



**MANSOURA UNIVERSITY
FACULTY OF ENGINEERING
ELECTRONICS & COMMUNICATION ENGINEERING DEPARTMENT**

**MACHINE-LEARNING BASED MODEL FOR CHEST
DISEASES DETECTION**

Project members

	ID number
Ahmed Gamal Mohamed Kamal Simeda	1000103832
Amr AboAlyazied Zaki AboAlyazied	1000151214
Ibrahim Mohamed Mahmoud Basal	1000103886
Mohamed Talaat Mohamed Qandeel	1000103599
Zakaria Abdallah Halim Rezk	1000103154

Supervisor

Assist. Prof. Doaa Adel Altantawy

Academic Year: 2021 – 2022

July 2022

TABLE OF CONTENTS

ABSTRACT	vi
LIST OF ABBREVIATIONS	vii
1 Chapter (1) Introduction	2
2 Chapter (2) Literature	7
3 Chapter (3) Pre-processing.....	23
3.1 Introduction	23
3.2 Top-hat and Bottom-hat Transform	24
3.3 Go-Dec based Radiographs sparsification.....	30
3.4 Fourier Transform	34
4 Chapter (4) Transfer Learning and SOTA models.....	38
4.1Transfer Learning	38
4.2 SOTA models.....	43
5 Chapter (5) Experimental Results and Discussion	54
5.1 Dataset.....	54
5.2 Experimental Results.....	57
5.2.1 The effect of image size on the performance	57
5.2.2 The effect of the proposed preprocessing techniques on the performance	63
5.2.3 The proposed ensemble model architecture	67
Conclusion.....	70
References.....	72

LIST OF FIGURES

Figure 1-1 Different Machine Learning tasks.....	4
Figure 2-1.1 Difference between M-class and Multi-label classification.	7
Figure 2-1.2 Difference between M-class and Multi-label classification.	7
Figure 2-2 O. Stephan's proposed architeture	8
Figure 2-3 DarkCovidNet.....	9
Figure 2-4 Workflow of proposed cascaded deep learning.....	10
Figure 2-5 Workflow of AXIR Net.	11
Figure 2-6.1 Center cropped image.....	12
Figure 2-6.2 multi-GAP's full model structure	12
Figure 2-7 The flowchart of MARnet identifies chest X-ray images.	13
Figure 2-8 The GS and Pre-trained model aided method.	14
Figure 2-9 The proposed COVID-19 X-ray classification model for training.	15
Figure 2-10 The DualNet architecture.....	16
Figure 2-11 The proposed COVID-19 X-ray classification model for training.	17
Figure 2-12 Full architecture of VDSNet.....	18
Figure 2-13 The architecture of the multi-label ne-grained network.....	19
Figure 2-14 Illustration of the work training strategy.....	19
Figure 2-15 A Two-stage Deep AUC Maximization Framework.....	20
Figure 3-1 Tanslation Operation.	25
Figure 3-2.1 illustration of Dilation and Erosion.	26
Figure 3-2.2 illustration of Dilation and Erosion	26
Figure 3-3 illustration of Opening and Closing	27
Figure 3-4 Effect of Top-hat and Bottom-hat Transformation	29
Figure 3-5 Image Sparsification by GoDec Algorithm	31
Figure 3-6 Effect of Applying HPF On the Fourier-transformed image.....	36
Figure 4-1 Illustration of Transfer learning concept.	38
Figure 4-2 How Transfer learning works.....	39
Figure 4-3 Pretraining models and Importance of Transfer learning.....	40
Figure 4-4 Extraction of Features.....	42
Figure 4-5 ResNet-18 Archetecture	44
Figure 4-6 ResNet-32 Architecture.....	45
Figure 4-7 ResNet-50 Architecture.....	46
Figure 4-8 ResNet-101 Architecture.....	47
Figure 4-9 Xception Architecture	47
Figure 4-10 MobileNet-V1 Architecture	48
Figure 4-11 MobileNet-V2 Architecture	49
Figure 4-12 MobileNet-V3 building block.....	49

Figure 4-13 Dense-121 Architecture	50
Figure 4-14 Dense-169 Architecture.	51
Figure 4-15 Vision Transformer building blocks.	51
Figure 4-16 Efficient Net Building blocks.	52
Figure 5-1 Percentage of Uncertainty Labels.....	52
Figure 5-2 Imputation Algorithms suggested by Stanford ML group.....	55
Figure 5-3 Age and Gender Distribution.....	56
Figure 5-4 Diseases Distribution.....	57
Figure 5-5 The Behaviour of Models when trained on image sizes of 224.....	59
Figure 5-6 The Behaviour of Models when trained on image sizes of 256.....	61
Figure 5-7 The Behaviour of Models when trained on image sizes of 320.....	63
Figure 5-8 Extracting the trained weights.....	64
Figure 5-9 ROC Curves of ensemble model's performance on the validation.....	68
Figure 5-10 The proposed ensemble model architecture	69

ABSTRACT

Chest radiography is one of the most common types of diagnostic radiology exams, which is critical for screening and diagnosis of many different thoracic diseases. Specialized algorithms have been developed to detect several specific pathologies such as lung nodules or lung cancer. However, accurately detecting the presence of multiple diseases from chest X-rays (CXRs) is still a challenging task. This study presents a supervised multi-label classification framework based on deep convolutional neural networks (CNNs) plus some proposed pre-preprocessing techniques for predicting the presence of 5 hard-detecting thoracic diseases. We tackled this problem by training state-of-the-art CNNs like ResNet, DenseNet, and EfficientNet. We also propose to use preprocessing techniques for a better handling feature extraction process in the framework flow. Our proposed ensemble model is trained on over 200,000 CXRs of the recently released CheXpert dataset and achieves a mean area under the curve (AUC) of 93.3% in predicting 5 selected pathologies from the validation set. This is among the highest AUC score yet reported to date.

LIST OF ABBREVIATIONS

CXR	Chest X-Ray
SOTA	State-Of-The-Art
AUC	Area Under the Curve
TB	Tob-hat and Bottom-hat transform
GoDec	Go Decomposition Algorithm
SVD	singular value decomposition
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
HPF	High-Pass Filter

CHAPTER 1

INTRODUCTION

1 Chapter (1) Introduction

Lung diseases threaten the global health because of the increasing rate of pollution and smoking especially in developing nations where billions face energy poverty and rely on polluting forms of energy. According to statistical data from the World Health Organization, lung diseases rank the third among all causes of death worldwide. Lung diseases lead to the death of more than five million people worldwide each year. As an important organ of the respiratory system, the lungs connect with the outside and the whole blood flows, so the lungs are vulnerable to internal and external microbial attack. Pneumonia is an inflammation of the lung parenchyma caused by pathogenic microbiological, physical and chemical factors, immune damage, allergies and drugs. Bacterial pneumonia is the most common one, and its chest X-ray image shows pulmonary exudation or solid shadow. Lung infiltration is there are some abnormal cells or foreign body in the lungs. Lung infiltration is caused by bacteria, fungi and viruses, and infiltrated shadow can be seen on chest X-ray images. Exudative lesions are acute inflammation, mostly caused by bacterial, viral or fungal infections, and its chest X-ray images will appear cloudy shadows. the WHO estimates that over 4 million premature deaths occur annually from household air pollution-related diseases including pneumonia. Over 150 million people get infected with Lung diseases on an annual basis especially children under 5 years old. in Africa's 57 nations, a gap of 2.3 million doctors and nurses exists. Chest X-ray image is the most commonly medical imaging technology in clinical diagnosis of lung diseases, which is very effective in the identification and detection of cardiothoracic, pulmonary and interstitial diseases and plays an important role in the treatment of lung diseases. Accurate analysis of patient's health information is a great challenge for radiologists. So computer-aided diagnosis system is proposed to control the differences among radiologists and provide reference for clinicians. In recent years. For these populations, accurate and fast diagnosis means everything. It can guarantee timely access to treatment and save much needed time and money for those already experiencing poverty according to the estimation of WHO[1]. The chest radiograph (chest X-ray, or CXR for short) is one of the most requested radiologic examinations for pulmonary/respiratory diseases, such as chronic obstructive pulmonary disease (COPD),

asthma, acute lower respiratory tract infections, tuberculosis (TB), and lung cancer. These five prementioned diseases are among the most common causes of severe illness and death worldwide[2].

X-ray screening is economical, making it more suitable for developing and low-income countries. Moreover, each scan can detect multiple suspected pathologies such as tuberculosis and pneumonia, etc. Computer-Aided Detection (CADe) and Diagnosis (CADx) has been a major research focus in medicine. CAD for chest X-ray could potentially become a cost-effective assistive tool for radiologists, and relatively fast as it takes no time, compared to other methods, such as Computed Tomography (CT) scan and Magnetic resonance imaging (MRI). Hence, an automated system that provides secondary readings for radiologists without any extra effort would help them not to miss any important details. In addition, some regions have limited access to trained radiologists, so making an accurate automated system that accurately detects thoracic diseases from chest x-rays would be of a great help.

Recently, deep learning has attracted lots of attention due to its high performance on different tasks e.g., detection, segmentation, and classification. for example in Fig. (1-1), The tasks of deep learning currently utilized in health care are divided into classification, detection, and segmentation. Noticeably, Deep learning-based approaches have proved their effectiveness in healthcare problems. Deep learning has been increasingly applied in the field of medical image, especially image classification. Related researches mainly include the automatic classification of lung tuberculosis, lung nodules detection, lung cancer detection, pneumonia detection, pneumothorax detection and COVID-19 prediction. Medical Imaging is not an exception, huge efforts were made to ease the process of medical image screening. One of these exerted efforts is an open competition launched by Stanford Machine Learning group, called CheXpert[3]. In addition, CheXpert is a large high-quality dataset contains over 220,000 chest x-ray images for more than 64,500 different patients. In total, the dataset contains 14 different diseases, but the competition is focusing on only 5 selected high similar and overlapping

ones: Cardiomegaly, Edema, Consolidation, Atelectasis, Pleural Effusion which makes this task a very challenging problem.

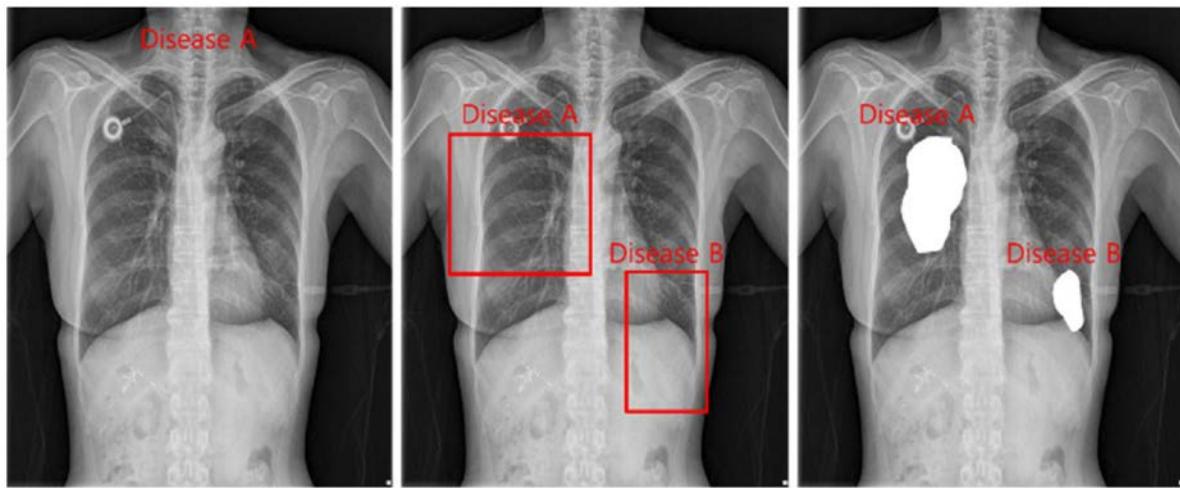


Figure (1-1) Classification, Detection, and Segmentation

Our motivation is to introduce an automated diagnostic model, based on a deep convolutional neural network that can differentiate between different lung diseases using chest X-ray images provided by CheXpert dataset. Hence, we are interested in a deep learning-based multi-label diagnostic problem. The contributions of Our work can be summarized as follows:

1. We proposed three different preprocessing techniques: Top-down Transform, “Go Decomposition” algorithm, and Fourier Transform-based technique
2. Using CheXpert as our case study, we trained many state-of-the-art (SOTA) models leaving a full analysis of the performance of different models on CheXpert and how changing image size would help in diseases detection. This analysis will be very insightful and helpful for anyone who is taking their first steps in CheXpert competition.

3. By making use of the insights provided by our analysis, we have proposed a deep learning-based ensemble model which surpassed the baseline provided by Stanford ML group and scored a very satisfying result compared to complex architectures on the leaderboard.
 4. The fine similarity and overlapping (multi-label diagnosis) of lung diseases make this problem a very challenging one. Consequently, our study is one of a few studies that proposes a solution for such an issue

The Book will be organized as follows: Chapter (2) describes the literature and how researchers were dealing with chest x-ray images. A detailed discussion of the proposed filters are presented in Chapter (3). A quick overview over the concept of Transfer Learning and investigation of different state-of-the-art (SOTA) models architectures are discussed in Chapter (4). A detailed description of CheXpert dataset and our full Analysis is given in Chapter (5) with the architecture of the proposed ensemble model. At the end, Section 6 concludes our work.

CHAPTER 2

LITERATURE

2 Chapter (2) Literature

This section will review the different approaches taken to tackle the problem of chest x-ray (CXR) diagnosis. We will divide our literature into two main parts: M-class classification and multi-label classification.

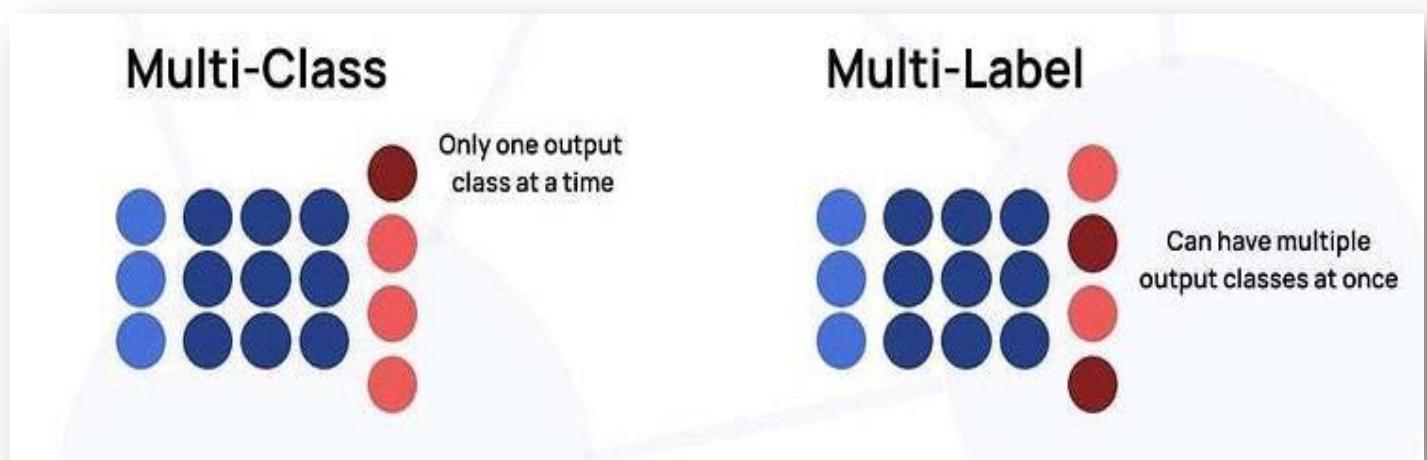


Figure (2-1.1) Difference between classification and multi-label classification Tasks

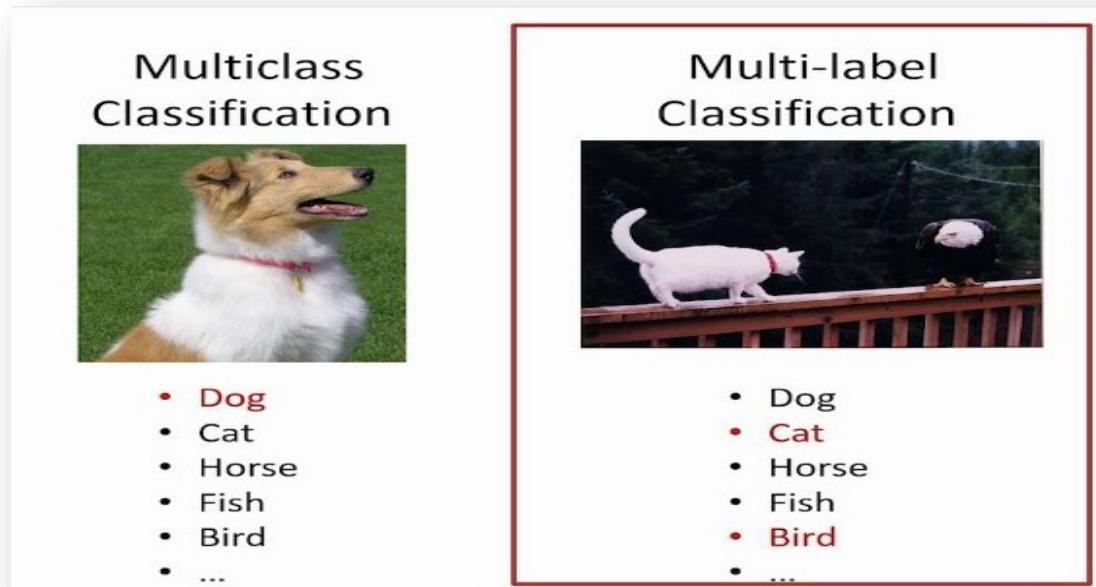


Figure (2-1.2) Difference between classification and multi-label classification Tasks

M-class problem: here, each training example has only one label out of M ones (depends on how many classes we are trying to classify). Many researchers had done a great effort to tackle such a problem. The first Computer-aided diagnosis (CAD) system for detecting lung nodules or affected lung cells was launched in the late 1980s but those efforts were not enough. This is due to the lack of efficient computational resources for the implementation of complex architectures and advanced image processing techniques at that time. After the successful invention of GPUs and CNNs, there was a revolutionary improvement of the performance of the CAD systems. S. Bhatia[4] Proposed an approach to detect lung cancer from CT scans using deep residual learning. They delineate a pipeline of preprocessing techniques starting from extracting the central region of interest of the lungs going through a feature extraction stage using residual networks and finally feeding these features to various classifiers in an ensemble manner to predict the likelihood of a CT scan being cancerous. They achieved an accuracy of 84% on the Lung Image Database Consortium image collection (LIDC-IDRI). O. Stephen[5] This study proposes a convolutional neural network model as shown Fig. (2-2) trained from scratch to classify and detect the presence of pneumonia from a collection of chest X-ray image samples. Unlike other methods that rely heavily on transfer learning approaches or traditional handcrafted techniques to achieve a remarkable classification performance, they constructed a convolutional neural network model from scratch to extract features from a given chest X-ray image and classify it to determine if a person is infected with pneumonia. Unlike other deep learning classification tasks with sufficient image repository, it is difficult to obtain a large amount of pneumonia dataset for this classification task; therefore, they deployed several data augmentation algorithms to improve the validation and classification accuracy of the CNN model and achieved remarkable validation accuracy. the highest accuracy achieved was 95.31% on the training set and 93.73% on the validation set.

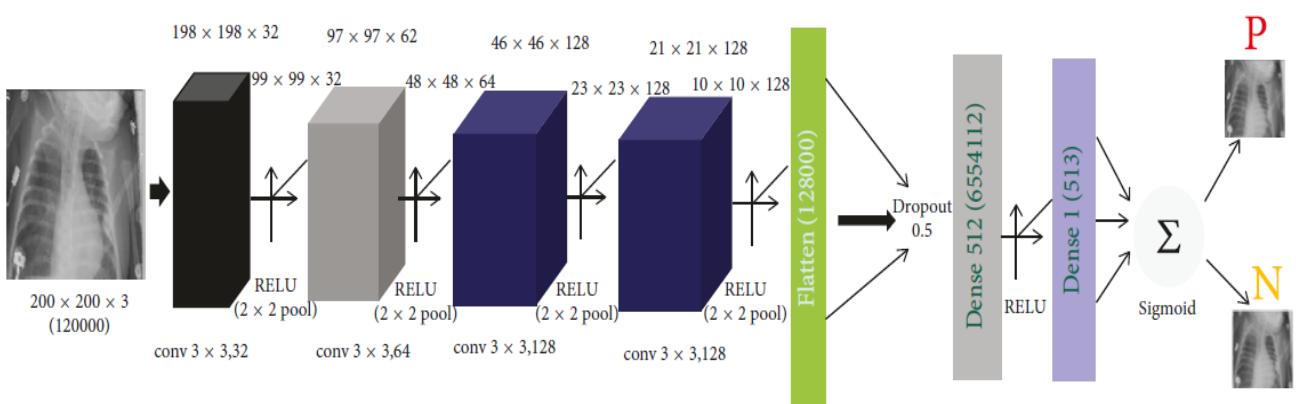
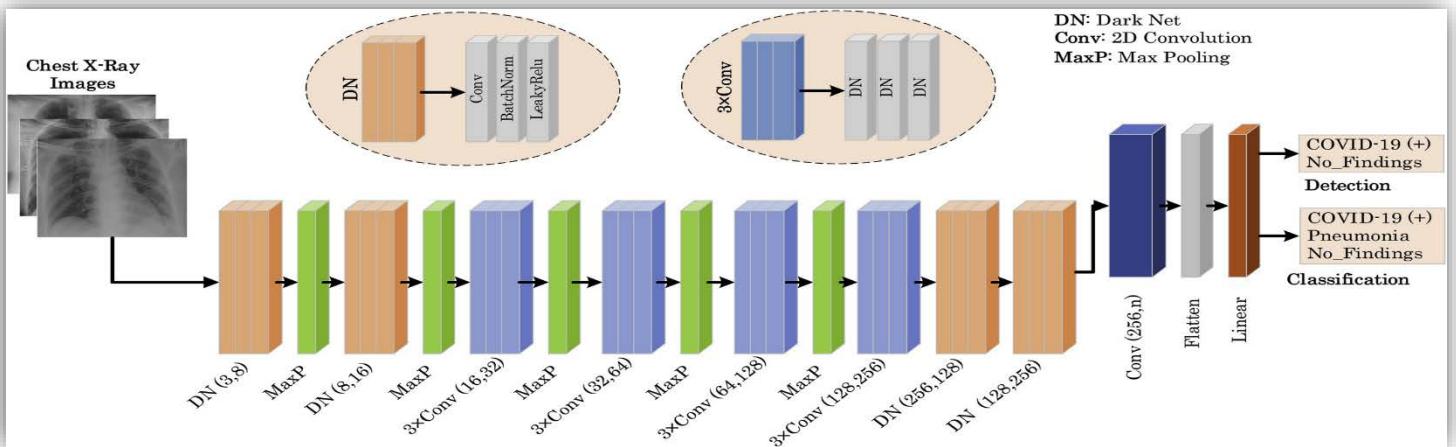


Figure (2-2) O. Stephan's proposed architecture

T. Ozturk et al.[6] proposed a 17-convolution-layer model to perform a binary classification task (e.g., normal and COVID-19) and a multiclass classification task (e.g., normal, pneumonia, and COVID-19). Their model is fully automated with an end-to-end structure without the need for manual feature extraction. Using a relatively small data, this framework managed to achieve up to 98.08% and 87.02% accuracy for binary and multi-class classification respectively.



Figure(2-3) DarkCovidNet.

M. E. Karar[7] This article proposes a new framework of cascaded deep learning classifiers to enhance the performance of these CAD systems for highly suspected COVID-19 and pneumonia diseases in X-ray images. Their proposed deep learning framework constitutes two major advancements as follows. First, complicated multi-label classification of X-ray images have been simplified using a series of binary classifiers for each tested case of the health status. That mimics the clinical situation to diagnose potential diseases for a patient. Second, the cascaded architecture of COVID-19 and pneumonia classifiers is flexible to use different fine-tuned deep learning models simultaneously, achieving the best performance of confirming infected cases. This study includes eleven pre-trained convolutional neural network models, such as Visual Geometry Group Network (VGG) and Residual Neural Network (ResNet). They have been successfully tested and evaluated on public X-ray image dataset for normal and three diseased cases. The results of proposed cascaded classifiers showed that VGG16, ResNet50V2, and Dense Neural Network (DenseNet169) models achieved the best detection accuracy of COVID-19, viral (non-COVID-19) pneumonia, and bacterial pneumonia images, respectively, they collected a very humble dataset and managed to achieve a mean accuracy of 99.9%.

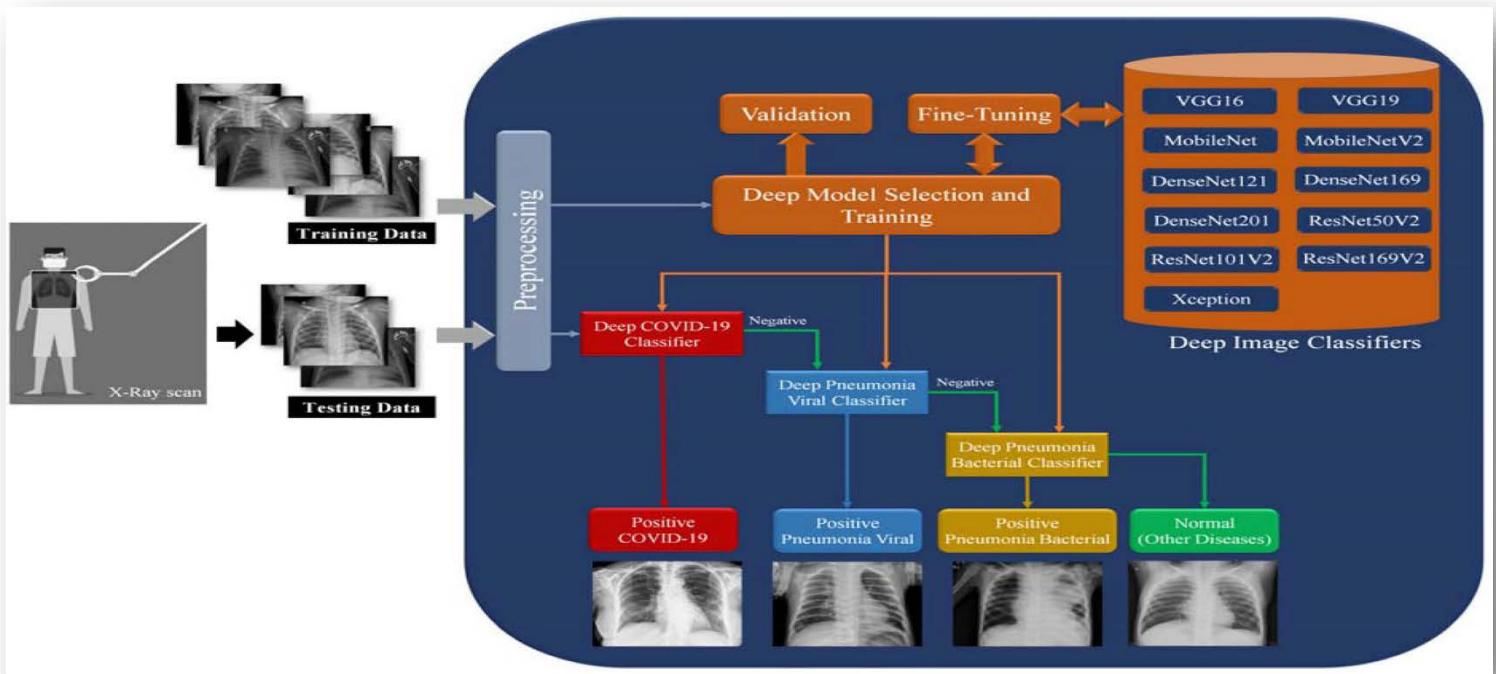


Figure (2-4) Workflow of proposed cascaded deep learning classifiers for confirming COVID-19, viral and bacterial pneumonia cases in chest X-rays.

S.H. Yoo et al[8] This study investigates the feasibility of using a deep learning-based decision-tree classifier for detecting COVID-19 from CXR images. The proposed classifier comprises three binary decision trees, each trained by a deep learning model with convolution neural network based on the PyTorch frame. The first decision tree classifies the CXR images as normal or abnormal. The second tree identifies the abnormal images that contain signs of tuberculosis, whereas the third does the same for COVID-19. The accuracies of the first and second decision trees are 98 and 80%, respectively, whereas the average accuracy of the third decision tree is 95% (100% for [COVID-19/TB classifier] and 89% for [COVID-19/non-TB classifier]).

M. Hong et al.[9] In this study, they propose a multi-class classification method by learning lung disease images with Convolutional Neural Network (CNN). As the image data for learning, the U.S. National Institutes of Health (NIH) dataset divided into Normal, Pneumonia, and Pneumothorax and the Cheonan Soonchunhyang University Hospital dataset including Tuberculosis were used. To improve performance, preprocessing was performed with Center Crop while maintaining the aspect ratio of 1:1. As a Noisy Student of EfficientNet B7, fine-tuning learning was performed using the weights learned from

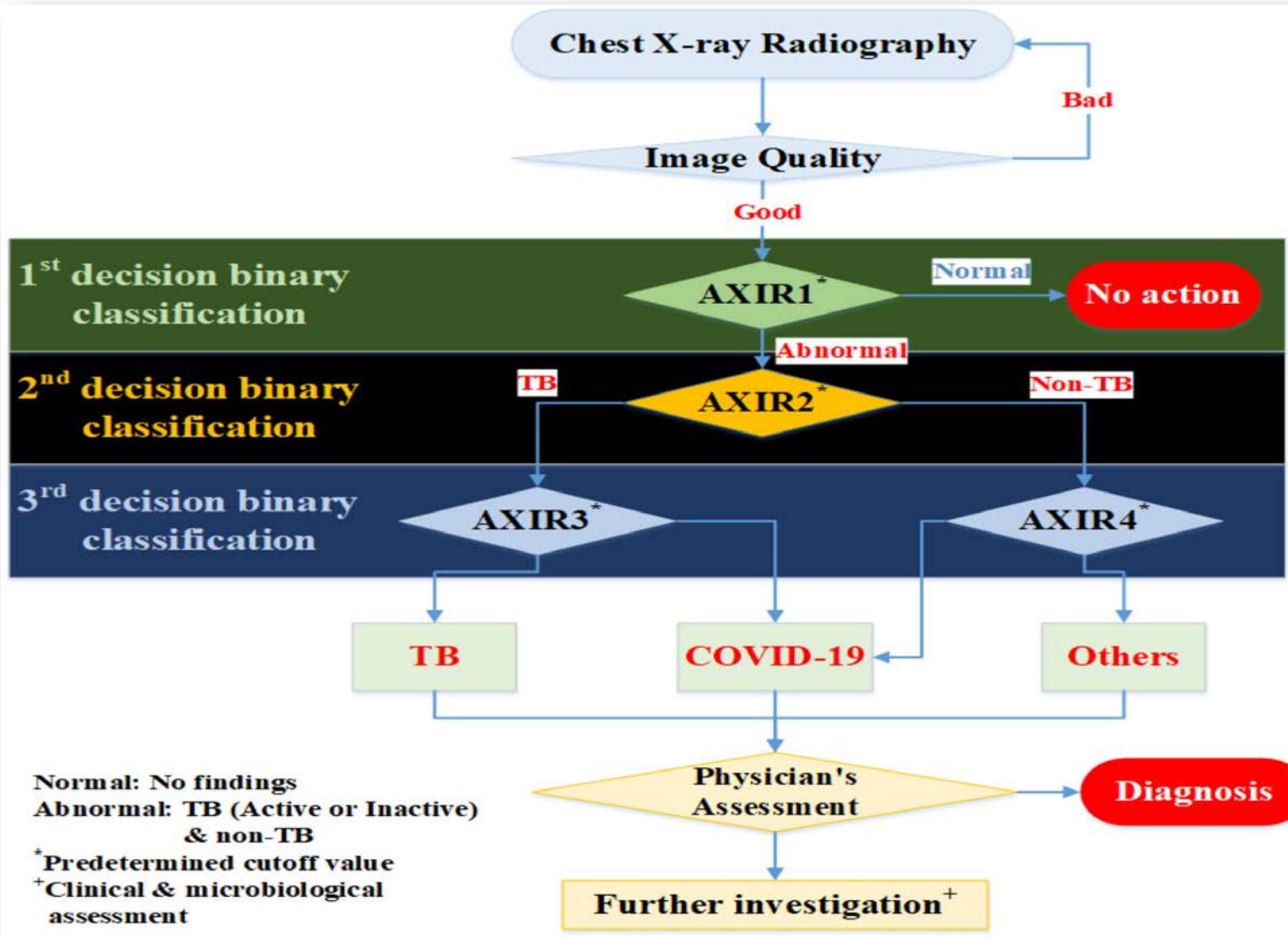


Figure (2-5) Workflow for determining whether chest X-ray image shows a normal, tubercular (TB), or COVID-19 infected lung. AXIR (Automated X-ray Imaging) Radiography system)..

ImageNet, and the features of each layer were maximally utilized using the multi-GAP structure. As a result of the experiment, Benchmarks measured with the NIH dataset showed the highest performance among the tested models with an accuracy of 85.32%, and the four-class predictions measured with data from Soonchunhyang University Hospital in Cheonan had an average accuracy of 96.1%.

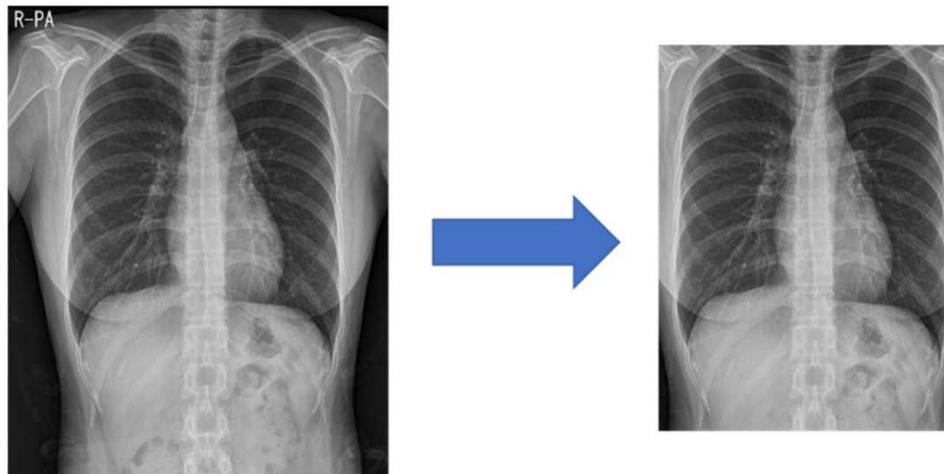


Figure (2-6.1) Center cropped image after processing in a 1:1 ratio.Radiography

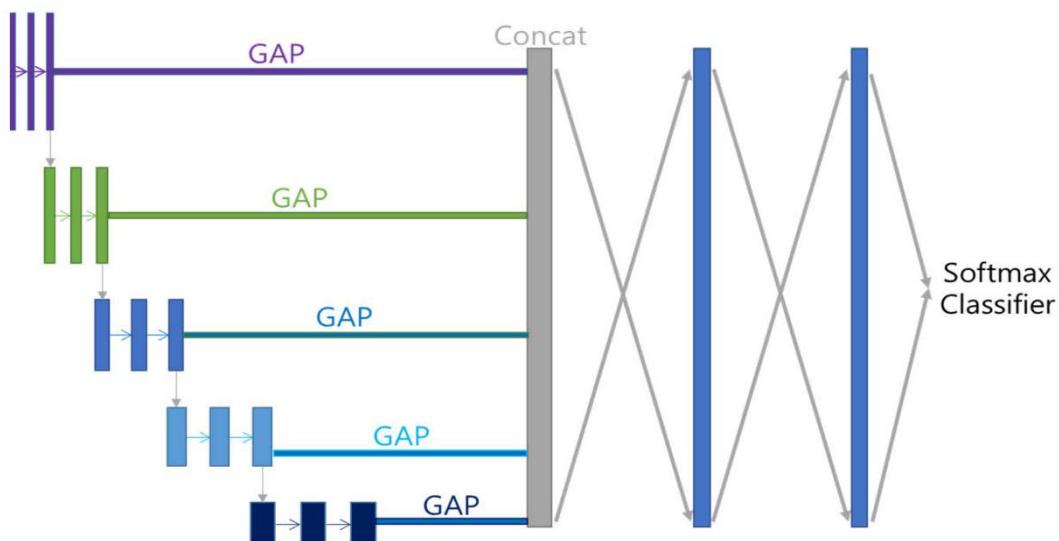


Figure (2-6.2) multi-GAP's full model structure.

B. Wang et al.[10] In this paper, they construct a multi-scale adaptive residual neural network (MARnet) to identify chest X-ray images of lung diseases. To make the model better extract image features, they cross-transfer the information extracted by residual block and the information extracted by adaptive structure to different layer, avoiding the reduction effect of residual structure on adaptive function. They compare MARnet with some classical neural networks, and the results show that MARnet achieves accuracy (ACC) of 83.3% and the area under ROC curve (AUC) of 0.97 in the identification of 4 kinds of typical lung X-ray images including nodules, atelectasis, normal and infection, which

are higher than those of other methods. Moreover, to avoid the randomness of the train-test-split method, 5-fold cross validation method is used to verify the generalization ability of the MARnet model and the results are satisfactory. Finally, the technique called Gradient-weighted Class Activation Mapping (Grad-CAM), is adopted to display significantly the discriminative regions of the images in the form of the heat map, which provides an explainable and more direct clinical diagnostic reference to lung diseases.

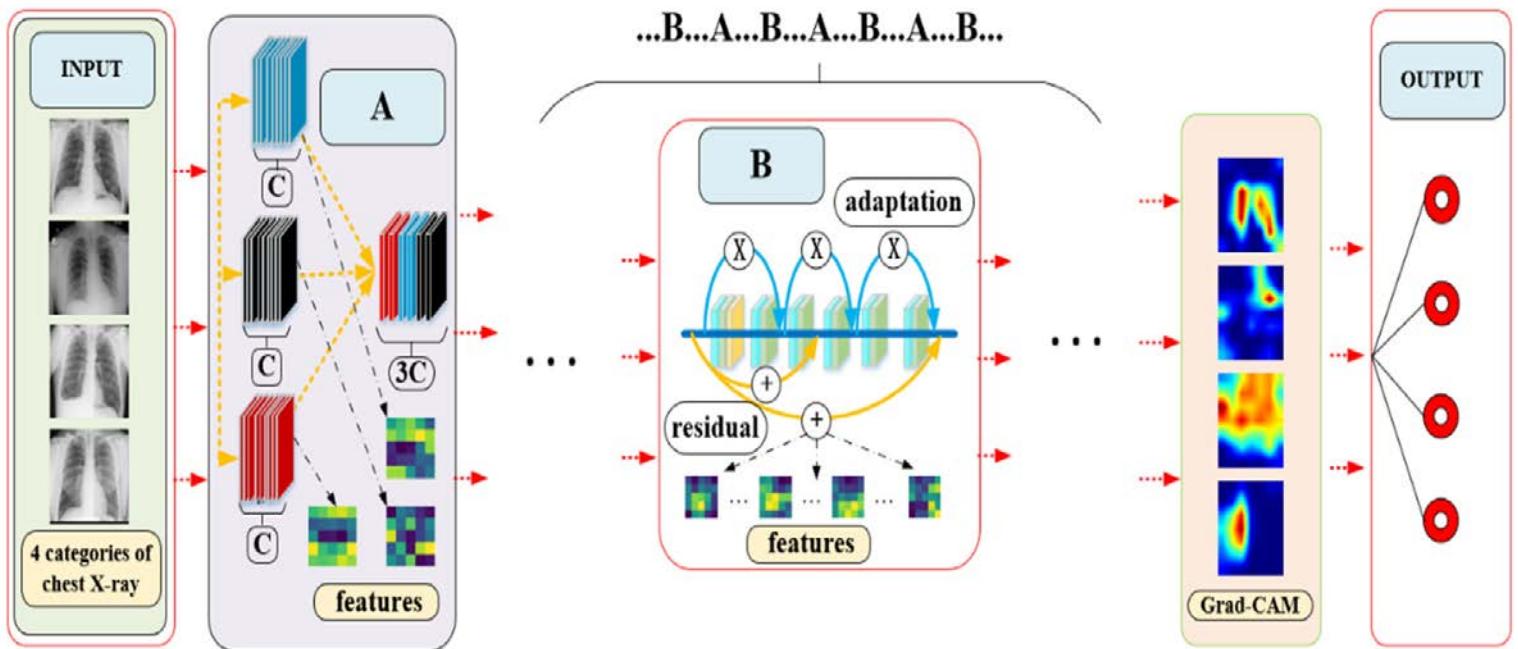


Figure (9) The flowchart of MARnet identifies chest X-ray images.

T. Ozcan[11] In this paper, a grid search (GS) and pre-trained model aided convolutional neural network (CNN) model is proposed to detect COVID-19 in X-Ray images. In the proposed method, the GS method is employed to optimize the hyperparameters of CNN, which directly affects classification performance. Three pre-trained CNN models (GoogleNet, ResNet18 and ResNet50), which can be used for classification, feature extraction and transfer learning purposes were used for transfer learning in this study. The proposed method was trained using the training and validation sub datasets of the collected dataset and detail evaluations are presented according to different performance metrics. The highest accuracy rate (97.69%) was obtained with the GS and ResNet50 aided proposed method.

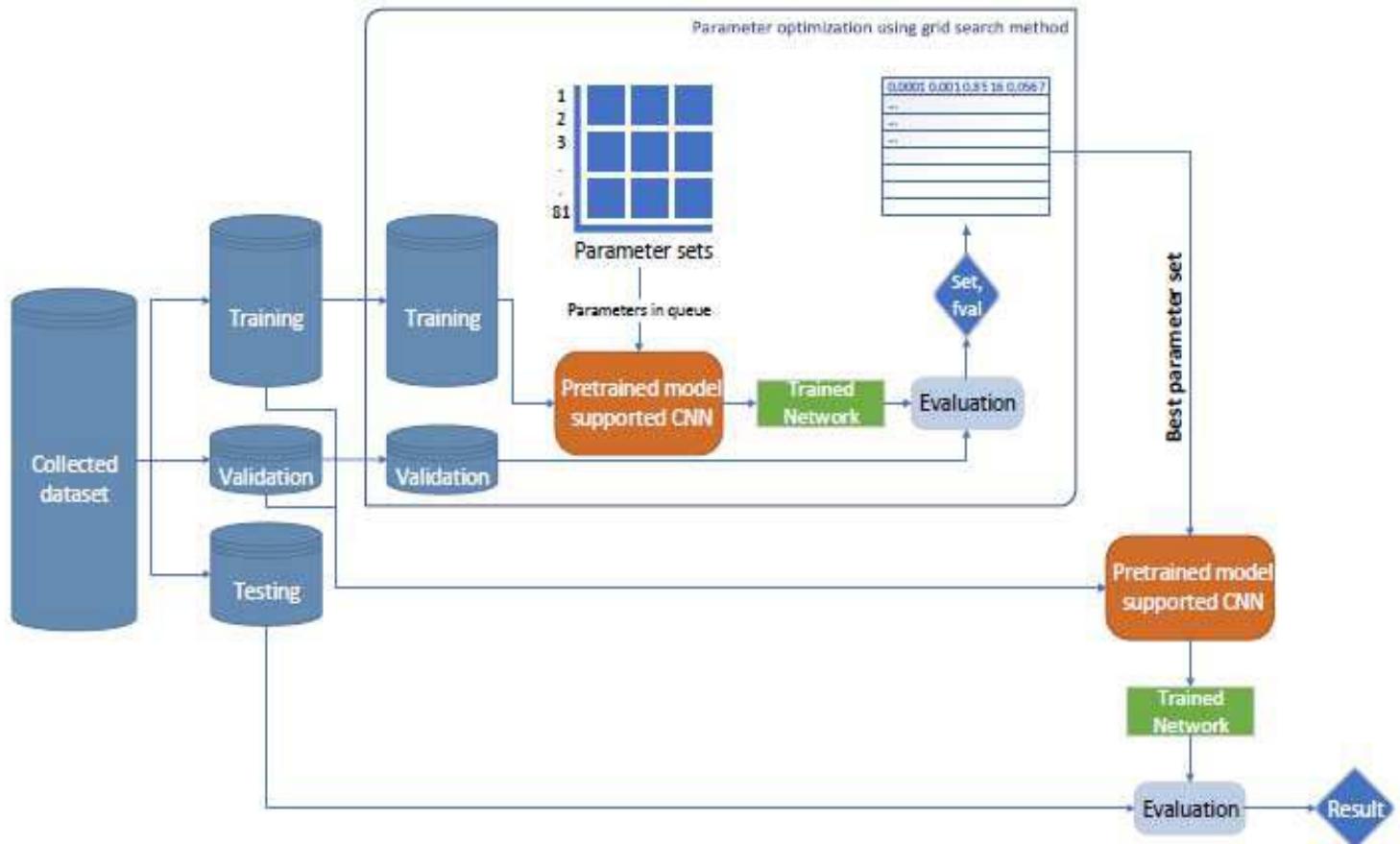


Figure (2-8) The GS and Pre-trained model aided method.

M. Loey, et al.[12] The purpose of this study is to classify chest X-ray images of COVID-19 artefacts in changed real-world situations. A novel Bayesian optimization-based convolutional neural network (CNN) model is proposed for the recognition of chest X-ray images. The proposed model has two main components. The first one utilizes CNN to extract and learn deep features. The second component is a Bayesian-based optimizer that is used to tune the CNN hyperparameters according to an objective function. The used large-scale and balanced dataset comprises 10,848 images (i.e., 3616 COVID-19, 3616 normal cases, and 3616 Pneumonia). In the first ablation investigation, they compared Bayesian optimization to three distinct ablation scenarios. They used convergence charts and accuracy to compare the three scenarios. They noticed that the Bayesian search-derived optimal architecture achieved 96% accuracy. To assist qualitative researchers, address their research questions in a methodologically sound manner, a comparison of research method and theme analysis methods was provided. The suggested model is shown to be more trustworthy and accurate in real world.

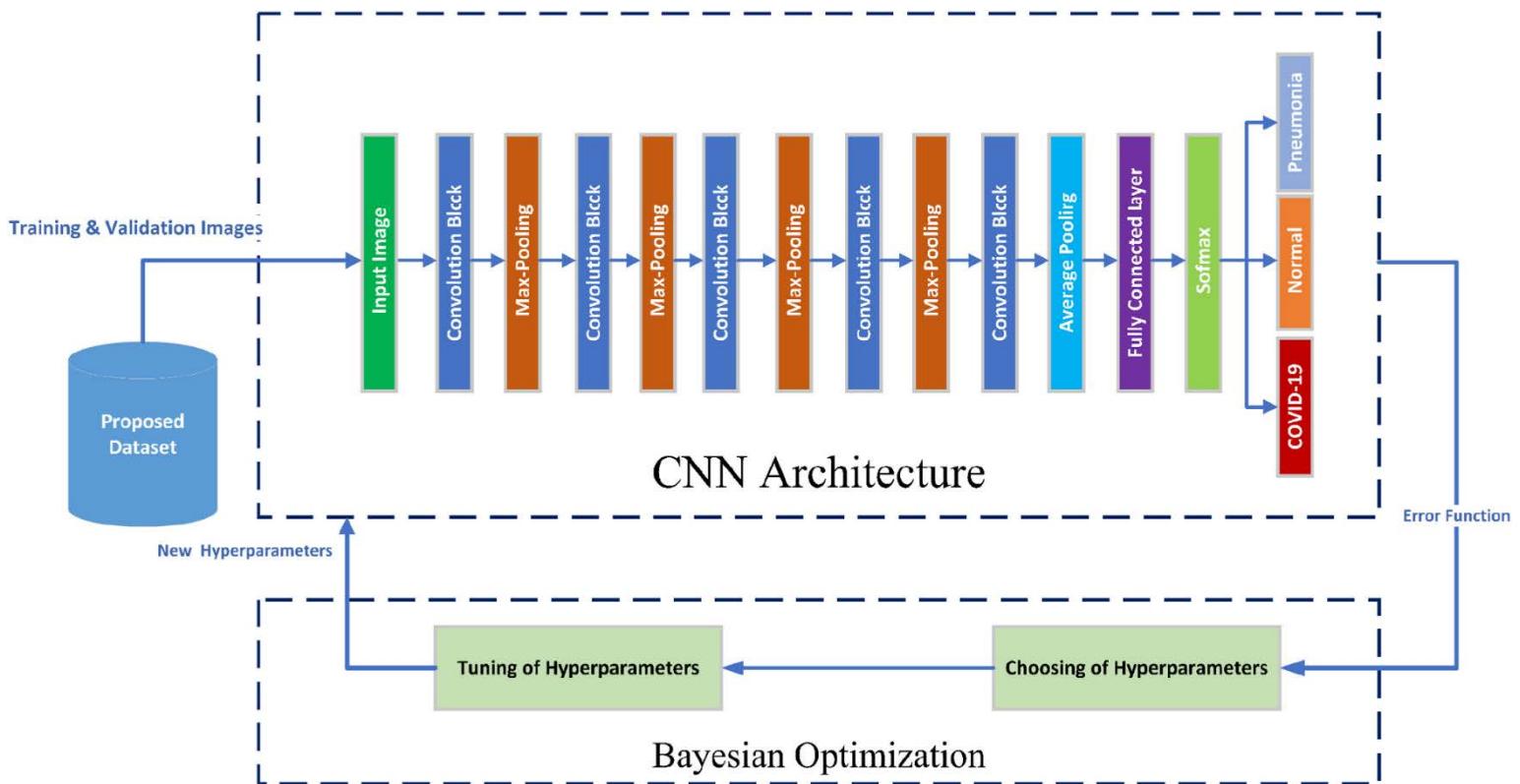


Figure (2-9) The proposed COVID-19 X-ray classification model for training.

Multi-label classification: the multi-label classification is a common issue we face in the diagnosis of lung diseases in which each training example is associated with possibly more than one label. Solving such an issue will play an important role in medical imaging. That's why a variety of deep learning-based approaches have been proposed in the literature. J. Rubin et al.[13] introduced 'DualNet' architecture which is a collection of deep convolutional neural networks trained on the largest released dataset of chest x-ray images -- the MIMIC-CXR dataset. that emulates routine clinical practice by simultaneously processing both frontal and lateral Chest-X ray (CXR) images of MIMIC-CXR dataset images obtained from a radiological exam. Their DualNet architecture shows improved performance in recognizing findings in CXR images when compared to applying separate baseline frontal and lateral classifiers. They achieved an average AUC of 72.1%.

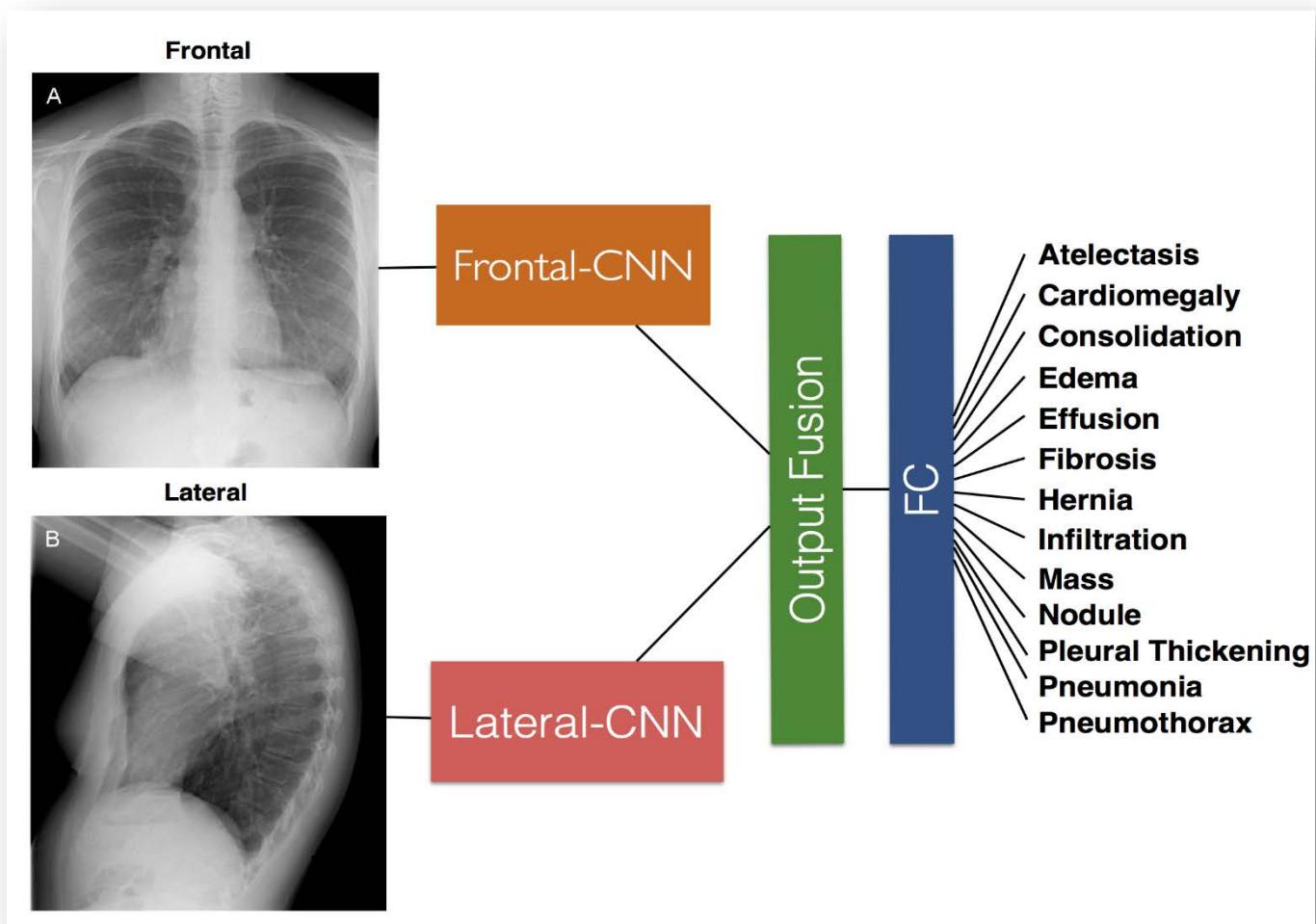


Figure (2-10) The DualNet architecture accepts a pair of frontals (PA or AP) and lateral input images. Two convolutional neural networks are trained in parallel for each input and their outputs are concatenated together before a final fully connected layer makes a multi-label prediction.

P. Kumar et al. [14] Proposed a set of deep learning models and present a cascaded deep neural network that can diagnose all 14 pathologies. to improve the performance of the multi-label classification task. They managed to score an average AUC of 90.55% on Chest X-ray 14 dataset.

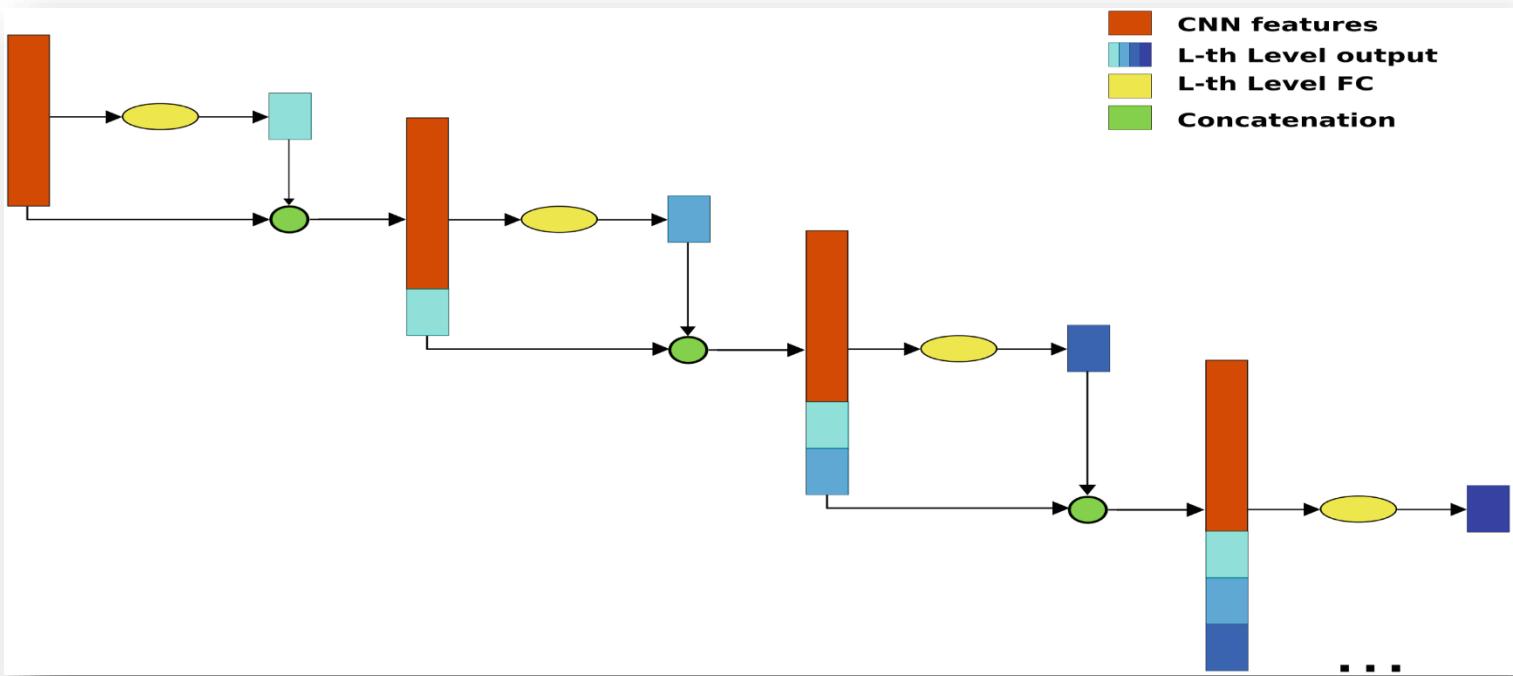


Figure (2-11) The proposed COVID-19 X-ray classification model for training.

S. Bharati et al. [15] Proposed a new hybrid deep learning framework by combining VGG, data augmentation and spatial transformer network (STN) with CNN. This new hybrid method is termed here as VGG Data STN with CNN (VDSNet). The new model is applied to NIH chest X-ray image dataset collected from Kaggle repository. Full and sample versions of the dataset are considered. For both full and sample datasets, VDSNet outperforms existing methods in terms of a number of metrics. For the case of full dataset, VDSNet exhibits a validation accuracy of 73%, while vanilla gray, vanilla RGB, hybrid CNN and VGG, and modified capsule network have accuracy values of 67.8%, 69%, 69.5% and 63.8%, respectively. When sample dataset rather than full dataset is used, VDSNet requires much lower training time at the expense of a slightly lower validation accuracy. Hence, the proposed VDSNet framework will simplify the detection of lung disease for experts as well as for doctors.

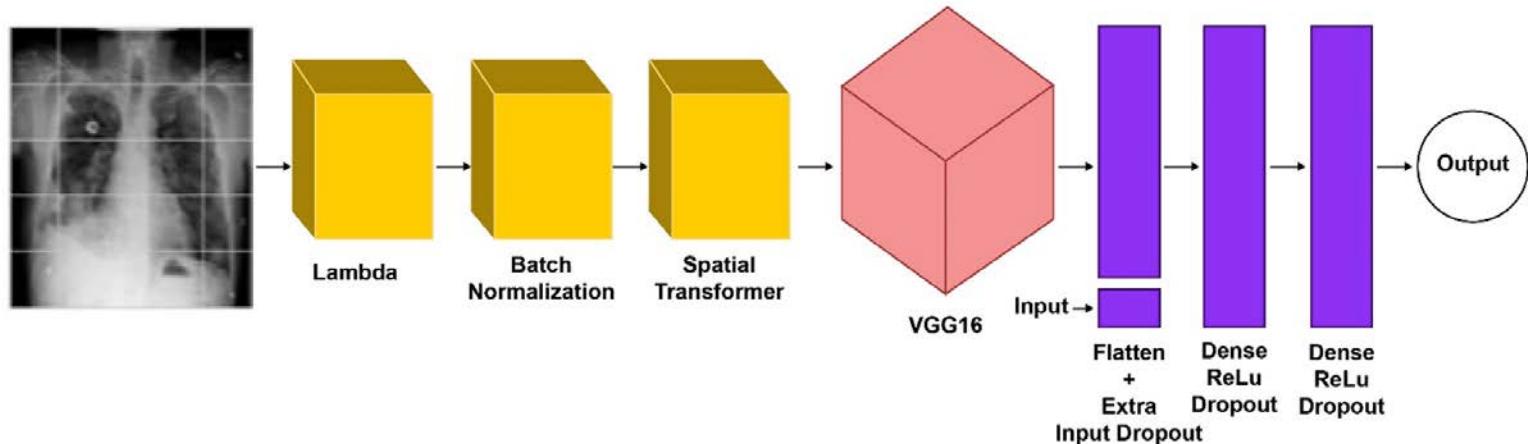


Figure (2-12) Full architecture of VDSNet.

Here also, the authors found that improving architecture is not all we need. Z. Ge [16] In this work they have proposed a novel error function based on SoftMax concept (Multi-label SoftMax Loss, MSML) to address the problems of multiple labels and imbalance data. They have designed convolutional deep network based on fine-grained classification methodology that incorporates MSML. They have evaluated their proposed method on various network backbones and showed performance improvements of AUC-ROC scores on the ChestX-ray14 dataset. The proposed error function provides a new direction to attain improved performance for wider medical data, they managed to get an AUC of 85.37% on 14 abnormalities from ChestX-ray14 dataset.

Hieu H. Pham et al.[17] This paper presents a supervised multi-label classification framework based on deep convolutional neural networks (CNNs) for predicting the presence of 14 common thoracic diseases and observations. They tackle this problem by training state-of-the-art CNNs that exploit hierarchical dependencies among abnormality labels. They also propose to use the label smoothing technique for a better handling of uncertain samples, which occupy a significant portion of many CXR datasets. their model is trained on over 200,000 CXRs of the recently released CheXpert dataset and achieves a mean area under the curve (AUC) of 0.940 in predicting 5 selected pathologies from the validation set. The proposed method is also evaluated on the independent test set of the CheXpert competition, which is composed of 500 CXR studies annotated by a panel of 5 experienced radiologists. The performance is on average better than 2.6 out of 3 other individual radiologists with a mean AUC of 0.930, which ranks first on the CheXpert leaderboard at the time of writing this paper.

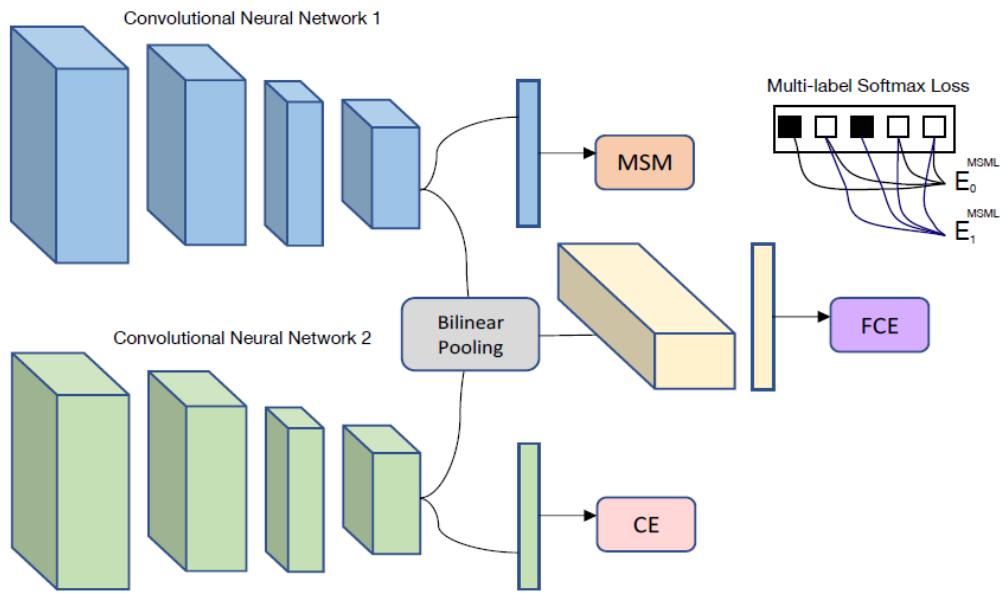


Figure (2-13) The architecture of the multi-label ne-grained network for chest x-ray disease detection. MSML loss operates on each positive classes from a sample and encourages them to consider independences between label being present and absent while minimizing the Oss across classes.

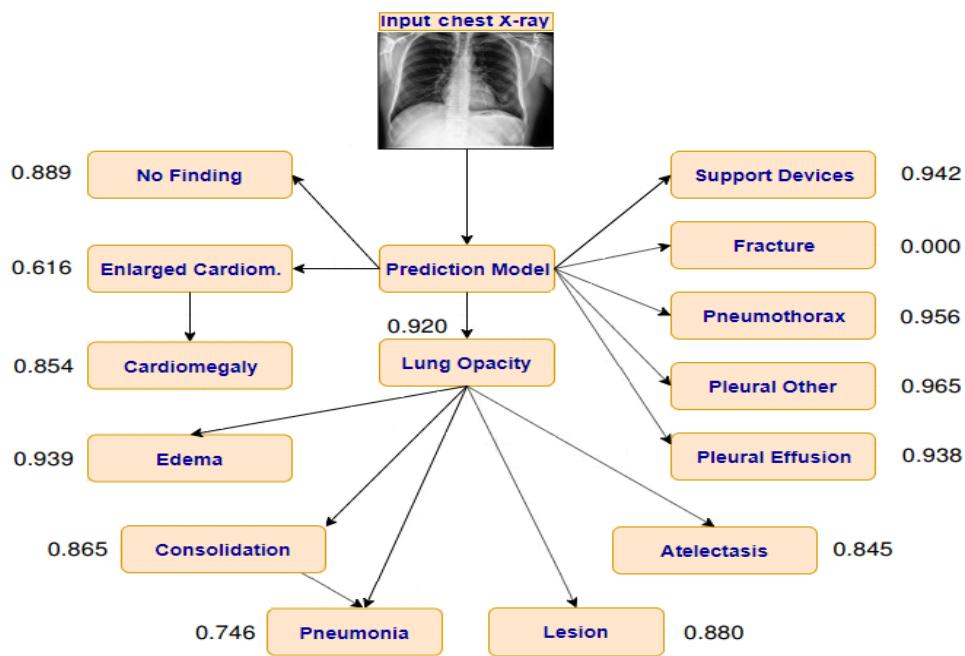


Figure (2-14) Illustration of their classification task, which aims to build a deep learning system for predicting probability of presence of 14 different pathologies or observations from the CXRs.

Z. Yuan et al [18] Deep AUC Maximization (DAM) is a new paradigm for learning a deep neural network by maximizing the AUC score of the model on a dataset. Most previous works of AUC maximization focus on the perspective of optimization by designing efficient stochastic algorithms, and studies on generalization performance of large-scale DAM on difficult tasks are missing. In this work, they aim to make DAM more practical for interesting real-world applications (e.g., medical image classification). First, they propose a new margin-based min-max surrogate loss function for the AUC score (named as the AUC min-max margin loss or simply AUC margin loss for short). It is more robust than the commonly used AUC square loss, while enjoying the same advantage in terms of large-scale stochastic optimization. Second, they conduct extensive empirical studies of their DAM method on four difficult medical image classification tasks, namely (I) classification of chest x-ray images for identifying many threatening diseases, (ii) classification of images of skin lesions for identifying melanoma, (iii) classification of mammogram for breast cancer screening, and (iv) classification of microscopic images for identifying tumor tissue. Their studies demonstrate that the proposed DAM method improves the performance of optimizing cross entropy loss by a large margin, and also achieves better performance than optimizing the existing AUC square loss on these medical image classification tasks. Specifically, their DAM method has achieved the 1st place on Stanford CheXpert competition on Aug. 31, 2020.

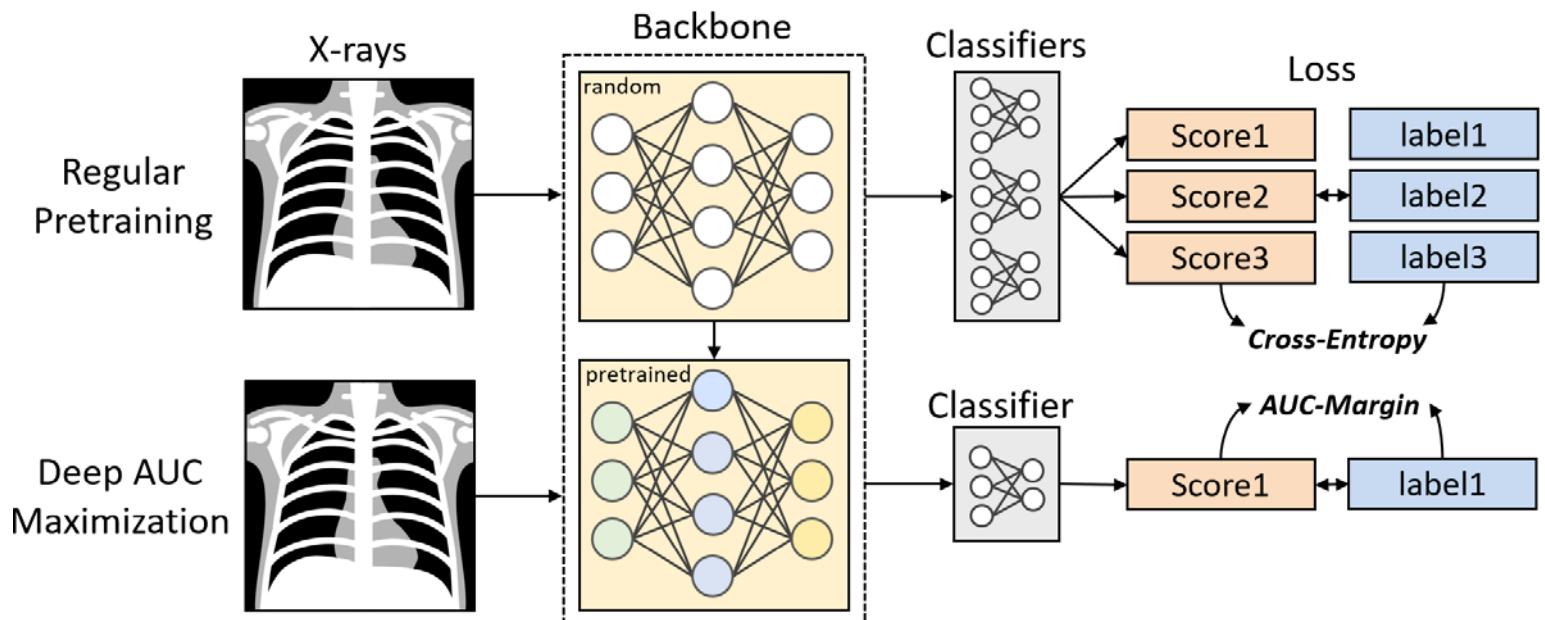


Figure (2-15) A Two-stage Deep AUC Maximization Framework. For the pretraining stage, they focus on learning representation by optimizing a standard Cross Entropy loss. For the AUC maximization stage, they focus on finetuning the decision boundary of classifier by optimizing AUC margin loss.

As we have seen, all the literature focuses on how to get a better “architecture” or how to improve the way an “architecture” learns. In this study, our solution is a more data-centric one where we care also about the “quality” of the data itself not only the architecture. We have done so by conducting a fully comprehensive analysis on CheXpert dataset. With the help of this analysis, we gained some intuition for what architecture and filter to use for each disease and consequently, proposed an ensemble model which surpassed the performance of the baseline provided by Stanford ML group.

CHAPTER 3

PRE-PROCESSING

3 Chapter (3) Preprocessing

All the basic knowledge needed to understand the science behind our project is covered In this chapter.

Machine's Perception of images is very different from that for humans i.e., Human vision is barely a beam of light hits the retina (a light-sensitive layer of tissue at the back of the eye), then some special cells in the eye called photoreceptors turn the light into electrical signals. These electrical signals travel from the retina through the optic nerve to the brain which, in turn, turns the signals into images. Because of this journey, humans' perception of an image is very high level. On the other hand, Machine can "see" an image after some sampling and digitization to convert it to a matrix of some elements called picture elements or "pixels" in short. Consequently, Machines can see, through these pixels, more details of an image.

This "detailed" view of an image by the machine can be exploited by an appropriate processing to extract only the features we are interested in. These features can be given at the end to a machine learning pipeline uses classical machine learning or deep leaning algorithms for a specific computer vision task.

Image processing is divided into three kinds of operations:

Point processing:

This kind of processing is concerned with the "value" of a pixel itself without paying any attention to the relationship between this pixel's value and its surroundings. So, by such a processing we are able to change the value of a pixel without any knowledge of the surroundings.

Neighbourhood processing:

This kind of processing is concerned with the relationship between a pixel and its surrounding. So, a pixel value can be changed only if the knowledge of the surroundings is provided.

Transformations:

This kind of processing treat the entire image matrix as a single large block. In other words, we are converting the image to a whole other domain so as to be able to look at it from a different angle may be this new domain highlights some important features which are suppressed in the old one.

We can think of an image as a function mapping coordinates in the image to pixel values. With this in mind:

From “Neighbourhood processing” perspective, we can apply a kernel to this image using convolution [*3*]. This will allow us to detect high and low frequencies in the image i.e., rapid or slow changes in a pixel’s values can be detected easily.

From “Transformation” perspective, we treat this image as an input to a system which, in turn, will output the same image in a whole other domain to highlight a specific feature of the image like, for an example, taking the Fourier transform to transform the image into the frequency domain to highlight the changes in the frequency over the whole image.

X-ray images is like any other images, it can be represented as a matrix of pixels but what distinguishes it from the other is that it has only one channel in a grayscale mode. Thus, we have chosen three image processing techniques to deal with such images trying to highlight the important features in the image before passing it to our deep learning-based pipeline.

3.2 Top-hat and Bottom-hat Transform (TB)

Before illustrating the idea behind top-hat and bottom-hat transform, let's mention first the two basic operations used in mathematical morphology:

(We will illustrate how use them on a binary image for simplicity)

The term morphology originally comes from the study of forms of plants and animals. In our context we mean the study of topology or structure of objects with respect to their images. So, Morphological processing refers to certain operations where an object i.e., an

image (I) is hit with a set of Euclidean-coordinate points known as structuring element (S).

Let S_o denote the translation of S so that its origin is located at O as shown in Fig. (3-1).

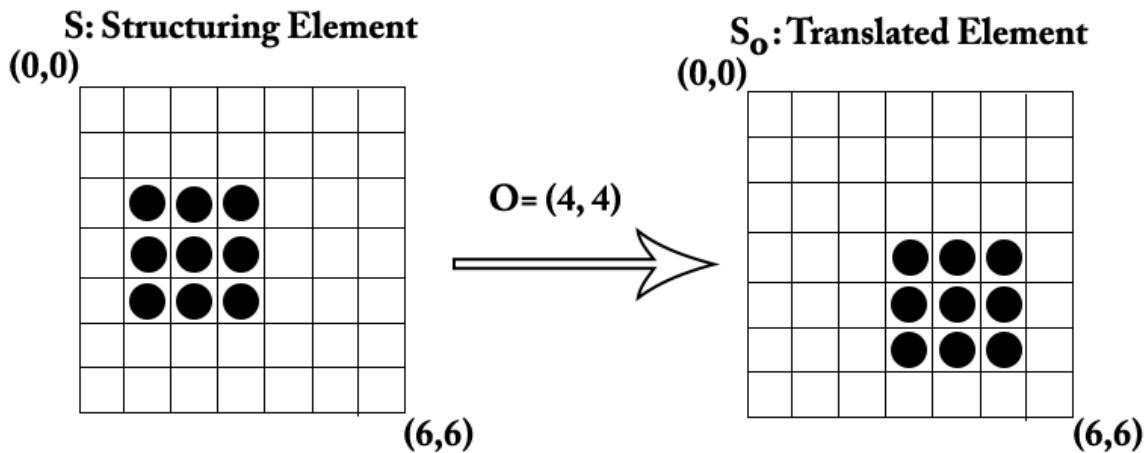


Figure (3-1) Tanslation Operation

The two basic morphological operations are:

1-Dilation

Dilation can be defined mathematically as follows:

$$I \oplus S \triangleq \{O : S_o \cap I \neq \emptyset\}$$

Simply put, the dilation of an image I by a structuring element S is the set of all points O such that S_o has a non-empty intersection with I .

2-Erosion

Erosion can be defined mathematically as follows:

$$I \ominus S \triangleq \{O : S_o \subset I\}$$

Similarly, the erosion of an image I by a structuring element S is the set of all points O such that S_o is included in I .

Fig. (3-2.1) and Fig. (3-2.2) illustrate both with an example to comprehend the intuition behind them.

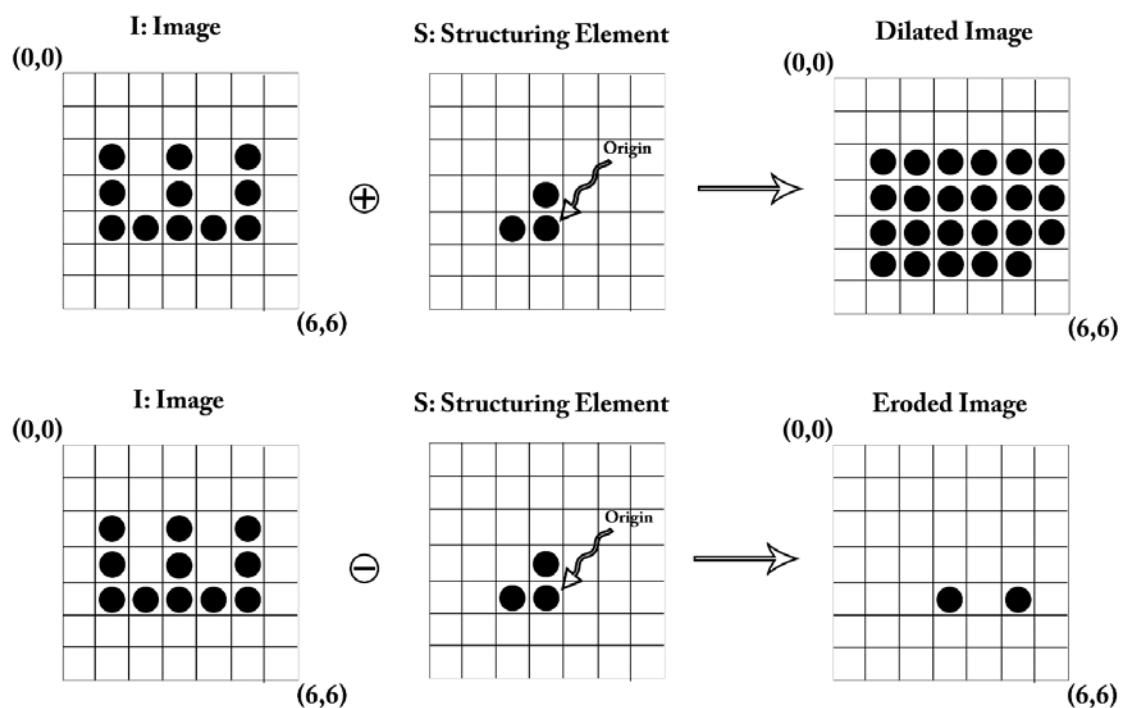


Figure (3-2.1) illustration of Dilation and Erosion

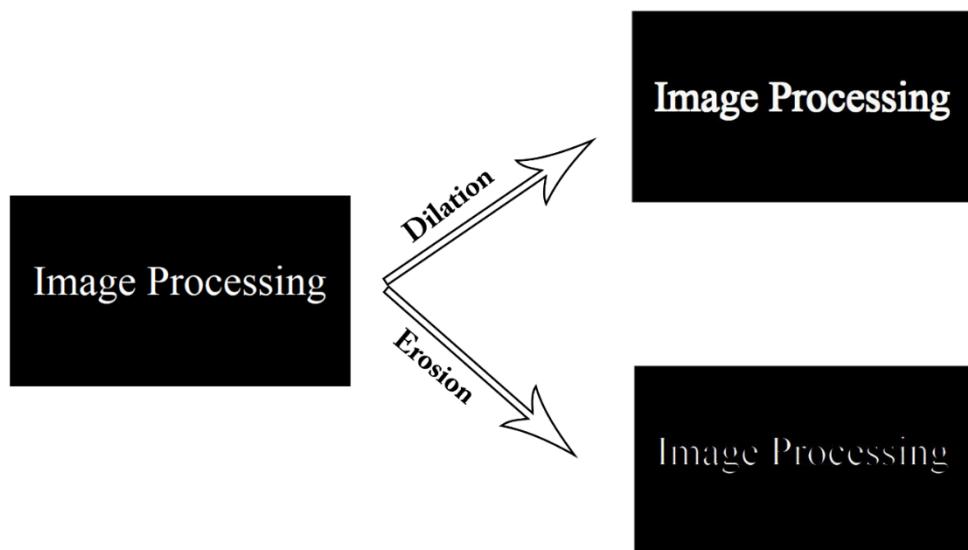


Figure (3-2.2) illustration of Dilation and Erosion

Based on these two, there are two basic “second-level” morphological operations:

1-Opening:

Dilation can be defined mathematically as follows:

$$I \circ S = (I \ominus S) \oplus S$$

So, Morphological opening is basically an erosion followed by a dilation.

2-Closing:

Erosion can be defined mathematically as follows:

$$I \bullet S = (I \oplus S) \ominus S$$

Similarly, Morphological closing is a dilation followed by an erosion i.e., the reverse of the operations for an opening.

Using a 11x11 square as our structuring element, we can see in Fig. (3-3) That The net effect of a morphological opening is to keep only the pixels of the input image which fits the structuring element. In other words, opening tends to remove pixels or “open gaps” in the image. On the other hand, the net effect of the closing operation is to remove background pixels that fit the structuring element. In other words, closing tends to “close gaps” in the image.

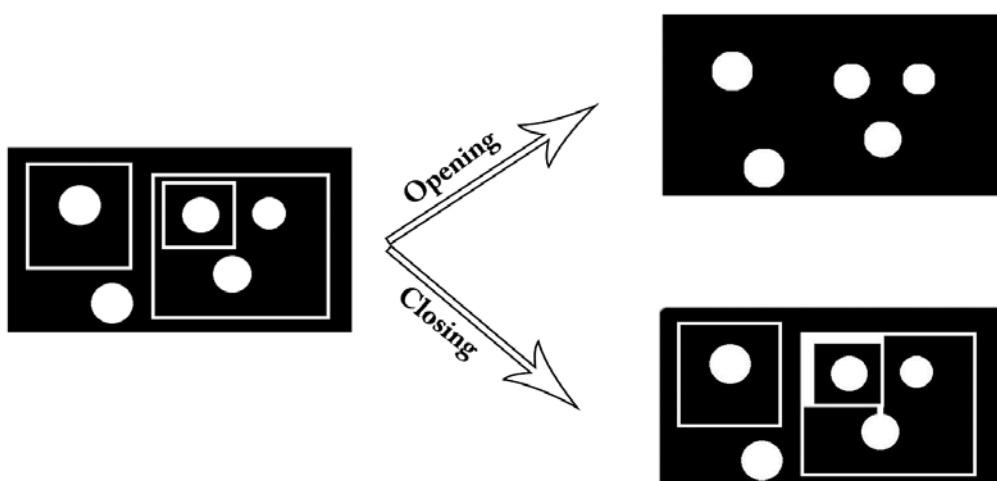


Figure (3-3) illustration of Opening and Closing

- **Top hat filtering:**

The top-hat filtering, also called top-hat transformation operator or white-top-hat filtering, in the mathematical morphology estimates the trend via morphological opening and then removes this trend from an image. The morphological opening, and the thus top-hat filter, works in the following steps:

- i) Replaces each pixel by the minimum value of the pixels in a square centered around it.
- ii) Replaces each pixel with the maximum value in the window of the created image in step(i)
- iii) Subtracts the morphological opening from the original image

Top hat equation:

$$I_t = I - (I \circ S)$$

The top-hat filter technique has the following properties:

- I. Strong background suppression effect.
- II. Smoothing effect.

- **Bottom hat filtering:**

The bottom-hat filtering, or called black-top-hat filtering, is given by:

$$I_b = (I \bullet S) - I$$

The bottom-hat filter has the property of enhancing "valleys" by applying the closing operator and is used in Morphological sharpening.

- **Top-hat and Bottom-hat combining:**

In the case of the top-hat transform, light objects are obtained, which are shorter than structuring element. On the other hand, the dark elements recovered with a bottom transform are smaller than structuring element. Thus, by combining the addition of the

top-hat output and the subtraction of the bottom-hat output with its original image, we were able to create an improved image where key features will be clearer.

Here is the mathematical formulation for the previous method:

$$I_t = I - (I \circ S)$$

$$I_b = (I \bullet S) - I$$

$$I_{out} = I + I_t - I_b$$

As shown in Fig. (3-4), this transformation highlighted the radiopaque regions in the chest x-ray image which will help our deep learning-based model.

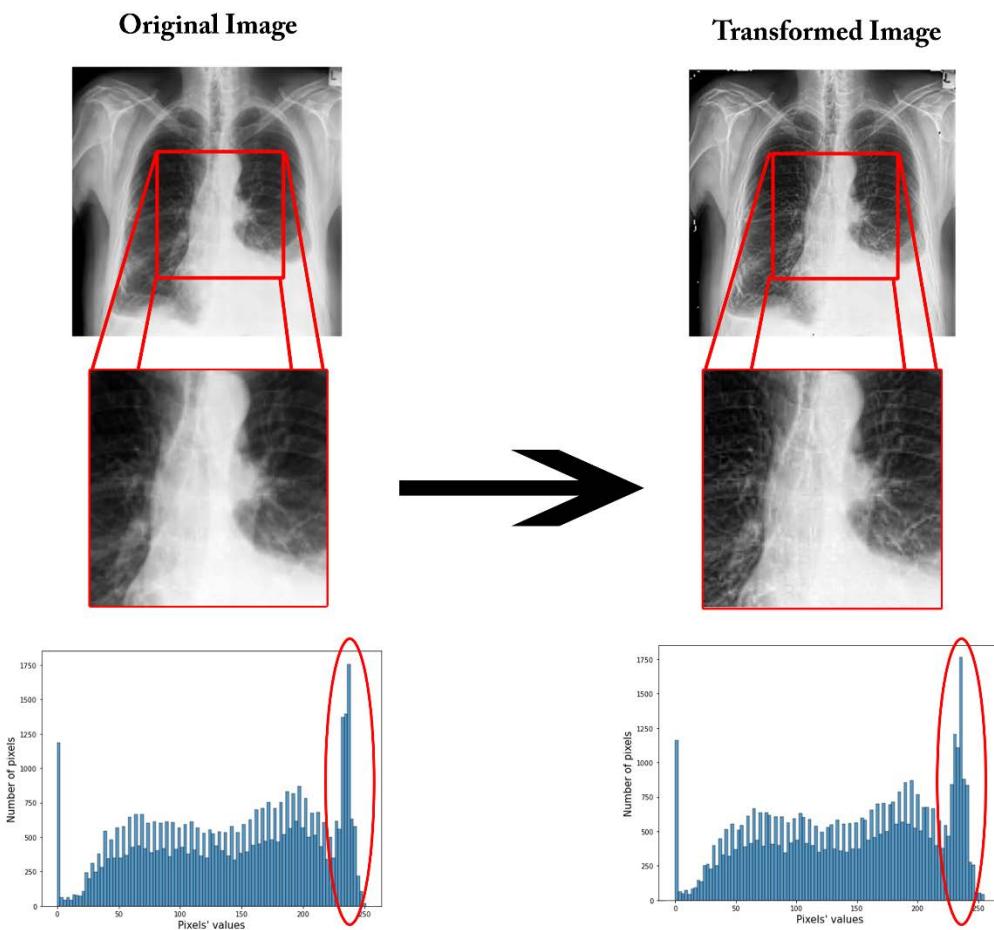


Figure (3-4) Effect of Top-hat and Bottom-hat Transformation

3.3 GoDec-based Radiographs sparsification.

Recent research about exploring low-rank and sparse structures (Zhou et al., 2011) concentrates on developing fast approximations and meaningful decompositions. Two appearing in Proceedings of the 28 th International Conference on Machine Learning, Bellevue, WA, USA, 2011. Copyright 2011 by the author(s)/owner(s). Notable representatives are the randomized approximate matrix decomposition (Halko et al., 2009) and the robust principal component analysis (RPCA) (Candes et al., 2009). The former proves that a matrix can be well approximated by its projection onto the column space of its random projections. This rank-revealing method provides a fast approximation of SVD/PCA. The latter proves that the low-rank and the sparse components of a matrix can be exactly recovered if it has a unique and precise “lowrank+sparse” decomposition. RPCA offers a blind separation of low-rank data and sparse noises.

In[19], the authors proposed a sparse-low-rank decomposition mechanism especially in noisy cases, denoted as “Go-Decomposition” (GoDec). GoDec efficiently and robustly estimate the low-rank part X_r , the sparse part X_s , and the noise part X_N of an inserted image as formulated in equation (1):

$$X = X_r + X_s + X_N \quad (1)$$

$$\text{rank}(X_r) \leq l, \quad \text{cardinality}(X_s) \leq k$$

By minimizing the error function stated in equation (2), the approximation function discussed in equation (1) can be solved.

$$\min_{X_r, X_s} \|X - X_r - X_s\|_F^2 \quad (2)$$

$$\text{rank}(X_r) \leq l, \quad \text{cardinality}(X_s) \leq k$$

The previous optimization problem (2) has an existing solution if it's divided into two subproblems (3-4). Alternatively, solving these two subproblems until convergence will lead to the desired solution:

$$\begin{cases} X_{r_t} = \underset{\text{rank}(X_r) \leq l}{\text{argmin}} \|X - X_r - X_{s_{t-1}}\|_F^2 \end{cases} \quad (3)$$

$$\begin{cases} X_{s_t} = \underset{\text{card}(X_s) \leq k}{\text{argmin}} \|X - X_{r_t} - X_s\|_F^2 \end{cases} \quad (4)$$

Although convexity is not satisfied here, there exists a global solution. particularly, the two subproblems in (3-4) can be solved by updating X_{r_i} via singular value hard thresholding of $X - X_{s_{i-1}}$ and updating X_{s_i} via entry-wise hard thresholding of $X - X_{r_i}$, respectively as illustrated in equations (5-6) where svd is the singular value decomposition operation, v_i is the eigen vector of $(X - X_{s_t})^T \cdot (X - X_{s_t})$ with a singular value σ_i , and u_i which is the eigenvector of $(X - X_{s_t})$. For the second subproblem, Ω is an index set, P_Ω is a linear operator that extracts the entries in Ω and fills the entries not in Ω with zeros.

$$\left\{ \text{svd}(X - X_{s_{t-1}}) = U\Sigma V^T \rightarrow X - X_{s_{t-1}} = X_{r_t} = \sum_{i=1}^l \sigma_i u_i v_i^T \right. \quad (5)$$

$$\left. X_{s_t} = P_\Omega(X - X_{r_t}), \Omega: |(X - X_{r_t})_{i,j \in \Omega}| \neq 0 \text{ and } \geq |(X - X_{r_t})_{i,j \in \bar{\Omega}}|, |\Omega| \leq k. \right. \quad (6)$$

Noticeably, the dominant computation in the previous two subproblems is the singular value decomposition (SVD) of $X - X_{s_{t-1}}$ in the updating X_{r_t} sequence. SVD needs $\min(mn^2, m^2n)$ flops, making it impractical when X is too large. The authors[19] proposed a low-rank approximation based on bilateral random projections (BRP)[20] in place of SVD in the sake of significantly reducing the time taken. Using BRP and its power scheme modification, it takes the algorithm only $l^2(2n + l) + 4mnl$ or $l^2(m + 3n + 4l) + (4q + 4)mnl$ flops for $q = 0$ or $q > 0$ respectively. the decomposition of a chest x-ray image is given in Fig. (3-5). In this study, we are more interested in the Sparse Matrix X_s only.

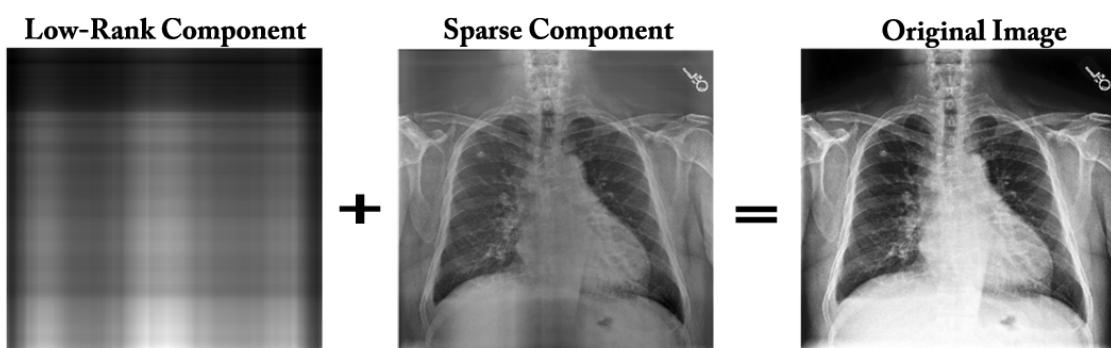


Figure (3-5) Image Sparsification by GoDec Algorithm

The algorithm summary is provided as a pseudo code for better understanding. In the sake of improving the approximation quality of X_r , a right random project Y_1 is used to build an improved projection matrix A_2 which, in turn, is used to generate Y_2 that, iteratively, is used to update and produce a better A_1 . Particularly, the first step is generating $Y_1 = XA_1$, then A_2 is updated $A_2 = Y_1$, then $Y_2 = X^T A_2$ is calculated, the last step of this iteration is updating $A_1 = Y_2$. After some number of iterations, a high-precision low rank approximation X_r can be obtained by applying the final Y_1 and Y_2 to the following formula:

$$X_r = Y_1 \left(A_2^T Y_1 \right)^{-1} Y_2^T \quad (7)$$

The previous formula (7) may not be enough especially when singular values of X decay slowly. Using power scheme[21], a more optimum modification is proposed. Instead of using just the pure image X , a modified version of it \tilde{X} is calculated by equation (8) whose singular values decay faster.

$$\tilde{X} = (XX^T)^q X : \begin{cases} q = 0 & \text{for dense } X \\ q > 0 & \text{otherwise} \end{cases} \quad (8)$$

The low-rank approximation of X is then given by:

$$L = Q_1 [R_1 (A_2^T Y_1)^{-1} R_2^T]^{\frac{1}{2q+1}} Q_2^T \quad (9)$$

Where QR decomposition of Y_1 and Y_2 is used to obtain the approximation of X with rank l :

$$Y_1 = Q_1 R_1, \quad Y_2 = Q_2 R_2 \quad (10)$$

GoDec Algorithm

Input: X, l, k, ε, q

Output: X_r, X_s

Initialize: $X_{r_0} := X, X_{s_0} := 0, A_1 = [\text{random matrix}]_{m \times n}, t := 0$

While $\frac{\|X - X_{r_t} - X_{s_t}\|_F^2}{\|X\|_F^2} > \varepsilon$ **do**

$t := t + 1;$

$$\widetilde{X_r} = [(X - X_{s_{t-1}})(X - X_{s_{t-1}})^T]^q (X - X_{s_{t-1}});$$

$$Y_1 = \widetilde{X_r} A_1, A_2 = Y_1;$$

$$Y_2 = \widetilde{A_1} = \widetilde{X_r} Y_1 = Q_2 R_2, Y_1 = \widetilde{X_r} Y_2 = Q_1 R_1;$$

If $\text{rank}(A_2^T Y_1) < l$

then $r := \text{rank}(A_2^T Y_1)$, go to the first step;

end if;

$$X_{r_t} = Q_1 [R_1 (A_2^T Y_1)^{-1} R_2^T]^{\frac{1}{(2q+1)}} Q_2^T;$$

$X_{s_t} = P_\Omega(X - X_{r_t}), \Omega$ is the nonzero subset of the first k largest entries of $|X - X_{r_t}|$;

end while

3.4 Fourier Transform (FT)

Converting the image into the frequency domain Using FT is a popular way in image processing due to its efficiency and speed. Investigating the frequency domain will allow us to apply low-pass, high-pass, or any other filter more accurately. Apparently, we are dealing with digital images, so we are concerned only with the Discrete Fourier Transform (DFT) in our discussion.

Let's begin our discussion with a quick reminder of the one-dimensional DFT, then we will expand our knowledge to the two-dimensional one and The Fast Fourier Transform (FFT).

The one-dimensional DFT:

For a length-N sequence $x[n]$, defined for $0 \leq n \leq N - 1$ only, N samples of its Discrete Time Fourier Transform (DTFT) [*8*] are required, which are obtained by uniformly sampling $X(e^{j\omega})$ on the $\omega - axis$ between $0 \leq \omega \leq 2\pi$ at $\omega_k = \frac{2\pi k}{N}$, $0 \leq k \leq N - 1$. We can formulate this definition as follows:

$$X[k] = X(e^{j\omega})|_{\omega=\frac{2\pi k}{N}} = \sum_{n=0}^{N-1} x[n] e^{-\frac{j2\pi k}{N}}, \quad 0 \leq k \leq N - 1$$

The Inverse Discrete Fourier Transform (IDFT) is given by:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{\frac{j2\pi k}{N}}$$

Note: One of the many aspects which make the DFT so attractive for image processing is the existence of very fast algorithms to compute it. There are several extremely fast and efficient algorithms for computing a DFT; one of these algorithms is called a fast Fourier transform, or FFT in short. The use of FFT vastly reduces the time needed to compute the DFT.

Taking this definition and expanding it to the two-dimensional space, we will be able to formulate the definition of the two-dimensional DFT.

The two-dimensional DFT:

Here in the 2-D space, the DFT takes an image (matrix) as an input and returns another matrix, of the same size, as an output. The formula can be simply put as follows:

$$F(l, k) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-2\pi i \left(\frac{xl}{M} + \frac{yk}{N}\right)}$$

The Inverse formula:

$$f(x, y) = \sum_{l=0}^{M-1} \sum_{k=0}^{N-1} F(l, k) e^{2\pi i \left(\frac{xl}{M} + \frac{yk}{N}\right)}$$

The DFT produces a complex-number valued output image which can be displayed with two images, either with the real and imaginary part or with magnitude and phase.

In image processing, we are more interested in the magnitude of the transformed image, as it contains most of the information we need.

Note: if we put $l = k = 0$

$$F(0, 0) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^0 = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$$

As we can see this term is calculated by summing all the values of the original matrix. This value is called the “DC Coefficient”. Consequently, the DC coefficient is much larger than other values. So, for the sake of a good display, the matrix is shifted so that the DC

coefficient will be in the center of the matrix. Thus a transformed image will likely be a single white dot surrounded by black background.

We can take advantage of this transformed image, which represents the change in frequency over the whole image and apply any filter to extract more features from the image.

In our project, we have used the high-pass filter (HPF) after transforming the input image into the frequency domain.

As shown in Fig. (3-6), after applying the HPF to the transformed image the high frequency components have multiplied by a certain gain bigger than one which makes radiopaque regions stand out more in the image.

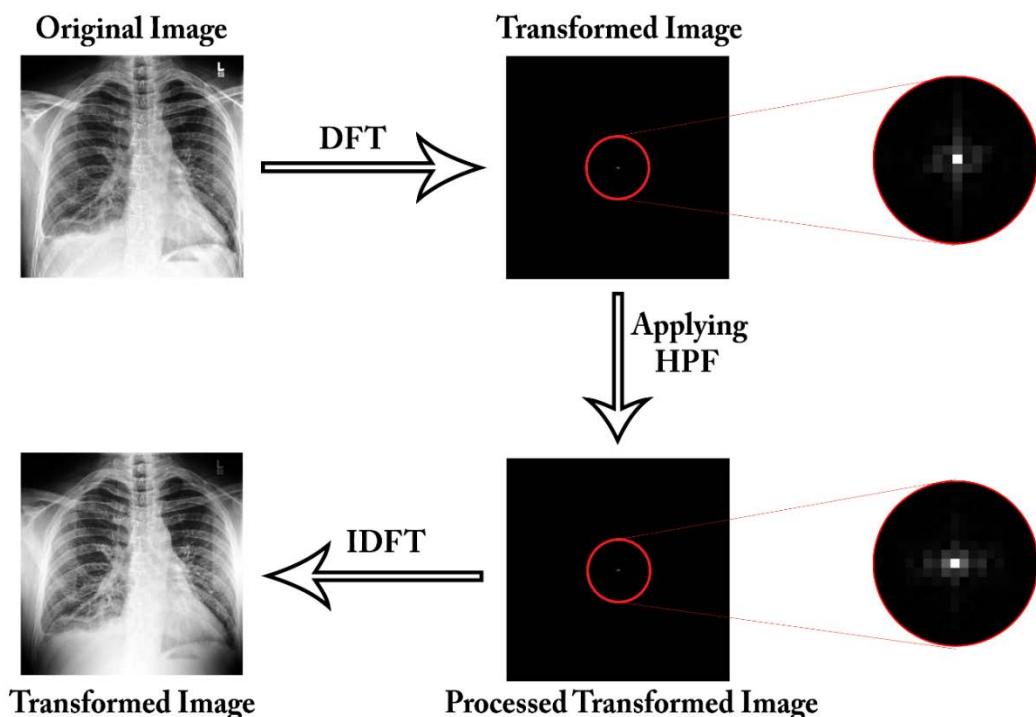


Figure (3-6) Effect of Applying HPF On the Fourier-transformed image

CHAPTER 4

TRANSFER LEARNING

AND

SOTA MODELS

4 Chapter (4) Transfer learning

4.1 Transfer Learning:

The reuse of a pre-trained model on a new problem is known as transfer learning in machine learning. A machine uses the knowledge learned from a prior assignment to increase prediction about a new task in transfer learning. You could, for example, use the information gained during training to distinguish beverages when training a classifier to predict whether an image contains cuisine.

The knowledge of an already trained machine learning model is transferred to a different but closely linked problem throughout transfer learning. For example, if you trained a simple classifier to predict whether an image contains a backpack, you could use the model's training knowledge to identify other objects such as sunglasses.

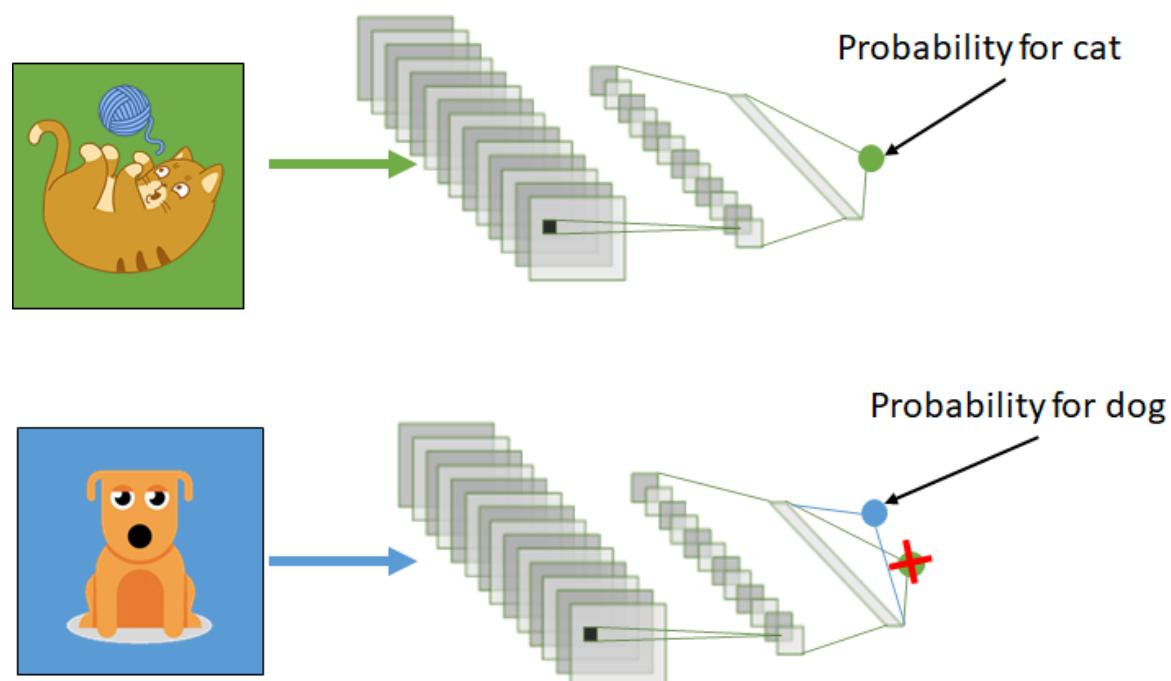


Figure (4-1) Illustration of Transfer learning concept

With transfer learning, we basically try to use what we've learned in one task to better understand the concepts in another. weights are being automatically being shifted to a network performing "task A" from a network that performed new "task B."

Because of the massive amount of CPU power required, transfer learning is typically applied in computer vision and natural language processing tasks like sentiment analysis.

How transfer learning works:

In computer vision, neural networks typically aim to detect edges in the first layer, forms in the middle layer, and task-specific features in the latter layers. The early and central layers are employed in transfer learning, and the latter layers are only retrained. It makes use of the labelled data from the task it was trained on.

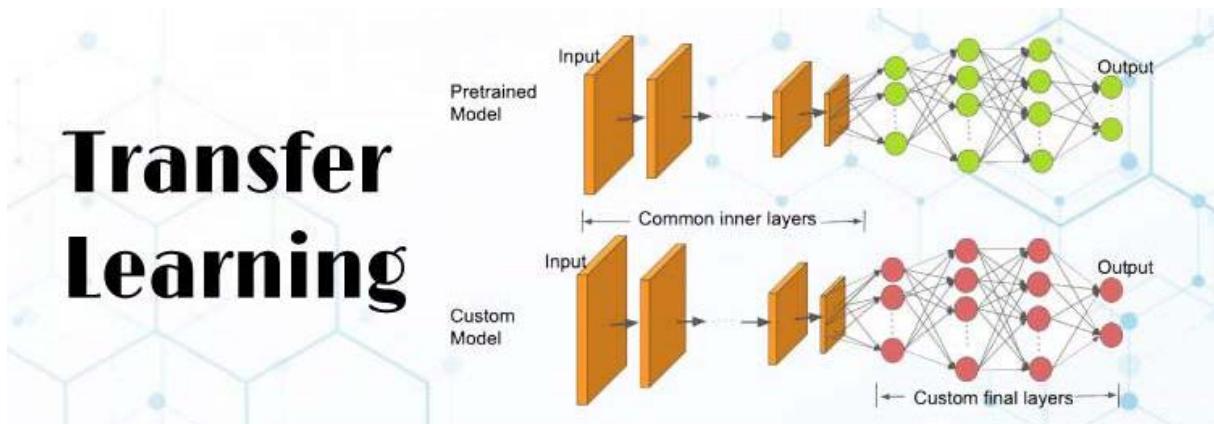


Figure (4-2) How Transfer learning works

Let's return to the example of a model that has been intended to identify a backpack in an image and will now be used to detect sunglasses. Because the model has trained to recognise objects in the earlier levels, we will simply retrain the subsequent layers to understand what distinguishes sunglasses from other objects.

Why to use transfer learning:

Transfer learning offers a number of advantages, the most important of which are reduced training time, improved neural network performance (in most circumstances), and the absence of a large amount of data.

To train a neural model from scratch, a lot of data is typically needed, but access to that data isn't always possible – this is when transfer learning comes in handy.

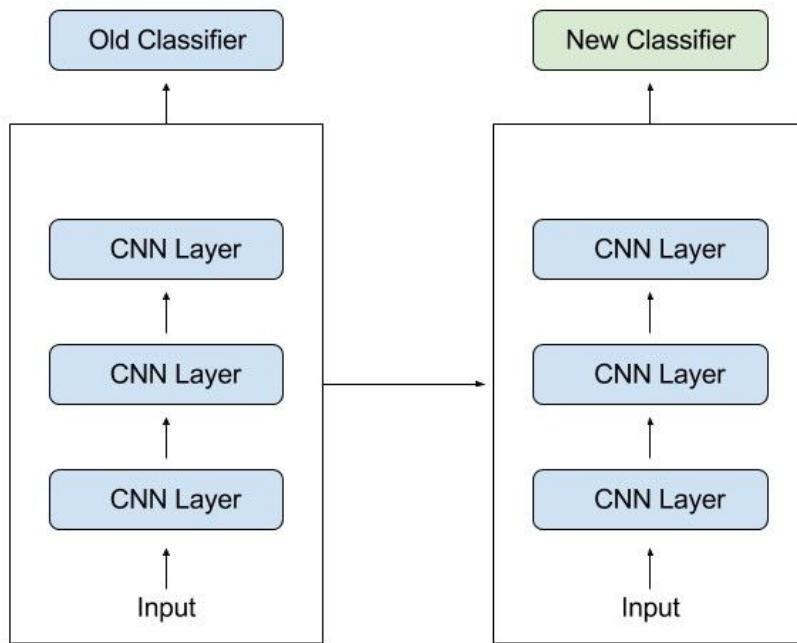


Figure (4-3) Pretraining models and Importance of Transfer learning

Because the model has already been pre-trained, a good machine learning model can be generated with fairly little training data using transfer learning. This is especially useful in natural language processing, where huge labelled datasets require a lot of expert knowledge. Additionally, training time is decreased because building a deep neural network from the start of a complex task can take days or even weeks.

When to Use Transfer Learning

When we don't have enough annotated data to train our model with. When there is a pre-trained model that has been trained on similar data and tasks. If you used TensorFlow to train the original model, you might simply restore it and retrain some layers for your job. Transfer learning, on the other hand, only works if the features learnt in the first task are general, meaning they can be applied to another activity. Furthermore, the model's input must be the same size as it was when it was first trained.

If you don't have it, add a step to resize your input to the required size.

1. TRAINING A MODEL TO REUSE IT

Consider the situation in which you wish to tackle Task A but lack the necessary data to train a deep neural network. Finding a related task B with a lot of data is one method to get around this.

Utilize the deep neural network to train on task B and then use the model to solve task A. The problem you're seeking to solve will decide whether you need to employ the entire model or just a few layers.

If the input in both jobs is the same, you might reapply the model and make predictions for your new input. Changing and retraining distinct task-specific layers and the output layer, on the other hand, is an approach to investigate.

2. USING A PRE-TRAINED MODEL

The second option is to employ a model that has already been trained. There are a number of these models out there, so do some research beforehand. The number of layers to reuse and retrain is determined by the task.

Keras consists of nine pre-trained models used in transfer learning, prediction, fine-tuning. These models, as well as some quick lessons on how to utilise them, may be found here. Many research institutions also make trained models accessible.

The most popular application of this form of transfer learning is deep learning.

3. EXTRACTION OF FEATURES

Another option is to utilise deep learning to identify the optimum representation of your problem, which comprises identifying the key features. This method is known as representation learning, and it can often produce significantly better results than hand-designed representations.

Feature creation in machine learning is mainly done by hand by researchers and domain specialists. Deep learning, fortunately, can extract features automatically. Of course, this

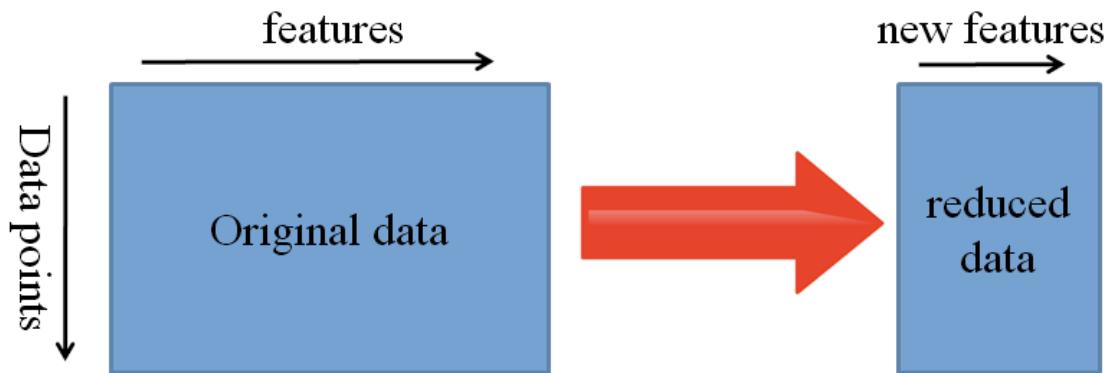


Figure (20) Extraction of Features

does not diminish the importance of feature engineering and domain knowledge; you must still choose which features to include in your network.

Neural networks, on the other hand, have the ability to learn which features are critical and which aren't. Even for complicated tasks that would otherwise necessitate a lot of human effort, a representation learning algorithm can find a decent combination of characteristics in a short amount of time.

The learned representation can then be applied to a variety of other challenges. Simply utilise the initial layers to find the appropriate feature representation, but avoid using the network's output because it is too task-specific. Instead, send data into your network and output it through one of the intermediate layers.

The raw data can then be understood as a representation of this layer.

This method is commonly used in computer vision since it can shrink your dataset, reducing computation time and making it more suited for classical algorithms.

4.2 SOTA models:

State-of-the-art (SOTA) DNNs are the best models you can use for any particular task. A DNN can be identified as SOTA based on its accuracy, speed, or any other metric of interest. However, in most computer vision areas, there is a trade-off between these metrics. That is, one can have a very fast DNN but its accuracy isn't up to the mark. Other times, we might be able to build a model with good performance metrics but it would lack the required latency or throughput across various tasks, such as image classification and detection.

The metrics we usually use to compare and evaluate DNNs are accuracy, precision, recall, F1-score for classification tasks, IoU, and mAP for object detection. A DNN will be declared state-of-the-art based on a combination of these metrics and additional performance metrics of interest, such as FLOPS, latency, throughput, and more.

Some SOTA models used in this project :

Several pre-trained models exist that can be applied using the TL such as ResNet50, ResNet101, VGG16, VGG19, Xception, MobileNet, MobileNetV2, DenseNet121, and DenseNet169. The usage of these pretrained models can result in an accuracy that is much better than the accuracy of a CNN built from scratch. The implementation of the TL can be done in one of two approaches, namely (1) Feature Extraction (Orenstein & Beijbom, 2017) and (2) Fine-Tuning (Guo et al., 2019).

In the first approach, the feature extractor that is part of the network is pretrained on the standard dataset (usually ImageNet) while the classifier is replaced and trained on the new data.

On the other hand, the second approach updates the weights of the entire pre-trained model, including the feature extractor part (Zhuang et al., 2020).

ResNet: ResNet stands for Residual Network; it means a deep network that is built upon the idea of residual learning. Residual learning is an interesting paradigm that is used to express a network that extracts residuals instead of features. This can help in solving the vanishing gradient problem.

ResNet-18: ResNet-18 is a convolutional neural network that is 18 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images

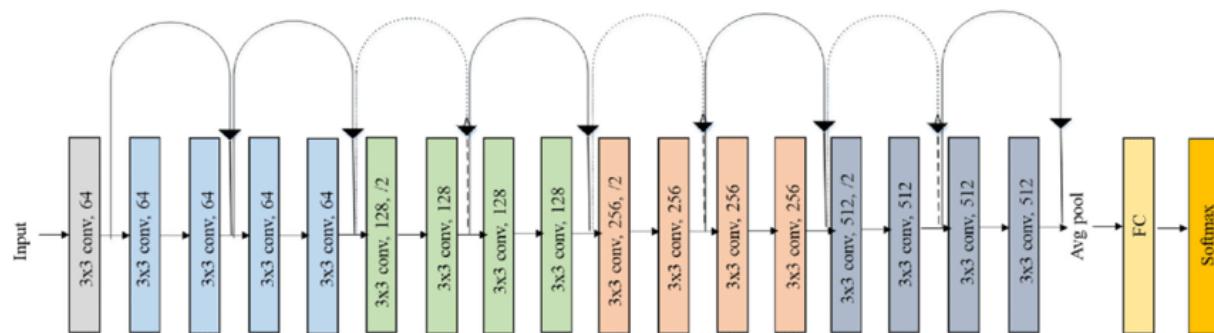


Figure (4-5) ResNet-18 Architecure

ResNet32: ResNet-32 is a convolution neural network backbone that is based off alternative ResNet networks such as ResNet-34, ResNet-50, and ResNet-101. As its name implies, ResNet-32 is has 32 layers. It addresses the problem of vanishing gradient with the identity shortcut connection that skips one or more layers.

ResNet50: ResNet50 is a version of ResNet that has 50 layers and 16 residual blocks (He et al., 2016).

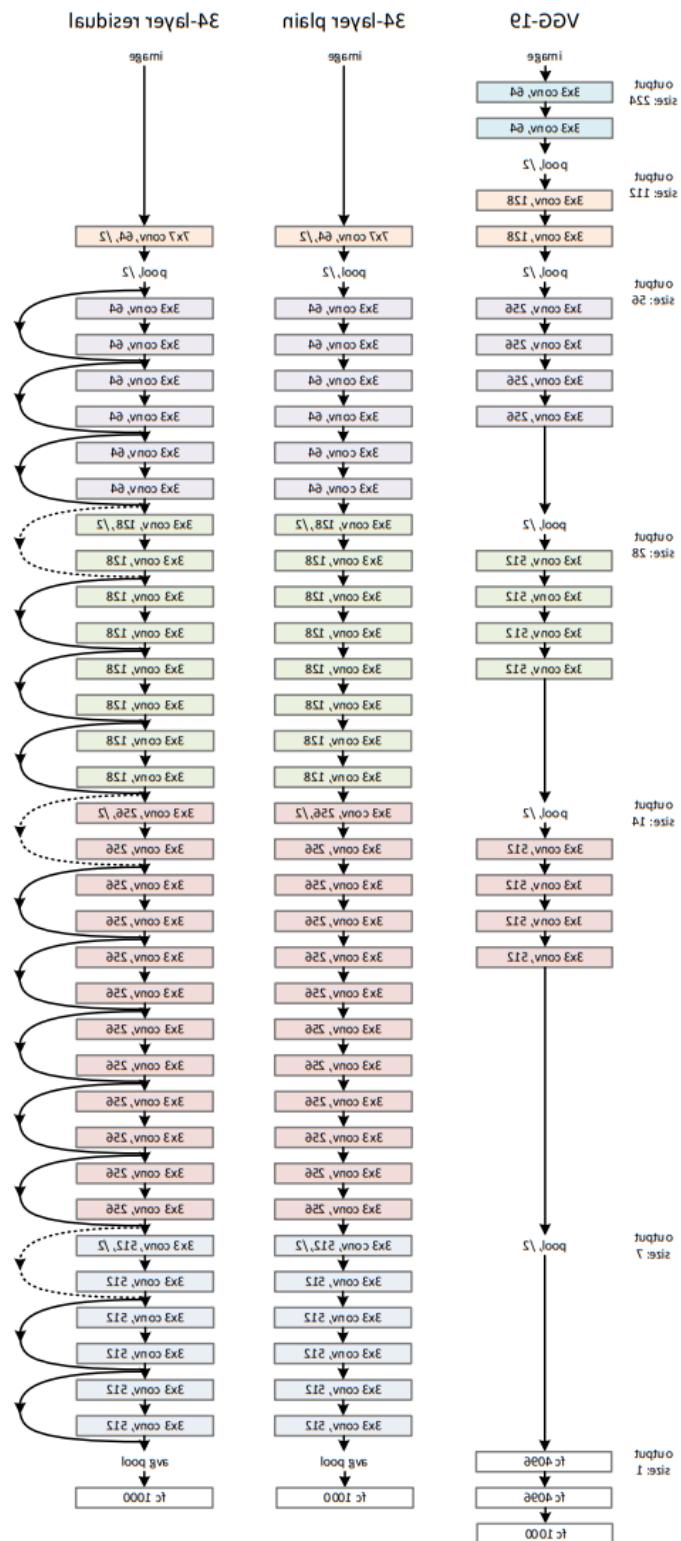


Figure (4-6) ResNet-32 Architecture

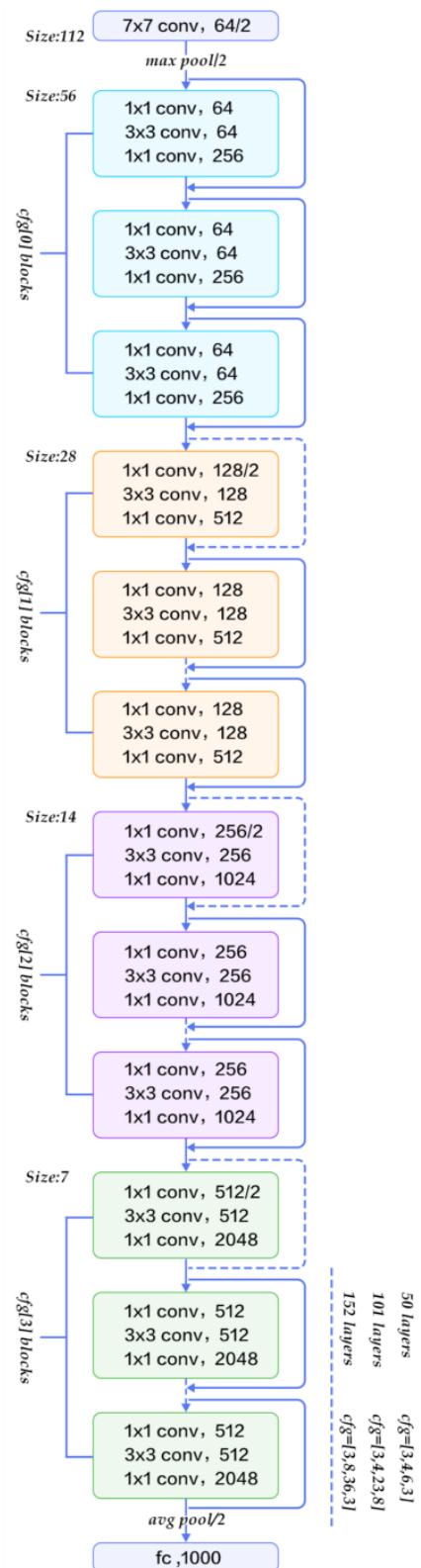


Figure (4-7) ResNet-50 Architecture

ResNet101: ResNet101 is another version of ResNet that applies the paradigm of residual learning. So, the vanishing gradient problem is solved in this type of network. This network contains 101 layers with 33 residual blocks (He et al., 2016).

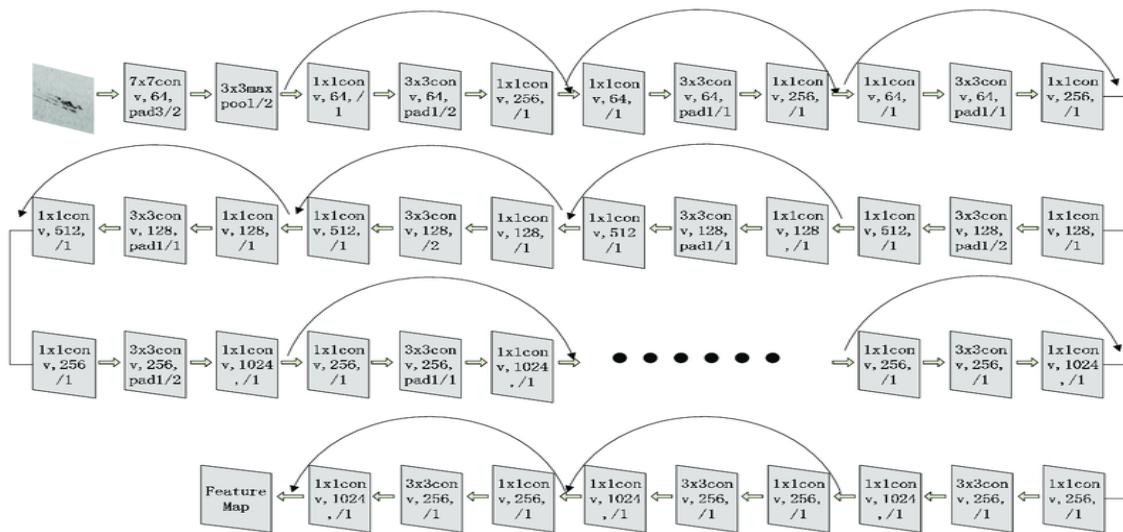


Figure (4-8) ResNet-101 Architecture

Xception: Xception stands for “extreme inception”. It is a deep CNN that is built on the idea of depth-wise separable convolution layers. It has 36 layers consisting of 2 convolution layers, depth-wise separable convolution layers, and 4 convolution layers. All the previous layers are followed by an FC layer at the end (Chollet, 2017).

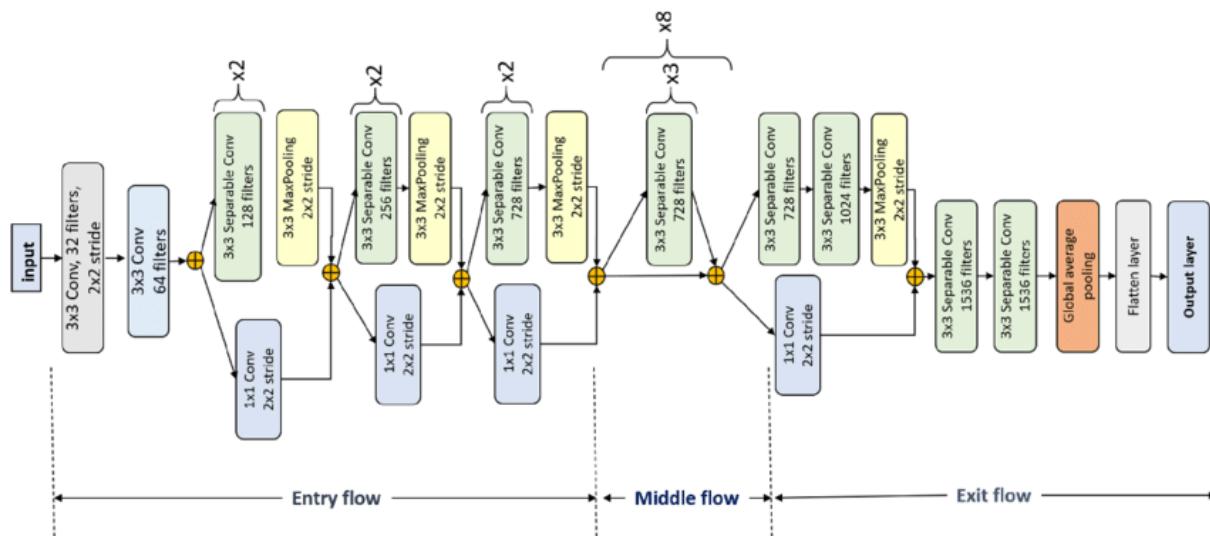


Figure (4-9) Xception Architecture

MobileNet: MobileNet is also based on the idea of depth-wise separable convolution layers. This is an efficient way to reduce the complexity and size of the model. It has 28 layers consisting of convolution layers, followed by depth-wise separable convolution layers. All the previous layers are followed by an FC layer at the end (Howard et al., 2017).

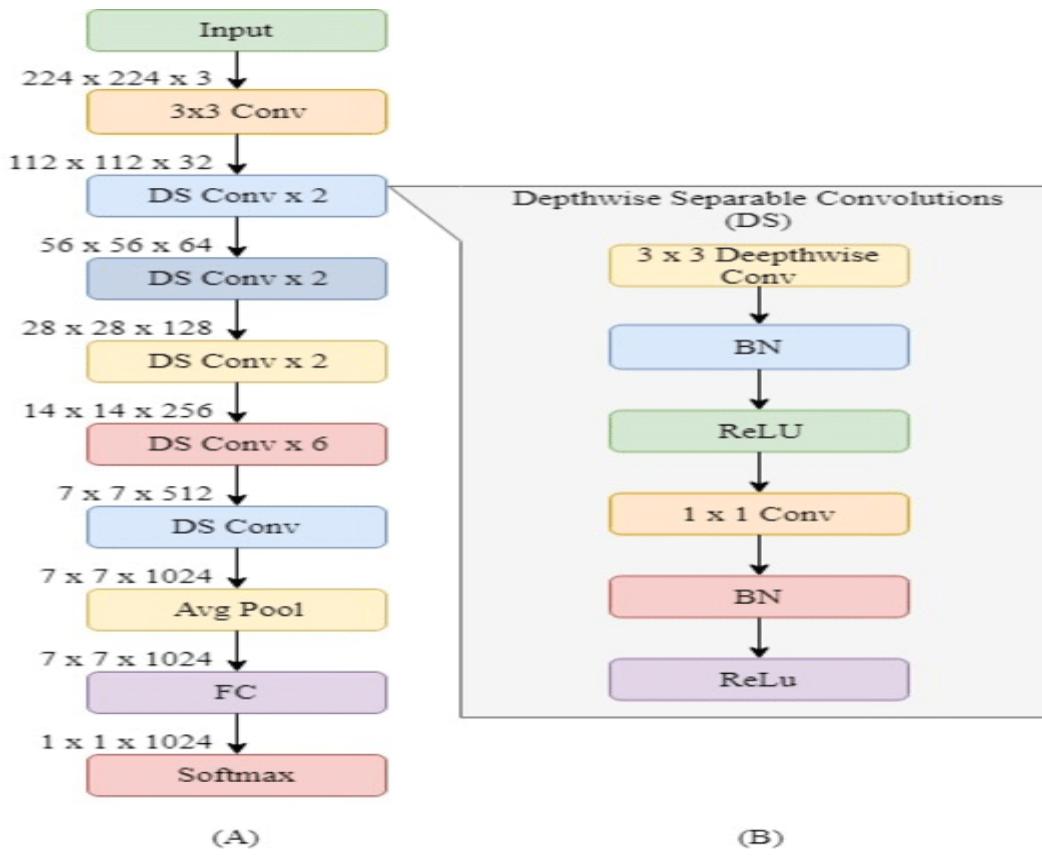


Figure (4-10) MobileNet-V1 Architecture

MobileNetV2: MobileNetV2 is a modified version of the MobileNet to include inverted residual blocks and linear bottlenecks. Therefore, this network is faster than the traditional MobileNet. It has 52 deep layers consisting of 3 convolution layers, 16 inverted residual and linear bottleneck blocks, and ends with a single convolution layer. All the previous layers are followed by an FC layer at the end (Sandler et al., 2018).

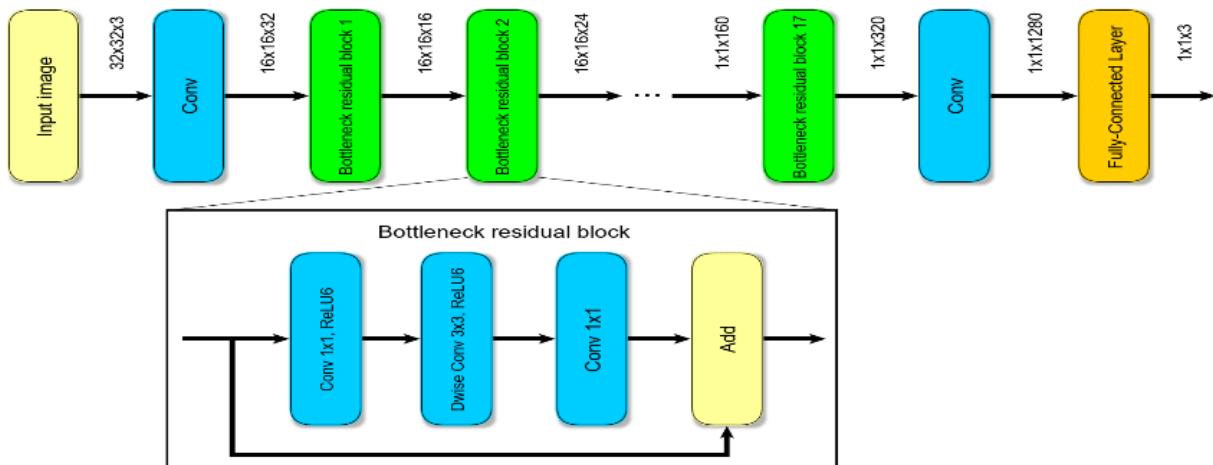


Figure (4-11) MobileNet-V2 Architecture

MobileNetV3: MobileNetV3 is a convolutional neural network that is tuned to mobile phone CPUs through a combination of hardware-aware network architecture search (NAS) complemented by the NetAdapt algorithm, and then subsequently improved through novel architecture advances.

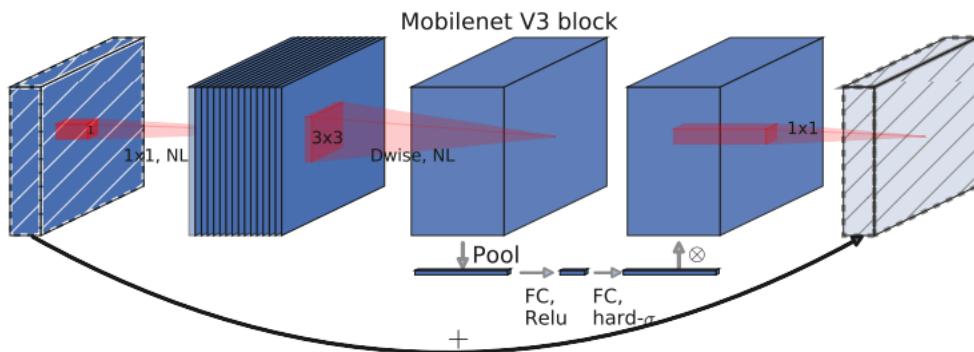


Figure (4-12) MobileNet-V3 building block

We can notice that: MobileNet-V3 is simply a MobileNet-V2 architecture plus squeeze-and-Excite [**]

DenseNet121: DenseNet stands for “Densely Connected Convolutional Networks”. It requires much fewer parameters than other CNN types. However, its architecture takes a long time for training because every layer is connected to all its following layers and as a result, every layer has to wait for the previous layers to take its input (Celik, Talo, Yildirim,

Karabatak, & Acharya, 2020). This problem was solved by introducing both the input image and the gradient values to all layers. DenseNet121 is a dense network with 121 layers. This type of networks contains 4 dense blocks. Transition layers consisting of convolution and pooling layers are also included between every two adjacent blocks to change the feature-map sizes (Huang et al., 2017).

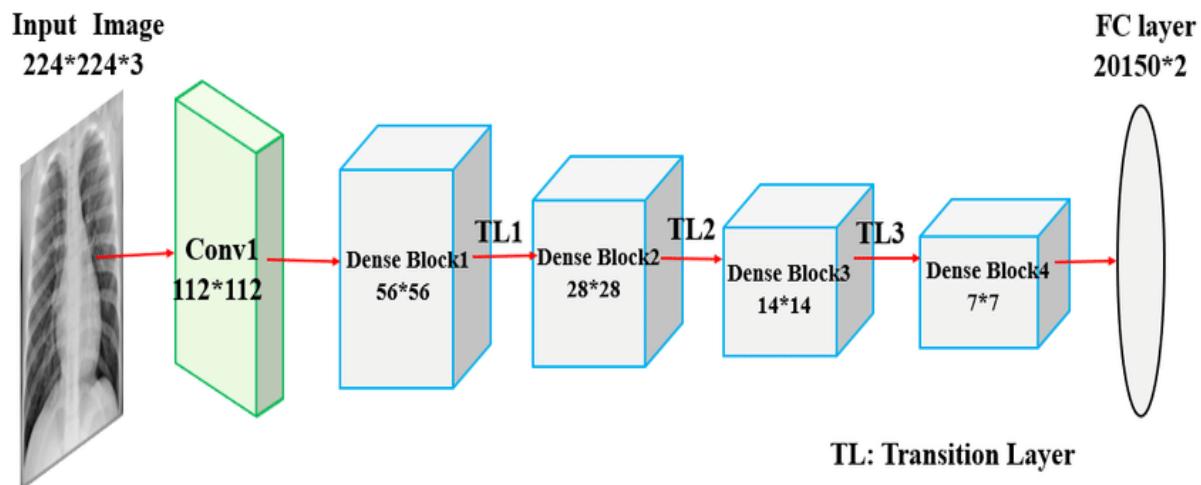


Figure (4-13) Dense-121 Architecture

DenseNet169: DenseNet169 is another dense network with 169 layers. This type of networks contains 4 dense blocks. Transition layers consisting of convolution and pooling layers are also included between every two adjacent blocks to change the feature-map sizes(Huang et al.,2017).

CoAtNets : (Convolution and Self-Attention Network) are a family of hybrid DNNs released in 2021 by Google Research. They were built mainly from two important principles, (i) Unifying depthwise convolution operations along with self-attention from Stacking Conv layers along with attention layers vertically to improve the overall generalization and efficiency of CNNs.

EfficientNet: is a CNN architecture that belongs to the family of models found automatically using Neural Architecture Search (NAS). EfficientNet models use a compound coefficient to uniformly scale the different dimensions of width, depth, and

resolution. This policy, unlike traditional approaches, does not scale the factors arbitrarily but follows a compound scaling method.

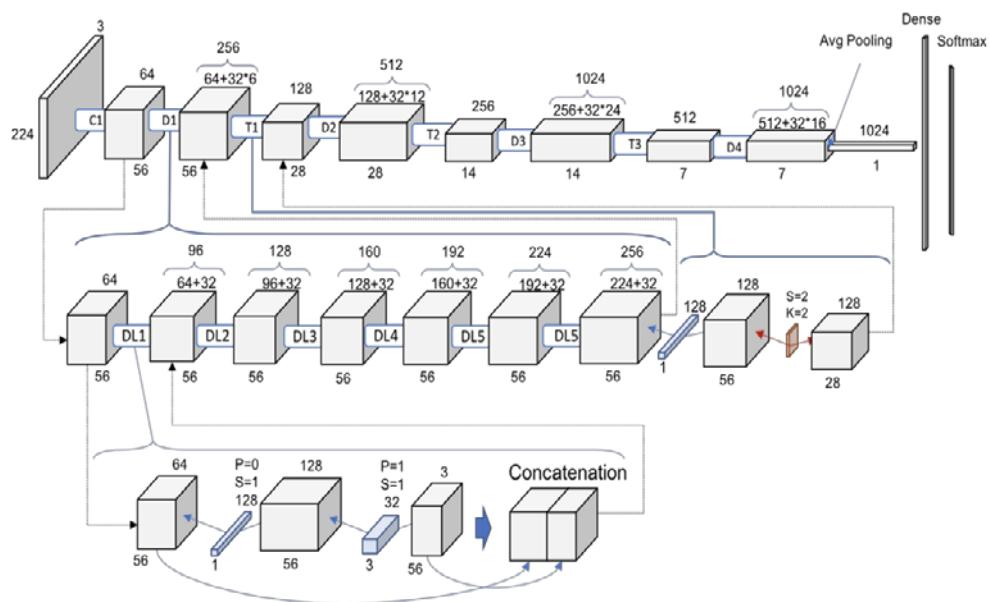


Figure (4-14) Dense-169 Architecture

Vision Transformer (ViT)

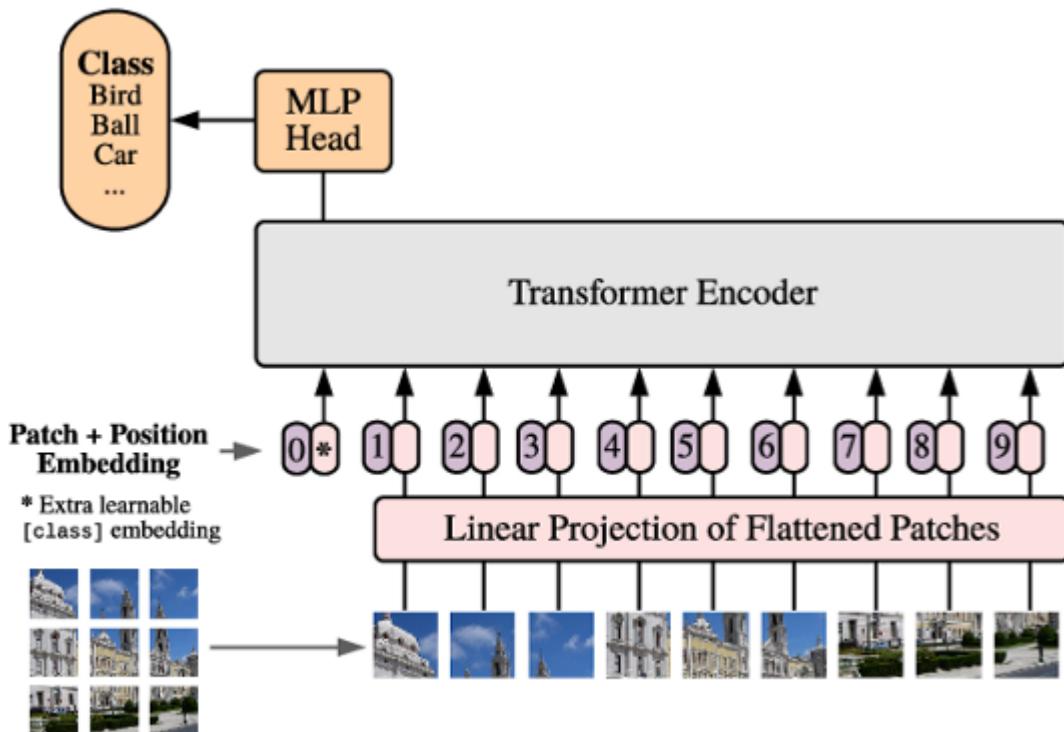


Figure (4-15) Vision-Transformer (Self-Attention Nets) Building blocks

Suppose the availability of computational resources increases by a factor of $2N$. This policy helps in increasing network depth by α , width by β and image resolution or size by γ , where these coefficients are determined by grid search. EfficientNet models also use a compounding coefficient Φ , scale the above coefficients in a uniform manner.

The models range from B0 to B7 depending on the number of model parameters. The model paper shows that the authors used Neural Architecture Search to increase the efficiency of the model and optimize both accuracy and FLOPS. The main block of EfficientNet is an MBConv (mobile inverted bottleneck convolution) to which squeeze-and-excitation optimization is added. The baseline B0 model of EfficientNet achieves a 77.3% accuracy on the ImageNet Dataset, using only 5.3M parameters and 0.39B FLOPs, a 90% decrease in the number of FLOPs compared to a ResNet-50 model. The largest EfficientNet model in the family of these models is EfficientNet-B7, which outperforms all the other models from B0-B6 on ImageNet with an 84.4% Top-1 accuracy and 97.1% Top-5 accuracy; it also uses approximately 66M parameters as part of the training with 37B FLOPs in total.

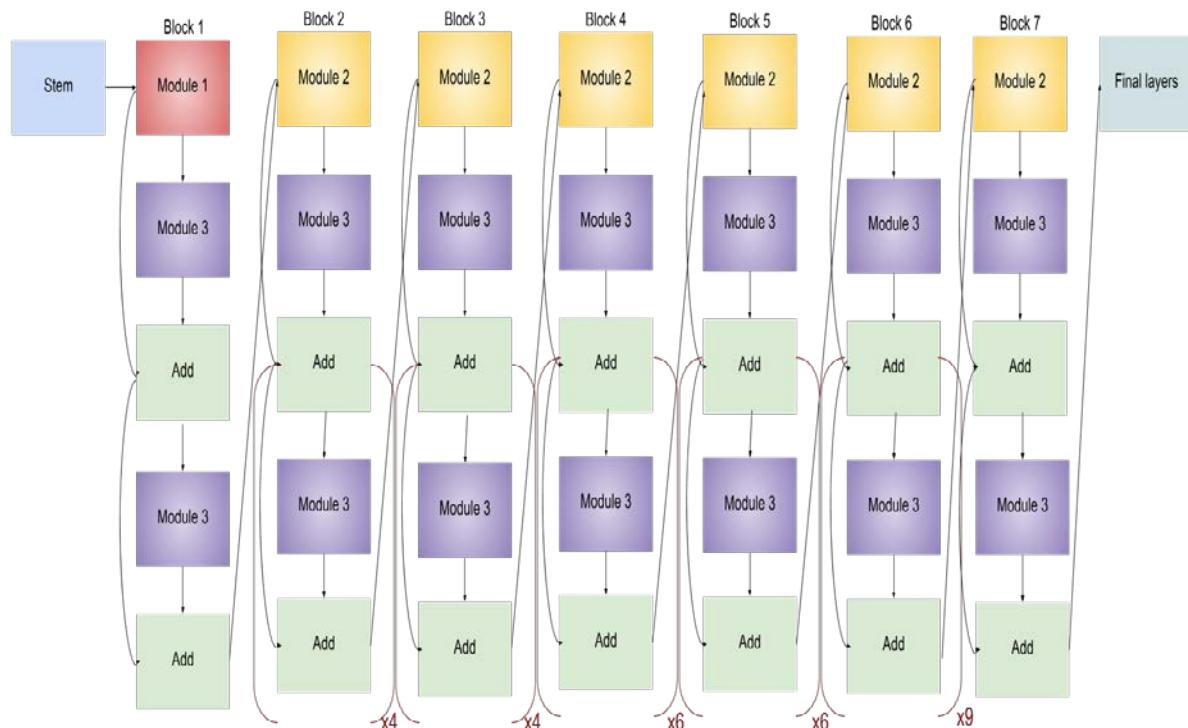


Figure (4-16) Efficient Net Building blocks

CHAPTER 5

EXPERIMENTAL

RESULTS AND

DISCUSSION

5 Chapter (5) Experimental Results and Discussion

5.1 Dataset

In this study, CheXpert dataset is used as the main source of chest X-ray images. CheXpert is one of the largest and trusted public datasets organized by Stanford ML group[3]. It consists of three main subsets. Firstly, a training set which consists of more than 220,000 high-resolution images from more than 64,500 different patients. Secondly, a validation set which consists of 234 images from 200 different patients. Lastly, a test set consists of 500 studies on which the consensus of 5 separate board-certified radiologists serves as ground truth. The radiographs in this dataset are a multi-view i.e., an image may belong to one of two main views: frontal view and lateral view. For the frontal-view images, an image may be anterior-posterior (AP) or posterior-anterior (PA). In addition, during label extraction from radiology reports, each image was labeled as follows: “1” for positive cases, “0” for negative ones, and “-1” for uncertainty. Despite having the labels for 14 different diseases, CheXpert competition is focusing on only 5 selected diseases: Cardiomegaly, Edema, Consolidation, Atelectasis, Pleural Effusion. Hence the main interest of this paper will be regarding only these 5 diseases. **This dataset is chosen for two main reasons.** First, CheXpert Competition is an Open medical AI competition so the results can be compared with others. Second, CheXpert labeler has proved its superiority over the method employed in Peng et al.[22] which was used to annotate both chestX-ray8[23] and its expansion chestX-ray14 as shown in the original paper by J. Irvin et al.[3]

Data Exploration

- Uncertainty Labels:

challenging issue in the dataset is that we do not have full access to the true labels of all images i.e., some images have “-1” as a label which indicates that the labeler can't specify whether it's a “0” or “1”. The percentage of these uncertainty labels is shown in Fig. (5-1). Hence, we need to devote extra effort to impute them.

Different approaches are suggested by J. Irvin et al.[3] for dealing with the uncertainty labels as shown in figure (5-2). We adopted the approach suggested In[18], who are the top performer in the competition, to map uncertain labels “-1” to negative “0” for three classes (Cardiomegaly, Consolidation and Pleural Effusion) and to positive “1” for the rest (Edema and Atelectasis).

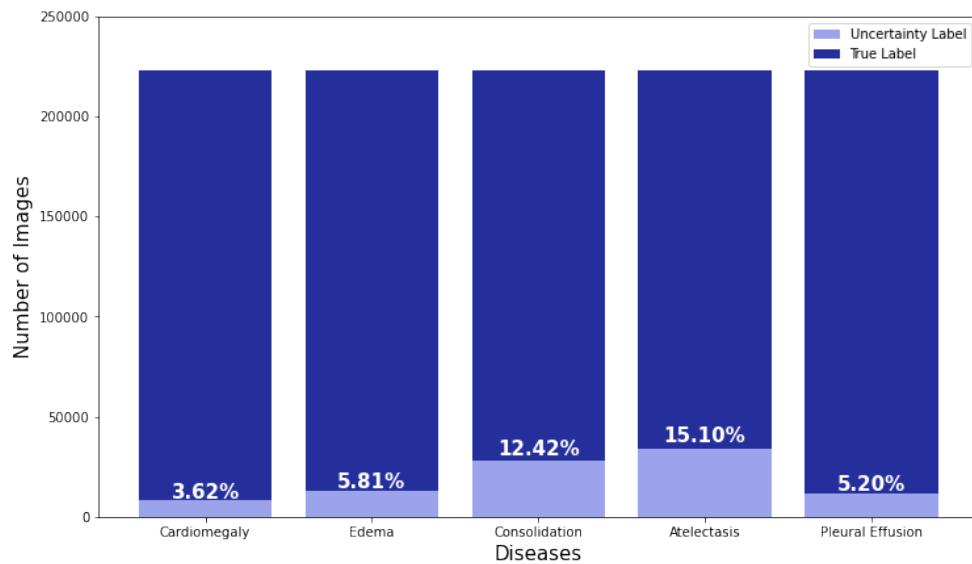


Figure (5-1) Percentage of Uncertainty Labels

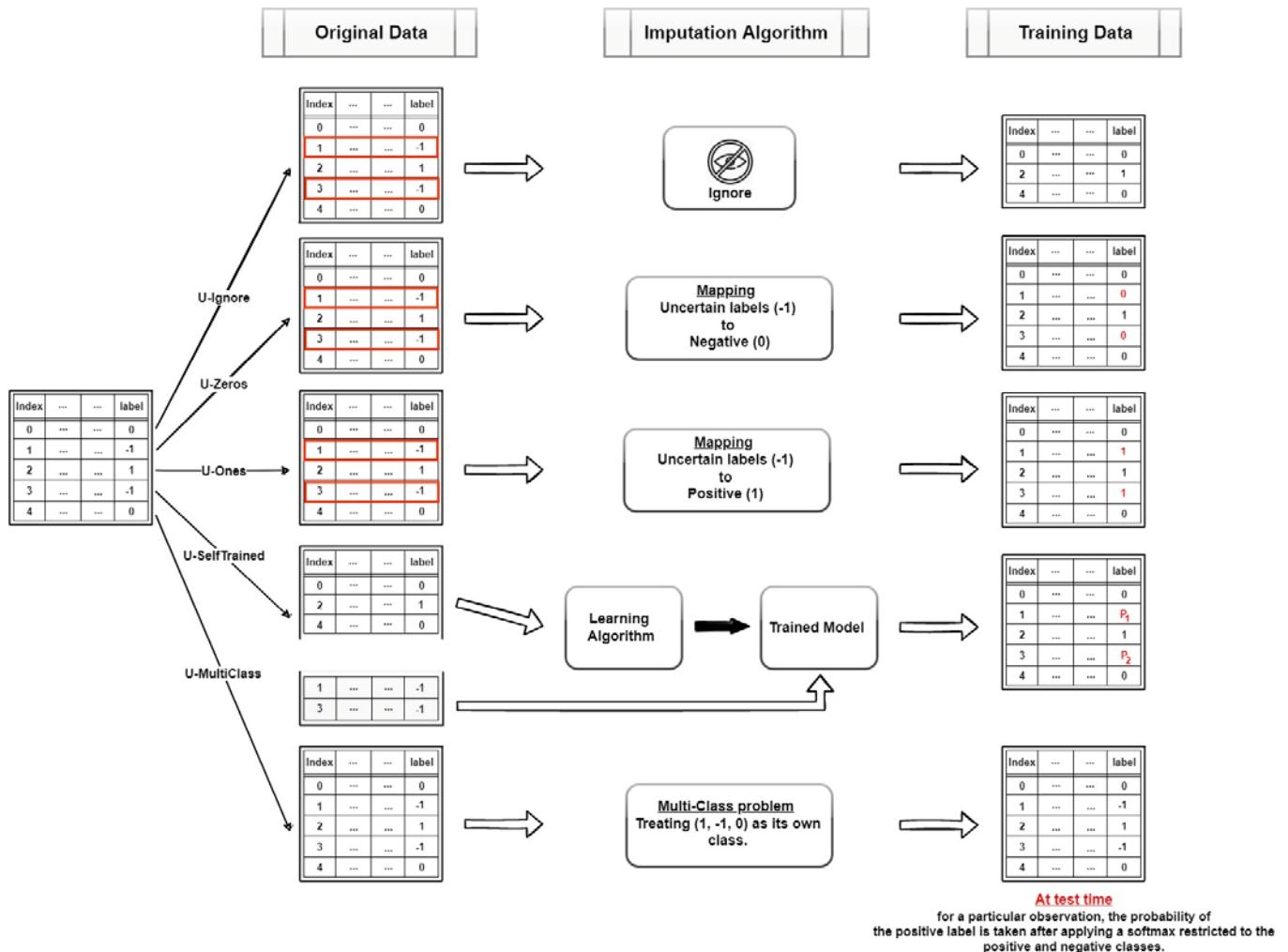


Figure (5-2) Imputation Algorithms suggested by Stanford ML group

- *Age and Gender Distribution*

The dataset is distributed between ages from a couple months to 90 years old. As indicated in Fig. (5-3), the age classes are defined as following: [Adolescent (<18), Young Adult (18-24), Adult (24-45), Mid-Age (45-64), Aged (64<)]. In addition, it is demonstrated that the “aged” group i.e., patients above 64 years old, represents the biggest portion which is something reasonable because the older we become the more likely we catch lung diseases as our lungs get weaker. Moreover, nearly 59% of the radiographs belong to Males and 41% to Females. Noticeably, Men are more likely to complain about their lungs due to bad habits like smoking that are more popular among men than women.

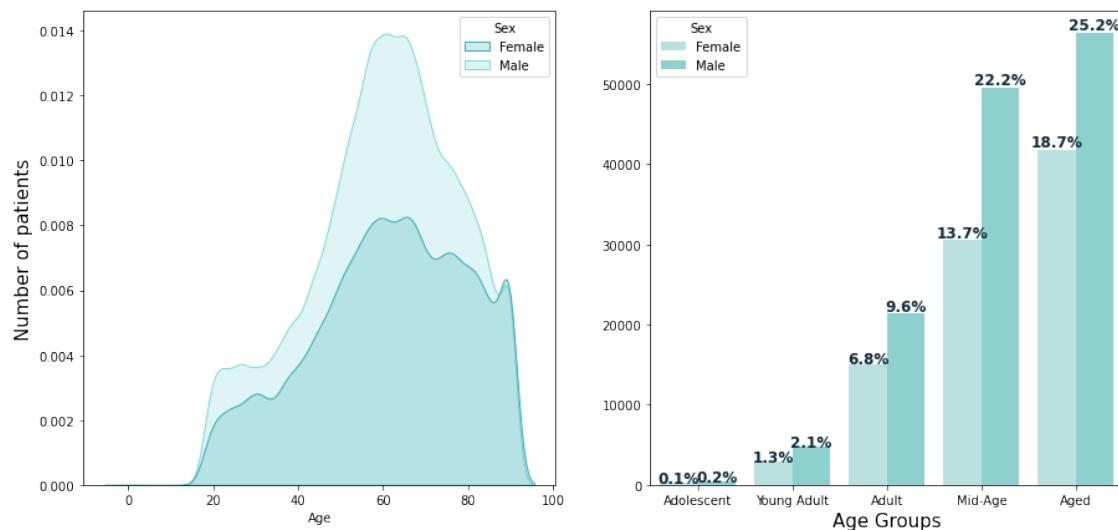


Figure (5-3) Age and Gender Distribution

- Diseases Distribution

The dataset contains two different types of Chest X-rays: Frontal views occupy about 85 % of the radiographs and Lateral views that represent about 15% of the data. According to Fig. (5-4), positive cases (1) represent 12.2% and 11.2% of Frontal and Lateral observations for Cardiomegaly, 32.2% and 11.5% for Edema, 6.8% and 5.6% for Consolidation, 31.2% and 23.3% for Atelectasis, and 40.3% and 28.7% for Pleural Effusion respectively.

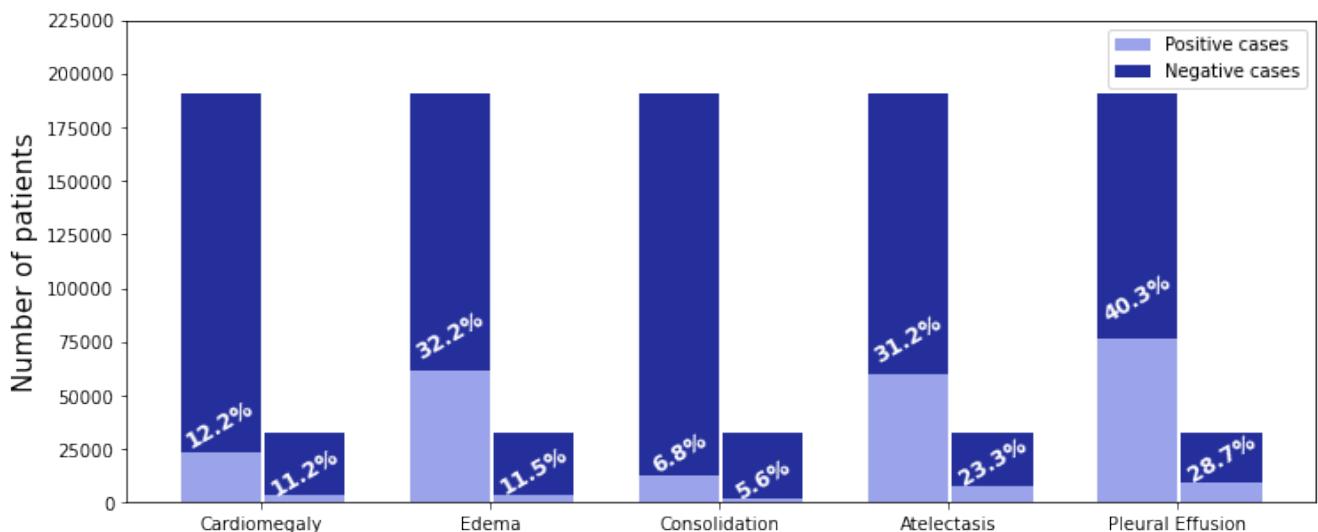


Figure (5-4) Diseases Distribution

5.2 Experiments and Results

different comparisons with the most SOTA models that employed CheXpert dataset in their classification problem are demonstrated and the results are indicated visually and computationally.. We have used SOTA models that came out between 2016 and 2022 pretrained on ImageNet and all training stages was done using Python (v3.7.13) with Pytorch framework (v1.11) and Cuda (v11.3) operating on Google Colab Pro service that provides Nvidia Tesla P100 with 16GB of dedicated memory as our main graphic processing unit (GPU). In addition, frontal view radiographs occupy around 85.5% of the dataset. Hence, solving the challenges of frontal view radiographs is going to have a remarkable impact on the stability of the model during evaluation and how the model is going to perform in real time scenarios. According to that, we have paid more attention to frontal view radiographs.

5.2.1 The effect of image size on the performance

We began our study by training each model on different image sizes: 224, 256, 320 to get more insights about the behavior of our models with different image resolutions. Models have been trained for 5 epochs each starting with same learning rate 1e-3 and being automatically decreased by half if there is no improvement in the performance for 2000 consecutive batches. Tables (1), (2), and (3) are showing the results for 224, 256, and 320 respectively.

**Table (1) AUC scores on the official validation set for different SOTA models trained on
input image size of 224**

Model Name	Cardiomegaly	Edema	Consolidation	Atelectasis	Pleural Effusion	Average
ResNet-18	0.858177	0.920089	0.921691	0.835906	0.91361	0.889895
ResNet-34	0.858846	0.935714	0.925368	0.814488	0.924706	0.891824
ResNet-50	0.84258	0.919196	0.92114	0.823412	0.942369	0.889739
ResNet-101	0.871324	0.934524	0.916728	0.7979	0.935688	0.891233
ResNet-152	0.879345	0.920089	0.917831	0.857008	0.919158	0.898686
Inception_v4	0.844029	0.94494	0.931618	0.861837	0.922328	0.90095
Xception	0.855615	0.912351	0.920956	0.850289	0.945312	0.896905
Inception_ResNet_v2	0.851604	0.91622	0.921875	0.830026	0.924932	0.888932
MobileNet_v2	0.854055	0.945387	0.90625	0.813753	0.943274	0.892544
MobileNet_v3	0.862634	0.928571	0.909375	0.836745	0.937274	0.89492
EfficientNet-B0	0.857509	0.94122	0.934375	0.835276	0.93048	0.899772
EfficientNet-B1	0.827317	0.93244	0.934375	0.821207	0.93014	0.889096
EfficientNet-B2	0.8584	0.940923	0.94136	0.834646	0.936481	0.90236
EfficientNet-B3	0.848485	0.933185	0.936213	0.826667	0.930707	0.895051
EfficientNet-B4	0.832442	0.9375	0.919118	0.815643	0.936821	0.888305
EfficientNet_v2	0.875	0.939435	0.922978	0.864987	0.943501	0.90918
ReXNet	0.821078	0.94003	0.927206	0.824462	0.921649	0.886885
DenseNet-121	0.870432	0.940327	0.921875	0.814803	0.929574	0.895402
DenseNet-161	0.834559	0.919048	0.931985	0.865827	0.925838	0.895451
DenseNet-201	0.830102	0.936161	0.945037	0.83622	0.935462	0.896596
DenseNet-169	0.861408	0.925	0.910662	0.828241	0.927083	0.890479
Total Average	0.852140	0.931540	0.924667	0.832826	0.931257	0.894486

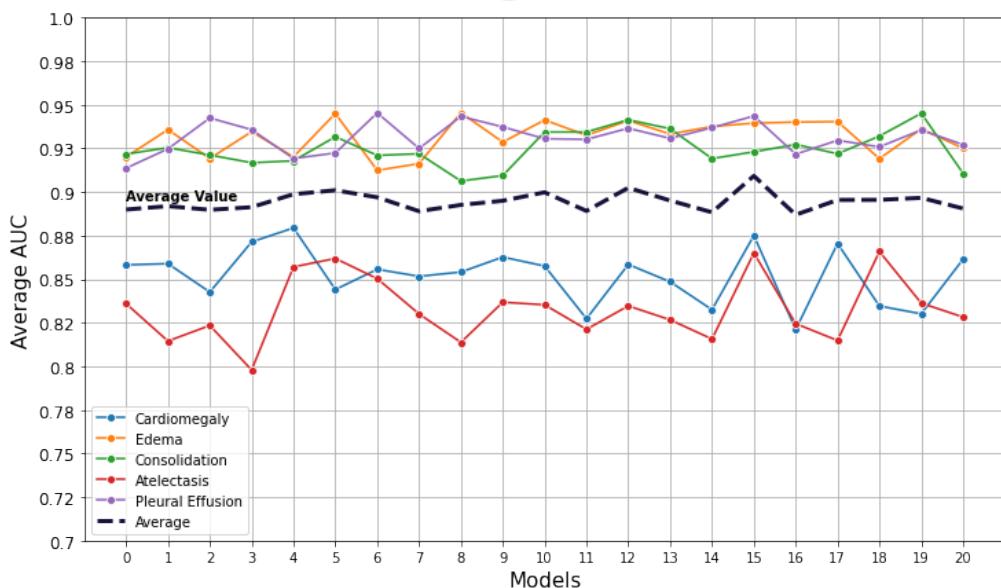


Figure (5-5) The Behaviour of Models when trained on image sizes of 224

As shown in table (1), the average AUC of all models trained on image size of 224 was **89.45%**. In addition, taking each model that has an outstanding result on a specific disease and combining them all through an ensemble model, the maximum result we could get was **91.62%** on average. As Fig. (5-5) illustrates, the general trend of the models trained on 224-size is that there is a **noticeable variance** among different models' performance leading to **less chances of getting high performances on the validation dataset**.

As shown in table (2), the average AUC of all models trained on image size of 256 was **89.58%**. In addition, taking each model that has an outstanding result on a specific disease and combining them all through an ensemble model, the maximum result we could get was **91.89%** on average. As illustrated in Fig. (5-6), the general trend of the models trained on 256 is that **the variance between the different models' performance has been reduced leading to high chances of getting better performance on the validation dataset**.

**Table (2) AUC scores on the official validation set for different SOTA models
trained on input image size of 256**

Model Name	Cardiomegaly	Edema	Consolidation	Atelectasis	Pleural Effusion	Average
ResNet-18	0.866644	0.931696	0.918566	0.820997	0.942595	0.8961
ResNet-34	0.862857	0.9375	0.916176	0.828451	0.925385	0.894074
ResNet-50	0.857732	0.930506	0.920588	0.844094	0.924026	0.895389
ResNet-101	0.855949	0.927827	0.924265	0.835276	0.938745	0.896412
ResNet-152	0.855169	0.941815	0.915257	0.822152	0.935122	0.893903
Inception_v4	0.850045	0.94375	0.943566	0.844304	0.929801	0.902293
Xception	0.815954	0.930208	0.923346	0.856168	0.936821	0.892499
Inception_ResNet_v2	0.824866	0.929613	0.927574	0.848609	0.928555	0.891843
MobileNet_v2	0.848819	0.954762	0.931066	0.84294	0.917912	0.8991
MobileNet_v3	0.829211	0.944792	0.926287	0.83622	0.915761	0.890454
EfficientNet-B0	0.833556	0.94628	0.944853	0.840315	0.921082	0.897217
EfficientNet-B1	0.818293	0.94494	0.936213	0.837585	0.908741	0.889155
EfficientNet-B2	0.855281	0.943006	0.907353	0.817638	0.929688	0.890593
EfficientNet-B3	0.883578	0.937798	0.930699	0.852388	0.931386	0.90717
EfficientNet-B4	0.853498	0.938988	0.917096	0.834226	0.922101	0.893182
EfficientNet_v2	0.879122	0.921577	0.926654	0.845144	0.931159	0.900732
ReXNet	0.867981	0.950298	0.923529	0.822362	0.932858	0.899406
DenseNet-121	0.838235	0.941071	0.928493	0.84273	0.933537	0.896813
DenseNet-161	0.850713	0.930506	0.925919	0.831286	0.934556	0.894596
DenseNet-201	0.85361	0.918155	0.911213	0.857743	0.935575	0.895259
DenseNet-169	0.811163	0.93125	0.925735	0.862467	0.948936	0.89591
Total Average	0.848204	0.936968	0.924974	0.839195	0.929731	0.895814

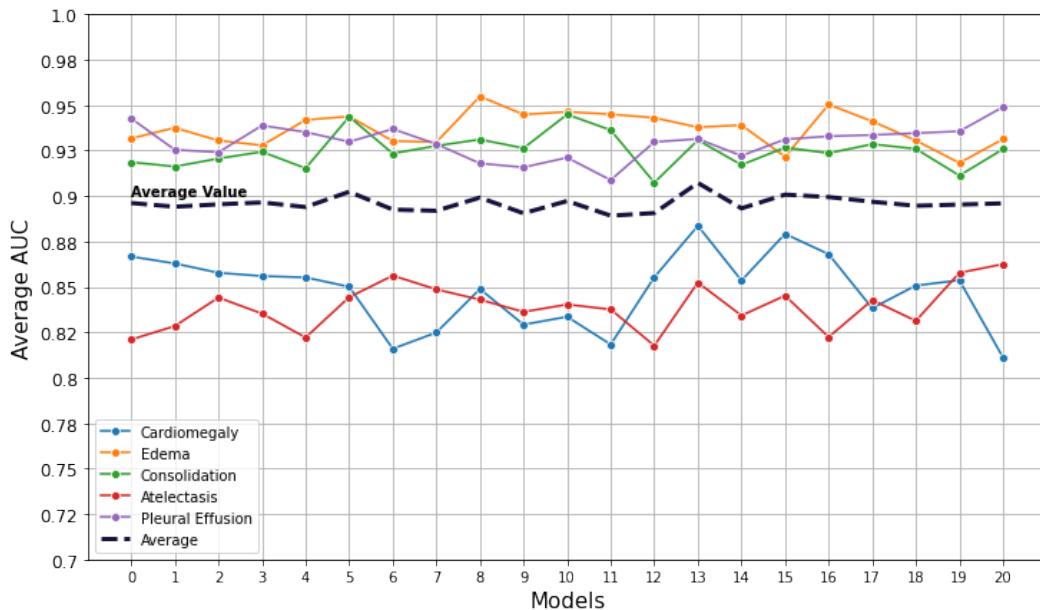


Figure (5-6) The Behaviour of Models when trained on image sizes of 256

From table (3), the average AUC of all models trained on image size of 320 was **89.94%**. Moreover, taking each model that has an outstanding result on a specific disease and combining them all through an ensemble model, the maximum result we could get was **92.33%** on average. As demonstrated by Fig. (5-7), the general trend of the models trained on 320 is that **the variance here is the least which proves the effectiveness of having higher image sizes** (except for Pleural Effusion which has the best results in small resolutions).

**Table (3) AUC scores on the official validation set for different SOTA models
trained on input image size of 320**

Model Name	Cardiomegaly	Edema	Consolidation	Atelectasis	Pleural Effusion	Average
ResNet-18	0.858846	0.917411	0.925551	0.867507	0.930933	0.90005
ResNet-34	0.864082	0.934226	0.913419	0.83769	0.927536	0.895391
ResNet-50	0.841689	0.931845	0.942463	0.879055	0.933877	0.905786
ResNet-101	0.868873	0.922768	0.909926	0.862887	0.921309	0.897153
ResNet-152	0.864416	0.937649	0.949265	0.854278	0.925611	0.906244
Inception_v4	0.861297	0.943452	0.925919	0.820787	0.924366	0.8952
Xception	0.847148	0.936458	0.918199	0.84399	0.932178	0.895595
Inception_ResNet_v2	0.861742	0.93006	0.925184	0.853123	0.935915	0.901205
MobileNet_v2	0.861074	0.936012	0.932721	0.837165	0.931499	0.899694
MobileNet_v3	0.896502	0.938839	0.894485	0.833596	0.930593	0.898803
EfficientNet-B0	0.861408	0.941964	0.933456	0.834541	0.924253	0.899124
EfficientNet-B1	0.839127	0.941667	0.935662	0.873281	0.925725	0.903092
EfficientNet-B2	0.876671	0.936905	0.940993	0.860262	0.917799	0.906526
EfficientNet-B3	0.883578	0.948363	0.898897	0.828241	0.937726	0.899361
EfficientNet-B4	0.843471	0.938988	0.917463	0.848084	0.91293	0.892187
EfficientNet_v2	0.875668	0.936458	0.923529	0.847979	0.936934	0.904114
ReXNet	0.834225	0.952083	0.927206	0.841575	0.930593	0.897136
DenseNet-121	0.859403	0.942262	0.938603	0.834751	0.927763	0.900556
DenseNet-161	0.850379	0.944048	0.937132	0.827087	0.939425	0.899614
DenseNet-201	0.848819	0.943899	0.932904	0.830341	0.930593	0.897311
DenseNet-169	0.851827	0.945982	0.917096	0.825827	0.928895	0.893925
Total Average	0.859535	0.938159	0.925718	0.844859	0.928879	0.899432

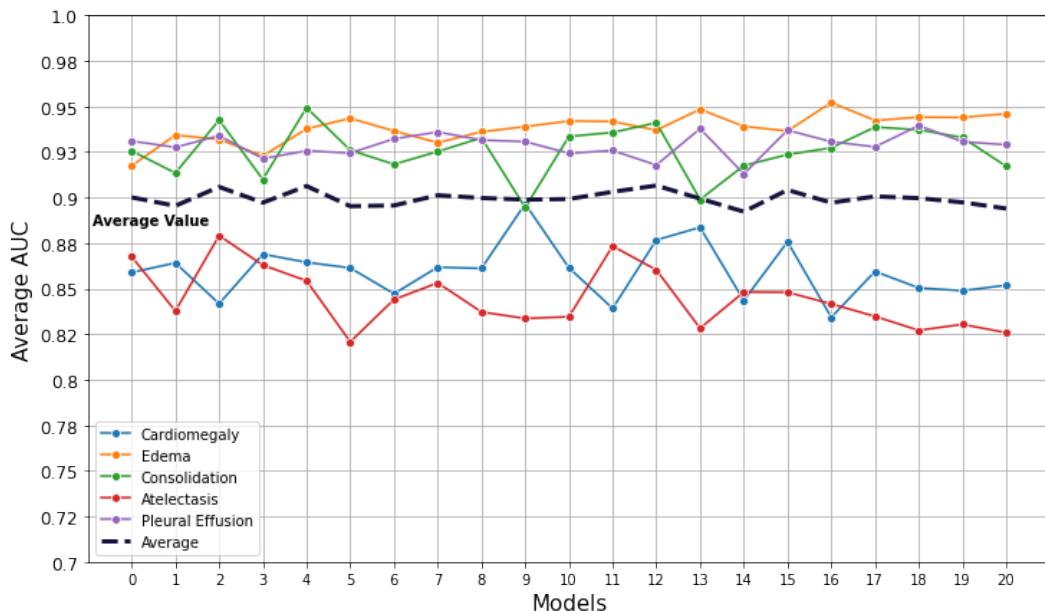


Figure (5-7) The Behaviour of Models when trained on image sizes of 320

5.2.2 The effect of the proposed preprocessing techniques on the performance

Based on the previous investigation, The top-5 models were selected for conducting an ablation study to investigate how beneficial it is to use each of the proposed preprocessing techniques individually i.e., training each model in 4 different scenarios with and without the proposed preprocessing technique: Top-Bottom transformation, GoDec algorithm, and Fourier Transform. Instead of training on the ImageNet-pretrained weights, we used the weights learned from the previous experiment. As the previous models were trained in a multi-label classification manner, we needed to extract these learned weights and use them to initiate our models in the current experiment. This extraction process is illustrated in Fig. (5-8).

After loading the weights to binary-classification-specific model, we trained each model for 5 epochs with the same specifications and setups as demonstrated previously. The final results for Cardiomegaly, Edema, Consolidation, Atelectasis, and Pleural Effusion are shown in Tables (4), (5), (6), (7), (8) respectively.

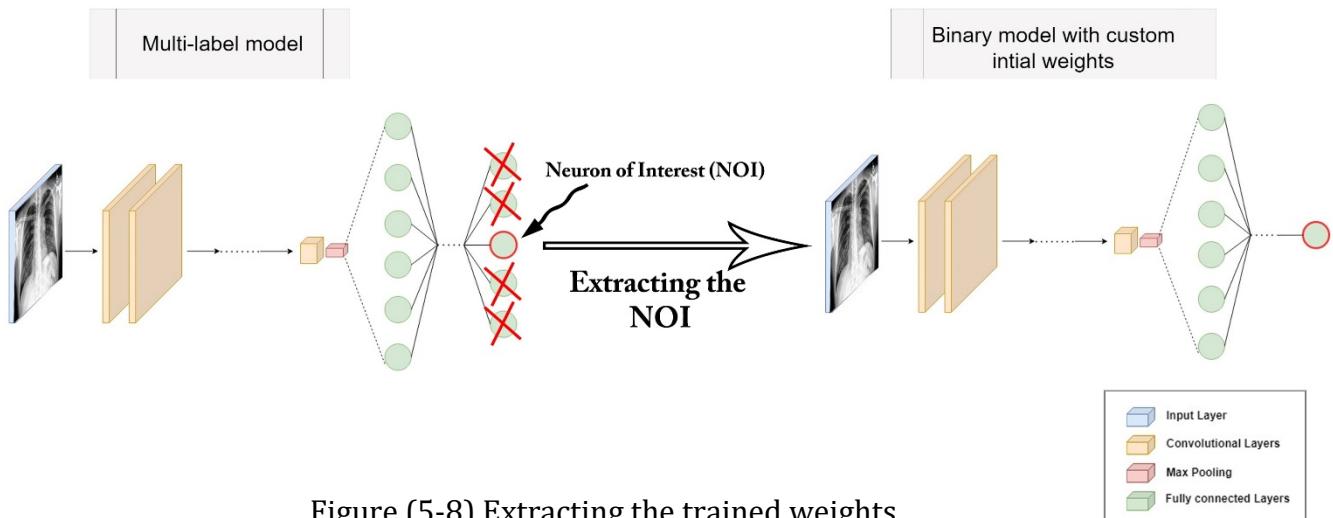


Figure (5-8) Extracting the trained weights

Table (4) Cardiomegaly Analysis

Model Name	Without Filter	Top-Bottom	GoDec	Fourier
MobileNet_v3	0.8811	0.8992	0.9002	0.8889
EfficientNet-B3	0.8864	0.8893	0.9013	0.8982
ResNet-152	0.88	0.8709	0.8847	0.8871
EfficientNet_v2	0.8871	0.8717	0.8963	0.8937
EfficientNet-B2	0.8787	0.8871	0.8929	0.8897
Total Average	0.88266	0.88364	0.89508	0.89152

As illustrated in Table (4), GoDec algorithm suppresses the unnecessary details in the image which helped the models in extracting the deep features that distinguishes "Cardiomegaly" (two models have surpassed 90% with GoDec algorithm which is relatively a very competitive score for a single model). In addition, the average AUC by Fourier Transform (FT) is very close to the one by GoDec, and that is because FT has a

HPF which removes the low frequency components in the image leaving the model to pay more attention to the deep details. On the other hand, contrast enhancement by Top-bottom (TB) transform didn't help the models as much as the other filters.

Table (5) Edema Analysis

Model Name	Without Filter	Top-Bottom	GoDec	Fourier
MobileNet_v2	0.9466	0.9488	0.942	0.9458
ReXNet	0.957	0.9613	0.9512	0.9557
EfficientNet-B3	0.9396	0.9519	0.9435	0.95
EfficientNet-B0	0.9396	0.9473	0.9435	0.9469
DenseNet-169	0.9496	0.9507	0.946	0.9478
Total Average	0.94648	0.952	0.94524	0.94924

As shown in table (5), Contrast enhancement, by TB transform, helped in exposing and highlighting the fluid trapped in the lung and made it easier for the models to adapt neurons' weights to detect "Edema" more accurately. Highlighting high frequency components in radiographs by FT didn't upraise the results like TB did but still ahead of GoDec. Unlike Cardiomegaly, Darkening the image or suppressing details by GoDec didn't help in improving the results as did the other two.

Table (6) Consolidation Analysis

Model Name	Without Filter	Top-Bottom	GoDec	Fourier
ResNet-152	0.9362	0.9382	0.9351	0.9438
DenseNet-201	0.9417	0.9265	0.936	0.9463
EfficientNet-B0	0.9311	0.9347	0.9463	0.9335
Inception_v4	0.9276	0.941	0.9289	0.9281
ResNet-50	0.9364	0.9123	0.9241	0.9369
Total Average	0.9346	0.93054	0.93408	0.93772

As shown in table (6), strengthening high frequency components by FT did help in exposing the main features of consolidation. In contrast, neither suppressing the details by GoDec or contrast enhancement, by TB transform, helped in highlighting these features.

Table (7) Atelectasis Analysis

Model Name	Without Filter	Top-Bottom	GoDec	Fourier
ResNet-50	0.8553	0.8641	0.8653	0.8581
EfficientNet-B1	0.8503	0.875	0.8433	0.8656
ResNet-18	0.8538	0.8512	0.8591	0.8557
DenseNet-161	0.8538	0.8791	0.8393	0.8584
EfficientNet_v2	0.854	0.8525	0.8506	0.8407
Total Average	0.85344	0.86438	0.85152	0.8557

As shown in table (7), Contrast enhancement, by TB transform, helped in exposing and highlighting the alveoli and made it easier for the models to adapt neurons' weights to detect "Atelectasis" more accurately. Highlighting high frequency components in radiographs by FT didn't upraise the results like TB did but still ahead of GoDec. In addition, Darkening the image or suppressing details by GoDec didn't help in improving the results here as did the other two.

Table (8) Pleural Effusion Analysis

Model Name	Without Filter	Top-Bottom	GoDec	Fourier
DenseNet-169	0.9467	0.9415	0.9462	0.9464
Xception	0.9428	0.9404	0.9449	0.9438
EfficientNet_v2	0.9495	0.9453	0.9412	0.9489
MobileNet_v2	0.9472	0.9428	0.9411	0.9477
ResNet-18	0.9476	0.9398	0.9406	0.9484
Total Average	0.94676	0.94196	0.9428	0.94704

As shown in table (8), weakening the low frequency components and strengthening the high frequency ones by FT did help here in highlighting the fluid trapped in lung pleura. On the other hand, Contrast enhancement, by TB transform, and suppressing details by GoDec didn't help in improving the results here as did Fourier Transform.

5.2.3 The proposed ensemble model architecture.

After the previous ablation studies, we were able to combine the best preprocessing techniques with the best stat-of-the-art models in one ensemble architecture. Our proposed ensemble architecture is fully demonstrated in figure (5-10). The results on the validation set are shown in table (9) with the comparison of the baseline provided by Stanford ML group [3] and the work [17] that have the 2nd place at the time of writing. The ROC curves are provided in Fig. (11). As illustrated in table (9), Our proposed ensemble surpassed the Baseline introduced by Stanford ML group but still not good enough compared to the other work [17]. Despite having lower AUC than that work, our proposed more is way simpler. First, they used 6 ensemble models we used only 5. Second, they used a conditional training strategy while on the other hand, we used an ordinary training strategy. Last but not least, they used test-time augmentation (TTA) i.e., for each test CXR, they applied a random transformation (amongst horizontal flipping, rotating ± 7 degrees, scaling $\pm 2\%$, and shearing ± 5 pixels) 10 times and then averaged the outputs of the model on the 10 transformed samples to get the final prediction. Thus, we don't have the best results, but we wanted to prove that just by selecting the most appropriate preprocessing techniques can compensate us for complex training strategies and slow real time inference.

Table (9) Comparison with other performances on the validation set

Work	Cardiomegaly	Edema	Consolidation	Atelectasis	Pleural Effusion	Average
U-Ignore [3]	0.828	0.934	0.938	0.818	0.928	0.8892
U-Zeros [3]	0.840	0.929	0.932	0.811	0.931	0.8886
U-Ones [3]	0.832	0.941	0.899	0.858	0.934	0.8928
U-SelfTrained [3]	0.831	0.935	0.939	0.833	0.932	0.894
U-MultiClass [3]	0.854	0.928	0.937	0.821	0.936	0.8952
Ensemble without TTA [17]	0.906	0.951	0.955	0.908	0.958	0.937
Ensemble with TTA [17]	0.910	0.958	0.957	0.909	0.964	0.940
Our Ensemble	0.901	0.962	0.958	0.890	0.953	0.933

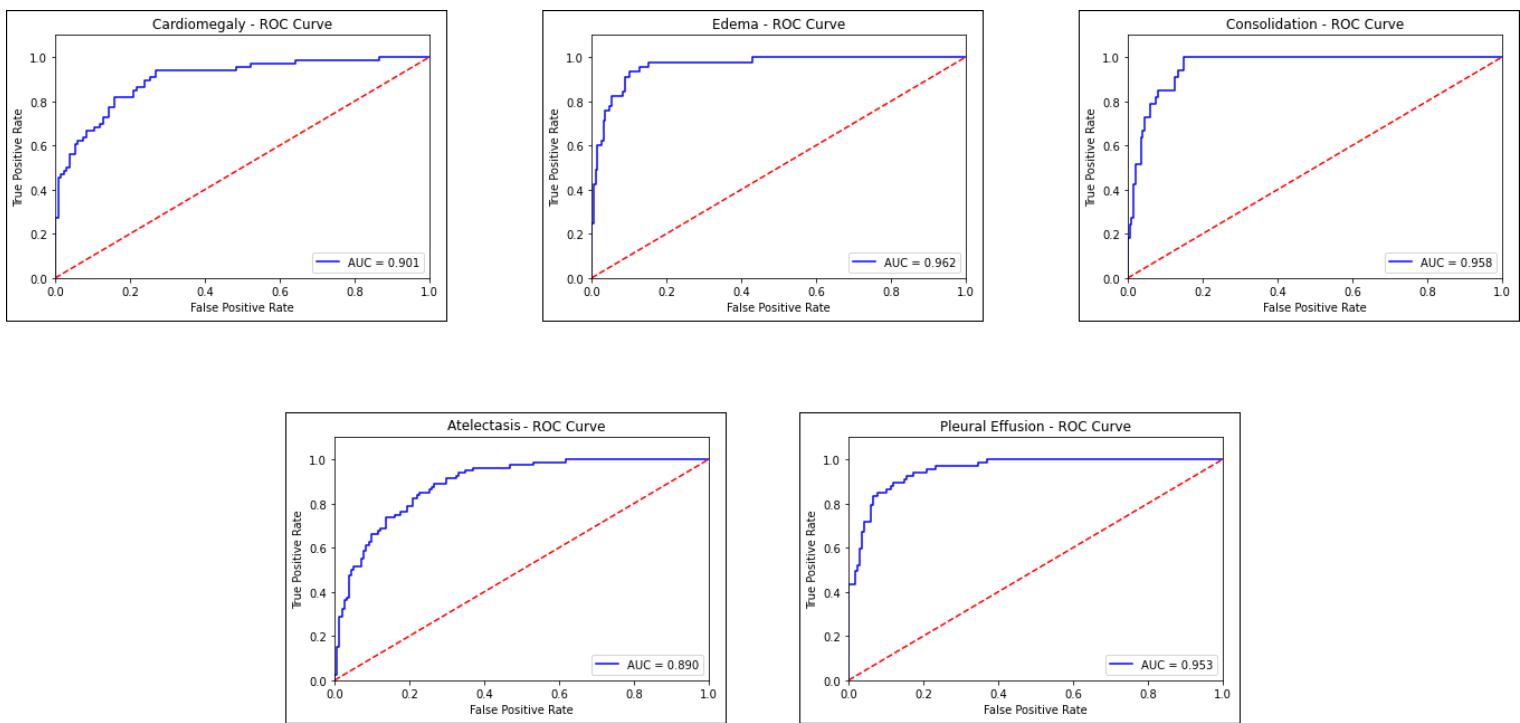


Figure (5-9) ROC Curves of ensemble model's performance on the validation

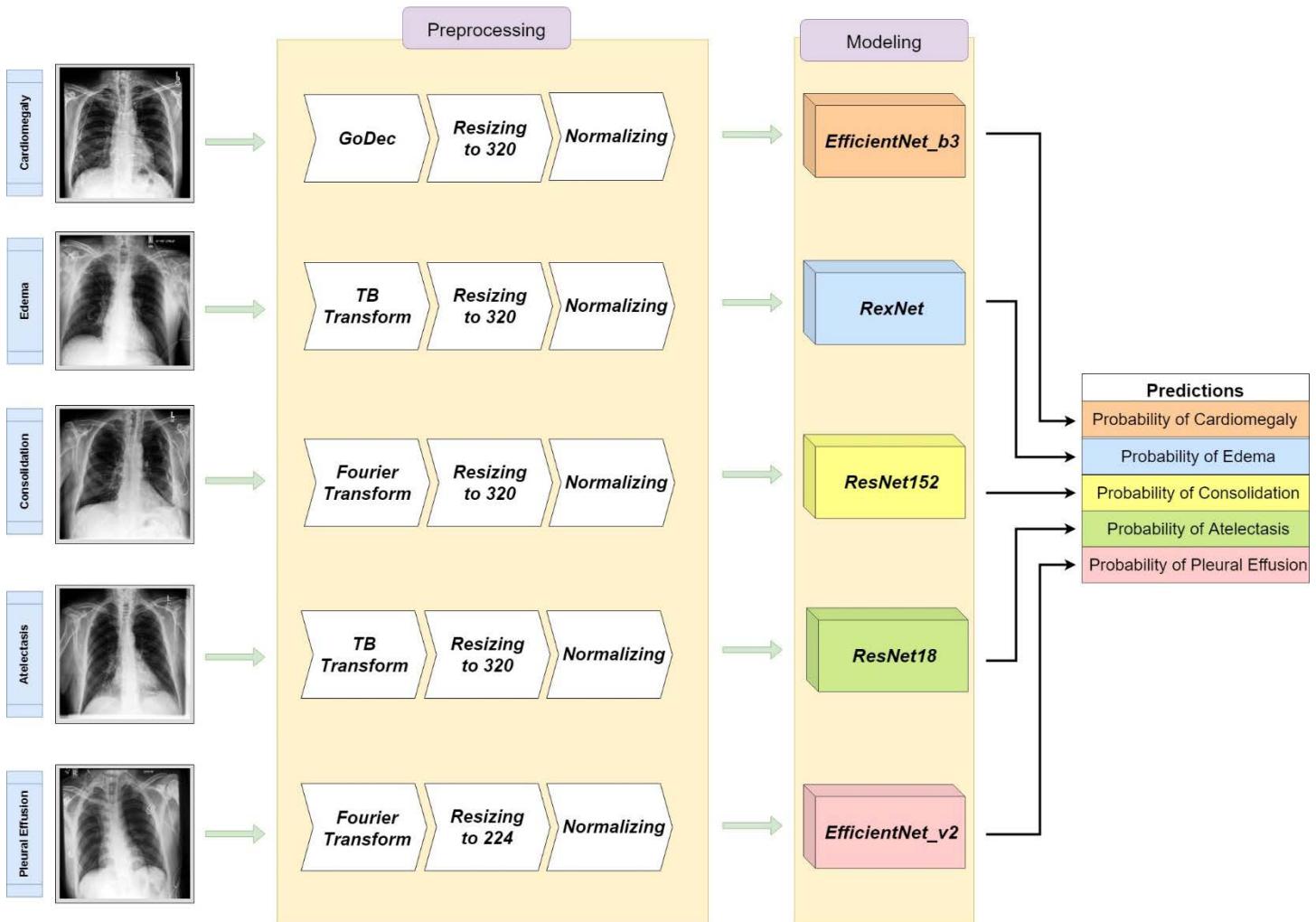


Figure (5-10) The proposed ensemble Architecture

Conclusion

We presented in this study a comprehensive approach for building a high-precision computer-aided diagnosis system for common thoracic diseases classification from CXRs. We investigated almost every aspect of the task including data cleaning, network design, training, and ensembling. In particular, we introduced a new preprocesing procedure by which deep features in the CXR are effectively exploited and integrated in training advanced CNNs. Extensive experiments demonstrated that the proposed preprocessing techniques can help in getting a high AUC score on CheXpert dataset which ,in turn, can be generalized to any chest x-ray dataset.

REFERENCES

REFERENCES

- [1] J. B. Soriano *et al.*, "Prevalence and attributable health burden of chronic respiratory diseases, 1990–2017: a systematic analysis for the Global Burden of Disease Study 2017," *Lancet Respir. Med.*, vol. 8, no. 6, pp. 585–596, Jun. 2020, doi: 10.1016/S2213-2600(20)30105-3.
 - [2] H. Wang *et al.*, "Global, regional, and national life expectancy, all-cause mortality, and cause-specific mortality for 249 causes of death, 1980–2015: a systematic analysis for the Global Burden of Disease Study 2015," *Lancet*, vol. 388, no. 10053, pp. 1459–1544, Oct. 2016, doi: 10.1016/S0140-6736(16)31012-1/ATTACHMENT/092770C9-1B7B-4CF3-9349-655862684147/MMC3.MP4.
 - [3] J. Irvin *et al.*, "CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 01, pp. 590–597, Jul. 2019, doi: 10.1609/AAAI.V33I01.3301590.
 - [4] S. Bhatia, Y. Sinha, and L. Goel, "Lung cancer detection: A deep learning approach," *Adv. Intell. Syst. Comput.*, vol. 817, pp. 699–705, 2019, doi: 10.1007/978-981-13-1595-4_55/FIGURES/3.
 - [5] O. Stephen, M. Sain, U. J. Maduh, and D. U. Jeong, "An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare," *J. Healthc. Eng.*, vol. 2019, 2019, doi: 10.1155/2019/4180949.
 - [6] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. Rajendra Acharya, "Automated detection of COVID-19 cases using deep neural networks with X-ray images," *Comput. Biol. Med.*, vol. 121, p. 103792, Jun. 2020, doi: 10.1016/J.COMPBIOMED.2020.103792.
 - [7] M. E. Karar, E. El-Din Hemdan, · Marwa, and A. Shouman, "Cascaded deep learning classifiers for computer-aided diagnosis of COVID-19 and pneumonia diseases in X-ray scans," *Complex Intell. Syst. 2020 71*, vol. 7, no. 1, pp. 235–247, Sep. 2020, doi: 10.1007/S40747-020-00199-4.
-

- [8] S. H. Yoo *et al.*, "Deep Learning-Based Decision-Tree Classifier for COVID-19 Diagnosis From Chest X-ray Imaging," *Front. Med.*, vol. 7, p. 427, Jul. 2020, doi: 10.3389/FMED.2020.00427/BIBTEX.
- [9] M. Hong, B. Rim, H. C. Lee, H. U. Jang, J. Oh, and S. Choi, "Multi-Class Classification of Lung Diseases Using CNN Models," *Appl. Sci. 2021, Vol. 11, Page 9289*, vol. 11, no. 19, p. 9289, Oct. 2021, doi: 10.3390/APP11199289.
- [10] B. Wang, W. Zhang, B. Wang, and W. Zhang, "MARnet: multi-scale adaptive residual neural network for chest X-ray images recognition of lung diseases," *Math. Biosci. Eng. 2022 1331*, vol. 19, no. 1, pp. 331–350, 2022, doi: 10.3934/MBE.2022017.
- [11] T. Ozcan and T. Ozcan, "A Deep Learning Framework for Coronavirus Disease (COVID-19) Detection in X-Ray Images," May 2020, doi: 10.21203/RS.3.RS-26500/V1.
- [12] M. Loey, S. El-Sappagh, and S. Mirjalili, "Bayesian-based optimized deep learning model to detect COVID-19 patients using chest X-ray image data," *Comput. Biol. Med.*, vol. 142, p. 105213, Mar. 2022, doi: 10.1016/J.COMPBIOMED.2022.105213.
- [13] J. Rubin Jonathan, D. Sanghavi, C. Zhao, K. Lee KathyLee, A. Qadir, and M. Xu-Wilson, "Large Scale Automated Reading of Frontal and Lateral Chest X-Rays using Dual Convolutional Neural Networks," Apr. 2018, doi: 10.48550/arxiv.1804.07839.
- [14] P. Kumar, M. Grewal, and M. M. Srivastava, "Boosted Cascaded Convnets for Multilabel Classification of Thoracic Diseases in Chest Radiographs," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10882 LNCS, pp. 546–552, 2018, doi: 10.1007/978-3-319-93000-8_62/TABLES/1.
- [15] S. Bharati, P. Podder, and M. R. H. Mondal, "Hybrid deep learning for detecting lung diseases from X-ray images," *Informatics Med. Unlocked*, vol. 20, p. 100391, Jan. 2020, doi: 10.1016/J.IMU.2020.100391.
- [16] Z. Ge, D. Mahapatra, S. Sedai, R. Garnavi, and R. Chakravorty, "Chest X-rays
-

Classification: A Multi-Label and Fine-Grained Problem," Jul. 2018, doi: 10.48550/arxiv.1807.07247.

- [17] H. H. Pham, T. T. Le, D. Q. Tran, D. T. Ngo, and H. Q. Nguyen, "Interpreting chest X-rays via CNNs that exploit hierarchical disease dependencies and uncertainty labels," *Neurocomputing*, vol. 437, pp. 186–194, May 2021, doi: 10.1016/J.NEUROCOM.2020.03.127.
- [18] Z. Yuan, Y. Yan, M. Sonka, and T. Yang, "Large-scale Robust Deep AUC Maximization: A New Surrogate Loss and Empirical Studies on Medical Image Classification," pp. 3020–3029, Dec. 2020, doi: 10.48550/arxiv.2012.03173.
- [19] T. Zhou and D. Tao, "GoDec: Randomized low-rank & sparse matrix decomposition in noisy case," *Proc. 28th Int. Conf. Mach. Learn. ICML 2011*, pp. 33–40, 2011.
- [20] T. Zhou and D. Tao, "Bilateral random projections," *IEEE Int. Symp. Inf. Theory - Proc.*, no. 1, pp. 1286–1290, 2012, doi: 10.1109/ISIT.2012.6283064.
- [21] S. Roweis, "EM algorithms for PCA and SPCA," *Adv. Neural Inf. Process. Syst.*, pp. 626–632, 1998.
- [22] Y. Peng, X. Wang, L. Lu, M. Bagheri, R. Summers, and Z. Lu, "NegBio: a high-performance tool for negation and uncertainty detection in radiology reports," *AMIA Summits Transl. Sci. Proc.*, vol. 2018, p. 188, 2018, Accessed: Jun. 05, 2022. [Online]. Available: /pmc/articles/PMC5961822/
- [23] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "dataset noisy labels ChestX-ray8 multidisease ChestX-ray14," pp. 2097–2106, [Online]. Available: <https://github.com/TRKuan/cxr8%0Ahttps://www.nih.gov/news-events/news-releases/nih-clinical-center-provides-one-largest-publicly-available-chest-x-ray-datasets-scientific-community%0Ahttps://cloud.google.com/healthcare-api/docs/resources/public-datasets/n>