**CSE-3200**

**System Development Project**

# **Pneumonia Detection from Chest X-Ray Images Using Convolution Neural Network**

# **Chest X-Ray Images (Pneumonia)**

# **Chest X-Ray Images (Pneumonia**

By

**Delowar Sikder**

Roll: 1607094

**MD**. **Mostafijur Rahman**

Roll: 1607117

****

**Department of Computer Science and Engineering**

**Khulna University of Engineering & Technology**

**Khulna 9203, Bangladesh**

**CSE-3200**

**System Development Project**

By

**Delowar Sikder**

Roll: 1607094

**MD**. **Mostafijur Rahman**

Roll: 1607117

**\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_**

**Supervisor: Mohammad Insanur Rahman Shuvo**

**Lecturer**

**Dept. of Computer Science and Engineering**

**Khulna University of Engineering & Technology**

**Khulna - 9203, Bangladesh**

**Department of Computer Science and Engineering**

**Khulna University of Engineering & Technology**

**Khulna - 9203, Bangladesh**

**ACKNOWLEDGEMENT**

We express our gratitude to Almighty ALLAH with due respect. We express our gratitude to our honorable supervisor **Mohammad Insanur Rahman Shuvo** (Lecturer, Department of Computer Science and Engineering, KUET) for his tremendous friendly and excellent guidance.

At the start of the term, it absolutely was nothing however a dream. We ‘re doubtful that we tend to might submit the project in time. Then within the tough moment, our honored teacher **Mohammad Insanur Rahman Shuvo** sir stood beside us, supervised absolutely for a decent work. He was there beside us all the time with huge heart, took pain, actuated fantastically, that has promoted our effort to check the sunshine.

And we conjointly categorical our feeling to all or any our teachers, senior students and cordial batch mates who have helped us in numerous ways in which throughout this analysis work with their valuable support.

**TABLE OF CONTENTS**

**Title Page No.**

Acknowledgement 1

Table of Contents 2

Chapter 1: Introduction

[1](#_Toc258456976).1 Introduction 4

[1.2 Statement of Problem 4](#_Toc258456978)

[1.3 Objectives 4](#_Toc258456980)

[Chapter 2: Background and Literature review](#_Toc258456981)

[2.1 Background 5](#_Toc258456982)

# [**Chapter 3**: **Tools :Convolutional Neural Network** 6](#_Toc258456983)

# [**Chapter 4:Workflow**](#_Toc258456983)

# 4.1: Implementation process9

# [**Chapter 5 : Implementation and Experimental Result**](#_Toc258456989) **of CNN**

5.1: Convolutional Neural architecture 10

# 5.2 Model Summary 10

# 5.3 Train Result 11

# 5.4 Test Accuracy 12

# [**Chapter 6 : Literature overview of CAE**](#_Toc258456989)

# 6.1 Convolutional auto encoder 13

6.2: Why the Convolutional Autoencoders are Preferred for Image Data 13

6.3: How Does the Convolutional Autoencoders Work

6.3.1: Encoder 14

6.3.2: Decoder 14

# [**Chapter 7 : Image**](#_Toc258456989) **Noising and Denoising Process**

# **7.**1 Implementation 15

# 7.2 Work flow 15

# 7.3 Model Description 16

# 7.3 Model Evaluation 16

[**Chapter 8 : Conclusion and Recommendation**](#_Toc258456998)

[8.1 Conclusion 19](#_Toc258456999)

[8.2 Limitation 19](#_Toc258457000)

8.3 Future Work 19

[References 20](#_Toc258457002)

**Chapter 1**

**Introduction**

* 1. **Introduction**

Pneumonia is respiratory organ inflammation caused by infection with virus, bacteria, fungi or different pathogens. According to National Institutes of Health (NIH), chest x-ray is that the best take a look at for respiratory disease identification. However, reading x-ray pictures are often difficult and needs domain experience and knowledge.

It would be very nice if we will simply raise a machine that scan the pictures and tell us the results. during this story, we'll use deep learning to train AI formula that analyzes chest x-ray pictures and detects respiratory disease.

**1.2 Statement of Problem**

Pneumonia is a horrible disease in Bangladesh. It is very difficult for the doctor to differentiate the pneumonia and normal diseases from x-ray image properly. Because human has some limitation. For example, human eyes are not capable to easily differentiate some change on nature. As result life of patient face a trouble issue due to doctor failure. Here machine is better than human. It can do any job properly. To solve this major problem, a system is developed that will detect the diseases from x-ray image whether it is pneumonia or not. For doing this job, we collect x-ray image from kaggle which is used for analysis.

**1.3 Objectives**

The main objective of our application is to save our valuable time to detect pneumonia diseases. Our project will recognize pneumonia diseases using a x-ray image obtained from a kaggle dataset. The background of our topics and applying it for the purpose of verifying pneumonia or not, using convolutional neural network.

**Chapter 2**

**Background**

**2.1 Background**

Nowadays all the countries are facing the growing and challenging difficulties of pneumonia diseases. Since the change of climate around the world a global challenge has been arise to stop health risk. Though there are several health organization has been incorporated to prevent the diseases from attack to aware the people, but due to some unawareness of human, it has become an easy infect to disease .The new England journal of medicine had been published that In children under the age of 5, the annual incidence of pneumonia in Europe and North America is 34 to 40 cases per 1000, higher than at any other time of life, except perhaps in adults over the age of 75 or 80.1-4 In the developing world, pneumonia is not only more frequent than in Europe and North America5-7; it is also more extreme and is the largest killer of children.

The United Nations Children's Fund (UNICEF) reports that 3 million children worldwide are kill ed every year by pediatric pneumonia. Many deaths occur in children with underlying conditions, such as chronic premature lung disease, congenital heart disease and immunosuppression, almost exclusively. Hence, pneumonia diseases detection is an emergence issue for a country to prevent its pestilence of this as well as people’s health.

Most of the people are face great health risk due to determine pneumonia diseases. The revolution of medical science, most of the medicine are invented arising with diseases. But before use of medicine we should be find the diseases as properly. A point to be noted that According to a recent study by Johns Hopkins, in the United States, more than 250,000 people die each year due to medical errors, making it the third leading cause of death after heart disease and cancer.

Because x-ray report is observed by doctor manually, lacking of doctor knowledge or optical illusion it may occur. So it is very import for that determine our diseases to take proper medicine.

**Chapter 3**

**Literature review of CNN**

**3.1 Convolutional Neural Networks**

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analyzing visual imagery. CNNs use a variation of multilayer perceptions designed to require minimal preprocessing. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics.

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field.

Every image is an arrangement of dots (a pixel) arranged in a special order. If you change the order or color of a pixel, the image would change as well. Let us take an example. Let us say, you wanted to store and read an image with a number 4 written on it.

The machine will basically break this image into a matrix of pixels and store the color code for each pixel at the representative location. In the representation below – number 1 is white and 256 is the darkest shade of green color (I have constrained the example to have only one color for simplicity).

## Defining a Convolutional Neural Network

We need three basic components to define a basic convolutional network.

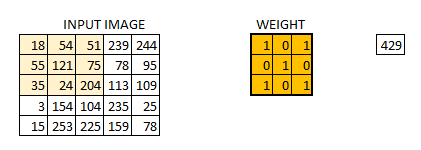
1. The convolutional layer
2. The Pooling layer[optional]
3. The output layer

### The Convolution Layer

We have initialized the weight as a 3\*3 matrix. This weight shall now run across the image such that all the pixels are covered at least once, to give a convolved output. The value 429 above, is obtained by the adding the values obtained by element wise multiplication of the weight matrix and the highlighted 3\*3 part of the input image. The 6\*6 image is now converted into a 4\*4 image.

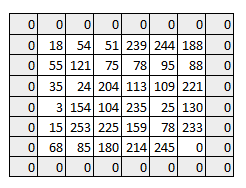
The weight matrix behaves like a filter in an image extracting particular information from the original image matrix. A weight combination might be extracting edges, while another one might a particular color, while another one might just blur the unwanted noise.

### The concept of stride and padding



As we saw above, the filter or the weight matrix, was moving across the entire image moving **one** pixel at a time. We can define it like a hyper parameter, as to how we would want the weight matrix to move across the image. If the weight matrix moves 1 pixel at a time, we call it as a stride of 1. Let’s see how a stride of 2 would look like.

As you can see the size of image keeps on reducing as we increase the stride value. Padding the input image with zeros across it solves this problem for us. We can also add more than one layer of zeros around the image in case of higher stride values.

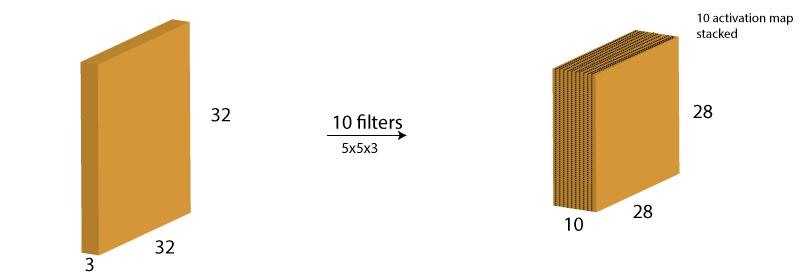


We can see how the initial shape of the image is retained after we padded the image with a zero. This is known as **same padding**since the output image has the same size as the input.

### Multiple filters and the activation map

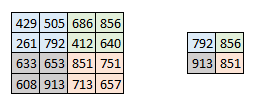
One thing to keep in mind is that the depth dimension of the weight would be same as the depth dimension of the input image. The weight extends to the entire depth of the input image. Therefore, convolution with a single weight matrix would result into a convolved output with a single depth dimension. In most cases instead of a single filter(weight matrix), we have multiple filters of the same dimensions applied together.

The output from the each filter is stacked together forming the depth dimension of the convolved image. Suppose we have an input image of size 32\*32\*3. And we apply 10 filters of size 5\*5\*3 with valid padding. The output would have the dimensions as 28\*28\*10.



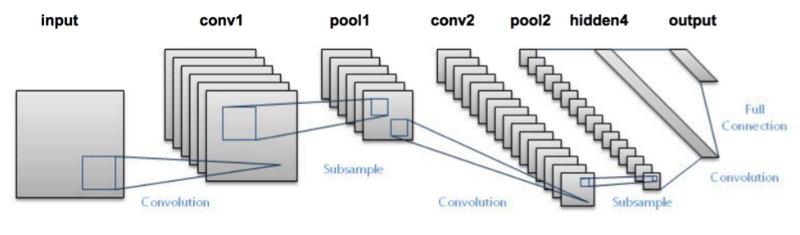
### The Pooling Layer

Sometimes when the images are too large, we would need to reduce the number of trainable parameters. It is then desired to periodically introduce pooling layers between subsequent convolution layers. Pooling is done for the sole purpose of reducing the spatial size of the image. Pooling is done independently on each depth dimension, therefore the depth of the image remains unchanged. The most common form of pooling layer generally applied is the max pooling.



Here we have taken stride as 2, while pooling size also as 2. The max operation is applied to each depth dimension of the convolved output. As you can see, the 4\*4 convolved output has become 2\*2 after the max pooling operation.

Putting it all together – How does the entire network look like



we are going to use Keras deep learning library in python to build our CNN (Convolutional Neural Network).

**Chapter 4**

**Flowchart**

**4.1: Implementation process:**

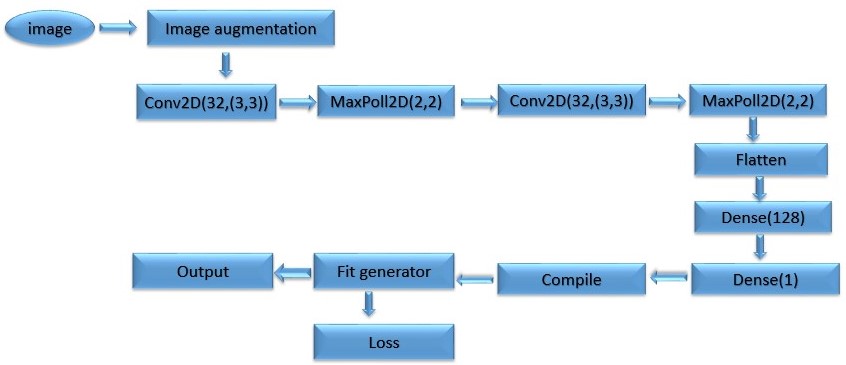
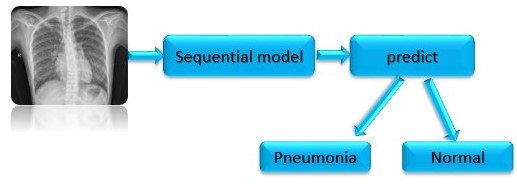
****

Fig 4.1: Training image of the system

This flowchart indicates model of the system. It takes an image and convoluted the image 32 filters where filter size 3\*3. After Convolution, we use maxpooling to reduce image size. A fully connected network would take this image as an array by flattening it. Then, we compile the image and fit the image in our model.



x-ray image

Fig 4.2: test image

First, an image is taken to check this normal or pneumonia. First of all, image is converted into our sequential model. Then, Prediction is done to check it pneumonia or normal

# **Chapter 5**

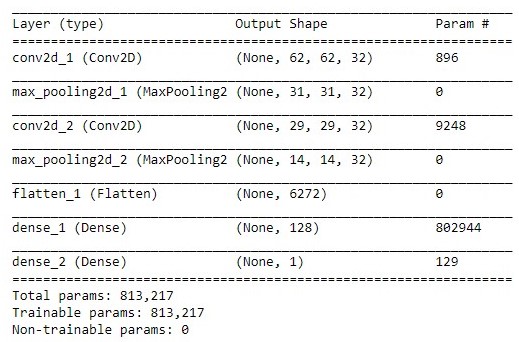
## Implementation and Experimental Result

**5.1: Convolutional Neural architecture:**

* we will define the Convolutional neural networks architecture as follows:
* The first hidden layer is a convolutional layer called a Convolution2D. We will use 32 filters with size 3×3 each.
* Then a Max pooling layer with a pool size of 2×2.
* Another convolutional layer with 32 filters with size 3×3 each.
* Then a Max pooling layer with a pool size of 2×2.
* Then next is a Flatten layer that converts the 2D matrix data to a 1D vector before building the fully connected layers.
* After that we will use a fully connected layer with 128 neurons and relu activation function.
* Finally, the output layer which has 1 neuron for the 1 classes and a sigmoid activation function to output probability-like predictions for each class.

**5.2: Model Summary:**

**Table 5.2.1:**

****

**5.3: Train Result:**

Train the image in jupyter to use keras.Kerasis a high level API built on Tensorflow which is used to build a simple convolutional neural network. The test image holds the image that needs to be tested on the CNN. Once we have the test image, we will prepare the image to be sent into the model by converting its resolution Then we are using on our classifier object to get the prediction. As the prediction will be in a binary form, we will be receiving either a 1 or 0, which will represent pneumonia or normal respectively.

**Validation accuracy of this Model after train:** 95.69%

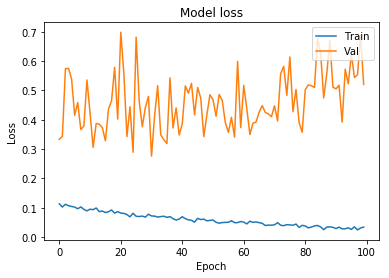


Fig 5.3.1: Model loss vs iteration

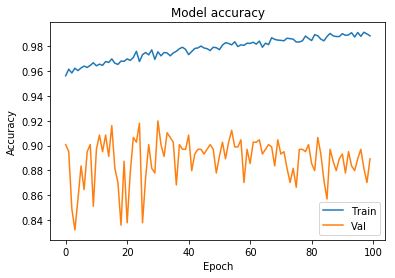


Fig 5.3.2: Model accuracy vs iteration

**5.4: Test Accuracy:**

Test accuracy is calculated using y-pred and y\_real array data. The data generate compare with normal and pneumonia. Here Let normal is 0 and Pneumonia is 1. This data is store in array and using c++ programming language we generate confusion matrix data. The result is calculate using online confusion matrix calculator.

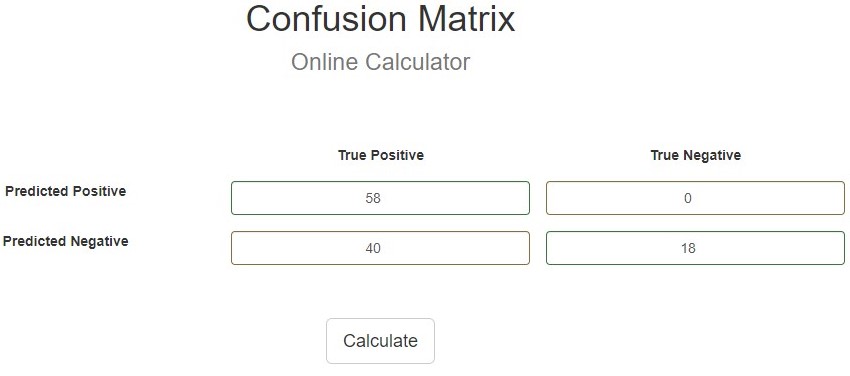
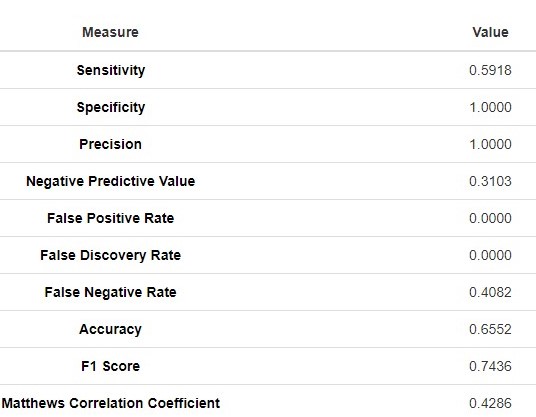


Fig 5.4.1: Confusion matrix

**Table 5.4.1: Evaluation of Confusion matrix:**



**Chapter 6**

**Literature overview of CAE**

**6.1 Convolutional auto encoder**

Convolutional AutoEncoders (CAEs) approach the filter definition task from a different perspective: instead of manually engineer convolutional filters we let the model learn the optimal filters that minimize the reconstruction error. These filters can then be used in any other computer vision task.

CAEs are the state-of-art tools for unsupervised learning of convolutional filters. Once these filters have been learned, they can be applied to any input in order to extract features. These features, then, can be used to do any task that requires a compact representation of the input, like classification.

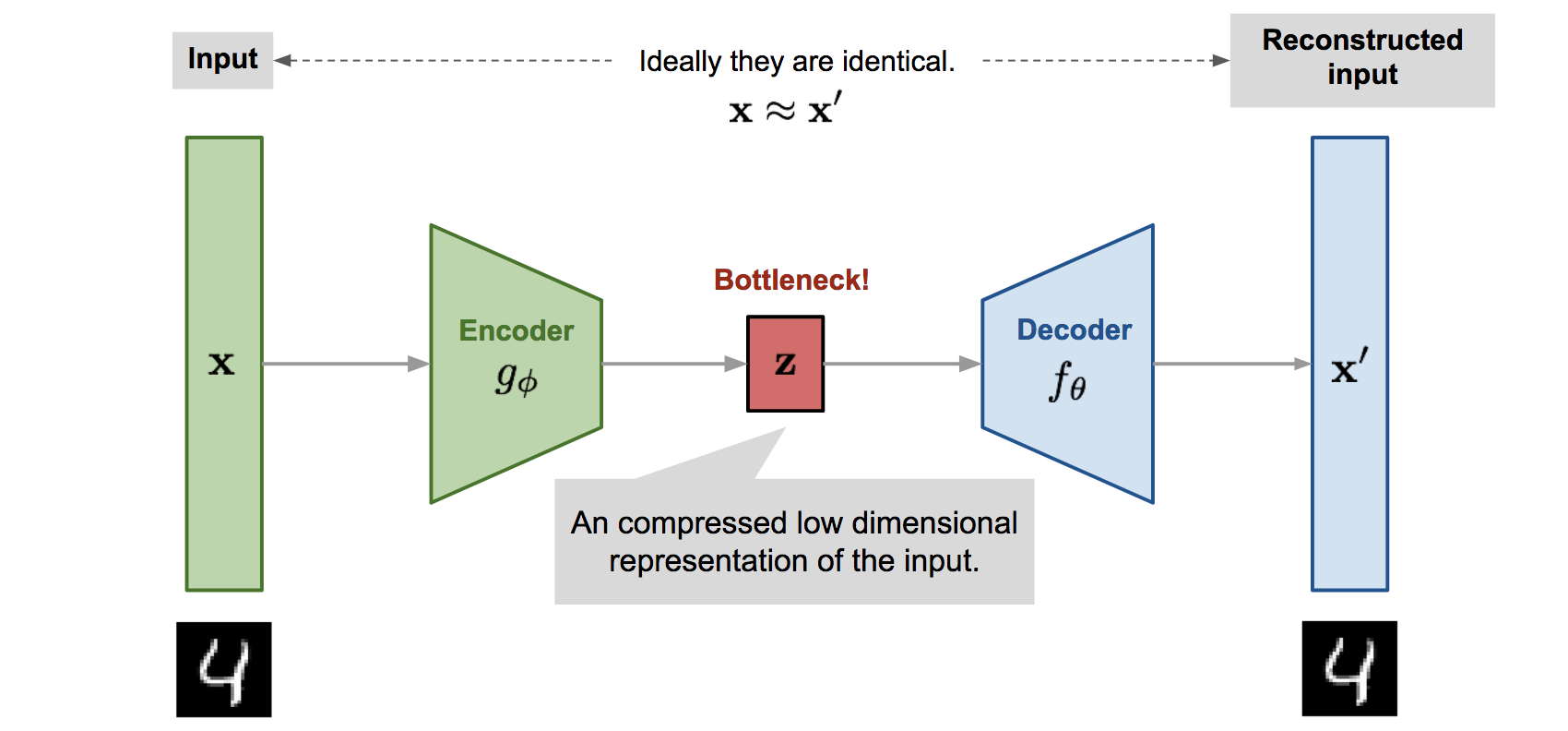
****

Fig 6.1: Sample image of an Autoencoder

**6.2: Why the Convolutional Autoencoders are Preferred for Image Data?**

The Convolution Autoencoders keep the spatial information of the input image data as they are, and extract information gently in what is called the **Convolution layer**. Figure (6.1) demonstrates that a flat 2D image is extracted to a thick square (Conv1), then continues to become a long cubic (Conv2) and another longer cubic (Conv3). This process is designed to retain the spatial relationships in the data. This is the encoding process in an Autoencoder. In the middle, there is a fully connected Autoencoder whose hidden layer is composed of only 10 neurons. After that comes with the decoding process that flattens the cubic, then to a 2D flat image. The encoder and the decoder are symmetric in Figure (6.1). They do not need to be symmetric, but most practitioners just adopt this rule as explained in “[Anomaly Detection with Autoencoders made easy](https://towardsdatascience.com/anomaly-detection-with-autoencoder-b4cdce4866a6)”.

**6.3: How Does the Convolutional Autoencoders Work?**

**6.3.1: Encoder**

this part of the network compresses or down samples the input into a fewer number of bits. The space represented by these fewer number of bits is often called the *latent-*space or bottleneck. The bottleneck is also called the "maximum point of compression" since at this point the input is compressed the maximum. These compressed bits that represent the original input are together called an “encoding” of the input.

 a convolution among an input volume I={I1,⋯,ID}I={I1,⋯,ID} and a set of n convolutional filters {F(1)1,⋯,F(1)n}{F1(1),⋯,Fn(1)}, each with depth DD, produces a set of n activation maps, or equivalently, a volume of activations maps whith depth n:

Om(i,j)=a(∑d=1D∑u=−2k−12k+1∑v=−2k−12k+1F(1)md(u,v)Id(i−u,j−v))m=1,⋯,n

**6.3.2: Decoder**

this part of the network tries to reconstruct the input using only the encoding of the input. When the decoder is able to reconstruct the input exactly as it was fed to the encoder, you can say that the encoder is able to produce the best encodings for the input with which the decoder is able to reconstruct well!

the reconstructed image I~I~ is the result of the convolution between the volume of feature maps Z={zi=1}nZ={zi=1}n and this convolutional filters volume F(2)F(2).

I~=a(Z∗F(2)m+b(2))I~=a(Z∗Fm(2)+b(2))

Padding II with the previously found amount of zeros, leads the decoding convolution to produce a volume with dimensions:

Ow=Oh=(Iw+(2k+1)−1)−(2k+1)+1=Iw=Ih

**Chapter 7**

## Image Noising and Denoising process

**7.1: Implementation:**

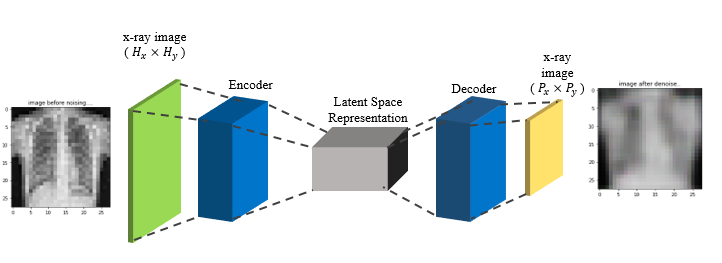
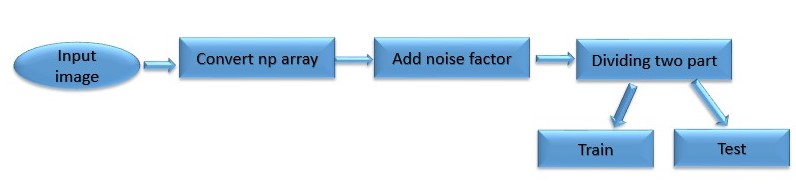


Fig 7.1: process of encoding to decoding

**7.2: Work flow:**

****

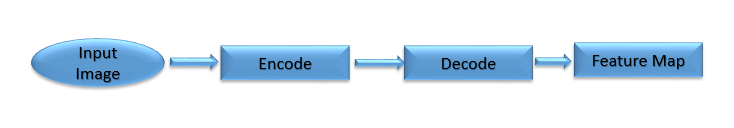
****

Fig 7.2: Noising and Denosing process

**7.3: Model Description:**

In Figure (7.1) there are three layers labeled Conv1, Conv2, and Conv3 in the encoding part. So we will build accordingly.

1. The code below input\_img = Input(shape=(28,28,1) declares the input 2D image is 28 by 28.
2. Then it builds the three layers Conv1, Conv2 and Conv3.
3. Notice that Conv1 is inside of Conv2 and Conv2 is inside of Conv3.
4. The padding specifies what to do when the filter does not fit the input image well. padding='valid' means dropping the part of the image when the filter does not fit; padding='same' pads the picture with zeros to fit the picture.

Use reverse technique for Denoise the image

**7.4: Model Evaluation**

**Before Noising:**

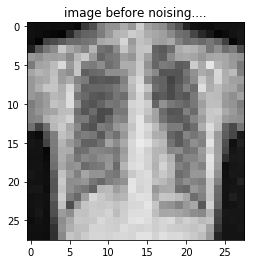


Fig 7.4.1: Before add noise

**Adding 0.5 noise factor with image:**

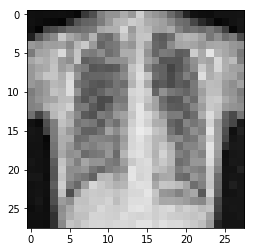
****

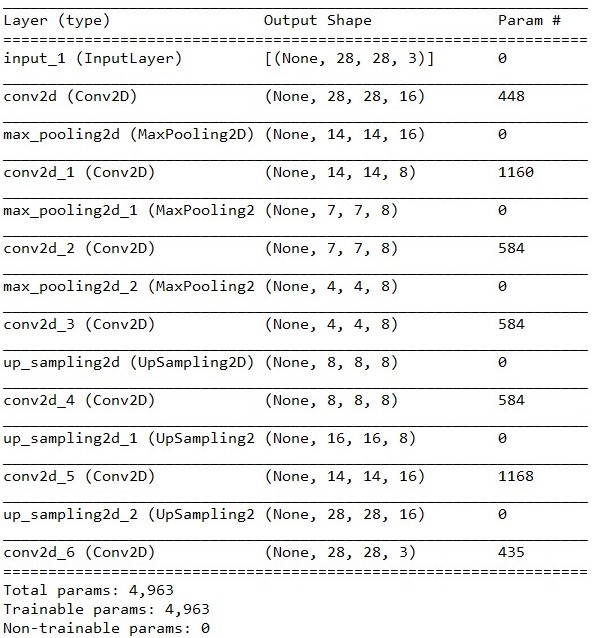
Fig 7.4.2: After Noise

**Denoise the Image:**



Fig 7.4.3: After Denoise

**Table 7.1: Summary of Convolutional Autoencoder model is given below:**

****

**Chapter 8**

**Conclusion and Future Work**

**8.1 Conclusion:**

In this paper, we have demonstrated the feasibility of deploying deep learning techniques for the task of identifying pneumonia from x-ray image using the Keras deep learning library in python to build our CNN. We noted how deep convolutional neural networks can work as feature extractors and thus no image processing techniques need to be applied to manually find the presence of security features in an image. Although the data-set did not represent the real-world scenario of pneumonia diseases and not helpful for real time, it was helpful for experiments. Under the availability of a data-set, the deep neural networks can be better trained.

**8.2 Limitation**

* Here, we do not have real time image from diagnostic center we collect from kaggle
* We just collect normal and pneumonia image not real time data.
* We cannot test our model with real time scenario
* Noisy image cannot fully detect.

**8.3 Future Works:**

We are interested to involve more features to detect pneumonia diseases and also extend the support for all kinds of scenario. Besides such a model may then be built into a software and can thus help doctor in detecting pneumonia in case of suspicion in real time with just an image taken through the x-ray image. In a machine with a with a high-end processor, initialization of the software takes anywhere between 2 to 10 seconds and classification time is around 0.25 to 0.3 seconds per image. Future avenues of research include examining various deep neural network architectures which are more efficient in terms of time and space complexity.

**REFERENCES**

1. Medical image denoising using convolutional denoising autoencoders

Lovedeep Gondara dept. of CSE Simon Fraser University

1. <https://medium.com/intuitive-deep-learning/build-your-first-convolutional-neural-network-to-recognize-images-84b9c78fe0ce>
2. <https://becominghuman.ai/detecting-pneumonia-with-deep-learning-3cf49b640c14>
3. <https://www.analyticsvidhya.com/blog/2017/06/architecture-of-convolutional-neural-networks-simplified-demystified/>
4. <https://pgaleone.eu/neural-networks/2016/11/24/convolutional-autoencoders/>
5. <https://www.analyticsvidhya.com/blog/2017/05/neural-network-from-scratch-in-python-and-r/>
6. <https://www.coursera.org/specializations/deep-learning>
7. <https://blog.keras.io/building-autoencoders-in-keras.html>
8. <https://www.cedars-sinai.org/health-library/diseases-and-conditions---pediatrics/p/pneumonia-in-children.html>
9. <https://www.cnbc.com/2018/02/22/medical-errors-third-leading-cause-of-death-in-america.html>