ISSN Number: 2454-2024(Online)
International Journal of Technical Research & Science

# DEEP LEARNING FOR PNEUMONIA CLASSIFICATION

**Yash Sharma, Gaurav Tiwari, Arti Vaish**
**E-Mail Id: yashsharma.btech18@sushantuniversity.edu.in,**
**gauravtiwari.btech18@sushantuniversity.edu.in**
**Department of Computer Science & Engineering, Sushant University, Gurugram, Haryana, India**

**Abstract:** Pneumonia is a common infectious disease that is responsible for over one million cases and tens of thousands of deaths annually in the US alone. As an example, in 2017, 1.3 million cases were reported, of which over 50000 died, resulting in a death rate of nearly four percent. To tackle this problem, we have used the RSNA pneumonia detecting challenge dataset provided by the US National Institutes of Health Clinical Centre.
It consists of 26684 X-ray images, of which about 20672 images do not show signs of pneumonia and remaining 6012 images do show signs of pneumonia. Though this project we tried to identify signs of pneumonia in X-Ray images with deep neural networks using PyTorch Lightning which is machine learning framework.
**Keywords:** Pneumonia, deep neural networks, PyTorch Lightning.

## 1. INTRODUCTION

Pneumonia is a lung infection causing inflammation of the air sacs(alveoli) filling them with pus making breathing difficult. Bacteria, viruses, and fungi cause this infection. Fungal Pneumonia is contracted from the environment and is not contagious, bacterial and viral Pneumonia are contagious and spread through inhalation of droplets from a sneeze or cough.

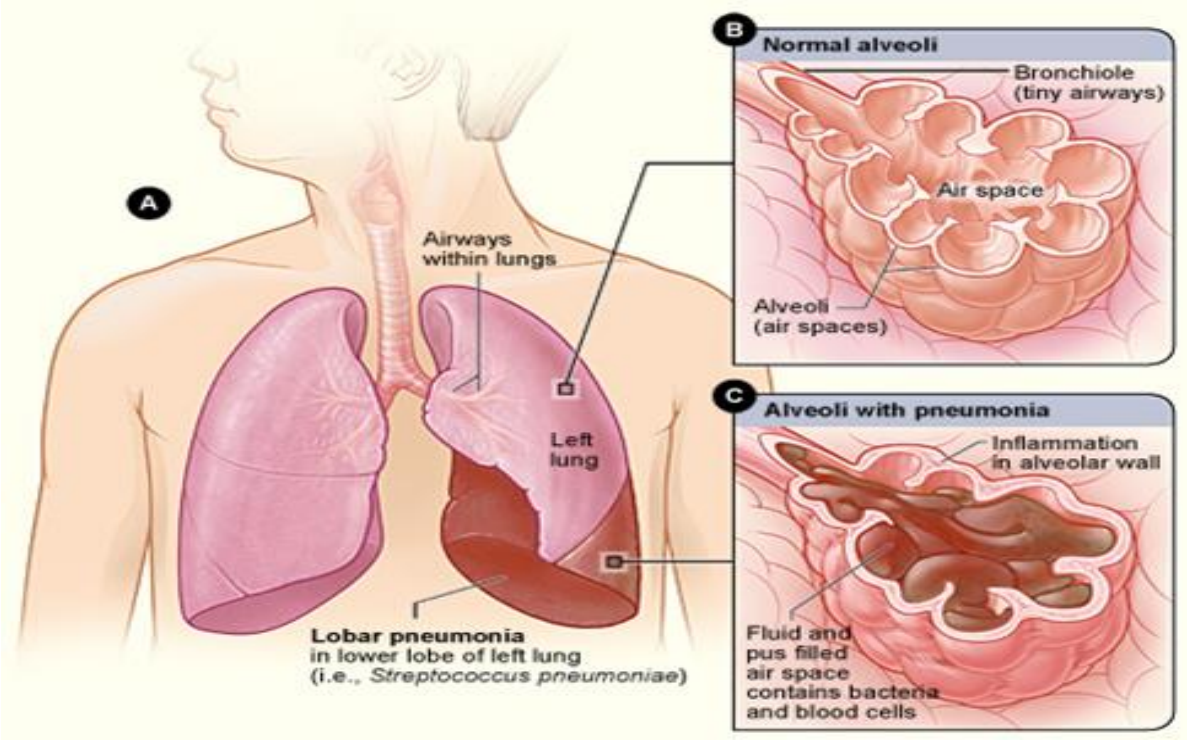


**Fig. 1.1 Alveoli**

The severity of the infection ranges from mild to life-threatening requiring ventilator support in extreme circumstances. One can contract and recover from this infection without even knowing about it, Doctors call this 'walking pneumonia' [2]. Senior citizens over 65 years of age and infants below 2 years are especially susceptible to this infection because of their weakened immunity.
Symptoms of the infection vary depending on the general state of health and age. Common symptoms include coughing that may produce phlegm (mucus), fever, sweating or chills, shortness of breath, chest pain, fatigue, loss of appetite, nausea, and headaches. [3]
Diagnosis: Blood Tests to check for bacterial infection, Chest X-Ray to determine the spread of infection, Pulse Oximetry to check blood oxygen level, Sputum test to establish the cause of infection. [1]
Most cases of viral pneumonia don't require Hospitalization and are cured by OTC medications prescribed by a doctor. However, if symptoms are severe then without proper medical care the infection may develop life-threatening complications including Lung abscess in which pockets of pus form in and around the lungs. Bacteraemia, in which blood gets infected with bacteria causing septic shock and organ failure. Fluid build-up between layers of tissue, lungs, and chest cavity. [4]
Pneumonia can be prevented by vaccination and annual flu shots. Maintaining a healthy lifestyle to strengthen the immune system, washing hands, and wearing masks in public areas. [5]

The chest X-ray is one of the most commonly accessible radiological examinations for screening and diagnosis of many lung diseases including pneumonia. Chest X-ray dataset used in this project was first provided by National Institutes of Health Clinical Centre. [6]

We found and download the dataset from Kaggle's community RSNA pneumonia detection challenge. [7]

## 2. IMPLEMENTATION

### 2.1 Pre-processing

First, we defined the data pre-processing. The original X-ray image of pneumonia detection dataset have a shape of 1024 x 1024. This is way too large for a machine learning algorithm. Thus, have to resize all images to 224 x 224. Next, standardize, all pixel values into the zero one interval by scaling them with a constant factor of 1/255. Or, in other words, we multiply all pixels in all images with this constant. Now, after applying the pre-processing steps, we split the dataset into 24000 trained images and 2684 validation images. This is crucial when tackling machine learning problems to make sure that the model is able to generalize to unseen data and not simply learn the label of all images by heart. Images that do not show signs of pneumonia are stored in directory zero, and all images with pneumonia are stored in directory one. There are four folders, zero and one in our trained directory and folders zero and one in our validation directory. Additionally, we compute mean and standard deviation of the training dataset for the purpose of normalization. For image in dataset, we calculated the sum of the individual pixels, as well as the sum of squared pixels. Then we add those two sums to two global variables called sums and sums squared. We can now compute the mean by simply dividing the sum variable by the number of images in the dataset. In this case, this would be 24000. Now we calculate the fraction between some squared and subtract the square mean from this expression. Subsequently, we take the square root of this result, which delivers us the standard deviation. So therefore, instead of loading the whole dataset into the memory, we only need to update two sums to compute mean and standard deviation.

### 2.2 Dataset

To create the dataset, we will make use of the torch vision dataset folder for a class which alleviates the need for a custom dataset. When loading an image, we set normalize the image pixel values with our pre computed mean and standard deviation. Normalization is performed by subtracting the mean from all pixel values and subsequent division by standard deviation. Additionally, we apply random rotations, random translations, random scales and random resized drops for the purpose of data augmentation or, in other words, we randomly rotate images, we randomly scale images, we randomly shift images, or we randomly crop a portion of the image and then resize it to the original image size of 224 x 224.

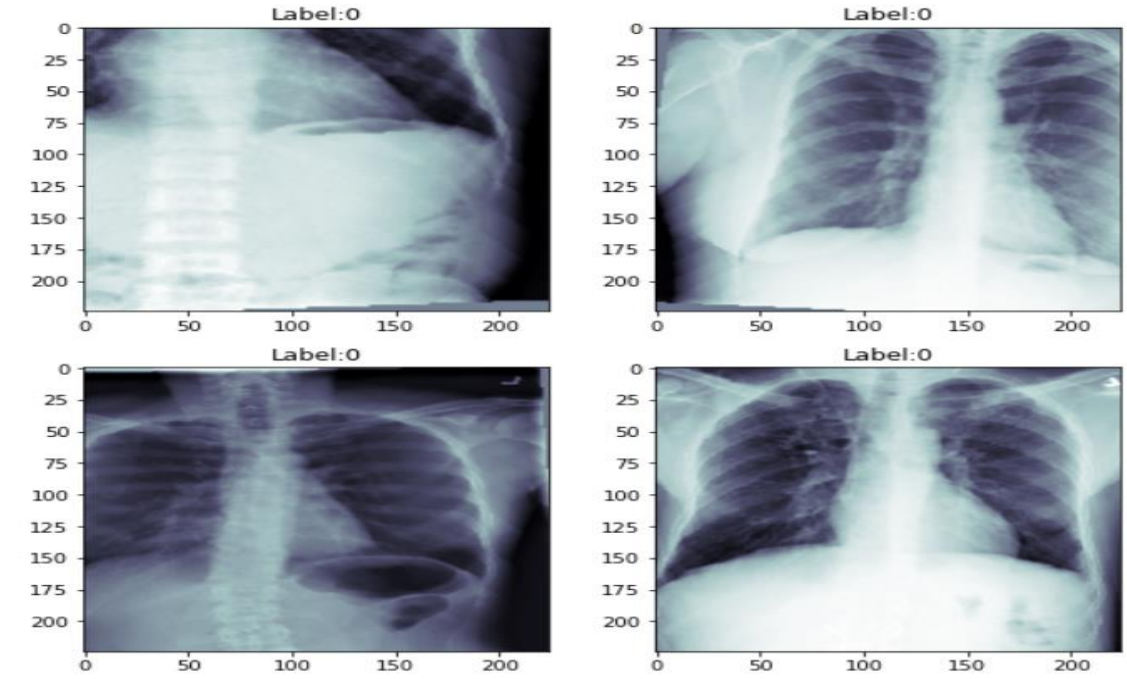


**Fig. 2.1 Dataset Images**

### 2.3 Training

Now while training, we use a library called PyTorch Lightning. PyTorch Lightning is a high-level python tracker for a simple and effective training that facilitates us some points. It eliminates the need of a manual implementation of the training loop. It automatically handles multiple GPU and comes with extensive logging and call-back interfaces. The great point of PyTorch lightning, we still have full and easy access to all internal variables and parameters. We use the famous ResNet18 network architecture, however, as we are working with medical images, which are typically not in the RGB format. We need to change the number of input channels from 3 to 1.

Additionally, as we only want to classify whether there are signs of pneumonia in an image, we change the output dimension from 1000 to 1. As loss function, we use the binary cross entropy with logits loss. This loss is directly applied to the logic or, in other words, the raw prediction values will be returned by a neural network and negative output, or a negative logic means that the network didn't see any signs of pneumonia in the corresponding image. In contrast, a positive output or launch it means that the network saw some signs of pneumonia.
We used a famous Adam Optimizer with a learning rate of 1e-4 and then trained the network for 30 epochs.

### 2.4 Interpretability

Now if we could answer to question why the model outputs pneumonia for a certain image, it would significantly increase the acceptance of such models in the general population and also amongst medical professionals.
Luckily, this is in fact absolutely possible by extracting the image region with the largest influence on the prediction. Or, in other words, we need to find out which part of the image supports the classifier the most in its prediction. One way to achieve this is to compute the class activation map. To compute class activation map, first, we need to grep the output of the last convolutional layer. So instead of only storing the prediction, we also store the output of the last convolution in the network. Those values are called features. Next, we need to extract the weights w off the fully connected layer. Those are the weights responsible to transform the output of the last convolutional layer into the raw prediction of our network. Now we simply compute the DOT product between our k feature vectors defined by A and our weights W. This produces the final class activation map.
However, there are some restrictions when we apply the class activation map. At first, it only works with convolutional neural networks. And in fact, it needs a specific network architecture. Namely, the last contribution must be followed by a global average pooling layer, which is then followed by a single, fully connected layer.
When applying class activation map-based methods to other network architecture or a not convolutional neural network, we can use some alternatives. As an example, GradCAM and GradCAM++, which are both extensions of the cam method.

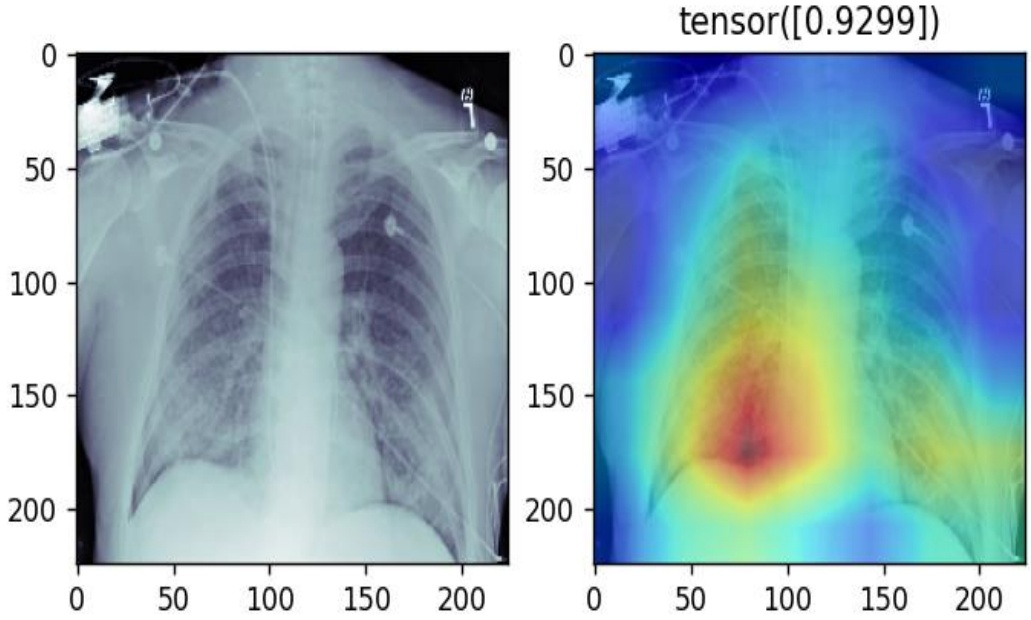


**Fig. 2.2 Computed Class Activation Map**

## CONCLUSION

We successfully build a classifier which is able to identify signs of pneumonia in X-ray images using RSNA Pneumonia dataset. First, we pre-processed the images by resizing the images to 224 x 224, standardize the pixel value into the interval [0,1] by scaling with 1/255, splitting the images into two parts, training images and validating images and normalized by computing mean and standard deviation. Then we created our dataset by making the use of torch vision. Dataset Folder. Additionally, also applied data augmentation by random rotations, random translation, random scales, random resized crops. To tackle the training, we used Pytorch lightning library. We also used ResNet18 network architecture by changing input channels from 3 to 1 and output channels from 1000 to 1. We used Binary entropy loss function with logits loss. This loss is directly applied to logits aka raw prediction values returned by the neural network. Negative logit means that the network didn't see the signs of pneumonia in the corresponding image. We used the famous Adam optimizer with the learning rate of 1e-4 and trained the network for 30 epochs. Finally, we calculated class activation map to interpret those the part of the image that support the classifier the most in its prediction.

## RESULT

The model yields an accuracy of 84.57%, precision of 70% and recall of 55.2%.

## REFERENCES

[1] "Pneumonia," Accessed on: Nov. 3, 2021. [Online]. Available: https://www.nhlbi.nih.gov/health/pneumonia#:~:text=A%20chest%20X%2Dray%20is,enough%20oxygen%20into%20your%20blood.

[2] Winchell JM, Mycoplasma pneumoniae – A national public health perspective, Curr Pediatr, Rev. 2013.

[3] "Pneumonia Symptoms," June 13, 2020. Accessed on: Nov. 3, 2021. [Online]. Available: https://www.mayoclinic.org/diseases-conditions/pneumonia/symptoms-causes/syc-20354204 .

[4] "What are the complications of pneumonia, Accessed on: Nov. 3, 2021. [Online]. Available: https://www.hopkinsmedicine.org/health/conditions-and-diseases/pneumonia .

[5] "Pneumonia Can Be Prevented—Vaccines Can Help," Oct 14, 2021. Accessed on: Nov. 3, 2021. [Online]. Available: https://www.cdc.gov/pneumonia/prevention.html

[6] Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM, "Chest-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases". IEEE CVPR 2017, Available: https://openaccess.thecvf.com/content_cvpr_2017/papers/Wang_Chest-ray8_Hospital-Scale_Chest_CVPR_2017_paper.pdf

[7] RSNA Pneumonia Detection Challenge, Radiological Society of North America, Aug 2018. [Online]. Available: https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data