AND EER POINT.

5.2.2 Performance Evaluation for Classification Models

Table 5.4 shows the results of the assessment metrics of the 15 architectures of models. For different backbone CNNs, the model with global max pooling always has the lowest score on evaluation metrics. It is essential to figure out the intuitive difference between global max pooling and other global pooling methods. Global max pooling encourages the network to identify one discriminative part rather than identify the extent of the object as other approaches (Zhou et al., 2016). Other methods consider all discriminative parts when doing the global pooling operation on the feature maps. However, global max pooling only performs the max score. Therefore, the possible lack of global information leads to the models with global max pooling being less performing than others. Other than that, the performance of the other models is similar, with balancing accuracies largely above 80% and relatively high F1-scores.

5.2.3 Performance Evaluation for Visualization Methods

To order to evaluate the discriminative localization of the proposed methods, we compare them with the popular methods, CAM and Grad-CAM. Due to lacking labeled data, we propose two kinds of experiments. The first one is to evaluate these methods with variable masked areas. This experiment focuses on exploring the effect of different size mask areas on the visualization methods. Besides, we want to figure out the appropriate size of the mask area when the visualization methods perform stably. The second one is to assess their performance in the semantic masked areas. We aim to evaluate the lesion localization of the proposed methods in the semantic mask areas rather than random bounding boxes.

Precision of Lesion Localization with Variable Masked Areas

We accept the lesions highlighted within the lungs as reasonable localizations. To figure out to what extent the lesion localizations are within the lungs precisely, we performed binary classification on the class activation maps of test images. Lesion regions are considered to be discriminative

Backbone Conv	Pooling Type	Balanced Accuracy	F1-score
ResNet-50	Average	85.5%	0.912
	Max	75.8%	0.825
	Exponential	**86.8%**	**0.914**
	Linear	81.0%	0.881
	LSE	79.6%	0.868
VGG-19 BN	Average	80.1%	0.866
	Max	79.2%	0.853
	Exponential	82.8%	0.890
	Linear	**83.3%**	**0.891**
	LSE	80.8%	0.876
MobileNet V2	Average	83.3%	0.893
	Max	77.8%	0.855
	Exponential	**85.0%**	**0.907**
	Linear	81.8%	0.878
	LSE	81.4%	0.878

Table 5.4: Summary of the Performance of Multi-class Classification Models

areas that are highly activated by the model. The area of the lungs is defined as masked areas that are labeled with bounding boxes. We consider the discriminative areas within the masked areas as label 1, whereas non-discriminative areas outside masked areas are labeled as 0. Since our dataset is not labeled with semantic segmentation of lungs, we cannot get access to the ground truth of lungs in CXR images. We implement the bounding box to approximately delineate the area of the lungs. In order to reduce the systematic error brought by the bounding box, we select seven sizes of bounding boxes for each image to apply the binary classification.

Before we perform the binary classification, we need to define the masked area and discriminative area in detail.

Various Sizes of Masked Areas

We adopt seven groups of bounding box sizes. The bounding box is defined as a rectangle with a height and width, respectively. They are 55×20 pixels, 75×30 pixels, 95×30 pixels, 115×30 pixels, 135×50 pixels, 155×70 pixels, and 175×90 pixels. In Figure 5.5, we show some examples of different bounding boxes. The height of the bounding box is generally increased by 20 in each group. We put two bounding boxes on bilateral lungs while trying to keep them within the lungs as much as possible. Meanwhile, we shift the place of the bounding box on the right side to avoid covering the heart.

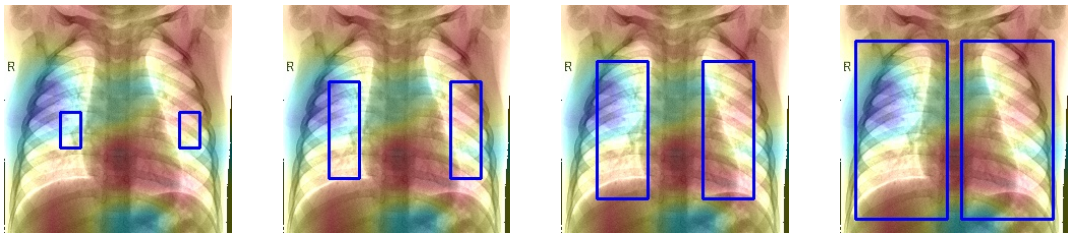
Discriminative Area

We define the total number of values in the class activation map as greater than the set threshold as the discriminative area.

Binary Classification on Test Images

To compare the overall performance among various visualization approaches, we apply binary classification to all test images. The following processes are applied:

First, the test images are preprocessed and classified by models. Then, we compute class activation maps on those test images that are correctly classified by the model based on different approaches. After that, we define the lung area on the CXR images by adding bounding boxes as masked areas. Furthermore, we define a group of thresholds for each class activation map to obtain the binary classification result. In this experiment, we set up 20 thresholds at the range within $[0, 1]$ with a gap of 0.05. Then, we define the data points of class activation maps which are within the masked areas as positive examples and the rest as negative examples. The detailed



(a) 55 × 20 (b) 95 × 30 (c) 135 × 50 (d) 175 × 90

Figure 5.5: EXAMPLES OF BOUNDING BOXES IN CXR IMAGES HEAPMAP. It illustrates how the bounding boxes are placed on the CXR images with the visualization heatmaps. The bounding boxes are placed on the bilateral lungs while the bounding box on the right side is shifted to avoid the heart. The red part of the heatmap is the discriminative area activated by a model.

meanings of satisfying the conditions of TP, TN, FP and FN are defined as follows:

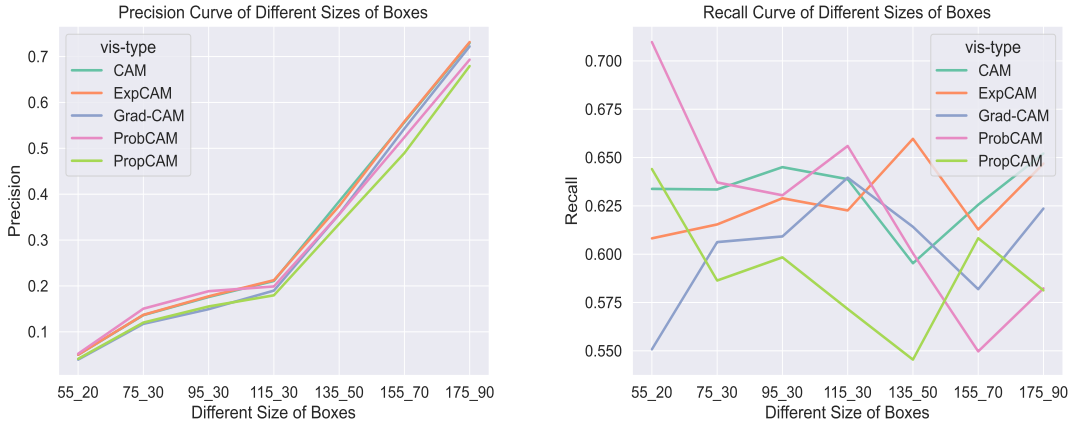
- **True Positive (TP)**: data point locates within the lung area and its value is $\geq$ threshold.
- **True Negative (TN)**: data point locates out of the lung area and its value is $<$ threshold.
- **False Positive (FP)**: data point locates out of the lung area and its value is $\geq$ threshold.
- **False Negative (FN)**: data point locates within the lung area and its value is $<$ threshold.

Finally, We compute the confusion matrix of the binary classification result.

Precision Curve and Recall Curve of Different Sizes of Boxes

To obtain the precision curve and recall curve related to different sizes of boxes, we need to choose an appropriate threshold for each visualization method. To control variables in the analysis, we use the EER threshold to compute the precision and recall of each group of bounding boxes since we consider the EER threshold as the optimal threshold for the binary classification of visualization approaches on the condition to box size. In this way, all visualization methods can be compared on the condition of their best performance in each group of bounding boxes. We go through the results among 15 models (Figure A.1 and A.2). It reveals that the precisions of all visualization methods positively correlate to the sizes of bounding boxes. When the size of bounding boxes increases, related precisions go up as our expectation. However, recalls of visualization approaches do not show regular variation in the size of bounding boxes. We take the results of the precision and recall curve of different sizes of boxes from the model with ResNet-50 and global average pooling as an example to explain detailedly. When the bounding boxes become larger, two situations come as follows. On the one hand, the areas that do not belong to the lungs are included. It reflects the changes in precision. On the other hand, more lung areas are contained. It reflects the difference in recall.

In Figure 5.6(a), the precision grows in general when the size of bounding boxes rises. When the bounding boxes enlarge, more discriminative areas that are out of the lungs will be included in the masked areas. As Figure 5.5(d) shows, the top right corner is the background of the CXR image, but it is adopted as the masked area in the bounding box. It means that this bounding box contained the related discriminative areas on this corner. During this process, the original false positives become true positives. Thus, the number of true positives rises. And the total number of positives is fixed on the condition to a specific threshold. Therefore, the precision increases (Equation (5.2)) as the bounding boxes get larger. In Figure 5.6(b), the recall does not keep decreasing as the size of bounding boxes changes. Only the recall of probabilistic CAM shows decreasing to



(a) ResNet-50 + Average Pooling

(b) ResNet-50 + Average Pooling

Figure 5.6: Precision Curves and Recall Curves of Various Sizes of Bounding Boxes. (a) and (b) are from the model with ResNet-50 as backbone CNN and global average pooling. ExpCAM indicates exponential CAM. ProbCAM indicates probabilistic CAM. PropCAM indicates proportional CAM.

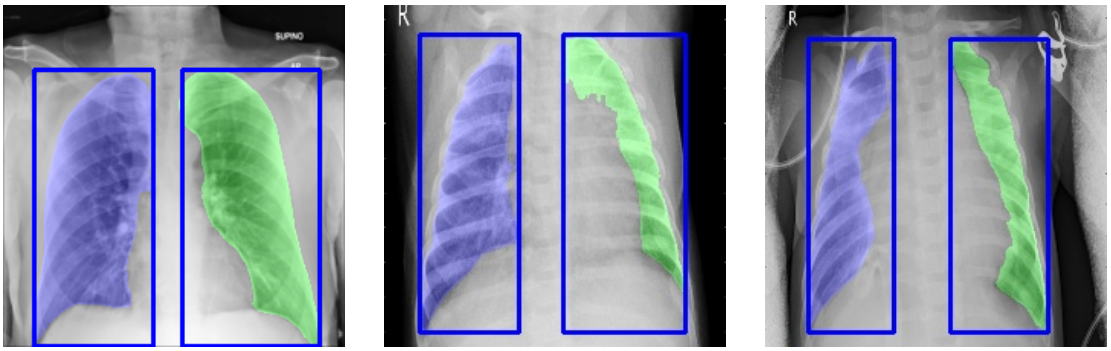
some extent. When the bounding boxes become larger, the recall values fluctuate. There are two reasons. First, we do not compare the recall with the same threshold for different sizes of bounding boxes. Second, the fluctuating recall is related to the performance of the model. Although the bounding boxes contain more lung areas, the discriminative areas from models are not always in the lung areas. In Figure 5.5, the discriminative areas are in the middle of the spine, lungs, or background. When the bounding boxes enlarge, they might contain more discriminative areas or more non-discriminative areas within the lungs. In Equation (5.3), when the recall values are high, the bounding boxes contain more discriminative areas (true positives). When the recall values are low, the bounding boxes contain more non-discriminative areas (false negatives). In our cases, we try to adjust the sizes of bounding boxes to cover the lung areas as much as possible. If the recall values are more unstable, the model makes decisions more arbitrarily which means the model does not tend to make predictions in the lung areas. In addition, the fluctuating recall curves may reveal that the hyper-parameter representations chosen by the visualization methods are lacking reliability to a certain extent. Other relevant results of 15 models are shown in the supplementary material (Figure A.2). Based on these results, it is difficult to find a general rule to determine the sizes of bounding boxes. Therefore, we will mainly analyze the performance of lesion localization based on the semantic masked area.

Precision of Lesion Localization with Semantic Masked Area

We also implement the binary classification of the proposed methods with the semantic masked area. The definition of discriminative areas and the process of binary classification are the same as in the previous section. The only difference is that we use the semantic bounding boxes instead of various sizes of the bounding boxes.

Semantic Masked Areas

In this experiment, we apply bounding boxes based on lung segmentation to define the semantic masked area. We choose to use bounding boxes rather than lung segmentation since the bounding boxes can capture more lung area than segmentation results when lungs are infected.



(a) COVID-19 (b) Virus (c) Bacteria

Figure 5.7: IMAGE SEGMENTATION ON TEST IMAGES TO LOCALIZE LUNG REGIONS. Lung areas segmentation of a COVID-19 patient(a), a viral lung infections patient (b), and a bacterial lung infections patient (c)

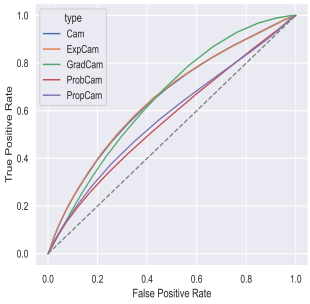
We apply a state-of-the-art automated lung segmentation model [7] to the test images and get the semantic lung masks. Some examples of lung masks are shown in Figure 5.7. Figure 5.7(a) detects the lung areas accurately in the CXR image. However, in the right part of Figure 5.7(b) and Figure 5.7(c), there are some missing parts of the lungs. When a patient has lung infections, the lungs will be filled with fluid or pus so that the affected parts will appear white on a chest X-ray. Therefore, the lung segmentation model fails to detect the affected part of the lungs. Therefore, the bounding boxes are used to capture as much of the lung area as possible in cases of infections. We leverage the highlighted lung masks in the original images to define our bounding boxes. The bounding boxes are considered lung regions which are masked areas used to perform the binary classification. Examples of bounding boxes with lung masks are shown in Figure 5.7.

ROC Curve

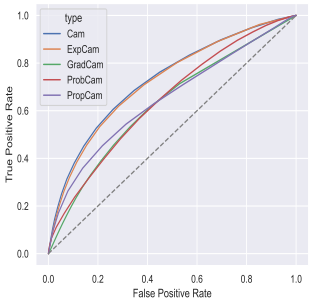
We use the ROC curve to evaluate the performance of the proposed methods on average. We go through all ROC curves of 15 models (Figure A.3). In general, the AUC values of all visualization methods are over 0.5 which means that the proposed methods are able to distinguish the significant areas and non-significant areas. Among them, Grad-CAM has the highest AUC values in the majority of models with the backbone CNN of MobileNetV2 and ResNet-50. However, Grad-CAM does not perform well among the models with the CNN architecture of VGG19 Batch Normalization. Then exponential CAM performs the second best stably among all models. The performance of exponential CAM is close to that of CAM which slightly outperforms CAM in most cases according to the AUC values. After that, the performance of proportional CAM and probabilistic CAM are the fourth and fifth with a tiny difference.

We analyse the results from Figure 5.8 and Figure 5.9 in details. Figure 5.8 illustrates the best performance of exponential CAM in the binary classification from each type of backbone CNN model. Figure 5.8(a) and Figure 5.8(c) show that the ROC curves of exponential CAM, CAM and Grad-CAM are close to each other. In Figure 5.8(b), CAM is better than exponential CAM to a small extent. However, Grad-CAM performs worst among all. Proportional CAM and probabilistic CAM have similar performances but are worse than the other three methods. Figure 5.9 shows the worst performance of Grad-CAM in the binary classification from each type of backbone CNN model. All the methods have similar performance in Figure 5.9(a). In Figure 5.9(b), the visualisation methods demonstrate sequential differences in performance. CAM performs the best, followed by exponential CAM, Grad-CAM, proportional CAM and probabilistic CAM.

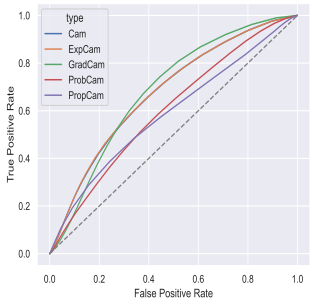
[7] https://github.com/alimbekovKZ/lungs_segmentation



(a) ResNet-50 + Exponential Pooling

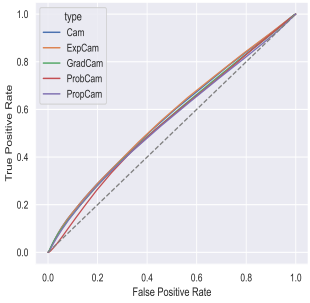


(b) VGG19 BN + LSE Pooling

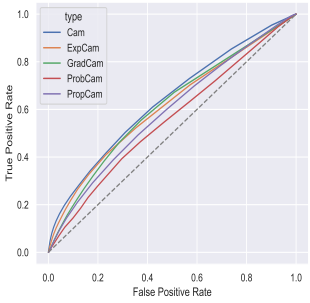


(c) MobileNetV2 + LSE Pooling

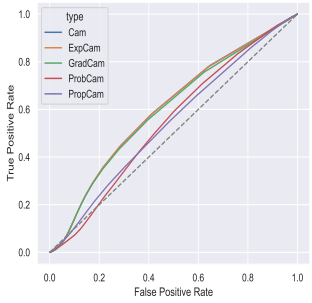
Figure 5.8: ROC CURVES. It illustrates the **best** performances of exponential CAM for each type of backbone CNN.



(a) ResNet-50 + Max Pooling



(b) VGG19 BN + Max Pooling



(c) MobileNetV2 + Linear Pooling

Figure 5.9: ROC CURVES. It illustrates the worst performances of exponential CAM for each type of backbone CNN.

In Figure 5.9(c), the performance among visualization approaches resembles that in Figure 5.8(c). Exponential CAM, CAM and Grad-CAM perform well and similarly. Proportional CAM and probabilistic CAM have similar performances and follow the other three methods.

Acceptable Mask Ratio Analysis

In this section, we evaluate the performance of the proposed methods on lesion localization. Since we focus on whether models can locate lesions in the CXR images of patients with lung infections, we only analyze the class of bacteria, virus, COVID-19, and SARS and exclude the normal class.

We assess the performance of the proposed methods based on the semantic masked areas by the acceptable mask ratio. To calculate the acceptable mask ratio, we first need to compute the discriminative area under specific thresholds. To make sure that all visualization methods are evaluated on the optimal classification condition, we choose the Equal Error Rate (EER) thresholds of ROC curves. Since the EER points are the optimal cut-off points for binary classification (Figure 5.4), we consider the threshold of the EER point as the optimal threshold. Then, we compute the discriminative area under the EER threshold. The number of values greater than the EER

threshold is considered the discriminative area. After that, we calculate the intersection between the discriminative area and the masked area. Finally, we compute the ratio of the intersection area to the whole discriminative area. The ratio is considered the acceptable mask ratio (Equation (4.6)). The acceptable mask ratio is used to evaluate the performance of visualization methods. The higher the ratio is, the more precisely the visualization method focuses on the lungs.

We will analyze the assessment in two aspects. Firstly, we will evaluate the average acceptable mask ratio of the proposed methods of all models. Secondly, we will evaluate the acceptable mask ratio of the proposed methods on models specific basis.

Acceptable Mask Ratio Analysis of All Models

We leverage box plots to help us understand the locality, spread and skewness groups of acceptable mask ratio values among visualization approaches. Figure 5.10 demonstrates the different parts of one kind of box plot. It displays data in a standardized way based on a five-number summary.

- **Minimum**: The smallest value in the data set or the critical value of the 1.5 times of Interquartile range (IQR) below the first quartile (Q1).
- **First Quartile (Q1)**: The middle number between the minimum and the median of the data set.
- **Median**: The middle value of the data set.
- **Third Quartile**: The middle value between the median and the maximum of the data set.
- **Maximum**: The largest value in the data set or the critical value of the 1.5 times of Interquartile range (IQR) over the first quartile (Q1).
- **Interquartile range (IQR)**: The difference between the first and third quartile. It contains 50% of the data.
- **Outliers**: The values that are more than 1.5 times the interquartile range from the first or third quartile.
- **Whiskers**: The lines that extend from the box to show the range of the data.
- **Box**: The box extends from the first quartile to the third quartile. The line in the box represents the median.

Figure 5.11(a) summarizes the overall acceptable mask ratio of four lung infections including bacteria, virus, COVID-19 and SARS, of the proposed methods on all models. The average acceptable mask ratios of all visualization methods are $\pm$ 0.7, which means around 70% of the discriminative areas activated by various models are within the semantic masked areas that we consider as the lungs. Probabilistic CAM has the highest mean value among all. Then it is followed by exponential CAM, CAM, Grad-CAM, and proportional CAM. However, probabilistic CAM has the largest interquartile range which means its acceptable mask ratios fluctuate more dramatically compared to other methods. For different models, the acceptable mask ratios from probabilistic CAM have higher dispersion. It might come from the difference among models or classes. Exponential CAM and CAM has alike median and interquartile range. Compared to CAM, the overall acceptable mask ratios of exponential CAM are slightly more compact in whiskers and outliers. Exponential CAM and CAM have relatively stable performance among models. Grad-CAM has a similar interquartile range to exponential CAM. However, Grad-CAM has the most outliers, which means its acceptable mask ratios are more scattered. It reveals that the visualization results of Grad-CAM vary considerably among models. Although Proportional

CAM has the smallest mean value and medium interquartile range, it has fewer outliers which shows that the discriminative areas of models fall more consistently in the lungs.

Figure 5.11(b) demonstrates the overall acceptable mask ratio for each class, namely bacteria, virus, COVID-19 and SARS individually, of 15 models. We gain more detailed insights here.

- **Probabilistic CAM:** It only outperforms others in the bacteria class but performs worst in another three classes. But its superior performance in the bacteria category contributes significantly to the average performance (Figure 5.11(a)). Although probabilistic CAM considers both dominant class probability and non-dominant class probability, bacteria as the majority class, has learned more features than others in the model training. When applying probabilistic CAM to visualize the bacteria category, the class activation maps from other categories has a small impact on its final visualization but provide valid edge information. It contributes to its performance in the visualization of the bacteria class. However, when applying probabilistic CAM to visualize other categories, the impact of the bacteria class cannot be ignored which influences the final visualization results of other categories to some extent. It leads to the inferior performance of probabilistic CAM in the other three classes.

- **Exponential CAM:** The overall performance of exponential CAM is close to CAM. Compared to CAM, exponential CAM enhances the differences between the discriminative areas and non-discriminative areas of class activation maps by an exponential function. It helps to separate the boundary values of the discriminative areas, contributing to the more compact acceptable mask ratios of exponential CAM. Although exponential CAM performs second best in the overall acceptable mask ratio, it has the more stable performance among models for each class. In the majority of classes, it has the top performance of acceptable mask ratio.

- **Proportional CAM:** Similar to exponential CAM, proportional CAM also enhances the differences between the discriminative areas and non-discriminative areas of class activation maps. It incorporates the proportion between each element and the overall feature map as a weight to compute the final visualization heatmap. For the element with similar values, it cannot separate them effectively, either classifying them as discriminative areas or non-discriminative areas. Especially for the boundary values, it cannot absorb the edge information of the discriminative areas which might lead to great differences with various thresholds. Furthermore, it leads to the inferior performance of proportional CAM in the overall acceptable mask ratio.

Acceptable Mask Ratio Analysis for Each Model

In this part, we focus on models with the best and worst performances. We will first evaluate the average performance of the acceptable mask ratio of each model on all visualization methods. Then we look into the distribution of the acceptable mask ratios of each model in each class.

Figure 5.13 displays the average acceptable mask ratio of four lung infections including bacteria, virus, COVID-19 and SARS, from the best three models. Figure 5.14 demonstrates the average acceptable mask ratio of four lung infections from the worst three models. Compared with Figure 5.13 and Figure 5.14, exponential CAM outperforms others in acceptable mask ratio in both best and worst models in most cases. Then CAM follows. Both exponential CAM and CAM have higher average acceptable mask ratios in models with better performances. For the other three methods, Grad-CAM, Probabilistic CAM and Proportional CAM, in general, the average acceptable mask ratios for all three methods increases with the overall performance of the model, with exceptions in the individual model options. It means that the visualization results of the best models are more accurate than the worst models. For special cases, the performance of Grad-CAM is moderate but fluctuates with the model using VGG19 Batch Normalization as CNN architecture.

[8] https://www.leansigmacorporation.com/box-plot-with-minitab/

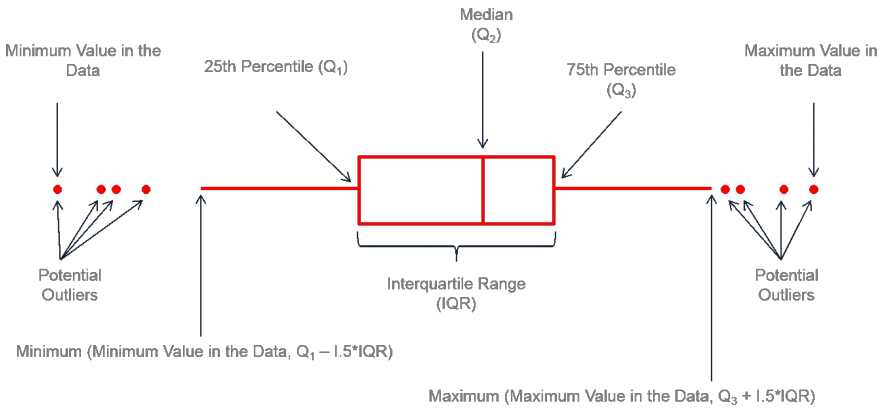


Figure 5.10: Box Plot Anatomy [8]

Probabilistic CAM and proportional CAM have similar fluctuation as Grad-CAM. Probabilistic CAM fluctuates in the models with the MobileNetV2 option while proportional CAM fluctuates in the models with the ResNet-50 option. It shows these three approaches are more sensitive to the models. It is more likely to be affected by the model training or the threshold. Other results can be found in Figure 5.12.

Then we look into the distribution of the acceptable mask ratios of each model in each class. Figure 5.15 shows the distribution of the acceptable mask ratios of the best three models in each class. Figure 5.16 demonstrates the distribution of the acceptable mask ratios of the worst three models in each class. In general, different types of CNN architecture do not make obvious differences in the acceptable mask ratio distribution. Compared to the difference among pre-trained models, various pooling operations on the global pooling layer have a more significant impact on the acceptable mask ratio distribution. In Figure 5.17, large fluctuations in results among different global pooling operations.

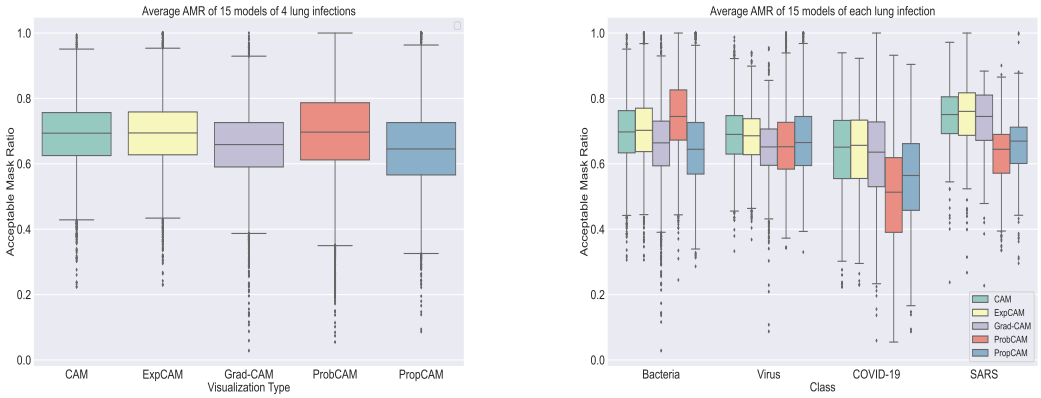
As Figure 5.12 and Figure 5.17 show, there is no significant correlation between the performance of different CNN architectures and the acceptable mask ratio of visualization results. The better performance of a model does not necessarily correspond to a high acceptable mask ratio. Furthermore, the higher performance of the model does not suggest that the model prefers to make decisions in the lung region. Furthermore, the impact of different global pooling methods on lesion localization appears to be more pronounced than using various network architectures. Within the same global pooling layer approach, the acceptable mask rates of models with different network structures exhibit similar patterns in two respects.

1. As shown in Figure 5.12, average acceptable mask ratios among different visualization methods have similar ranking distributions.

2. As shown in Figure 5.17, its acceptable mask ratios among lung infections maintain a similar ranking for the same visualization method. For example, in the global average pooling, probabilistic CAM has the ranking of class bacteria, SARS, virus and COVID-19 in performance of acceptable mask ratio.

Statistical Distribution Heatmap for Each Class

To observe and compare the distribution of discriminative areas of different classes, we calculate the statistical distribution heatmap based on various class activation maps for each class. We



(a) Overall Acceptable Mask Ratio of Lung Infections

(b) Overall Acceptable Mask Ratio for Each Class

Figure 5.11: OVERALL ACCEPTABLE MASK RATIO OF 15 MODELS. (a) illustrates the overall acceptable mask ratio of four lung infections including bacteria, virus, COVID-19 and SARS, of 15 models (b) demonstrates the overall acceptable mask ratio for each lung infection class of 15 models.

go through all the test images classified correctly by the model for a specific class to count its distribution of the discriminative areas. We implement this experiment in the following steps:

1. We divide the 224×224 pixels of visualization heatmaps generated by various visualization approaches into 196 grids, and each grid is 16×16 pixels in size.
2. We examine the values of each grid based on the EER threshold of the model. If over half of the values in the grid are greater than the threshold, we count this grid as a discriminative cell and mark the grid as 1; otherwise, we mark the grid as 0.
3. We count all visualization heatmaps for each class to get a distribution map of the total statistically significant regions for respective classes.
4. We normalize the distribution maps to get the statistical distribution heatmaps for each class.

To show the statistical distribution heatmap intuitively and understandably, we pick up one test image from each lung infection category as a representation to demonstrate its corresponding statistical distribution heatmap. Figure 5.18 lists the selected test images.

Figure 5.19 and 5.20 show the statistical distribution heatmaps based on class activation maps generated by various visualization methods of the best and worst models, respectively. These two figures are both from the model with ResNet-50 but with different global pooling methods. Figure 5.19 shows the results from the model with exponential pooling. Figure 5.20 lists the results from the model with max pooling. Since Global max pooling only considers the localization with max scores, compared to Figure 5.19, the results from Figure 5.20 are more compact and each category corresponds to a specific range of area for class bacteria, virus and SARS. However, for class COVID-19, the results from the worst model (Figure 5.20) are scattered which shows that the model predicts the COVID-19 class arbitrarily rather than detecting it in significant areas. The results from Figure 5.19 show that the model with exponential pooling detects the COVID-19 class from most areas within the lungs. Besides, both models perform better and more concentrated visualizations in the majority of the classes, bacteria and virus, respectively.

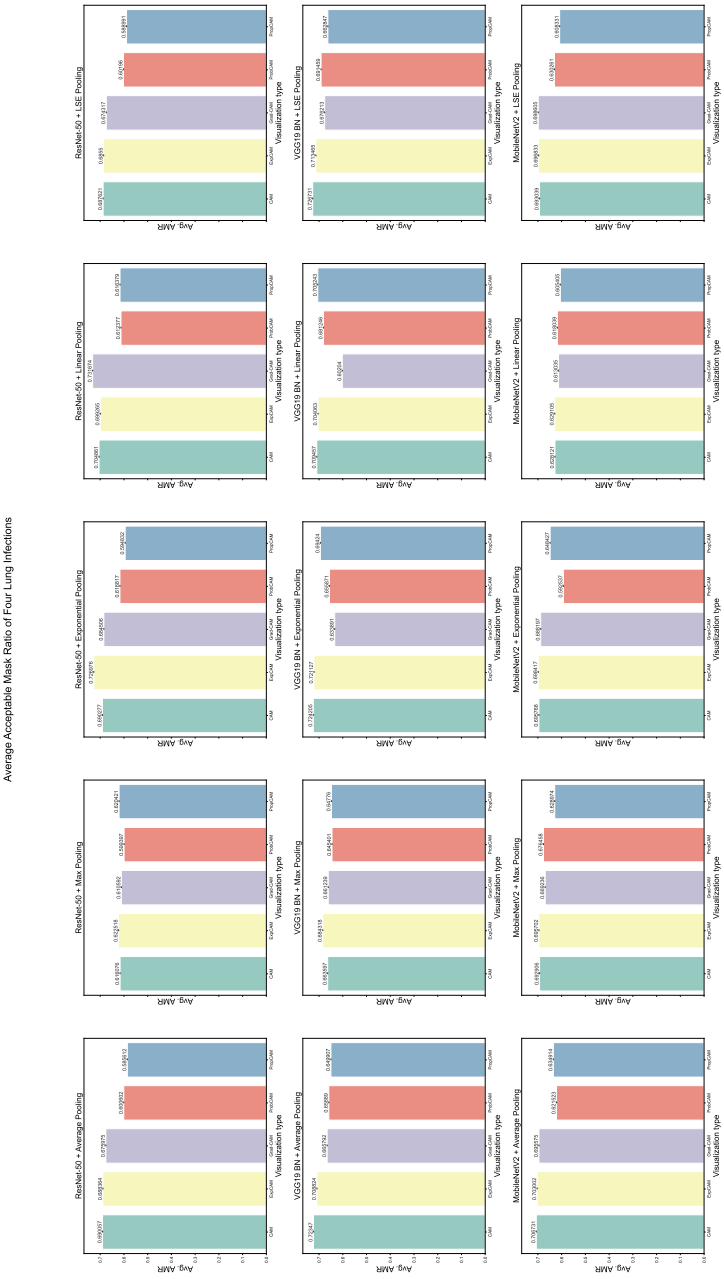
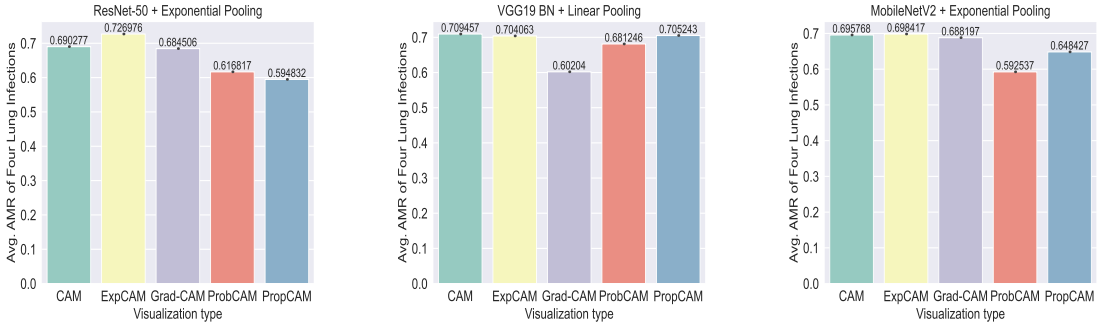
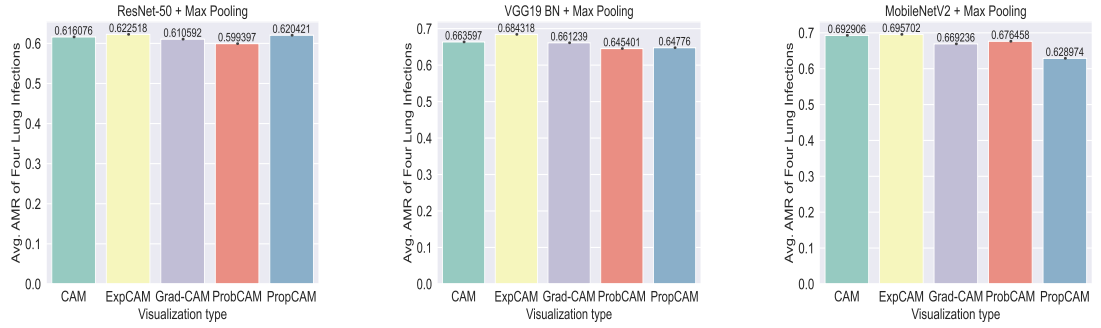


Figure 5.12: Average Acceptable Mask Ratio of Four Lung Infections of 15 models



(a) (b) (c)

Figure 5.13: PARTIAL RESULTS OF AVERAGE ACCEPTABLE MASK RATIO OF ONE MODEL. It illustrates the average acceptable mask ratio of four lung infections including bacteria, virus, COVID-19 and SARS, from the **best** models with different backbone CNN options.



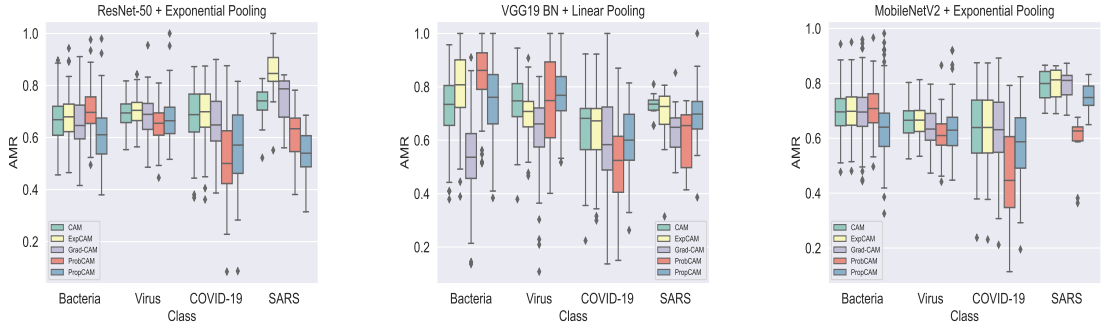
(a) (b) (c)

Figure 5.14: PARTIAL RESULTS OF AVERAGE ACCEPTABLE MASK RATIO OF ONE MODEL. It illustrates the average acceptable mask ratio of four lung infections including bacteria, virus, COVID-19 and SARS, from the worst models with different backbone CNN options.

For different visualization methods, exponential CAM locates lesions mainly within the lungs but does not show obvious differences from CAM. Proportional CAM has similar statistical distribution heatmaps as exponential CAM. Probabilistic CAM performs better in the majority of the classes. Besides, it shows to have the ability to detect the lesion on both sides of the lungs. However, it is hard to focus on the lesion in the minority of classes, like COVID-19 and SARS, indicating that probabilistic CAM is sensitive to the model performance of non-dominant classes.

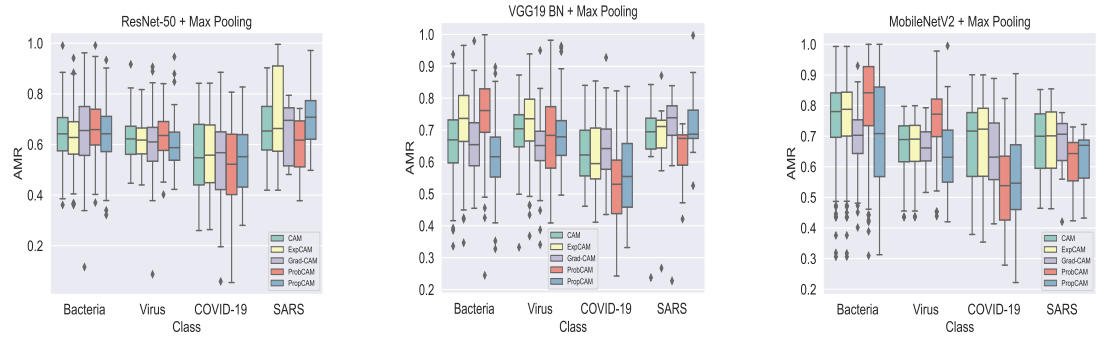
5.2.4 Summary of Lesion Localization Results

In this chapter, we implement two kinds of binary classification to evaluate the average performance of different visualization approaches under various thresholds. Compared to other methods, exponential CAM is relatively stable in recall and precision in most cases. And it has consistently maintained the best or second best performance in ROC curves and AUC among different models. For probabilistic CAM and proportional CAM, their performance fluctuates relatively,



(a) (b) (c)

Figure 5.15: PARTIAL RESULTS OF ACCEPTABLE MASK RATIO. It illustrates the acceptable mask ratio for each class from the **best** models with different backbone CNN options.



(a) (b) (c)

Figure 5.16: PARTIAL RESULTS OF ACCEPTABLE MASK RATIO. It illustrates the acceptable mask ratio for each class from the worst models with different backbone CNN options.

resulting in their poor average performance.

In addition, we proposed the analysis of acceptable mask ratios to assess the visualization methods quantitatively. Probabilistic CAM has the best performance on overall acceptable mask ratios for four lung infections of 15 models. Its superior performance in the bacteria class contributes to its overall performance to a great extent. For more stable performance among various, exponential CAM is the better choice. In most cases, exponential CAM has the best performance among classes of models. The performance of proportional CAM is not stable enough among classes since it cannot well separate the difference in class activation maps.

Furthermore, we count the discriminative areas of each class to generate the statistical distribution heatmaps for each class. Compared to the variable results from Grad-CAM, the results of exponential CAM and proportional CAM mainly concentrate on the lungs among classes. And Probabilistic CAM shows its ability to detect the lesion on both sides of the lungs.

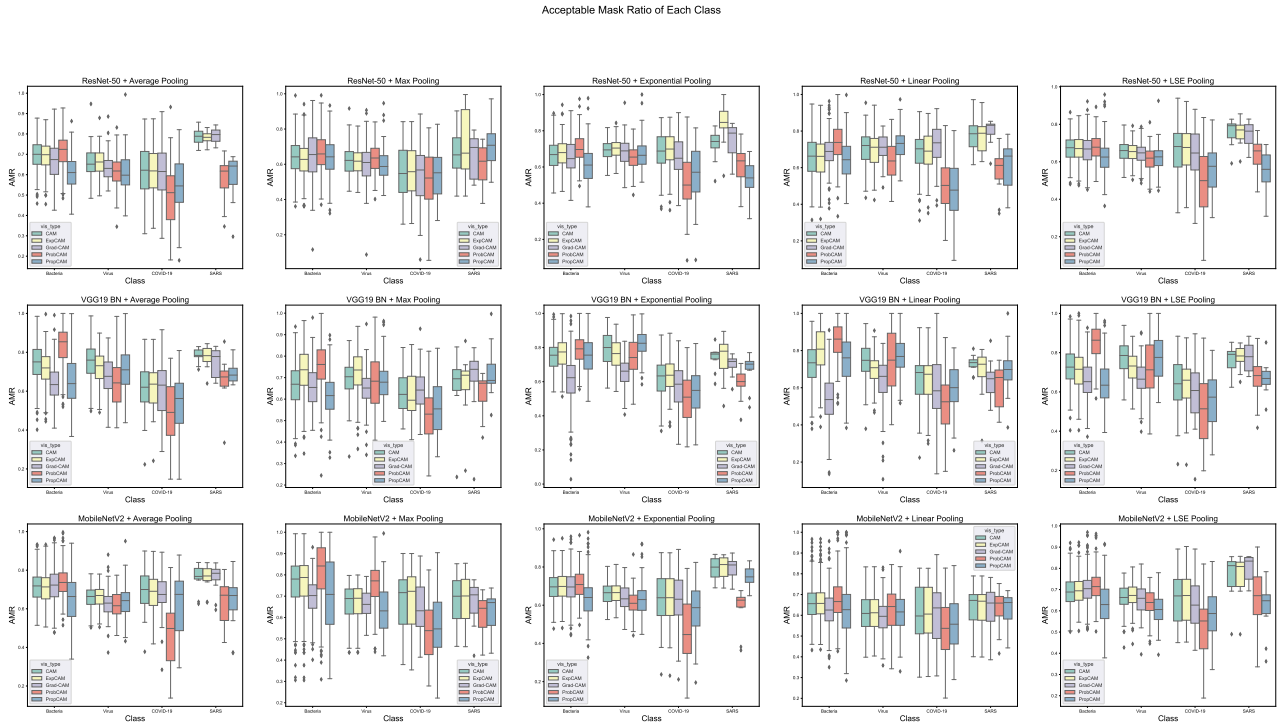
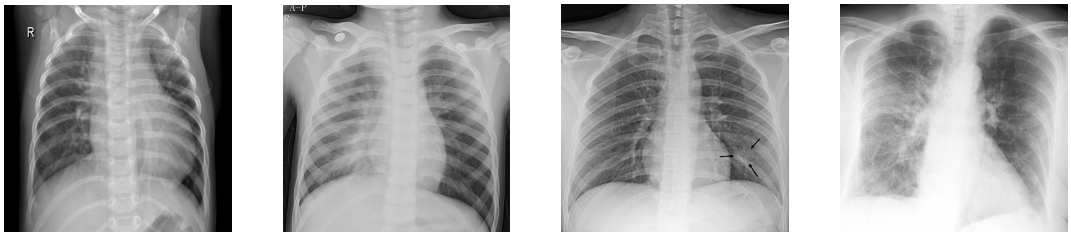


Figure 5.17: Acceptable Mask Ratio of Each Lung Infection of 15 models



(a) Bacteria (b) Virus (c) COVID-19 (d) SARS

Figure 5.18: TEST IMAGES USING FOR STATISTICAL DISTRIBUTION HEATMAP. Test images from four lung infections demonstrate the statistical distribution heatmaps for their corresponding class.

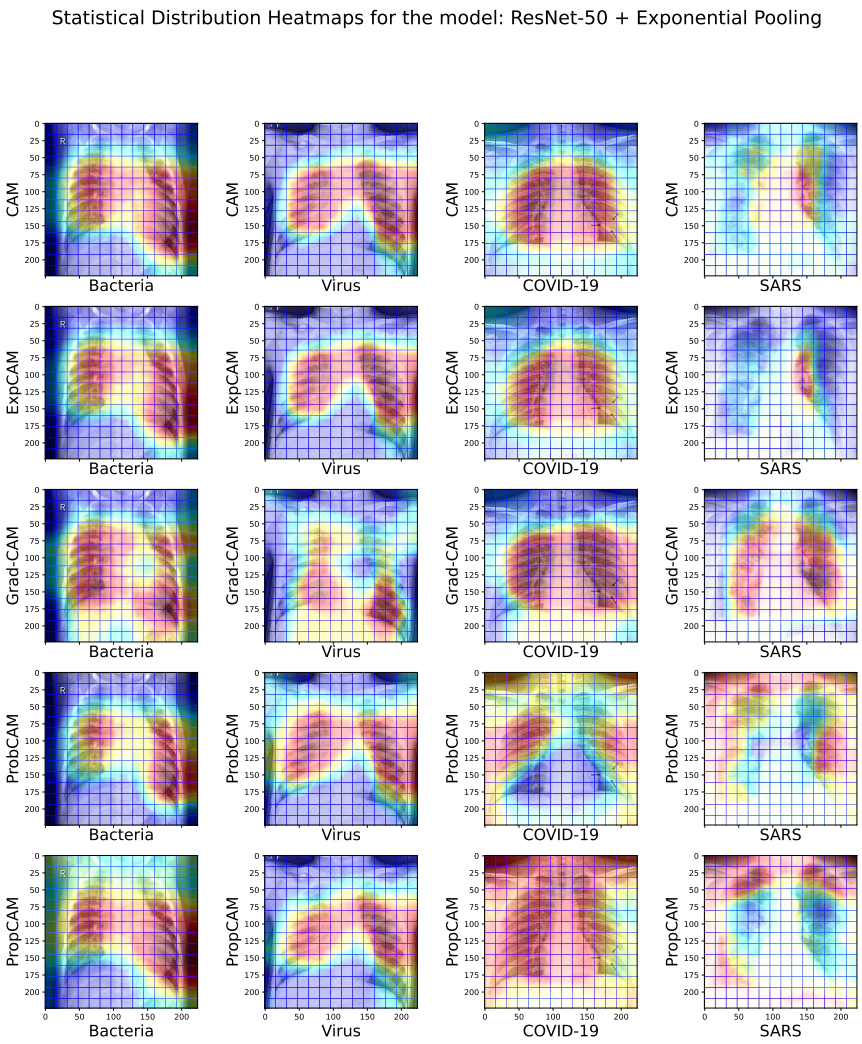


Figure 5.19: Statistical Distribution Heatmaps of the Model with ResNet-50 and Exponential Pooling

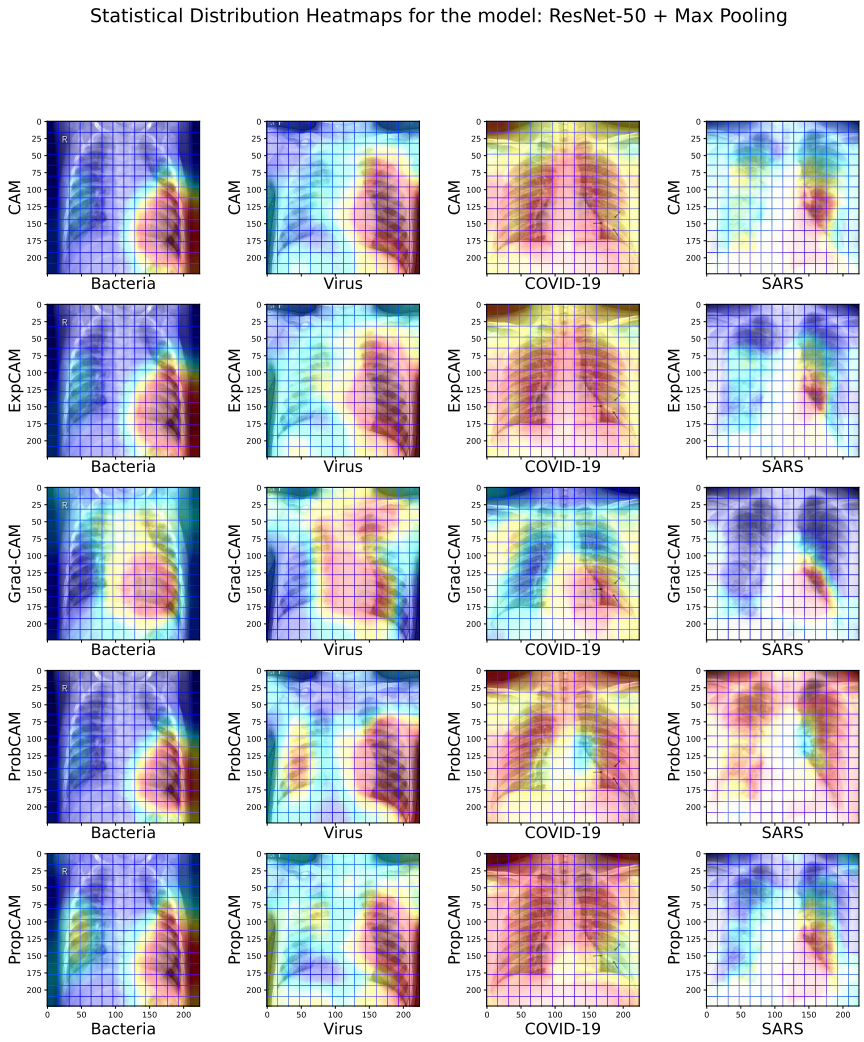


Figure 5.20: Statistical Distribution Heatmaps of the Model with ResNet-50 and Max Pooling

Chapter 6

Discussion and Future Work

In this section, we will discuss some observations from the experiments. We will also discuss the limitations of the proposed approaches and the metric.

Exponential CAM calculates the exponential of the feature values based on class activation from the dominant category. Overall, the performance of exponential CAM is rather close to that of CAM. Figure 6.1 shows different visualizations of a test image from each of the four classes. These four test images are activated by the model with ResNet-50 and global average pooling, visualized by different methods, and labeled with the semantic masked areas. As shown in Figure 6.1, the distribution of discriminative regions of exponential CAM is similar to that of CAM. It demonstrates that exponential CAM cannot significantly enhance the differences among feature values. In most cases, the performance of the acceptable mask ratio in exponential CAM is slightly better than in CAM. It is because the exponential operation on the class activation map can filter out some values nearby the thresholds to some extent. This generates more compact discriminative areas, which reflect a better performance of the acceptable mask ratio. However, exponential CAM cannot guarantee the performance of the acceptable mask ratio. In some cases, the performance of the acceptable mask ratio in exponential CAM is worse than that in CAM. This is because the exponential operation on the class activation map can also amplify the noise in the discriminative regions. This generates more noisy discriminative areas. In addition, the performance of the acceptable mask ratio in exponential CAM is sensitive to the threshold. The performance of the acceptable mask ratio in exponential CAM is better than that in CAM when the threshold is small. However, the performance of the acceptable mask ratio in exponential CAM is worse than that in CAM when the threshold is large.

Probabilistic CAM considers the probability of classification as the weight of the feature values and also incorporates the class activation maps from all classes. As shown in Figure 6.1, probabilistic CAM can highlight the discriminative regions within the lungs in the majority of classes, bacteria, and virus, separately. And distributions of significant areas locate within a similar range of areas as exponential CAM. For the minority classes of COVID-19, the most discriminative areas (red parts) decrease significantly. If we view from Figure 5.19 and 5.20 as a whole, the discriminative areas of the COVID-19 class are more highlighted on the sides. These might reduce the probability that discriminative areas are captured by the masked areas, resulting in poor performance. For another minority class of SARS, the result from Figure 6.1 is good and focuses on the lower parts of the lungs. However, if we view from Figure 5.19 and 5.20 as a whole, the discriminative areas distribute over a wide area of variation. This results in its performance in terms of acceptable mask rates varying widely among models. From the fluctuating performance of the acceptable mask ratio in the minority of classes, we can conclude that probabilistic CAM is sensitive to the dataset imbalance. It might reflect that models do not learn the discriminative features of the minority classes well. Meanwhile, the performance of probabilistic CAM would rely more on the quality of models.

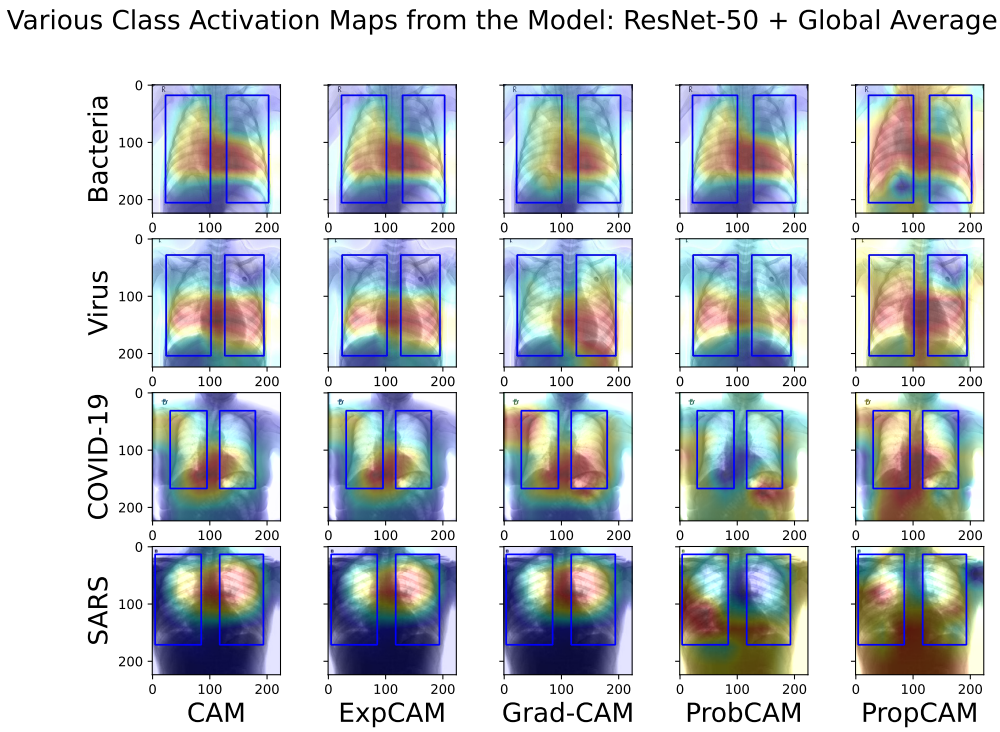


Figure 6.1: Examples of Different CAMs Generated from the Model with ResNet-50 and Global Average Pooling. Different methods visualize four images from the class of bacteria, virus, COVID-19, and SARS, respectively. These images are labeled with semantic bounding boxes. The first column shows the visualization results of CAM among four classes, and so on.

Proptional CAM considers the proportion of the feature values as the weight of the feature values from the dominant category. As shown in Figure 6.1, the significant regions distribute more evenly compared to other visualization methods. A wider regional distribution might lead to greater variance in the calculation of acceptable mask rates. Similarly, proportional CAM has decentralized discriminative regions of the COVID-19 class as shown in Figure 5.19 and 5.20. In addition, the discriminative regions of the SARS class distribute mainly on the top of the background in Figure 5.20 when the model does not perform well. However, when the model has better performance, the significant areas distribute more compactly within the lungs. Therefore, the performance of the acceptable mask ratio in proportional CAM is also sensitive to the model performance, especially for minority classes.

Besides, we observed that all visualization methods perform in the following order in terms of acceptable mask ratios: bacterial, viral, SARS and COVID-19 classes (Figure 5.17). In some cases, the SARS class can outperform the viral class and even the bacterial class. The dataset for training is imbalanced, the bacteria class has the most training and test images, followed by the virus class, then the COVID-19 class and the SARS class. Since the SARS class has the least training images, models are supposed to learn fewer features in the SARS class. Therefore, the performances of the SARS class would be the worst among all categories. We try to explain the reasons from the visualization results. However, it is difficult to make conclusions from them. For this issue, we need more information to investigate.

As mentioned above, the performance of models is important to interpretable visualization.

Compared to different CNN architectures, the pooling methods shed more light on the localization of the lesions. The effect of different global pooling methods on lesion localization appears to be more pronounced. However, the global pooling method is limited to only the remaining one value from the feature maps after processing. It loses limited spatial information, which might be useful for lesion localization. For global max pooling, it is easy to get stuck in one special area. If this area is not related to the lesion, the model will not be able to jump out of this area and learn about the lesion. For other pooling methods which consider the average values of the feature map, they might ignore certain difference. We can consider leveraging the local pooling methods to help capture more spatial information on a small scale.

In the future, we can also focus on following directions. First, we need to collect more chest X-ray images for minority classes, such as COVID-19 and SARS. In our work, we use the imbalanced dataset to train models. Although we apply sample weight, the imbalance still exists, and it reflects the interpretability of the deep learning model. The performance of Probabilistic CAM in each category from best to worst maintains the same rank as dataset imbalance from largest to smallest. It performs best in the bacteria class and moderately in the virus class but poorly in the COVID-19 class and SARS class. Second, it is important to collect more clinical information about patients. The chest X-ray dataset we use does not have corresponding patient information. Thus, we cannot learn about the diversity of the data, e.g., whether the CXR images are from babies, children, or adults. It is more important to know whether this radiograph is from a patient with early onset. The difficulty of distinguishing between models varies because of the intensity of the infection. Different viral infections can be difficult to distinguish in the early stages if the symptoms are mild. Third, we can explore more operations on visualization methods to improve the interpretability of the deep learning model. We can take the activation function or cross-entropy loss into account to allocate the weights of class activation maps. Besides, saliency maps can also be used to supplement spatial information. Saliency information can be preserved even in low-resolution grayscale images (Yohanandan et al., 2018). A saliency map is an image that highlights the areas that people's eyes first pay attention to. The human visual system first pays attention to the differences between different pictures, such as brightness. A saliency map can help to find discrepancies between X-ray images. Fourth, we can explore the interpretability of the deep learning model in other medical fields, such as medical image segmentation and medical image registration.

Our main performance evaluation of visualization methods is based on the acceptable mask ratio. In Figure 5.12, the average of acceptable mask ratios are over 0.7 in best cases and around 0.6 in worse cases. According to these results, at least 60% of lesion localizations happen within the lungs among four kinds of lung infections. On the condition of the models with around 80% of balanced accuracy on average, the performance of visualization methods is relatively good. However, there is a systematic error in the acceptable mask ratio, which might increase its value to be higher than the true value. One of the basic components of an acceptable mask ratio is masked areas. Although we leverage the segmentation of lungs, masked areas still cannot avoid containing the background of CXR images. Lesion localization happening in the background of CXR images is not acceptable but included in the measurement of acceptable mask ratio. Therefore, it is important to improve the approximation of the lung areas to reduce the system error in the acceptable mask ratio. There are two ways to improve the approximation of the lung areas.

- One is to use a more accurate segmentation method to segment the lungs as more as possible. We can consider training a segmentation model based on our dataset, taking advantage of transfer learning.
- The other is to use a more accurate bounding box of the lungs. We can leverage polygons to approximate the lung areas. The polygons can be obtained by the segmentation of the lungs. We can use the polygons to replace the bounding box of the lungs. The polygons can help to reduce the areas from the background.

In addition, we can also try to locate the masked areas more precisely for each category. For instance, coronavirus more often infects the right lower lobe, and the left lower lobe of the lungs (Zhang et al., 2022). Therefore, we can divide the masked areas into several parts according to lung anatomy and allocate different weights for each part based on the clinical characteristics of every category. The weights can be obtained by the clinical knowledge of the experts or researchers. Then, we calculate the acceptable mask ratio for each category with weights to obtain a more accurate evaluation of lesion localization.

Chapter 7

Conclusion

To cope with the respiratory pandemic, a quick diagnosis is key. It is still a challenge to develop a system that can automatically detect COVID-19 or other pneumonia from CXR images without human intervention when the system lacks interpretability of the deep learning model. In this work, we focus on the explainable classification of chest X-ray images. We pay attention not only to COVID-19 but other kinds of lung infections. There are two contributions to this work. On the one hand, we propose three new and simple visualization methods to improve the interpretability of the deep learning model. The proposed methods are based on the class activation map (CAM) framework. On the other hand, we propose a quantitative metric, the acceptable mask ratio, to evaluate the interpretability of the deep learning model rather than visual observation. A higher score of acceptable mask ratio means that the highlighted discriminative regions are more precisely located within the lungs. Therefore, we can assess different methods intuitively. We evaluate the proposed methods on our Chest X-ray dataset. The experimental results show that the proposed methods can improve the interpretability of the deep learning model and highlight the discriminative regions more precisely located within the lungs to some extent.

In the acceptable mask ratio analysis, the average performance of exponential performance is slightly better than CAM. Compared to Grad-CAM, exponential CAM performs more stably in various models and classes. Proportional CAM performs worst in general. However, it outperforms other methods in different classes on specific models (Figure 5.17). Probabilistic CAM is superior to other approaches in the bacteria class most of the time. The bacteria class has the most training images. Compared to exponential CAM, proportional CAM and probabilistic CAM are more sensitive to the models.

In the scenario of clinical decision-making, the interpretability of the deep learning model is very important. With the help of the acceptable mask ratio, we can evaluate the interpretability of deep learning models to some extent. Therefore, it assists in choosing a model for an automated diagnosis system with both high accuracy and interpretability. In addition, the proposed methods can be used to explain the classification results of the deep learning model and help clinicians to have a more trustworthy diagnosis model.

However, it is still difficult to conclude to what extent the classification results of machine learning on CXR images are reasonable. Different scenarios can have different requirements. For example, in the scenario of distinguishing normal people and pneumonia patients, we can accept the results from a model happening within the lungs in more than half cases. However, in the scenario of distinguishing COVID-19 and other pneumonia, we need to be more careful. We expect the detections not only to happen within the lungs as much as possible but also to show different lesion localizations according to different infections.

Besides, we compare the visualization with multiple CNN architectures and global pooling methods. There is no significant correlation between the performance of different CNN architectures and the interpretability of the models. Better performance of the model does not necessarily

correspond to better interpretability. Furthermore, the higher performance of the model does not indicate that the model prefers to make decisions in the lung region. Meanwhile, the effect of different global pooling methods on lesion localization seems to be more pronounced than using various network architectures. For different global pooling methods with the same CNN architecture, the interpretability of models does not show a significant difference (Figure A.4).

Appendix A

Attachments

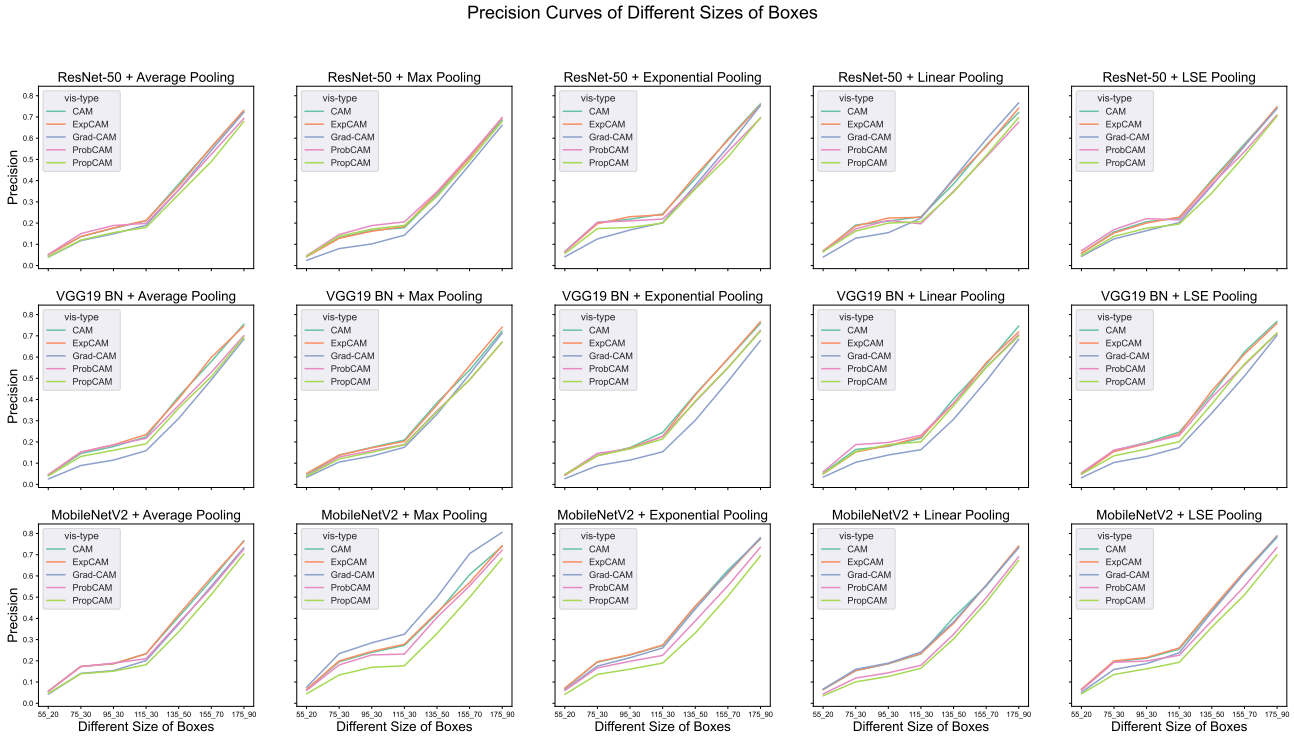


Figure A.1: Precision Curves of Various Visualization Methods among 15 Models

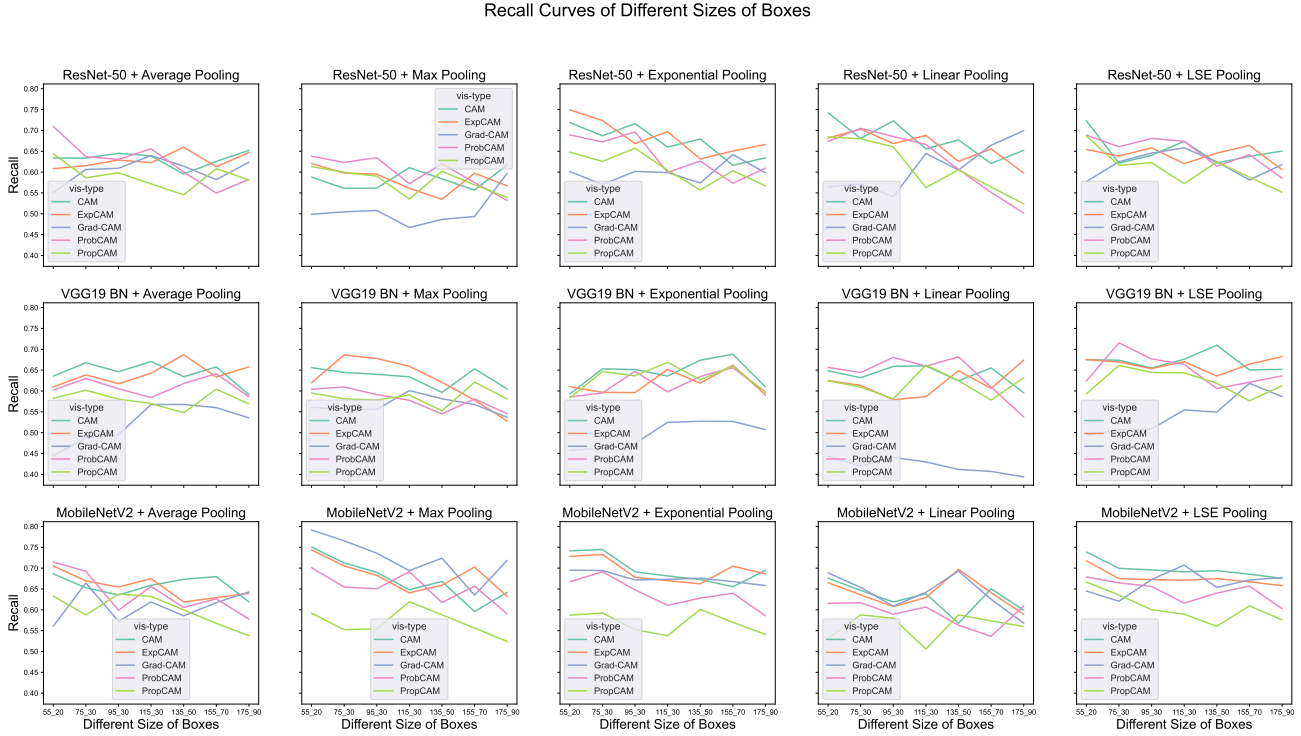


Figure A.2: Recall Curves of Various Visualization Methods among 15 Models

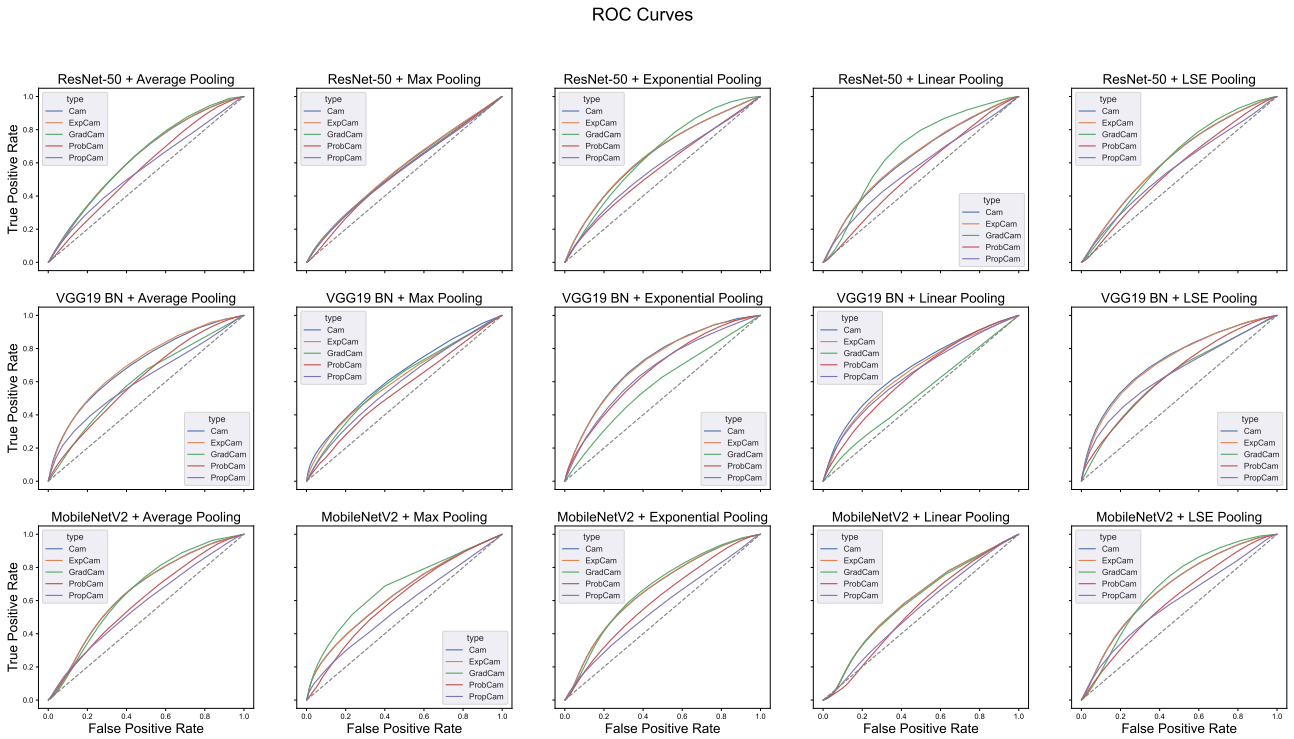


Figure A.3: ROC Curves of Various Visualization Methods among 15 Models

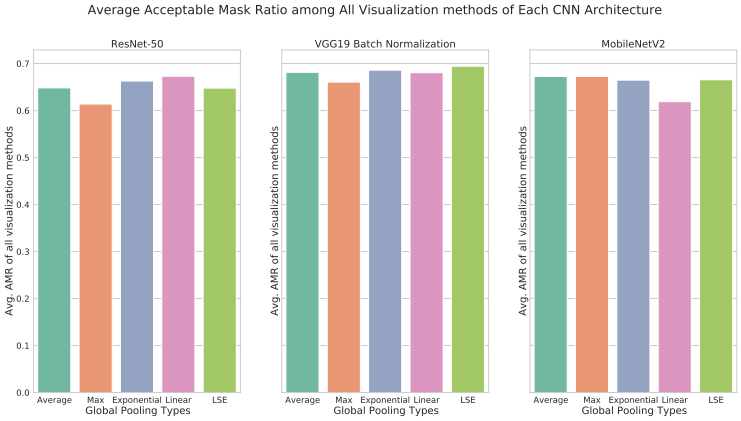


Figure A.4: Average Acceptable Mask Ratio among All Visualization methods of Each CNN Architecture. It illustrates the performance of different visualization methods on different global pooling methods with the same CNN architecture.

List of Figures

1.1 Partial results from (Tang et al., 2021). Dark red regions highlight the discriminative area used for image classification. . . . 3

2.1 Number of Studies Utilizing Explanatory Techniques for COVID-19 Detection in CXR Images among literature Reviewed . . . 6

3.1 Structure of Global Pooling [1] . . . 9
3.2 Different global pooling examples . . . 10

4.1 The Architecture of Classification Model . . . 14
4.2 Pipeline of Visualization Using Grad-CAM . . . 14
4.3 Pipeline of Visualization Using CAM . . . 15
4.4 Pipeline of Visualization Using Probabilistic CAM . . . 16
4.5 Examples of AMRs. The blue box is a masked area. The red box is the discriminative area. . . . 17

5.1 CXR images examples . . . 20
5.2 Distribution of Training Dataset . . . 22
5.3 Confusion matrix for Binary Classification . . . 25
5.4 Illustration of ROC Curve and EER point . . . 27
5.5 Examples of bounding boxes in CXR images heapmap . . . 29
5.6 Precision Curves and Recall Curves of Various Sizes of Bounding Boxes . . . 30
5.7 Image Segmentation on Test Images to Localize Lung Regions . . . 31
5.8 ROC Curves . . . 32
5.9 ROC Curves . . . 32
5.10 Box Plot Anatomy [2] . . . 35
5.11 Overall Acceptable Mask Ratio of 15 models . . . 36
5.12 Average Acceptable Mask Ratio of Four Lung Infections of 15 models . . . 37
5.13 Partial Results of Average Acceptable Mask Ratio of One Model . . . 38
5.14 Partial Results of Average Acceptable Mask Ratio of One Model . . . 38
5.15 Partial Results of Acceptable Mask Ratio . . . 39
5.16 Partial Results of Acceptable Mask Ratio . . . 39
5.17 Acceptable Mask Ratio of Each Lung Infection of 15 models . . . 40
5.18 Test Images Using for Statistical Distribution Heatmap . . . 41
5.19 Statistical Distribution Heatmaps of the Model with ResNet-50 and Exponential Pooling . . . 42
5.20 Statistical Distribution Heatmaps of the Model with ResNet-50 and Max Pooling . . . 43

6.1 Examples of Different CAMs Generated from the Model with ResNet-50 and Global Average Pooling. Different methods visualize four images from the class of bacteria, virus, COVID-19, and SARS, respectively. These images are labeled with semantic bounding boxes. The first column shows the visualization results of CAM among four classes, and so on. . . . 46

A.1 Precision Curves of Various Visualization Methods among 15 Models . . . 52
A.2 Recall Curves of Various Visualization Methods among 15 Models . . . 53
A.3 ROC Curves of Various Visualization Methods among 15 Models . . . 54

A.4 Average Acceptable Mask Ratio among All Visualization methods of Each CNN Architecture. It illustrates the performance of different visualization methods on different global pooling methods with the same CNN architecture. 55

List of Tables

5.1 Dataset Summary 20
5.2 The Number of Images in Each Set 20
5.3 Summary of Defined Parameter of Proposed Models 24
5.4 Summary of the Performance of Multi-class Classification Models 28

List of Listings

Bibliography

Afshar, P., Heidarian, S., Naderkhani, F., Oikonomou, A., Plataniotis, K. N., and Mohammadi, A. (2020). COVID-CAPS: A capsule network-based framework for identification of COVID-19 cases from X-ray images. *Pattern Recognition Letters*, 138:638–643.

Alghamdi, H. S., Amoudi, G., Elhag, S., Saeedi, K., and Nasser, J. (2021). Deep learning approaches for detecting COVID-19 from chest X-ray images: A survey. *Ieee Access*, 9:20235–20254.

Altaf, F., Islam, S. M., Akhtar, N., and Janjua, N. K. (2019). Going deep in medical image analysis: concepts, methods, challenges, and future directions. *IEEE Access*, 7:99540–99572.

Basu, S., Mitra, S., and Saha, N. (2020). Deep learning for screening COVID-19 using chest X-ray images. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2521–2527. IEEE.

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Chattopadhay, A., Sarkar, A., Howlader, P., and Balasubramanian, V. N. (2018). Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE.

Dixit, A., Mani, A., and Bansal, R. (2021). COVIDetect-DESVM: Explainable framework using differential evolution algorithm with SVM classifier for the diagnosis of COVID-19. In *2021 4th International Conference on Recent Developments in Control, Automation & Power Engineering (RDCAPE)*, pages 339–334. IEEE.

Drain, P. K. (2022). Rapid diagnostic testing for SARS-CoV-2. *New England journal of medicine*, 386(3):264–272.

Funer, F. (2022). Accuracy and interpretability: Struggling with the epistemic foundations of machine learning-generated medical information and their practical implications for the doctor-patient relationship. *Philosophy & Technology*, 35(1):1–20.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Haibo, H. and Yunqian, M. (2013). Imbalanced learning: foundations, algorithms, and applications. *Wiley-IEEE Press*, 1:27.

Hasan, M. J., Alom, M. S., and Ali, M. S. (2021). Deep learning based detection and segmentation of COVID-19 & pneumonia on chest X-ray image. In *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*, pages 210–214. IEEE.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Karim, M. R., Döhmen, T., Cochez, M., Beyan, O., Rebholz-Schuhmann, D., and Decker, S. (2020). DeepCOVIDExplainer: Explainable COVID-19 diagnosis from chest X-ray images. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1034–1037.

Khosravi, M. R. and Samadi, S. (2021). BL-ALM: A blind scalable edge-guided reconstruction filter for smart environmental monitoring through green IoMT-UAV networks. *IEEE Transactions on Green Communications and Networking*, 5(2):727–736.

Kim, H. E., Cosa-Linan, A., Santhanam, N., Jannesari, M., Maros, M. E., and Ganslandt, T. (2022). Transfer learning for medical image classification: a literature review. *BMC medical imaging*, 22(1):1–13.

Kim, M.-J., Kang, D.-K., and Kim, H. B. (2015). Geometric mean based boosting algorithm with over-sampling to resolve data imbalance problem for bankruptcy prediction. *Expert Systems with Applications*, 42(3):1074–1082.

Lai, C.-C., Shih, T.-P., Ko, W.-C., Tang, H.-J., and Hsueh, P.-R. (2020). Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) and coronavirus disease-2019 (COVID-19): The epidemic and the challenges. *International journal of antimicrobial agents*, 55(3):105924.

Lam, T. T.-Y., Jia, N., Zhang, Y.-W., Shum, M. H.-H., Jiang, J.-F., Zhu, H.-C., Tong, Y.-G., Shi, Y.-X., Ni, X.-B., Liao, Y.-S., et al. (2020). Identifying SARS-CoV-2-related coronaviruses in Malayan pangolins. *Nature*, 583(7815):282–285.

Lin, C., Chen, Z., Xie, B., Sun, Z., Ding, Y., Li, X., Niu, M., Guo, S., and Lei, J. (2020). COVID-19 pneumonia patient without clear epidemiological history outside Wuhan: An analysis of the radiographic and clinical features. *Clinical imaging*, 65:82–84.

Lin, M., Chen, Q., and Yan, S. (2014). Network in network. In Bengio, Y. and LeCun, Y., editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Minaee, S., Kafieh, R., Sonka, M., Yazdani, S., and Soufi, G. J. (2020). Deep-COVID: Predicting COVID-19 from chest X-ray images using deep transfer learning. *Medical image analysis*, 65:101794.

Montavon, G., Binder, A., Lapuschkin, S., Samek, W., and Müller, K.-R. (2019). Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 193–209.

Narin, A., Kaya, C., and Pamuk, Z. (2003). Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks. arxiv 2020. *arXiv preprint arXiv:2003.10849*.

Oran, D. P. and Topol, E. J. (2021). The proportion of SARS-CoV-2 infections that are asymptomatic: a systematic review. *Annals of internal medicine*, 174(5):655–662.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Pinheiro, P. O. and Collobert, R. (2015). From image-level to pixel-level labeling with convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1713–1721.

Prechelt, L. (1998). Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer.

Ratul, M. A. R., Elahi, M. T., Yuan, K., and Lee, W. (2020). RAM-Net: a residual attention MobileNet to detect COVID-19 cases from chest X-ray images. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 195–200. IEEE.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Salamon, J., McFee, B., Li, P., and Bello, J. P. (2017). Multiple instance learning for sound event detection. *Detection and Classification of Acoustic Scenes and Events*, 2017.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote sensing of Environment*, 62(1):77–89.

Tang, G. S., Chow, L. S., Solihin, M. I., Ramli, N., Gowdh, N. F., and Rahmat, K. (2021). Detection of COVID-19 using deep convolutional neural network on chest X-Ray (CXR) images. In *2021 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–6. IEEE.

Tronci, R., Giacinto, G., and Roli, F. (2009). Dynamic score combination: A supervised and unsupervised score combination method. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 163–177. Springer.

Van Rossum, G. and Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA.

Wang, B. X. and Japkowicz, N. (2010). Boosting support vector machines for imbalanced data sets. *Knowledge and information systems*, 25(1):1–20.

Wang, Y., Li, J., and Metze, F. (2019). A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 31–35. IEEE.

Yohanandan, S., Song, A., Dyer, A. G., and Tao, D. (2018). Saliency preservation in low-resolution grayscale images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 235–251.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27.

Zhang, J., Chen, N., Zhao, D., Zhang, J., Hu, Z., and Tao, Z. (2022). Clinical characteristics of COVID-19 patients infected by the omicron variant of SARS-CoV-2. *Frontiers in Medicine*, 9.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929.