
Pneumonia Detection from Chest X-ray using Transfer Learning

Author:

Sourav Dutta

Examination Roll No. : **M4CSE19020**

Registration No. : **140751 of 2017 - 2018**

Class Roll No. : **001710502012**

Under Guidance and Supervision of

Prof. Mahantapas Kundu

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Engineering

in the

Department of Computer Science and Engineering

Jadavpur University

May 29, 2019

Declaration of Originality and Compliance of Academic Ethics

I, Sourav Dutta, declare that this thesis titled, "**Pneumonia detection from Chest X-ray using Transfer Learning**" and the work presented in it are my own. I confirm that:

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

Name : Sourav Dutta

Examination Roll No. : M4CSE19020

Registration No. : 140751 of 2017 - 2018

Class Roll No. : 001710502012

Signature:

Date:

Faculty of Engineering and Technology

Jadavpur University

Certificate of Recommendation

This is to certify that the dissertation entitled “Penumonia detection from Chest X-ray using Transfer Learning” has been carried out by Sourav Dutta (University Registration No.: 140751 of 2017-2018, Examination Roll No.: M4CSE19020) under my guidance and supervision and be accepted in partial fulfillment of the requirement for the Degree of Masters in Computer Science and Engineering. The research results presented in the thesis have not been included in any other paper submitted for the award of any degree in any other University or Institute.

Supervisor:

Prof. Mahantapas Kundu

Dept. Of Computer Science and Engineering

Jadavpur University, Kolkata-700032

Countersigned:

Prof. Mahantapas Kundu (Head of Department)

Dept. Of Computer Science and Engineering

Jadavpur University, Kolkata-700032

Prof. Chiranjib Bhattacharjee (Dean)

Faculty of Engineering and Technology

Jadavpur University, Kolkata-700032

Faculty of Engineering and Technology
Jadavpur University

Certificate of Approval

This is to certify that the thesis entitled “Pneumonia detection from Chest x-ray using Transfer Learning” is a bona fide record of work carried out by Sourav Dutta (University Registration No.: 140751 of 2017-2018, Examination Roll No.: M4CSE19020) in partial fulfillment of the requirements for the award of the degree of Master in Computer Science and Engineering in the Department of Computer Science and Engineering, Jadavpur University during the period of June 2017 to May 2019. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, the opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

Examiner:

Signature:

Date:

Prof. Mahantapas Kundu
Dept. of Computer Science and Engineering
Jadavpur University, Kolkata-700032
Signature:

Date:

"To me, mathematics, computer science, and the arts are insanely related. They're all creative expressions."

Sebastian Thrun

Acknowledgements

I would like to express my deepest gratitude to my advisor, Prof. **Mahantapas Kundu**, Department of Computer Science and Engineering, Jadavpur University for his guidance, care, patience, motivation and for providing an excellent working environment. His ability to make any scientific discussions casual and to provide constructive, logical, experienced comments on any issue helped me in improving my project. Without his enthusiasm, encouragement, and continuous support this thesis would hardly have been completed.

I would like to thank Prof. **Mita Nasipuri**, Department of Computer Science and Engineering, Jadavpur University for her patience, guidance, suggestions and moral support in times of need.

I am indebted to **Prof. Subhadip Basu**, Department of Computer Science and Engineering, Jadavpur University for his kindness in providing me the facilities to continue my work uninterrupted and his moral support.

I would also like to thank my seniors **Mr. Supratim Ghosh, Mr. Sandeep Gupta** for providing valuable help at times of need.

I am grateful to all my friends who are always by my side in times of need, for their faith in me and for providing constant moral support.

I extend my whole hearted thanks to my family for having faith in me and supporting me through this endeavor because none of these would have been possible without their love and support.

This thesis would not have been completed without the inspiration and support of a number of wonderful individuals. My thanks and appreciation to all of them for being part of this journey and making this thesis possible.

Name : Sourav Dutta
Examination Roll No. : M4CSE19020
Registration No. : 140751 of 2017 - 2018
Class Roll No. : 001710502012

Contents

List of Tables

List of Figures

1. Introduction

The work presented in this thesis mainly concentrates on detection of Pneumonias from X-ray images through application of Deep Convolutional Neural Networks (CNNs). Considering complexity of the problem related to detection of Pneumonias, deep CNNs with automatic feature learning capabilities are employed here.

1.1. Pneumonia Detection/Diagnosis Techniques

Doctors generally diagnose a pneumonia patient by asking their medical history and doing a physical exam, including listening to their lungs with a stethoscope to check for abnormal bubbling or crackling sounds that suggest pneumonia.

If pneumonia is suspected, the doctor may recommend the following tests:

- **Blood tests.** Blood tests are used to confirm an infection and to try to identify the type of organism causing the infection. However, precise identification isn't always possible.
- **Chest X-ray.** This helps your doctor diagnose pneumonia and determine the extent and location of the infection. However, it can't tell your doctor what kind of germ is causing the pneumonia.
- **Pulse oximetry.** This measures the oxygen level in your blood. Pneumonia can prevent your lungs from moving enough oxygen into your bloodstream.
- **Sputum test.** A sample of fluid from your lungs (sputum) is taken after a deep cough and analyzed to help pinpoint the cause of the infection.

1.2. Chest X-Ray

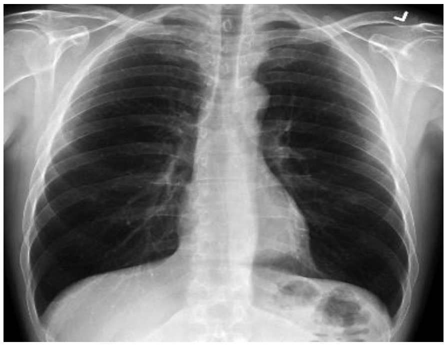


Figure 1: Chest X-ray image

X-rays are a type of radiation called electromagnetic waves. **X-rays** make up **X-radiation**, a form of electromagnetic radiation. Most X-rays have a wavelength ranging from 0.01 to 10 nanometers, corresponding to frequencies in the range 30 petahertz to 30 exahertz (3×10^{16} Hz to 3×10^{19} Hz) and energies in the range 100 eV to 100 keV. X-ray wavelengths are shorter than those of UV rays and typically longer than those of gamma rays. In many languages, X-radiation is referred to with terms meaning **Röntgen radiation**, after the German scientist Wilhelm Röntgen who discovered these on November 8, 1895. X-ray imaging creates pictures of the inside of human body. The images show the parts of human body in different shades of black and white. This is because different tissues absorb different amounts of radiation. Calcium in bones absorbs x-rays the most, so bones look white. Fat and other soft tissues absorb less, and look gray. Air absorbs the least, so lungs look black. X-Ray has been being used in medical imaging since the time it was discovered. Some examples of it’s application in medical domain are : Computed Tomography, Mammography, Angiography, Fluoroscopy etc.

Chest X-Ray is X-Ray images of chest region. It is used to evaluate the lungs, heart and chest wall and may be used to help diagnose shortness of breath, persistent cough, fever, chest pain or injury. It also may be used to help diagnose and monitor treatment for a variety of lung conditions such as pneumonia, emphysema and cancer. Because chest x-ray is fast and easy, it is particularly useful in emergency diagnosis and treatment.

1.3. Automated pneumonia detection

Automated Pneumonia detection can be done from X-rays, cough sounds, ultrasound videos/images of chest areas. Out of these, automated pneumonia detection from x-rays is the most popular practice[14].

Detecting pneumonia with chest X-rays is a challenging task that relies on the availability of expert radiologists. Moreover, detecting pneumonia with chest x-rays sometimes becomes difficult for the radiologists for the following reasons.

Appearance of pneumonial patches in X-ray images is not always clear and precise, may also be indicative of other diseases, and can mimic many other benign abnormalities. These may cause considerable variations in diagnosis, performed by different radiologists. These are the challenges related to automated pneumonia detection from chest x-rays. There have been many ventures in Medical Image Analysis, such as Lesion Detection and Segmentation [35-39], diseases classification [40-43], registration[44], regression[45] and so on. Wang et al [13] used four hugely successful CNN architectures, AlexNet[46], VGGNet[47], GoogleNet[48], ResNet[49] to detect pathologies in CXR images, the lesion areas are also detected in weakly supervised manner. Yao et al. [26] explore the correlation among the 14 pathologic labels with global images in ChestX-ray14 [9]. Using a variant of DenseNet [50] as an image encoder, they adopt the Long-short Term Memory Networks (LSTM) [51] to capture the dependencies. Kumar et al. [27] investigate that which loss function is more suitable for training CNNs from scratch and present a boosted cascaded CNN for global image classification. The recent effective method consists in CheXNet [9]. It fine-tunes a 121-layer DenseNet on the global chest X-ray images, which has a modified last fully-connected layer.

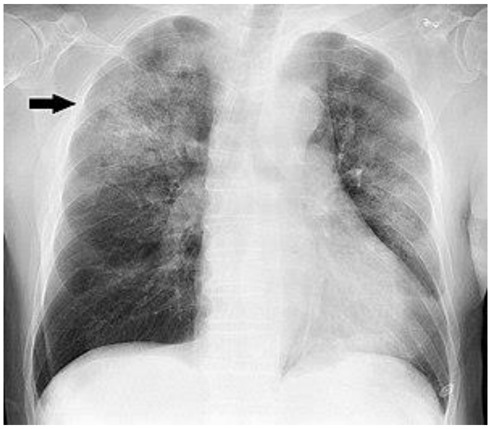


Figure 2: Pneumonia affected Lungs

1.4. Motivation

In spite of the complexity of the problem relating to automated pneumonia detection, there has been a tremendous need for an automated system capable of detecting pneumonia and other lung diseases from chest x-rays. According to an estimate published by the World Health Organization (Mollura et al., 2010), two thirds of the global population lacks access to radiology diagnostics. Even when imaging equipment is available, there is a shortage of experts who can interpret X-rays, leading to increased mortality from treatable diseases (Kesselman et al., 2016). All these has motivated the past work in automated pneumonia detection.

After a considerable number of attempts[references] have been made by the researchers, performances of the systems developed for automated pneumonia detection are not up to the mark. It is most likely due to creeping of uncertainty with possibilities of various chest diseases occurring with the same penumonial patches present in chest x-rays. In addition, impreciseness of chest x-rays also makes the problem complex. So till date, there is enough scope for investigating for better methods employing powerful machine learning techniques, especially deep learning methodologies, for automated pneumonia detection from chest x-rays.

2. Pneumonia

2.1. What is Pneumonia?

Pneumonia is an infection in one or both lungs. It can be caused by bacteria, viruses, or fungi. Bacterial pneumonia is the most common type in adults. Pneumonia causes inflammation in the air sacs in your lungs, which are called alveoli. The alveoli fill with fluid or pus, making it difficult to breathe. Pneumonia affects approximately 450 million people globally (7% of the population) and results in about 4 million deaths per year. Pneumonia was regarded by William Osler in the 19th century as "the captain of the men of death". With the introduction of antibiotics and vaccines in the 20th century, survival rate in Pneumonia is improved. Nevertheless, in developing countries, and among the very old, the very young, and the chronically ill, pneumonia is till a leading cause of death. Pneumonia often shortens suffering among those already close to death and has thus been called "the old man's friend".

2.2. Types and Causes of Pneumonia

Pneumonia is due to infections caused primarily by bacteria or viruses and less commonly by fungi and parasites. Although there are over 100 strains of infectious agents identified, only a few are responsible for the majority of the cases. Mixed infections with both viruses and bacteria may occur in roughly 45% of infections in children and 15% of infections in adults. A causative agent may not be isolated in approximately half of cases despite careful testing. Factors that predispose to pneumonia include smoking, immunodeficiency, alcoholism, chronic obstructive pulmonary disease, asthma, chronic kidney disease, liver disease, and old age. Additional risks in children includes not being breastfeed, exposure to cigarettes or air pollution, malnutrition, and poverty. The use of acid-suppressing medications – such as proton-pump inhibitors or H2 blockers – is associated with an increased risk of pneumonia.

2.2.1. Types by germ

1. ***Bacterial pneumonia:*** The most common cause of bacterial pneumonia is *Streptococcus pneumoniae*. *Chlamydophila pneumonia* and *Legionella pneumophila* can also cause bacterial pneumonia.

2. ***Viral pneumonia:*** Respiratory viruses are often the cause of pneumonia, especially in young children and older people. Viral pneumonia is usually not serious and lasts for a shorter time than bacterial pneumonia.

3. ***Mycoplasma pneumonia:*** Mycoplasma organisms are not viruses or bacteria, but they have traits common to both. Mycoplasmas generally cause mild cases of pneumonia, most often in older children and young adults.

4. ***Fungal pneumonia:*** Fungi from soil or bird droppings can cause pneumonia in people who inhale large amounts of the organisms. They can also cause pneumonia in people with chronic diseases or weakened immune systems.

One kind of fungal pneumonia is called *Pneumocystis jirovecii* pneumonia (PCP). This condition generally affects people with weakened immune systems, such as those with AIDS. In fact, PCP can be one of the first signs of infection with AIDS.

2.2.2. Types by location

Pneumonia is also classified according to where it was acquired.

1. ***Hospital-acquired pneumonia (HAP):*** This type of bacterial pneumonia is acquired during a hospital stay. It can be more serious than other types, because the bacteria involved may be more resistant to antibiotics.
2. ***Community-acquired pneumonia (CAP):*** This refers to pneumonia that is acquired outside of a medical or institutional setting.

2.2.3. Types by how they are acquired

Pneumonia can also be classified according to how it is acquired.

1. ***Aspiration pneumonia:*** This type of pneumonia occurs when you inhale bacteria into your lungs from food, drink, or saliva. This type is more likely to occur if you have a swallowing problem or if you become too sedate from the use of medications, alcohol, or some types of illicit drugs.

2. ***Ventilator-associated pneumonia (VAP):*** When people who are using a ventilator get pneumonia, it’s called VAP.

2.3. Symptoms

Pneumonia symptoms can be mild to life-threatening. The most common symptoms of pneumonia can include:

- coughing that may produce phlegm (mucus)
- fever, sweating, and chills
- shortness of breath
- chest pain

Other symptoms can vary according to the cause and severity of the infection, as well as the age and general health of the individual.

2.3.1. Symptoms by cause

- Viral pneumonia may start with flu-like symptoms, such as wheezing. A high fever may occur after 12–36 hours.
- Bacterial pneumonia may cause a fever as high as 105°F along with profuse sweating, bluish lips and nails, and confusion.

2.3.2. Symptoms by age

- Children under 5 years of age may have fast breathing.
- Infants may vomit, lack energy, or have trouble drinking or eating.
- Older people may have a lower-than-normal body temperature.

3. Past Work

In the field of Automatic Pneumonia Detection, there has been a number of attempts[references] in the past. Earlier works are centered on analysis of cough sounds, Chest X-rays or Ultrasound videos/images of chest areas and Chest tomography. Out of these, chest x-rays are the most popular ones especially for pneumonia detection, although Lung-Sonography is considered harmless because of being Non-radioactive. It is worth mentioning that methods involving neural networks and other computationally expensive techniques could not be effectively attempted until massive gain in computational power was achieved in the last decade.

3.1. Ultra-Sound Video Analysis

In early days, the work on automated pneumonia detection was mostly carried out with Ultra sound videos/images of chest areas [1-4] with handcrafted features. In [4], image processing techniques were used to segment pleural line and lung consolidation from ultra sound video frames and based on some observations and criteria, a patient is categorized in one of 3 classes - Pneumonia patient, Normal patient and Probable Pneumonia patient. In [5], features like average intensity of pixel values from of a Region of Interest in Ultra sound image of chest are calculated and fed to a Neural network for classification. In [6], Chest radiographs are used to extract geometrical features such as area, perimeter, equivalent diameter and irregularity index, and statistical features like mean, standard deviation, and entropy by intensity and continuity-based image processing techniques. Then these features are fed to an Artificial Neural Network (ANN) to classify the subject as Pneumonia patient or Normal patient.

3.2. Deep Learning based approaches applied on Chest X-ray

Recent surveys [19], [20], [21], [22] have demonstrated that deep learning technologies have been extensively applied to the field of chest X-ray image annotation [33], classification [23], [12], [13], [9], and detection (localization) [24], [25]. Islam et al. [12] explore different Convolutional Neural Network (CNN) architectures and find that a single CNN does not perform well across all abnormalities. Therefore, they leverage model ensemble to improve the classification accuracy, at the cost of increased training and testing time. The most significant work in this area has been done by Pranav Rajpurkar in [9]. They used a 121-layer Dense network for classification of Chest X-ray images into 14 possible pneumonial classes including the normal one. So far, this is the best result obtained on Chest X-ray diagnosis where a Deep Network has performed better than practicing radiologists. Besides that, various CNN based works have also emerged. Yao et al. [26] and Kumar et al. [27] classify the chest X-ray images by investigating the potential dependencies among the labels from the aspect of multi-label problems, in which an X-ray image may be labelled with more than one abnormality class. Rajpurkar et al. [8] has employed a CNN to address the multi-label classification problem. Bejnamin et al.[28] showed attempted Pneumonia detection from NIH Chest X-ray dataset by performing K-means, Logistic Regression as well as Deep Neural Network(Desenet-121). They used each pixel of the images as individual dimensions for K-means and Logistic Regression.

3.3. Attention-based approaches

The significant part of an X-ray image, responsible for demonstration of pathology is often very small and the shape, size and position of the part change for each kind of pathology. That's why it is not always good to let a neural network decide on the pathology based on observation of the entire image, rather the decision to be taken on extracting the significant part from an X-ray image. Multiple ventures have already put this idea to practice.

Ypsilantis et al. [29] explore where to look in chest X-rays for pneumonia detection with recurrent attention model (RAM) [30]. The RAM learns to sample the entire X-ray image sequentially and focuses on informative areas. Only one disease, viz., enlarged heart, is considered in their work. Pesce et al. [31] explore a soft attention mechanism from the saliency map of CNN features to locate lung nodule position in radiographies. And a localization loss is calculated by comparing the predicted position with the annotated position for propagate the loss backward. In [34], Resnet-152 (a 152-layer Convolutional Neural Network with skip connections) was used along with attention branch to help the network concentrate on specific areas of the image for determining the pathology. The attention branch uses Grad-CAM (Gradient based Class Activation Mapping) which learns to create heatmap[ref] per image using the gradient coming from the last layer during training. In [32], The feature maps from the last convolutional layer is examined to detect significant parts of an image, containing pathology. Then that part was cropped out and another model was trained on the several such cropped images. Finally, an ensemble of both of the networks is used for final classification.

4. Neural Network and Transfer Learning

4.1. Artificial Neural Network

Just like our brain, where billions of neurons are interconnected with each other and by working parallelly, they helps us in decision making and cognitive thinking throughout our day to day life, a neural network also has many neurons/nodes interconnected with each other and unitedly they specialize on solving one or multiple specific problems. A typical neural network has anything from a few dozens to hundreds, thousands, or even millions of artificial neurons which are arranged in a series of layers. Each of the layer connects to the layers on either side. Some of them which are at one end of the network and are designed to receive various forms of information from the outside world are called **Input Units**. The network will attempt to learn about, recognize, or otherwise process the inputs received by the input Units. Other units which sit on the opposite side of the network and signal how it responds to the information it's learned; those are known as **output units**. In between the input units and output units are one or more layers of **hidden units**, which, together, form the majority of the artificial brain and helps in the computational process of decision making. Most neural networks are **fully connected**, which means each hidden unit in a particular layer is connected to every unit in the layers on either side. The connections between one unit and another are represented by a number called a **weight**, which can be either positive (if one unit excites another) or negative (if one unit suppresses or inhibits another). The higher the weight, the more influence one unit has on another. (This corresponds to the way actual brain cells trigger one another across tiny gaps called synapses.). A neuron receives weighted outputs of neurons, sitting on the previous layer and how it responds to the input is dictated by the activation function it is using. The difference between 'how it is responding right at this moment' and 'how it should respond' decreases with time as the network is trained.

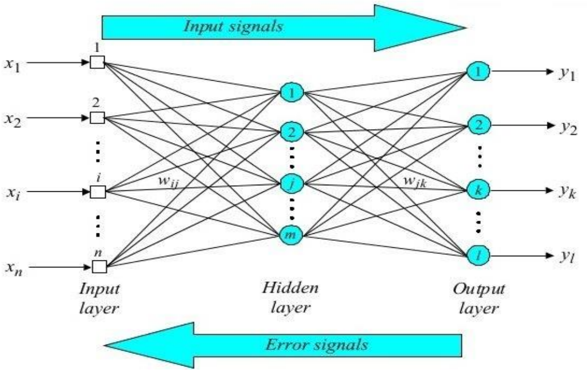


Figure 3(a) : Artificial Neural Network structure[54]

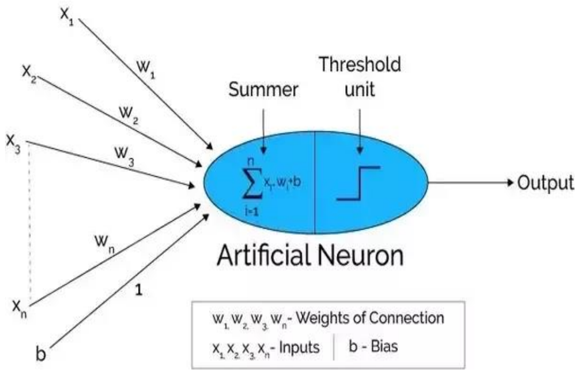

X_1
X_2
X_3
X_n
b
W_1
W_2
W_3
W_n
1
Summer
Threshold unit
$\sum_{i=1}^{n} x_i w_i + b$
Output
Artificial Neuron
W_1, W_2, W_3, W_n - Weights of Connection
X_1, X_2, X_3, X_n - Inputs
b - Bias

Figure 3(b): Artificial Neurone[54]

4.1.1. Architecture

Depending on how neurons are physically connected in an ANN, different architectures are proposed. Architecturally ANN's can be grouped into two major categories viz., feed-forward networks as shown in Figure 3.1a and recurrent (or feedback) networks as shown in Figure 3.1b. Feed-forward networks have no loops whereas recurrent (or feedback) networks has loops to enable feedbacks.

All the neurons of each layer except output layer in a neural network have activation functions associated with it. This is because the last output layer is usually taken to represent the class scores (e.g. in classification), which are arbitrary real-valued numbers, or some kind of real-valued target (e.g. in regression).

4.1.2. Commonly used activation functions

In general, a Neural Network performs a sequence of linear mappings with interwoven non-linearities. Every activation function (or non-linearity) takes a single number and performs a certain fixed mathematical operation on it. There are several activation functions viz., Sigmoid (Figure3.2a, Equation 3.1), Tanh (Figure3.2b, Eqaution3.2), ReLU (Figure 3.2c, Eqaution3.3), LeakyReLU.

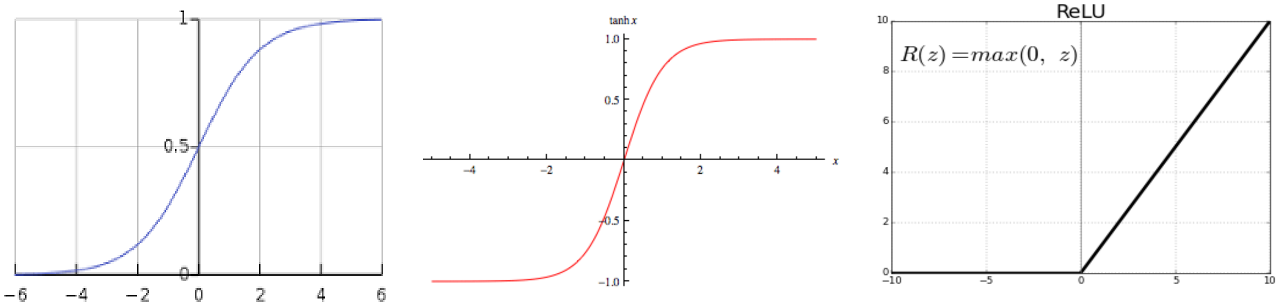
Chapter 3. Neural Networks and Transfer Learning 19

(A) Sigmoid non-linearity squashes real numbers to range between [0,1]

(B) The tanh non-linearity squashes real numbers to range between [-1,1]

(C) Rectified Linear Unit (ReLU), which is zero when x < 0 and then linear with slope 1 when x > 0

Definitions of some activation functions are shown below,



(a) Sigmoid non-linearity

(b) TanH non-linearity

(c) ReLU non-linearity

Sigmoid, $f(x) = \frac{1}{1+e^{-x}}$ Tanh, $f(x) = \tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$ ReLU, $f(x) = \max(0,x)$

4.1.3. How Neural Networks learn

Every living creature in this world learns from experiences. Gathering experience is not a single step process. First, an action is taken, an act is performed by the agent on the environment. In return the agent receives either a reward or penalty depending on whether the resultant state, after performing the action took the agent closer or farther from the destination state. Depending on the feedback (a reward or a penalty), the agent adjusts it’s mechanism or parameters in such a way that next time it can perform better when the same situation appears. By repeating the same process (by facing numerous situations/seeing many examples and improving itself a little bit every time, the agent finally can gather enough knowledge such that it knows how to act in a situation, it has never experienced. It’s called Generalization. Every neuron in the network has some parameters which contributes to the final decision of the network. These neurons receive weighted input from previous layer-outputs and the neuron fires when the weighted sum of the inputs becomes greater than a certain threshold.

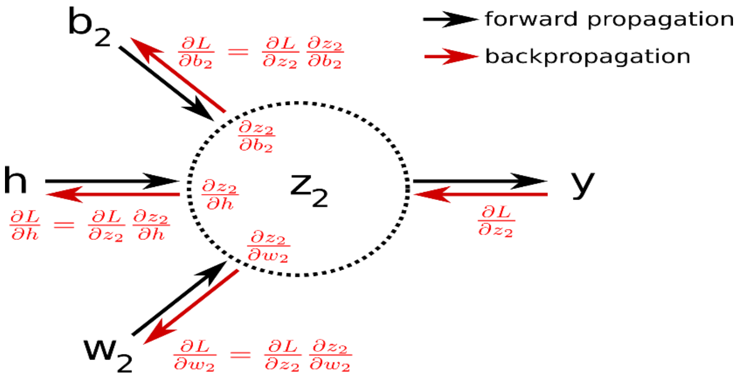
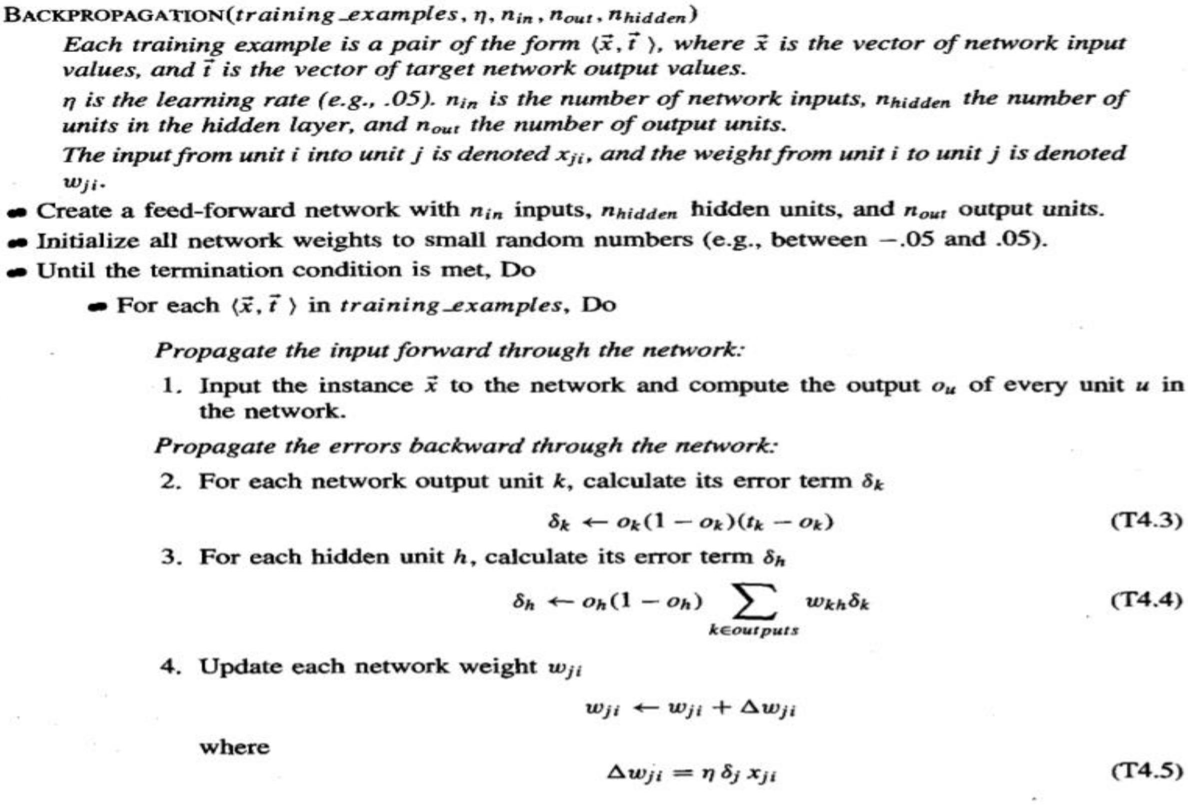

b_2
$\frac{\partial L}{\partial b_2} = \frac{\partial L}{\partial z_2} \frac{\partial z_2}{\partial b_2}$
forward propagation
backpropagation
$\frac{\partial z_2}{\partial b_2}$
h
$\frac{\partial z_2}{\partial h}$
z_2
y
$\frac{\partial L}{\partial h} = \frac{\partial L}{\partial z_2} \frac{\partial z_2}{\partial h}$
$\frac{\partial L}{\partial z_2}$
$\frac{\partial z_2}{\partial w_2}$
w_2
$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial z_2} \frac{\partial z_2}{\partial w_2}$

Figure 4 : Backpropagation technique

For a neural network to learn, there has to be an element of feedback involved—just as children learn by being told what they're doing right or wrong. In fact, we all use feedback, all the time. Think back to when you first learned to play a game like ten-pin bowling. As you picked up the heavy ball and rolled it down the alley, your brain watched how quickly the ball moved and the line it followed, and noted how close you came to knocking down the skittles. Next time it was your turn, you remembered what you'd done wrong before, modified your movements accordingly, and hopefully threw the ball a bit better. So you used feedback to compare the outcome you wanted with what actually happened, figured out the difference between the two, and used that to change what you did next time ("I need to throw it harder," "I need to roll slightly more to the left," "I need to let go later," and so on). The bigger the difference between the intended and actual outcome, the more radically you would have altered your moves.

Neural networks learn things in exactly the same way, typically by a feedback process called **backpropagation** (sometimes abbreviated as "backprop"). This involves comparing the output a network produces with the output it was meant to produce, and using the *difference* between them to modify the weights of the connections between the units in the network, working from the output units through the hidden units to the input units—going backward, in other words. In time, backpropagation causes the network to learn, reducing the difference between actual and intended output to the point where the two exactly coincide, so the network figures things out exactly as it should.

The Backpropagation Algorithm:



```
BACKPROPAGATION($training\_examples, \eta, n_{in}, n_{out}, n_{hidden}$)
    Each training example is a pair of the form $\langle \vec{x}, \vec{t} \rangle$, where $\vec{x}$ is the vector of network input
    values, and $\vec{t}$ is the vector of target network output values.
    $\eta$ is the learning rate (e.g., .05). $n_{in}$ is the number of network inputs, $n_{hidden}$ the number of
    units in the hidden layer, and $n_{out}$ the number of output units.
    The input from unit i into unit j is denoted $x_{ji}$, and the weight from unit i to unit j is denoted
    $w_{ji}$.
• Create a feed-forward network with $n_{in}$ inputs, $n_{hidden}$ hidden units, and $n_{out}$ output units.
• Initialize all network weights to small random numbers (e.g., between −.05 and .05).
• Until the termination condition is met, Do
    • For each $\langle \vec{x}, \vec{t} \rangle$ in $training\_examples$, Do
        Propagate the input forward through the network:
        1. Input the instance $\vec{x}$ to the network and compute the output $o_u$ of every unit $u$ in
           the network.
        Propagate the errors backward through the network:
        2. For each network output unit $k$, calculate its error term $\delta_k$
                $\delta_k \leftarrow o_k(1 - o_k)(t_k - o_k)$                          (T4.3)
        3. For each hidden unit $h$, calculate its error term $\delta_h$
                $\delta_h \leftarrow o_h(1 - o_h) \sum_{k \in outputs} w_{kh}\delta_k$        (T4.4)
        4. Update each network weight $w_{ji}$
                $w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$
           where
                $\Delta w_{ji} = \eta \, \delta_j \, x_{ji}$                            (T4.5)
```

Figure 5: Backpropagation Algorithm [58]

4.2. Convolutional Neural Network

The conventional-neural-networks is a type of neural-networks which has at least one convolution layer. They are mainly used for capturing local information (eg. neighbor pixels in

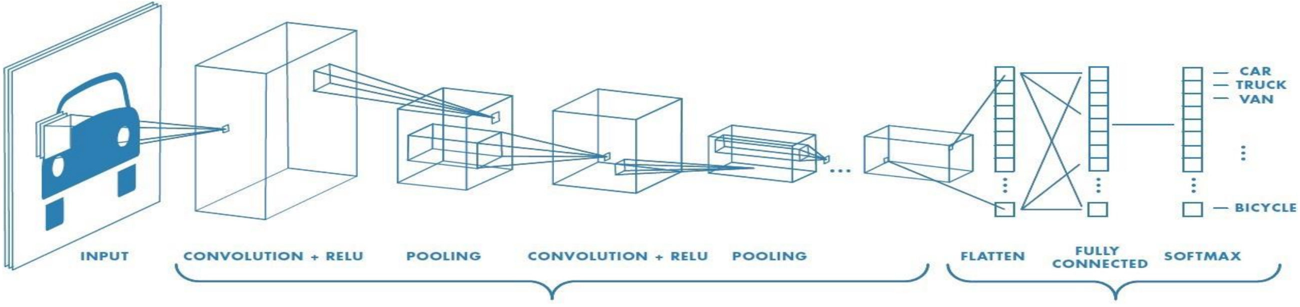


Figure 6 : Convolutional Neural Network

an image or surrounding words in a text) as well as reducing the complexity of the model (faster training, needs fewer samples, reduces the chance of overfitting).

4.2.1. Convolution

Convolution layer is a layer which multiple neurons shares the same weights. It is required especially in those cases where the input is an image. Because, need to look for a specific feature in multiple positions in the image, it might be as simple as a line, a round shape, an arc, color blob or some high level features, like an eye, an ear, a mat etc. which can be formed by arranging the combination of low level features in a specific way. A Convolution layer consists of multiple filters/kernels which is usually a square shaped set of numbers. This kernel convolves over the input feature map / image and generates new higher-level feature maps by multiplying the values in the input feature maps with the corresponding feature map weights. Kernels in the early layers in the network learns to detect pretty low-level features. As the depth of the network increases kernels learns high level features using previous layer's features. Height and Width of learned feature maps generally decrease with depth of the network whereas the depth increases. Last few layers of the network is generally fully connected layers, which are responsible for decision making. Dense layer(s) at the end of a CNN performs computations necessary to classify the input image to two or more classes.

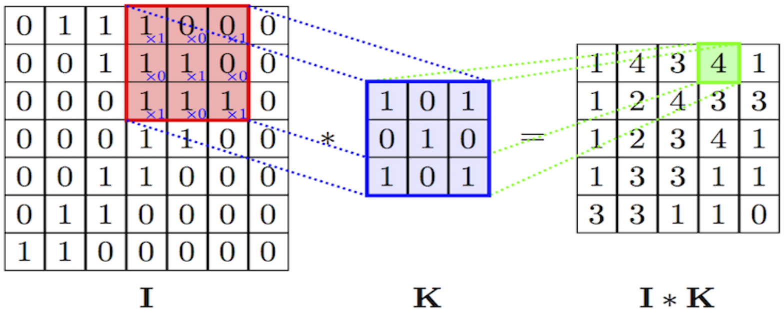


Figure 7 : Convolution in CNN

Here are the advantages of using Convolutional Neural Network over Artificial Neural Network:

1. The usage of CNNs are motivated by the fact that they can capture / are able to learn relevant features from an image /video at different levels similar to a human brain. Conventional neural networks cannot do this.

2. Another main feature of CNNs is **weight sharing**. Lets take an example to explain this. Say we have a one layered CNN with 10 filters of size 5x5. Now you can simply calculate parameters of such a CNN, it would be 5*5*10

3. weights and 10 biases i.e. **5* 5*10 + 10 = 260 parameters**. Now lets take a simple one layered NN with 250 neurons, here the number of weight parameters depending on the size of images is ‘250 x K’ where size of the image is P X M and K = (P *M). Additionally, you need ‘M’ biases. For the MNIST data as input to such a NN we will have **(250*784+1 = 19601) parameters.** Clearly, CNN is more efficient in terms of memory and complexity. Imagine NNs and CNNs with billions of neurons, then CNNs would be less complex and saves memory compared to the NN.

3. CNNs are very good **feature extractors** for completely new task / problem**.** This means that you can extract useful attributes from an already trained CNN with its trained weights by feeding your data and tune the parameters a bit for the specific task. Eg : Add a classifier after the last layer with labels specific to the task. This is also called **pre-training** and CNNs are very efficient in such tasks compared to NNs. Another advantage of this pre-training is we avoid training of CNN and save memory, time. The only thing you have to train is the classifier at the end for your labels.

4.3. Transfer Learning

“**I think Transfer Learning is the key to general intelligence. And I think the key to doing transfer learning will be the acquisition of conceptual knowledge that is abstracted away from perceptual details of where you learned it from**.” – Demis Hassahis.

4.3.1. What is Transfer Learning

Transfer Learning[53] is reuse of knowledge learned while solving a problem. It has many advantages. First, it requires quite a small amount of time to train Deep neural networks. Secondly, we don’t need vast amount of data to train a DL model as it normally requires while training from scratch.

In Transfer learning the knowledge of a trained Neural Network Model is applied in solving a different but problem of same kind. It’s basically an effort to increase generalization utilizing the knowledge gained in one task. It is very useful in those cases where loads of data is not available to train. Irrespective the kind of training images, every CNN learns some low-level features like edges in the earlier layers, shapes in the middle layers and task-specific high-level features at the later layers. So, if we already have a trained-model we can use its low and mid-level features in our current task. The task-specific features are only learned using the training images. This process is called Fine-tuning. We don’t change the weights in the early layers much, rather we try to learn task-specific features using those low-level features. We try to transfer as much knowledge as possible, and by doing that we can save hours or even days.

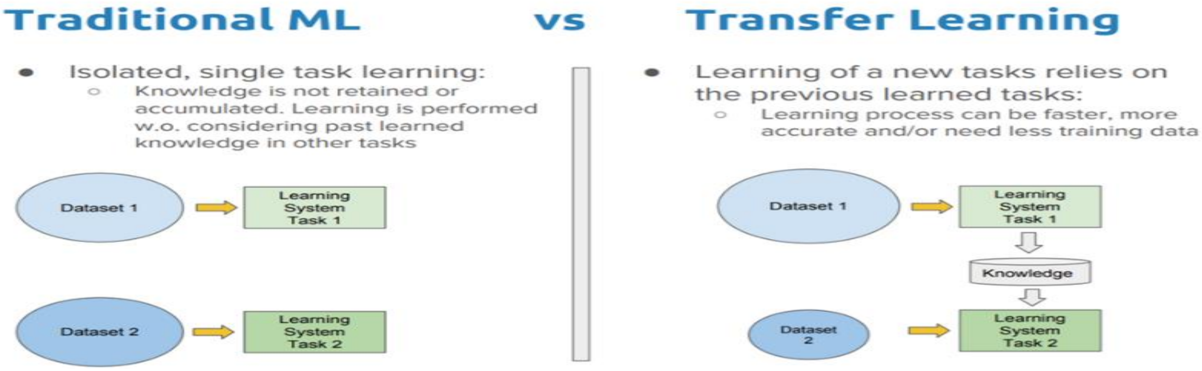

Traditional ML
vs
Transfer Learning
Isolated, single task learning:
Knowledge is not retained or accumulated. Learning is performed w.o. considering past learned knowledge in other tasks
Learning of a new tasks relies on the previous learned tasks:
Learning process can be faster, more accurate and/or need less training data
Dataset 1
Learning System Task 1
Dataset 2
Learning System Task 2
Dataset 1
Learning System Task 1
Knowledge
Dataset 2
Learning System Task 2

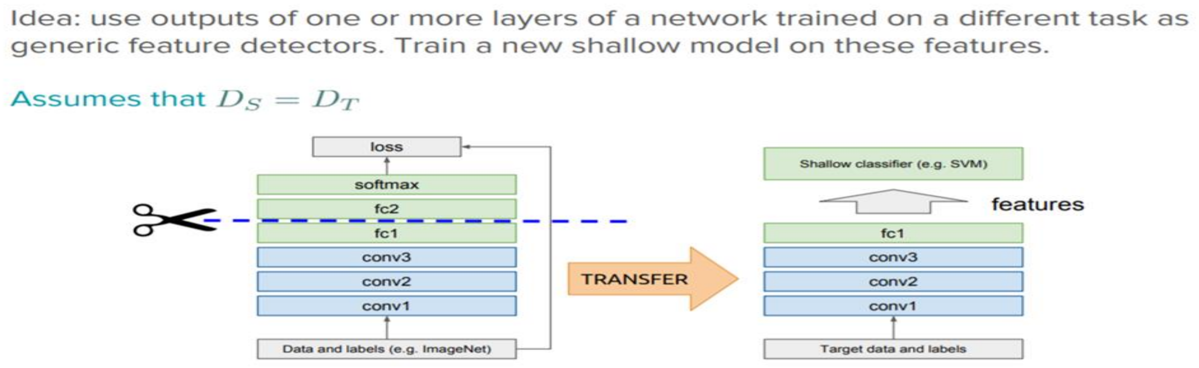
Figure 8: Tradiotional Machine Learning VS Transfer Learning

4.3.2. Use of Transfer Learning

The training time and the amount of data required for such deep learning systems are much more than that of traditional ML systems. There are various deep learning networks with state-of-the-art performance (sometimes as good or even better than human performance) that have been developed and tested across domains such as computer vision and natural language processing (NLP). In most cases, teams/people share the details of these networks for others to use. These pre-trained networks/models form the basis of transfer learning in the context of deep learning. The two most popular strategies for deep transfer learning are:

• Off-the-shelf Pre-trained Models as Feature Extractors

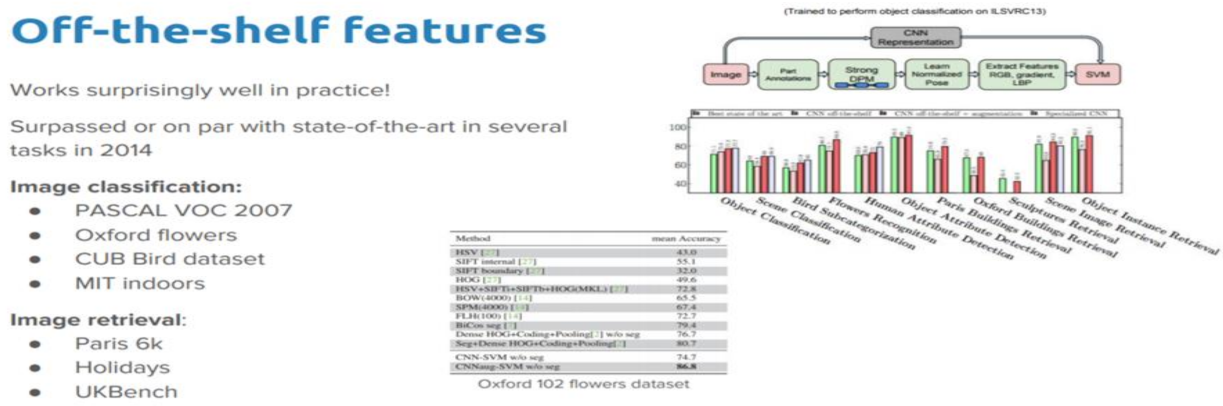
Deep learning systems and models are layered architectures that learn different features at different layers (hierarchical representations of layered features). These layers are then finally connected to a last layer (usually a fully connected layer, in the case of supervised learning) to get the final output. This layered architecture allows us to utilize a pre-trained network (such as Inception V3 or VGG) without its final layer as a fixed feature extractor for other tasks. The key idea here is to just leverage the pre-trained model's weighted layers to extract features but not to update the weights of the model's layers during training with new data for the new task. The performance of these off-the-shelf pre-trained models are good enough to beat even task focused specialized Deep Learning Models. The figure above shows it.



Idea: use outputs of one or more layers of a network trained on a different task as generic feature detectors. Train a new shallow model on these features.

Assumes that $D_S = D_T$

Figure 9: Pre-trained Models as Feature Extractor [57]



Off-the-shelf features

Works surprisingly well in practice!

Surpassed or on par with state-of-the-art in several tasks in 2014

Image classification:

- PASCAL VOC 2007
- Oxford flowers
- CUB Bird dataset
- MIT indoors

Image retrieval:

- Paris 6k
- Holidays
- UKBench

Method	mean Accuracy
HSV [27]	43.0
SIFT internal [27]	55.1
SIFT boundary [27]	32.0
HOG [27]	49.6
HSV+SIFTi+SIFTb+HOG(MKL) [27]	72.8
BOW(4000) [14]	65.5
SPM(4000) [14]	67.4
FLH(100) [14]	72.7
BiCos seg [7]	79.4
Dense HOG+Coding+Pooling[2] w/o seg	76.7
Seg+Dense HOG+Coding+Pooling[2]	80.7
CNN-SVM w/o seg	74.7
CNNaug-SVM w/o seg	**86.8**

Oxford 102 flowers dataset

Figure 10: Performance of off-the-shelf pre-trained models vs. specialized task-focused deep learning models [57]

- **Fine Tuning Off-the-shelf Pre-Trained Models**

This is a more involved technique, where we do not just replace the final layer (for classification/regression), but we also selectively retrain some of the previous layers. Deep neural networks are highly configurable architectures with various hyperparameters. As discussed earlier, the initial layers have been seen to capture generic features, while the later ones focus more on the specific task at hand. An example is depicted in the following figure on a face-recognition problem, where initial lower layers of the network learn very generic features and the higher layers learn very task-specific features. Using this insight, we may freeze (fix weights) certain layers while retraining, or fine-tune the rest of them to suit our needs. In this case, we utilize the knowledge in terms of the overall architecture of the network and use its states as the starting point for our retraining step. This, in turn, helps us achieve better performance with less training time.

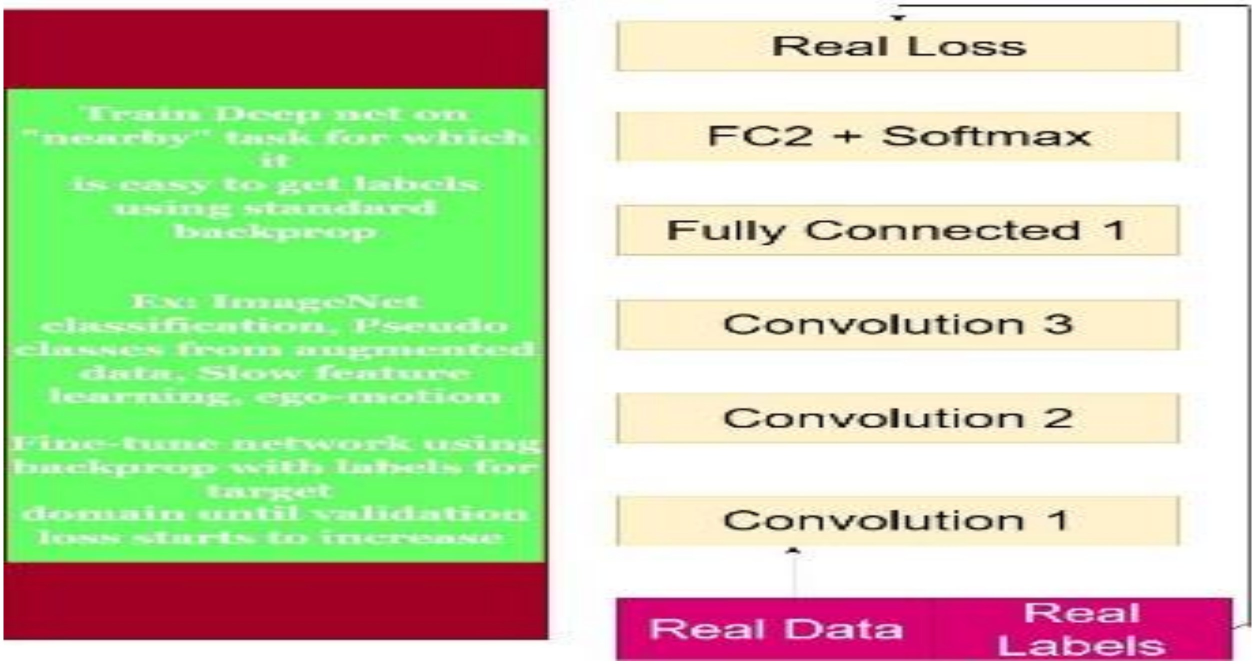

Train Deep net on "nearby" task for which it is easy to get labels using standard backprop
Ex: ImageNet classification, Pseudo classes from augmented data, Slow feature learning, ego-motion
Fine-tune network using backprop with labels for target domain until validation loss starts to increase
Real Loss
FC2 + Softmax
Fully Connected 1
Convolution 3
Convolution 2
Convolution 1
Real Data
Real Labels

Figure 11: Fine-tuning Pre-trained models

4.3.3. When to use Transfer Learning

Depending on both the size of the new dataset and the similarity of the new dataset to the original dataset, the approach for using transfer learning will be different. Keeping in mind that CNN features are more generic in the early layers and more original-dataset specific in the later layers, here are some common rules of thumb for navigating the four major scenarios:

1. The *target* dataset is **small** and **similar** to the *base* training dataset. Since the target dataset is small, it is not a good idea to fine-tune the CNN due to the risk of overfitting. Since the *target* data is similar to the *base* data, we expect higher-level features in the CNN to be relevant to this dataset as well. Hence, we:
    - Remove the fully connected layers near the end of the pretrained *base* CNN
    - Add a new fully connected layer that matches the number of classes in the *target* dataset
    - Randomize the weights of the new fully connected layer and freeze all the weights from the pre-trained network
    - Train the network to update the weights of the new fully connected layers
2. The *target* dataset is **large** and **similar** to the *base* training dataset. Since the *target* dataset is large, we have more confidence that we won't overfit if we try to fine-tune through the full network. Therefore, we:
    - Remove the last fully connected layer and replace with the layer matching the number of classes in the *target* dataset
    - Randomly initialize the weights in the new fully connected layer
    - Initialize the rest of the weights using the pre-trained weights, i.e., unfreeze the layers of the pre-trained network
    - Retrain the entire neural network
3. The *target* dataset is **small** and **different** from the *base* training dataset. Since the data is small, overfitting is a concern. Hence, we train only the linear layers. But as the *target* dataset is very different from the *base* dataset, the higher-level features in the CNN would not be of any relevance to the *target* dataset. So, the new network will only use the lower level features of the *base* CNN. To implement this scheme, we:
    - Remove most of the pre-trained layers near the beginning of the CNN
    - Add to the remaining pre-trained layers new fully connected layers that match the number of classes in the new dataset
    - Randomize the weights of the new fully connected layers and freeze all the weights from the pre-trained network
    - Train the network to update the weights of the new fully connected layers
4. The *target* dataset is **large** and **different** from the *base* training dataset. As the *target* dataset is large and different from the *base* dataset, we can train the CNN from scratch. However, in practice, it is beneficial to initialize the weights from the pre-trained network and fine-tune them as it might make the training faster. In this condition, the implementation is the same as in case 3.

4.3.4. Advantages of using Transfer Learning

Transfer learning is an optimization, a short cut to saving time or getting better performance. As shown in Figure 3.5, Lisa Torrey and Jude Shavlik described in [99] three possible benefits viz.,

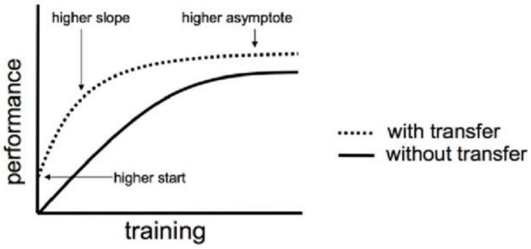


Figure 12: Benefits of Transfer Learning

• Higher start. The initial skill (before refining the model) on the source model is higher than it otherwise would be.

• Higher slope. The rate of improvement of skill during training of the source model is steeper than it otherwise would be.

• Higher asymptote. The converged skill of the trained model is better than it otherwise would be.

4.4. Data and Model setup

4.4.1. Data Preprocessing

In machine learning, it is often the case that the values of input features of sample data points are not with in a specific range. In that case, the gradients of a neural network become unstable and jumps back and forth. Consequently, the convergence become too slow, or does not converge at all. Normalization is a process to restrict the values of different features to same scale without disturbing the difference between the values of same feature. By making different features have same range of values, we ease the task of the network of finding Global Minimum. Two most popular normalization techniques are:

- **Min-max**

  Minimum value of a particular feature is subtracted from the value and then divided by the difference between the maximum and minimum value. Thus, every column now has values in range 0 and 1.

- **Standardization Scaling**

  Subtract the mean from the value and divide by the Standard Deviation. Now, each feature now has 0 as mean and 1 as standard deviation.

4.4.2. Weight initialization

There are several weight initialization techniques, among which three most popular techniques are:

- **Zero Initialization:**

  In general practice, biases are initialized with zero and weights are initialized with random numbers. Because, if the wrights are initialized with zeroes, the derivative of all weights with respect to the loss will be same and it will be as good as a linear model.

- **Random Initialization:**

  In this technique, weights are initialized with random numbers. But very large or very small numbers create problems as, the result of w.x + b becomes very high and for activation functions like sigmoid, the slope of the gradient is very small and thus learning takes a lot of time. Same is the scenario for very small weights.

- **He initialization :**

  The equation for initializing the weights is : $\sqrt{\frac{2}{size^{[l-1]}}}$

  W[l] = np.random.randn(size_l,size_l-1)*np.sqrt(2/size_l-1)

- **Xavier Initialization:**

  Same as He-initialization, except in the formula, the 2 is replaced with 1 and it is used in tanH activation function.

4.4.3. Regularization

Every supervised problem can be formulated as, given some data points in the feature space, we have to find an optimal function, that can follow the trend. Every problem has its own level of complexity and almost every problem contains noise. If the degree of the polynomial of the fitting function is too high, then it will have enough freedom to fit all the datapoints perfectly, thus incorporating noise. As noise is stochastic, the function might perform very good on training dataset, but testing performance will be very poor. On the other hand, if the degree of the polynomial is too low, it will not be able to follow the trend and might end up being a linear function which performs poorly on fitting the data points. It will do bad in both training and testing data. So, the challenge is to find the optimal function with enough freedom to match the complexity of the problem. Now, there is no optimal algorithm for it. So, all the functions starting from the linear to the one with very high complexity is tested parallelly, but now the goal is to minimize the loss (as before) as well as penalize the freedom of the function. The penalty is a monotonically increasing function with the degree of freedom. Thus, by incorporating penalization on the hyperparameters, we force it to fit the datapoints with minimum degree of freedom. The tradeoff between the training accuracy and complexity of the function is the goal of Regularization.

4.4.3.1. L1 Regularization

L1 Regularization adds the summation of the parameters to the loss function. This penalization can push some values to zero, resulting in offing/rejecting some parameters.

$$\mathrm{Error}_{L1} = \mathrm{Error} + \sum_{i=0}^{N} | \beta i |$$

4.4.3.2. L2 Regularization

It adds squared sum of the parameters to the loss term. This enforces higher penalty to large parameters compared to smaller ones, and forces the parameter to reach zero slowly unlike L1 Regularization, where it is reduced to zero rapidly.

$$\mathrm{Error}_{L2} = \mathrm{Error} + \sum_{i=0}^{N} | \beta i^2 |$$

4.4.3.3. *Dropout*

The idea of Dropout [33] is to turn off random set of neurons of a network randomly. The reason being, densely connected neurons over the time of training starts to depend on each other strongly, developing interdependency and eventually overfit, as dense layers consume most of the parameters of the network. By employing Dropout, neurons are forced to work with randomly selected other neurons, incorporating more robust features. Due to this, the final network becomes an ensemble of networks, each network being a network formed with a subset of all the neurons. This reduces forward propagation time, but requires almost double the number of epochs to train the network. At the time of evaluation, output of each neuron is adjusted according to it’s contribution in training.

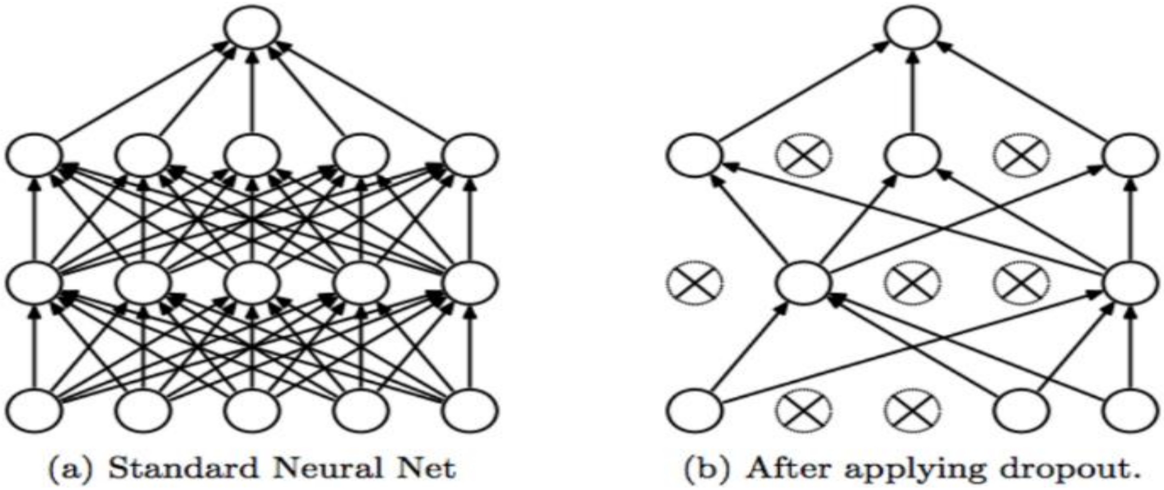


Figure 13 : Dropout [33]

5. Pneumonia detection using Transfer Learning

5.1. Overview of the work

It’s quite evident from past works that predicting pneumonia cases from Chest x-ray or Ultra sound images is a difficult task, if feasible at all. Deep Learning is a paradigm of Machine Learning which is a proven aid in this challenging scenario, making rule based complex arithmetic approach obsolete. While there have been some efforts in the past on detecting pneumonia using Artificial Neural Networks, but most of them used handcrafted features extracted from the images to feed into the ANN to perform classification. In contrast to these, Among the Deep Learning ventures which have the capability to learn features from the input images and classify them as well, Rajpuurkar et al. [9] and Qingji Guan et al. [32] and Shaanxi Key et al. [34] are significant.

Major steps followed for the present work:

1. Lung Segmentation with Chest X-ray images
2. Cropping the lung segments from X-ray images
3. Training a neural network on Lung Images, with 14 pathologies including Pneumonia
4. Displaying the pathological areas in the lung images using Gradient based Class Activation Mapping (Grad-CAM)

5.2. Lung Segmentation

X-Ray images of chest areas generally do not contain only the Lungs. Almost all chest X-ray images contain additional parts of body, like hands, belly, background, faces. While it’s possible for a network to recognize the lungs (significant part) from these images, it is not easy to detect 14 kind of pathologies of lungs from such images for the following reasons.

Pathological areas sometimes cover only 2-3% of complete X-ray images. Knowledge about the pixels of ‘non-lung’ areas is always a distraction and unnecessarily increases the complexity of the network. So, in such cases, the network should be made focused on lung areas. It requires Lung Segmentation prior to training for pathology detection.

In the present work a U-net is used for this purpose. After it detects a lung area in an X-ray image, it is to be extracted by identifying a minimum rectangular area big enough to contain both the lungs in the image.

5.2.1. U-Net

The U-Net[52] architecture is built upon the Fully Convolutional Network (a special type of CNN consisting of only Convolutional Layers) and modified in a way that it yields better segmentation result in medical imaging. Compared to FCN-8, the two main differences are:

1. U-net is symmetric

2. the skip connections between the **downsampling** path and the **upsampling** path apply a concatenation operator instead of a sum.

These skip connections intend to provide local information to the global information while upsampling. Because of its symmetry, the network has a large number of feature maps in the upsampling path, which allows to transfer information. By comparison, the basic FCN architecture only had '*number of classes*' feature maps in its upsampling path. Being a fully convolutional network, it does not depend on the shape of input image. The architecture of U-net is given below:

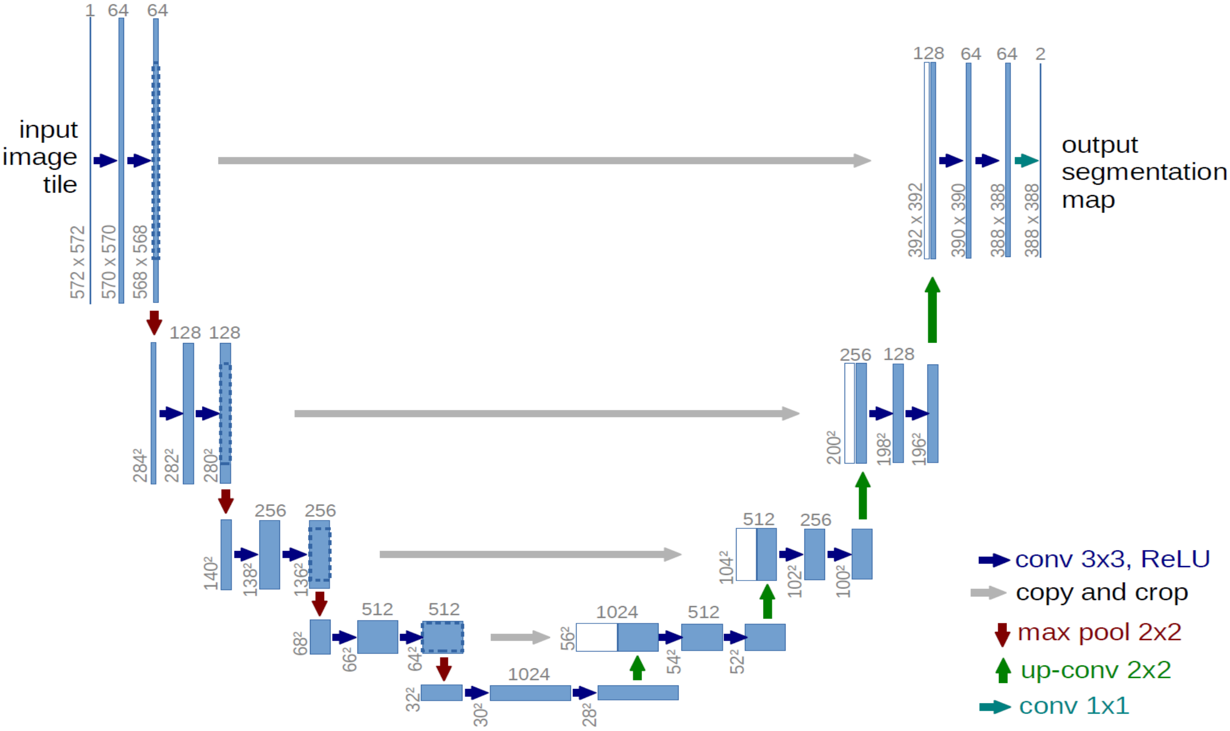


Figure 14: U-net architecture [55]

5.2.2. Databases

The two databases, **Montegomary**[17] and **Schenzen**[16], are used here for Lung segmentation with an U-net. **Montegomary** and **Schenzen** contains **456** and **234** images of lungs respectively along with masks for lungs. In total we had 704 Chest X-ray images with corresponding masks. **Montegomaty** dataset had separate masks for left and right lung. Naturally, we had to merge them in order to form the ground truth image. While training we used 80-20 approach, that is 80% of the images were used to train the network and the rest 20% were used for validation.

Images were resized to **224x224** and normalized before feeding it to the U-net. Thus, the segmented images were also of size **224x224**. The U-net used in this task had nearly 30 million parameters, but we had only 704 images of lungs for segmentation purpose. So naturally we had to take resort of Image Augmentation to increase the number of training samples. Some of the augmentations used for this purpose were:

1. Translation in random direction by random number of pixels in the range 0 – 20 % of 224.
2. Rotation by random degrees in between -15 to 15 degrees
3. Random zoom in the range of 0 to 20% of the image size
4. Mapping pixel values in the range 0 to 1.

5.2.3. Preprocessing

Due to presence of considerable amount of opacities in pathological X-ray images, the network trained with normal Chest X-ray images detects a huge number of false positives (mainly near the region between the body and hands) and false negatives (mainly in the regions with lung-opacity) on pathological X-ray images, to be dealt with the pneumonia classifier network. Occurrences of False Positives and False Negatives on some pathological sample X-ray images processed by the U-net trained for Lung area segmentation, are shown in figure 15[a-d]. This is mainly because of the fact that few of the normal images in the training set used for U-net contained opacities so the network didn’t really learn well to segment lungs in presence of opacity and it is bound to perform worse on the pathological images, to be used for training the ‘Pneumonia classifier’. So, just by cropping out the lung areas segmented by the U-net cannot solve the problem here.

To address this problem, Histogram Equalization is performed on the training images before feeding it to the U-net to be trained for Lung area segmentation. Performing histogram equalization over the entire image is not a good choice because Lung opacities could occur anywhere on the lung regions and only in those particular regions, contrast is much lower than rest of the image. So, ‘**Adaptive Histogram Equalization’** which essentially performs histogram equalization separately on different regions of an image and stiches the regions together using **Bilinear Interpolation**, is an obvious choice here. Using this technique, contrast in the images with lung opacities is enhanced. Lung area segmentation performed by the U-net on some samples of X-ray images with pathologies are shown in figure 15[a-j]. It is evident from the figure that contrast enhancement has resulted in significant improvement on Segmentation results, compared to those shown in figure 15[a-d], which shows lung segmentation results without prior contrast enhancements. It is also evident from the figure 15[a-j] that while False Negatives have significantly been reduced with prior contrast enhancement, False Positives are still present in the images due to appearance of black regions in between patient's arms and chests in Chest X-ray images. Such segmentation result is not acceptable so it requires   post processing once more to remove False Positives from the lung-segmented images.

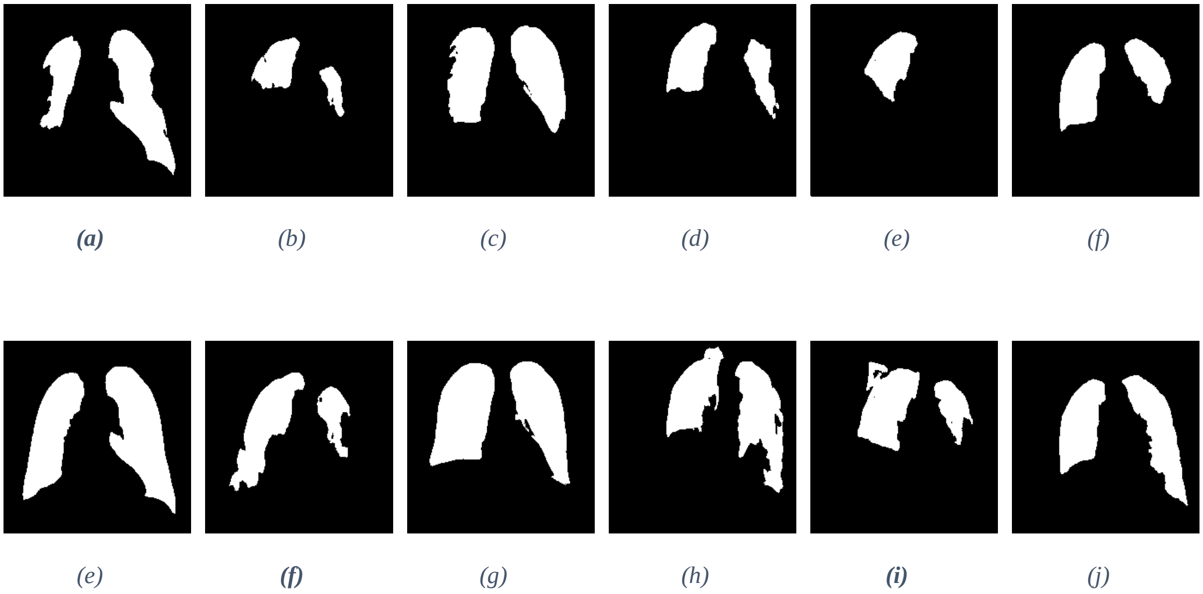


Figure 15[a-f] shows the Lung Segmentation result on normal Chest X-ray images.

[e-j] shows corresponding results on Histogram Equalized images.

5.3. Removing False Positives

The procedure devised here for removing False Positives from lung X-ray images is as follows:

First, certain threshold (0.4) is chosen through trial and error, based on which the pixel values of the lung-segmented images are thresholded between 0 and 1. In this process each thresholded X-ray image contains white pixels signifying **True Positives (lungs) mixed with False positives**, and black pixels signifying **everything else but lungs, such as other body parts and background as well as False Negatives**. Although nothing could be done to eliminate False Negatives, some Image Processing approaches can be of help to eliminate False Positives.

Some of the False Positives appear in the images due to noise. These are represented with tiny white regions there as shown in figure 16[e-h]. To remove such noise, a morphological operation namely Opening (Erosion followed by Dilation) is performed. As a result, more or less noise free lung segmented images consisting of a number of (usually 3-4) candidates for lung regions shown white in figure[i-l] are obtained. Out of these candidates for lung regions only and exactly two represent True Positives and the rest False Positives. In rare cases, where one of the lungs or both the lungs are barely visible due to the presence of opacity, only one or no candidate regions represent True Positive. For the images where more than two candidates for lung regions are present, True Positives are identified by considering the candidate regions with highest and second highest areas. If only one candidate region is found in the lung segmented noise eliminated X-ray images, choice for True Positive is unique. Figures below shows the sequence of operations through which lungs are extracted from the original image:

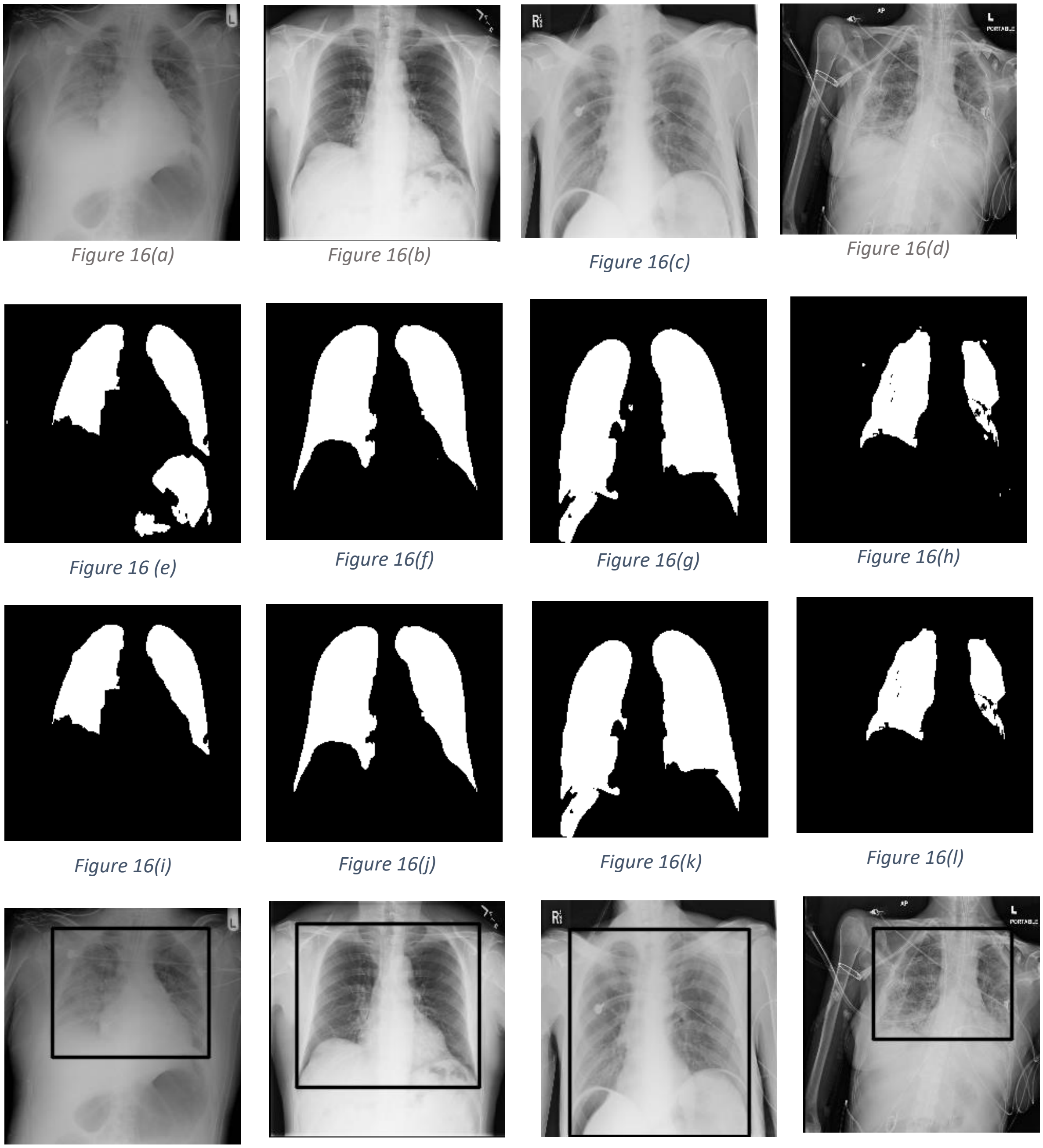


Figure 16(a) Figure 16(b) Figure 16(c) Figure 16(d)

Figure 16 (e) Figure 16(f) Figure 16(g) Figure 16(h)

Figure 16(i) Figure 16(j) Figure 16(k) Figure 16(l)

Figure 16(m) Figure 16(n) Figure 16(0) Figure 16(p)

Figure 16(a-d) shows the original images of chest X-ray. Figure 16(e-h) shows the result of Lung segmentation with a U-net. Figure 3(i-l) shows the post processed images, where everything but the lungs have been eliminated. Figure 4(m-p) shows bounding box around the lungs on original images

After the lung regions are so detected, they are extracted by using minimum bounding boxes. With all such extracted lung images the training dataset is formed.

The chest x-ray images for which one or no lung regions can be detected, the entire images are also included in the training set.

Since the sizes of the lung images which formed the training set are not same, they are all normalized to a size 224x224 before submitting them as input to the Densenet-121.

5.4. Training the Classifier

The classifier we used here was DenseNet-121. It is a 121-layer Convolutional Neural Network each layer directly connected to every layer after it.

5.4.1. Why DenseNet?

Traditional deep Convolutional Neural Network perform poorer when the depth is increased. Because, the gradients that flows backward from starting from the last layer gets vanished mid-way before reaching the other side. DenseNets tried to **simplify the connectivity pattern** between layers introduced in architectures like:

- Highway Networks
- Residual Networks
- Fractal Networks

The authors solve the problem ensuring **maximum information (and gradient) flow**. To do it, they simply connect every layer directly with each other.

Instead of drawing representational power from extremely deep or wide architectures, DenseNets exploit the potential of the network through feature reuse. It has been observed that some layers in ResNet architecture has redundant feature which unnecessarily contributes to a high number of parameters without learning anything new. DenseNet solves this problem by feature reuse, a particular layer in DenseNet can use any lower level feature when/if required.

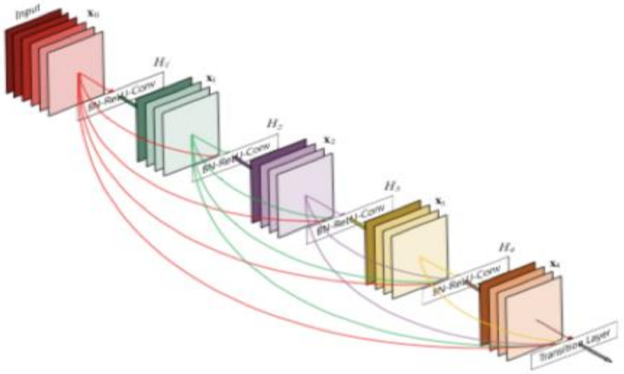
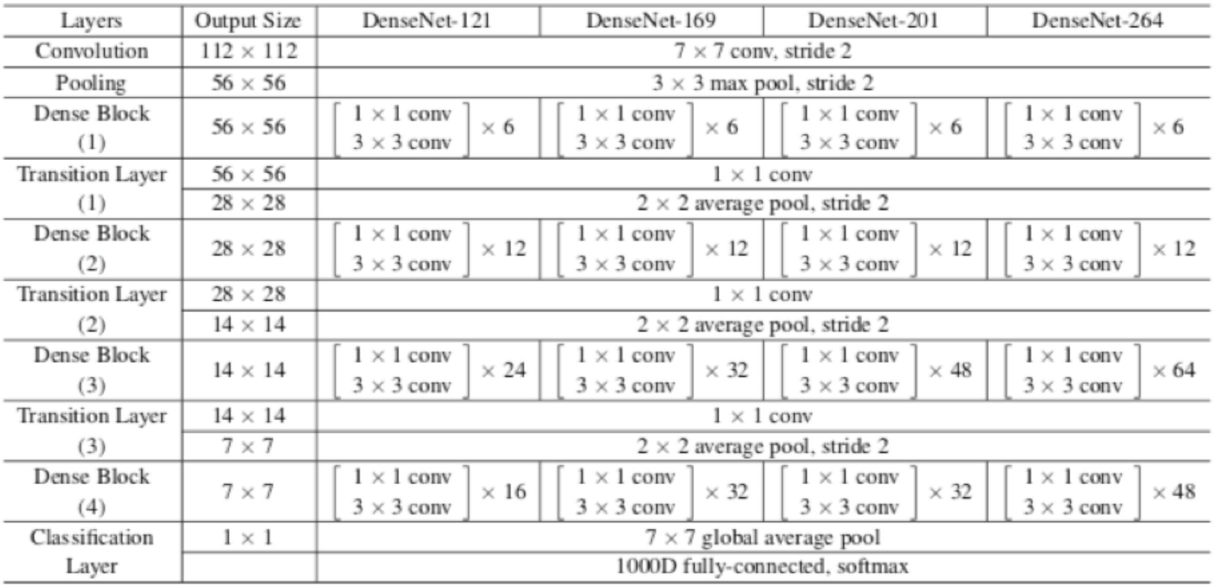


Figure 16: Simplified view of DenseNet [56]



Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	[1 × 1 conv 3 × 3 conv] × 6	[1 × 1 conv 3 × 3 conv] × 6	[1 × 1 conv 3 × 3 conv] × 6	[1 × 1 conv 3 × 3 conv] × 6
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	[1 × 1 conv 3 × 3 conv] × 12	[1 × 1 conv 3 × 3 conv] × 12	[1 × 1 conv 3 × 3 conv] × 12	[1 × 1 conv 3 × 3 conv] × 12
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	[1 × 1 conv 3 × 3 conv] × 24	[1 × 1 conv 3 × 3 conv] × 32	[1 × 1 conv 3 × 3 conv] × 48	[1 × 1 conv 3 × 3 conv] × 64
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	[1 × 1 conv 3 × 3 conv] × 16	[1 × 1 conv 3 × 3 conv] × 32	[1 × 1 conv 3 × 3 conv] × 32	[1 × 1 conv 3 × 3 conv] × 48
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

Figure 17: Different kinds of Densenet and their consisting blocks [56]

5.4.2. Database Overview

Several publicly available datasets are present regarding Chest X-ray image classification, ex: JSRT dataset [15], Shenzhen Chest X-ray set [16], Montegomary County Chest X-ray[17], Indiana University Chest X-ray Collection dataset[18]. However, the problem with these datasets are, the number of samples are too few to train a Deep Neural Model effectively.

There are two databases that we used.

1. We use the ChestX-ray14 dataset released by Wang et al. (2017) which contains 112,120 frontal-view X-ray images of 30,805 unique patients. Wang et al. (2017) annotate each image with up to 14 different thoracic pathology labels using automatic extraction methods on radiology reports. This is a Multi-label dataset.
2. Another one is a database from Radiological Society of North America (RSNA®). In collaboration with US National Institutes of Health, The Society of Thoracic Radiology, and MD.ai they prepared this dataset, which consisted of 3875 images of frontal Chest X-ray images, with 1346 images of Pneumonia affected lungs and rest of patient with no abnormalities.

To stress on the task of detecting pneumonia in Chest X-ray images with high precision, we merged these two datasets to increase the count of images with Pneumonia affected lungs. After merging the datasets, we ended up with 1,13,751 images of frontal Chest X-ray images. One mentionable point here is, the images in Chest X-ray 14 dataset had multiple labels associated with it, whereas this was not the case with Kaggle dataset. As described in the section above, all the images, after some preprocessing, were resized to 224x224 before using it for classification. Number of images in the final dataset, associated with a particular pathology is given below.

Table 1 : Number of samples corresponding to each pathological case in two datasets

Pathologies	**Chest X-ray 14**	**RSNA**
Atelectasis	11535	-
Cardiomegaly	2772	-
Effusion	13307	-
Infiltration	19870	-
Mass	5746	-
Nodule	6323	-
Pneumonia	5235	3875
Pneumothorax	5298	-
Consolidation	4667	-
Edema	2303	-
Emphysema	2516	-
Fibrosis	1686	-
Pleural Thickening	3385	-
Hernia	227	-
Norma / No Finding	61761	1346

5.4.3. Training

Random 80% of the dataset was used as training samples. Among the rest, random 12% were used as Test set and rest of the images were used as Validation set. A Densenet-121, pretrained on ImageNet[11] dataset was used for training. We ripped off the classification layer and added a new fully connected layer of 1000 nodes. The network had 15 output nodes, each responsible for spitting the probability of the input image to have certain kind of pathology(among 14 pathologies) or being a normal case. Dropout with probability 0.4 was incorporated in the FC-layer. As for optimizer, we used Adam with standard parameters (lr=0.001, β1=0.99, β2=0.9). The network was finetuned for 150 epochs. Learning rate was decreased by a factor of 0.1 every time the loss plateaued for 8 epochs. Batch size of 16 was used both for training and validation. The dataset being highly imbalanced, modified Weighted Binary Cross entropy Loss was used as loss function.

$L(X,y) = \sum_{c=1}^{n} [-y_c \log p(Y_c = 1|X) - (1-y_c)\log p(Y_c = 0|X)]$,

where $p(Yc = 1|X)$ is the predicted probability that the image contains the pathology c and $p(Yc = 0|X)$ is the predicted probability that the image does not contain the pathology c.

5.5. Localization

5.5.1. Grad-CAM

Grad-CAM [53] or Gradient weighted Class Activation Mapping answers a very important question about Deep Neural Networks: Are they trustworthy? Deep Neural Networks are still a black box to us. Given inputs, sometimes it gives us results, exceeding human level performance. But it's very hard to interpret what's going on behind the scene, leaving a vital question unanswered – how it reached the decision. Grad CAM solves this mystery to some extent. Given some input image and a predicted class, Grad CAM reveals where the network was looking in the image while making the decision. It provides us the heatmap, signifying the regions of the image which are important to the network.

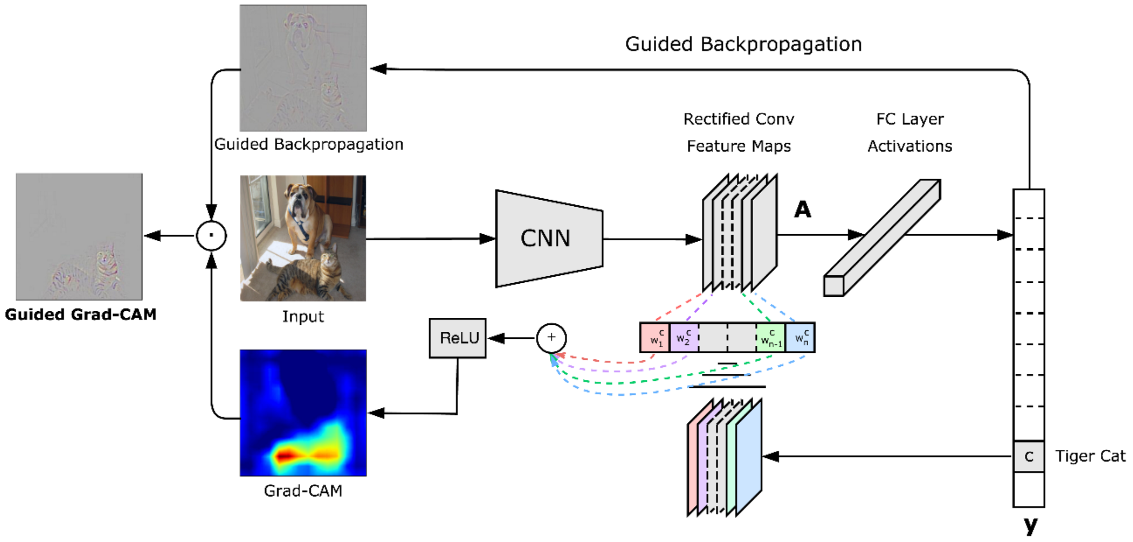


Figure 19: Idea of Grad-CAM [53]

5.5.2. Intuition

Quoting from the original paper[53]:

Gradient-weighted Class Activation Mapping (Grad-CAM), uses the gradients of any target concept (say logits for 'dog' or even a caption), flowing into the final convolutional layer to produce a coarse localization map highlighting the important regions in the image for predicting the concept.

The intuition behind is, at the very last convolution layer of a CNN, it learns Class specific features. For example, for a network which detects human in an image, it might be looking for hands and legs and eyes. While it is true that, some localization data is lost at the last conv layer, but it is the best trade-off between class specific semantic features and localization of features. Given an input image to the network, gradient of y_c(score for class c) is calculated with respect to the values in the feature maps A, i.e., $\frac{\partial yc}{\partial Akij}$, and then Global average pooling is calculated. Global Average Pool reveals how much a particular feature is important for the network for making the decision. So, the elementwise sum of the feature maps weighted by the corresponding gradient gives a coarse localization map of the important areas of an image. The values are normalized and passed through ReLU to get the final localization map.

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}}$$

$$L^c_{\text{Grad-CAM}} = ReLU \underbrace{\left(\sum_k \alpha_k^c A^k\right)}_{\text{linear combination}}$$

The localization map so obtained is essentially of the same size of the feature maps. Bilinear Interpolation is applied on the map to enlarge it to the size of the original image. A combination of Guided Backpropagation and Grad-CAM can give pixel level localization map, although only Grad-CAM is used in our task.

5.5.3. Why it is useful

Once a deep network is trained and tested for detection of pathologies in Chest X-ray images, the next question comes how it interpretes. Grad-CAM, as mentioned previously may be of help in answering this question. Under the present work, the Grad-CAM is applied for finding the heatmaps from the pathological cases of different Chest X-ray images. Although the heatmaps cannot be confirmed by the expert physicians at this moment, they may be useful with availability of the experts in near future.

5.5.4. Application of Grad-CAM on Chest X-rays with pathology

In spite of the fact that the network (Densenet-121) performs well on the Test set, it is unclear how the network is making it's decision. So, in order to locate the pathological regions, Grad-CAM is applied on the network. Unlike CAM[59], where the last Dense layers of a CNN is replaced with Convolutional layers and the network is retrained with the new architecture, Grad-CAM depends on the gradient of the feature maps with respect to a specific class, hence no architectural changes or retraining is required.

Densenet-121 is distinctly divided into two sequential blocks, **Features** and **Classifiers. Features** consists of **Dense Blocks** and **Transition Blocks**. **Classifiers** only consists of one Fully Connected layer. In the last convolution layer, there are 128 feature maps. With this arrangement, the following steps are followed for generating the heatmaps of the chest X-ray images of the pathological cases of chest X-ray images, used here.

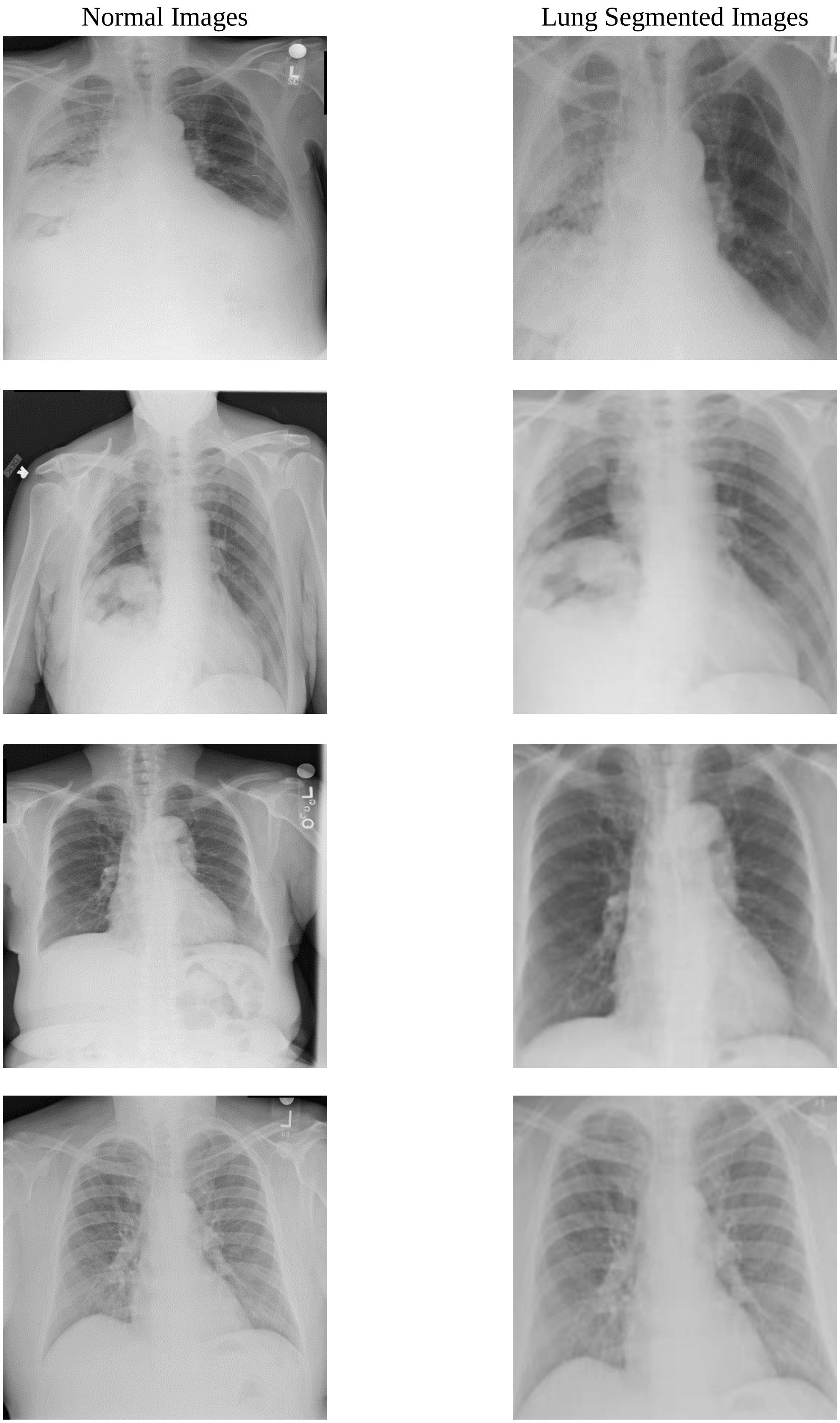
1. An image from the Test set is fed to the network after applying required preprocessing.
1. The gradient of the respective class is backpropagated till the last convolutional layer consisting of 128 feature maps, is reached.
2. Mean of the gradient are calculated for each channel in the Convolutional Feature Map, denoting how important that feature is for the particular class.
3. Sum of the feature maps, weighted by respective gradients is calculated and then normalized.
4. The normalized values are passed through ReLU to obtain the final Localization map.

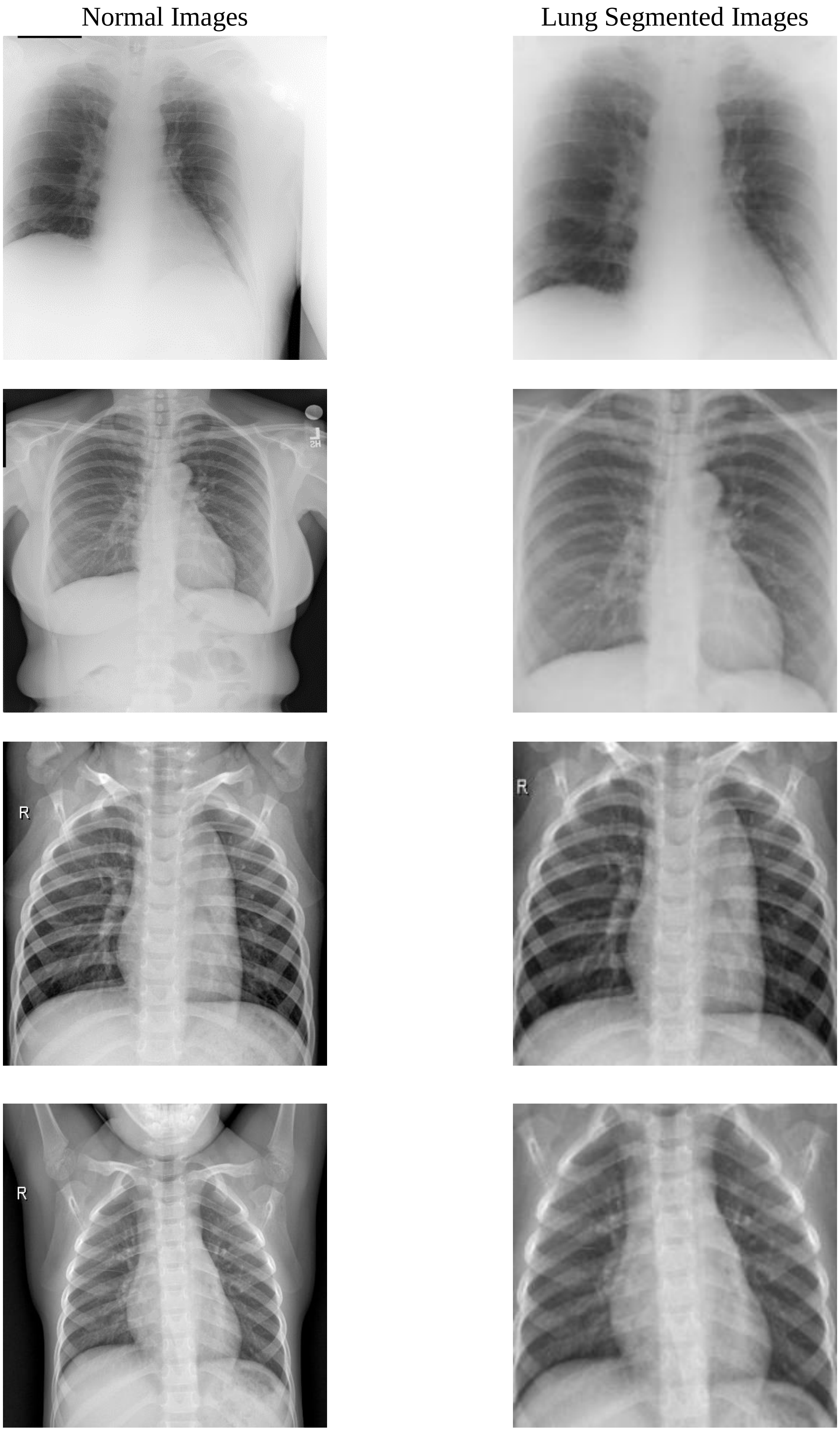
Here are some images of Chest X-ray with pathologies. The pathological regions are highlighted using Grad-CAM.

6. RESULTS

6.1. Lung Segmentation Result

Almost all the images in our training set contained No Lung-opacity, unlike lungs with pathologies - which normally has some kind of opacity which helps determining the kind of lung-disease. Hence, lungs were fairly visible and it was not difficult task for a network with parameters in range of millions to segment the lungs. Consequently, our network converged pretty fast (after almost 10 epochs) which accuracy of 95%. But we let it train for 30 more epochs, but accuracy did not change significantly whereas loss seemed to be decreasing linearly. We ended up with an accuracy of 97.6% after 40 epochs. Below are the results of some images along with the result of their Lung Segmented result.


Normal Images
Lung Segmented Images

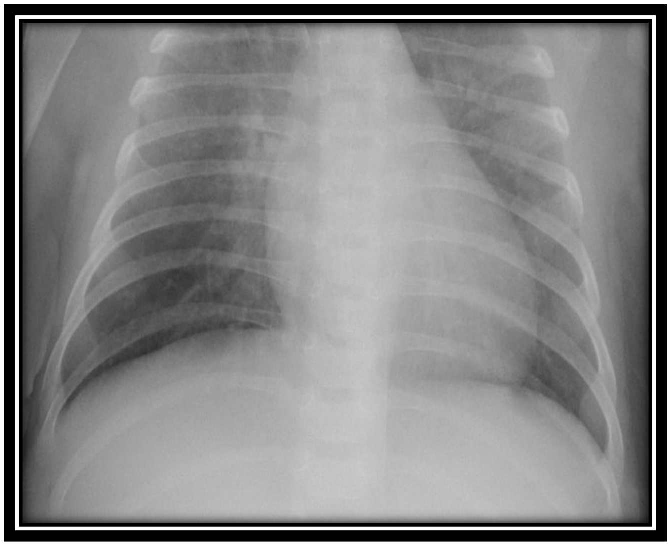
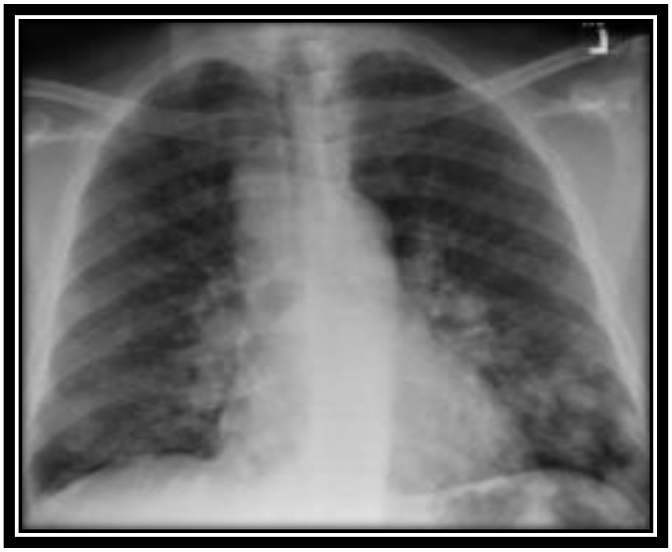


Normal Images

Lung Segmented Images

6.2. Pneumonia Detection Result

As mention in the Training section, a DenseNet-121 was used to train the network with Lung Segmented Images with Adam Optimizer for around 100 epochs. Here are a few results.

Images	Predicted probabilities for Pathologies
	Atelectasis: 1.2885e-03, Cardiomegaly: 3.2946e-05, Consolidation: 9.6575e-04, Edema: 2.5951e-05, Effusion: 1.7689e-04, Emphysema: 6.6695e-08, Fibrosis: 2.6602e-05, Hernia: 1.5976e-14, Infiltration: 1.9219e-03, Mass: 1.1354e-04, No Finding: 7.0516e-02, Nodule: 1.4201e-04, Pleural Thickening: 1.7663e-05, Pneumonia: 8.338e-01, Pneumothorax: 2.2161e-06
	Atelectasis: 4.8955e-02, Cardiomegaly: 4.7844e-03, Consolidation: 1.1590e-01, Edema: 5.7171e-02, Effusion: 1.6199e-01, Emphysema: 3.3764e-02, Fibrosis: 9.1335e-02, Hernia: 2.9168e-04, Infiltration: 3.9655e-01, Mass: 6.3888e-01, No Finding: 3.0347e-01, Nodule: 5.5854e-01, Pleural Thickening: 9.3124e-02, Pneumonia: 3.0334e-02, Pneumothorax: 6.4811e-02

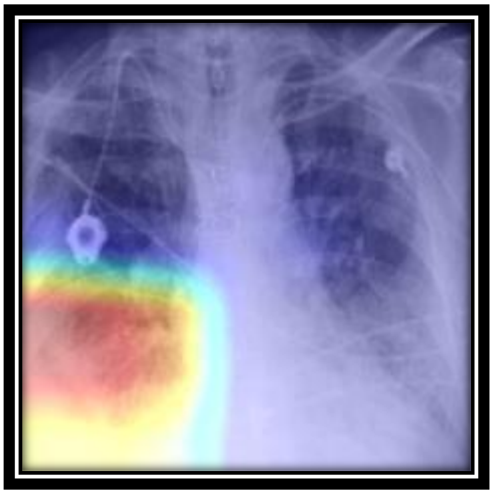
A comparative result of AUC obtained by various methods on Chest X-ray 14 datasets is presented in the table below:

Table 2: Comparison of the AUC of different methods on Chest X-ray 14 dataset

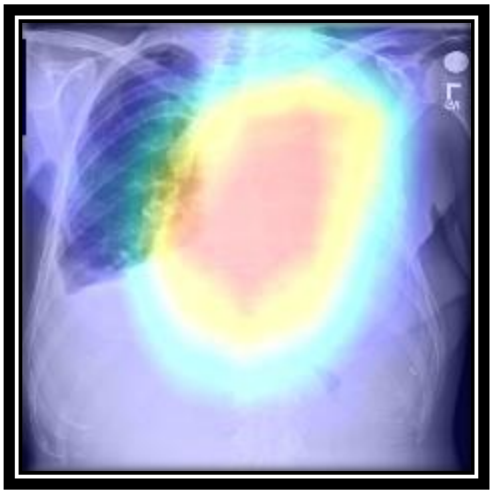
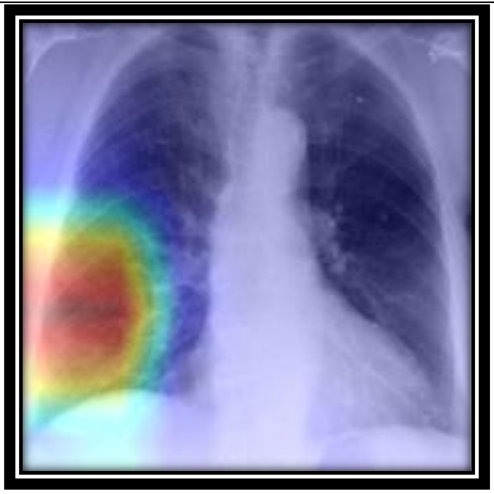
Method	AUC
Wang et al.[9]	0.738
Yao et al.[19]	0.803
Rajpurkar et al.[8]	0.842
Kumar et al.[7]	0.795
Our appproach	0.810

6.3. Grad-CAM Result

Gradient-weighted Class Activation Mapping or Grad-CAM helps building a coarse localization map from an input image, using the gradient flowing to the last convolutional layer. As for the task of detecting pathologies from Chest X-rays, Grad-CAM helps in determining the pathological area in the lungs. Here are some of the images with the pathological areas highlighted.



PNEUMONIA

	EFFUSION
	INFILTRATION

7. Future Scope

Future work may be carried out on studying the possibility of improving the performance of Densenet-121, used here for classification of pathological cases from Chest X-ray images, in the following directions.

1. Significant parts or patches of the Chest X-ray images containing pathologies may be extracted using attention mechanism and then be used to train a separate classifier so that the classification of pathology can be done in a more focused manner.
2. Not only Chest X-rays, but also incorporating symptoms of patients, like cough sound, fever, headache as features may help build more accurate classifiers.

8. Bibliography

[1] G. Volpicelli, “Lung Sonography,” in Journal of Ultrasound in Medicine, vol. 32, pp. 165–171, 2013.

[2] R. Copetti and L. Cattatossi, “Ultrasound diagnosis of pneumonia in children,” in La radiologia medica, vol. 113, pp. 190–198, 2008.

[3] D. Lichtenstein, “Ultrasound examination of the lungs in the intensive care unit,” in Pediatric Critical Care Medicine, vol. 10, pp. 693–698, 2009.

[4] D. Ashton-Cleary, “Is thoracic ultrasound a viable alternative to conventional imaging in the critical care setting?,” in British Journal of Anaesthesia, vol. 111, pp. 152–160, 2013.

[5] Pedro Cisneros-Velarde, Malena Correa, Holger Mayta, Cynthia Anticona, Monica Pajuelo, Richard Oberhelman, William Checkley, Robert H Gilman, Dante Figueroa, Mirko Zimic, Roberto Lavarello, Benjamín Castañed, “Automatic Pneumonia Detection based on Ultrasound Video Analysis”

[6] O. Zenteno et al., “An explorative childhood pneumonia analysis based on ultrasonic imaging texture features”, in Proc. SPIE, vol. 9681, 2015.

[7] M. Pereda, M. Chavez, C. Hooper-Miele, R. Gilman, M. Steinhoff, L. Checkley. “Lung Ultrasound for the Diagnosis of Pneumonia in Children: A Meta-analysis”, Pediatrics, 135(4), 714-722, 2015.

[8] Ronald Barrientos, Avid Roman-Gonzalez, Franklin Barrientos, Leonardo Solis, Malena Correa, Monica Pajuelo, Cynthia Anticona, Roberto Lavarello, Benjamin Castañeda, Richard Oberhelman, William Checkley, Robert H. Gilman, Mirko Zimic. “Automatic detection of pneumonia analyzing ultrasound digital images”

[9] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Robyn L. Ball, Curtis Langlotz, Katie Shpanskaya, Matthew P. Lungren, Andrew Y. Ng, “CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning”,2017

[10] Berbaum, K, Franken Jr, EA, and Smith, WL. The effect of comparison films upon resident interpretation of pediatric chest radiographs. Investigative radiology, 20(2):124–128, 1985.

CDC, 2017. URL https://www.cdc.gov/features/ pneumonia/index.html.

[11] Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Li, Kai, and Fei-Fei, Li. Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pp. 248–255. IEEE, 2009.

[12] Islam, Mohammad Tariqul, Aowal, Md Abdul, Minhaz, Ahmed Tahseen, and Ashraf, Khalid. “Abnormality detection and localization in chest x-rays using deep convolutional neural networks”. arXiv preprint arXiv:1705.09850, 2017.

[13] Wang, Xiaosong, Peng, Yifan, Lu, Le, Lu, Zhiyong, Bagheri, Mohammadhadi, and Summers, Ronald M. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. arXiv preprint arXiv:1705.02315, 2017.

[14] WHO. Standardization of interpretation of chest radiographs for the diagnosis of pneumonia in children. 2001.

[15] J. Shiraishi, S. Katsuragawa, J. Ikezoe, T. Matsumoto, T. Kobayashi, K.i. Komatsu, M. Matsui, H. Fujita, Y. Kodera, and K. Doi, “Development of a digital image database for chest radiographs with and without a lung nodule: receiver operating characteristic analysis of radiologists’ detection of pulmonary nodules,” American Journal of Roentgenology, vol. 174, no. 1, pp. 71–74, 2000.

[16] B. Van Ginneken, M. B. Stegmann, and M. Loog, “Segmentation of anatomical structures in chest radiographs using supervised methods: a comparative study on a public database,” Medical image analysis, vol. 10, no. 1, pp. 19–40, 2006.

[17] S. Jaeger, S. Candemir, S. Antani, Y.-X. J. W´ang, P.-X. Lu, and G. Thoma, “Two public chest x-ray datasets for computer-aided screening of pulmonary diseases,” Quantitative imaging in medicine and surgery, vol. 4, no. 6, p. 475, 2014.

[18] D. Demner-Fushman, M. D. Kohli, M. B. Rosenman, S. E. Shooshan, L. Rodriguez, S. Antani, G. R. Thoma, and C. J. McDonald, “Preparing a collection of radiology examinations for distribution and retrieval,” Journal of the American Medical Informatics Association, vol. 23, no. 2, pp. 304–310, 2015.

[19] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. S´anchez, “A survey on deep learning in medical image analysis,” arXiv preprint arXiv:1702.05747, 2017.

[20] A. Qayyum, S. M. Anwar, M. Majid, M. Awais, and M. Alnowami, “Medical image analysis using convolutional neural networks: A review,” arXiv preprint arXiv:1709.02250, 2017.

[21] D. Shen, G. Wu, and H.-I. Suk, “Deep learning in medical image analysis,” Annual Review of Biomedical Engineering, no. 0, 2017.

[22] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, “Deep convolutional neural networks for computer-aided detection: Cnn

architectures, dataset characteristics and transfer learning," IEEE transactions on medical imaging, vol. 35, no. 5, pp. 1285–1298, 2016.

[23] M. Anthimopoulos, S. Christodoulidis, L. Ebner, A. Christe, and S. Mougiakakou, "Lung pattern classification for interstitial lung diseases using a deep convolutional neural network," IEEE transactions on medical imaging, vol. 35, no. 5, pp. 1207–1216, 2016.

[24] S. Hwang and H.-E. Kim, "Self-transfer learning for weakly supervised lesion localization," in International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2016, pp. 239–246.

[25] C. Payer, D. ˇStern, H. Bischof, and M. Urschler, "Regressing heatmaps for multiple landmark localization using cnns," in International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2016, pp. 230–238.

[26] L. Yao, E. Poblenz, D. Dagunts, B. Covington, D. Bernard, and K. Lyman, "Learning to diagnose from scratch by exploiting dependencies among labels," arXiv preprint arXiv:1710.10501, 2017. 1, 2, 5, 7

[27] P.Kumar, M.Grewal, and M.M.Srivastava,"Boosted cascaded convnets for multilabel classification of thoracic diseases in chest radiographs," arXiv preprint arXiv:1711.08760, 2017. 1, 2, 5, 6, 7, 8

[28] Benjamin Antin, Joshua Kravitz, Emil Martayan, "Detecting Pneumonia in Chest X-Rays with Supervised Learning", 2017

[29] P.-P. Ypsilantis and G. Montana, "Learning what to look in chest x-rays with a recurrent visual attention model," arXiv preprint arXiv:1701.06452, 2017. 2

[30] V. Mnih, N. Heess, A. Graves et al., "Recurrent models of visual attention," in Advances in neural information processing systems, 2014, pp. 2204–2212. 2

[31] E. Pesce, P.-P. Ypsilantis, S. Withey, R. Bakewell, V. Goh, and G. Montana, "Learning to detect chest radiographs containing lung nodules using visual attention networks," arXiv preprint arXiv:1712.00996, 2017. 2

[32] Qingji Guan, Yaping Huang, Zhun Zhong, Zhedong Zheng, Liang Zheng and Yi Yang, "Diagnose like a Radiologist: Attention Guided Convolutional Neural Network for Thorax Disease Classification", 2018.

[33] Nitish Srivastava, Geofferey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", 2014.

[34] Shaanxi Key, "ChestNet: A Deep Neural Network for Classification of Thoracic Diseases on Chest Radiography",

[35] P. Liskowski and K. Krawiec, “Segmenting retinal blood vessels with¡? pub newline?¿ deep neural networks,” IEEE transactions on medical imaging, vol. 35, no. 11, pp. 2369–2380, 2016. 1

[36] Y. Yuan, M. Chao, and Y.-C. Lo, “Automatic skin lesion segmentation using deep fully convolutional networks with jaccard distance,” IEEE Transactions on Medical Imaging, 2017. 1

[37] H. R. Roth, L. Lu, A. Farag, H.-C. Shin, J. Liu, E. B. Turkbey, and R. M. Summers, “Deeporgan: Multi-level deep convolutional networks for automatedpancreassegmentation,”inInternationalConferenceonMedical Image Computing and Computer-Assisted Intervention. Springer, 2015, pp. 556–564. 1

[38] H. Fu, J. Cheng, Y. Xu, D. W. K. Wong, J. Liu, and X. Cao, “Joint optic disc and cup segmentation based on multi-label deep network and polar transformation,” IEEE Transactions on Medical Imaging, 2018. 1

[39] H. Fu, Y. Xu, S. Lin, X. Zhang, D. W. K. Wong, J. Liu, A. F. Frangi, M. Baskaran, and T. Aung, “Segmentation and quantification for angleclosureglaucomaassessmentinanteriorsegmentoct,”IEEEtransactions on medical imaging, vol. 36, no. 9, pp. 1930–1938, 2017.

[40] M. Anthimopoulos, S. Christodoulidis, L. Ebner, A. Christe, and S. Mougiakakou, “Lung pattern classification for interstitial lung diseases using a deep convolutional neural network,” IEEE transactions on medical imaging, vol. 35, no. 5, pp. 1207–1216, 2016. 1, 2

[41] P.Kumar,M.Grewal,andM.M.Srivastava,“Boostedcascadedconvnets for multilabel classification of thoracic diseases in chest radiographs,” arXiv preprint arXiv:1711.08760, 2017. 1, 2, 5, 6, 7, 8

[42] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya et al., “Chexnet: Radiologistlevel pneumonia detection on chest x-rays with deep learning,” arXiv preprint arXiv:1711.05225, 2017. 1, 2, 5, 6, 7, 8

[43] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, “Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases,” arXiv preprint arXiv:1705.02315, 2017. 1, 2, 4, 5, 6, 7

[44] R. Liao, S. Miao, P. de Tournemire, S. Grbic, A. Kamen, T. Mansi, and D. Comaniciu, “An artificial agent for robust image registration.” in AAAI, 2017, pp. 4168–4175. 1

[45] D. Yang, S. Zhang, Z. Yan, C. Tan, K. Li, and D. Metaxas, “Automated anatomical landmark detection ondistal femur surface using convolutional neural network,” in Biomedical Imaging (ISBI), 2015 IEEE 12th International Symposium on. IEEE, 2015, pp. 17–21. 1

[46] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, 2012, pp. 1097–1105. 1

[47] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014. 1

[48] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9. 1

[49] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778. 1, 3, 6

[50] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, “Densely connected convolutional networks,” arXiv preprint arXiv:1608.06993, 2016. 1, 6

[51] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997. 1

[52] Olaf Ronneberger, Philipp Fischer, Thomas Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation”, 2015.

[53] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, Chunfang Liu, “A Survey on Deep Transfer Learning”, 2018.

[53] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization, 2017

[54] https://www.xenonstack.com/blog/artificial-neural-network-applications/

[55] https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

[56] https://towardsdatascience.com/understanding-and-visualizing-densenets-7f688092391a

[57] https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a

[58] https://www.cs.ubbcluj.ro/~gabis/ml/ml-books/McGrawHill%20-%20Machine%20Learning%20-Tom%20Mitchell.pdf

[59] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba, “Learning Deep Features for Discriminative Localization”,