

# Project Title:

# Convolutional Neural Network - Pneumonia Prediction Using X-Ray Images

**Submitted To:**

Ma’am Mehreen Sirshar

**Submitted By:**

Mahrukh Shakoor-033

Saima Nasir-021

**Subject:**

Digital Image Processing

**Section:**

BSE-6A

**Date:**

15-05-2020

# Convolutional Neural Network - Pneumonia Prediction Using X-Ray Images

**Convolutional neural network (CNN)**

In [deep learning](https://en.wikipedia.org/wiki/Deep_learning), a convolutional neural network (CNN) is a class of [deep neural networks](https://en.wikipedia.org/wiki/Deep_neural_network), most commonly applied to analyzing visual imagery. The name convolutional neural network indicates that the network employs a mathematical operation called [convolution](https://en.wikipedia.org/wiki/Convolution). Convolution is a specialized kind of linear operation. Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers.

A convolutional neural network consists of an input and an output layer, as well as multiple [hidden layers](https://en.wikipedia.org/wiki/Multilayer_perceptron#Layers). The hidden layers of a CNN typically consist of a series of convolutional layers that convolve with a multiplication or other [dot product](https://en.wikipedia.org/wiki/Dot_product). The activation function is commonly a [RELU layer](https://en.wikipedia.org/wiki/Rectifier_(neural_networks)), and is subsequently followed by additional convolutions such as pooling layers, fully connected layers and normalization layers, referred to as hidden layers because their inputs and outputs are masked by the activation function and final [convolution](https://en.wikipedia.org/wiki/Convolution).

**Convolutional**

When programming a CNN, the input is a [tensor](https://en.wikipedia.org/wiki/Tensor) with shape (number of images) x (image width) x (image height) x ([image depth](https://en.wikipedia.org/wiki/Image_depth)). Then after passing through a convolutional layer, the image becomes abstracted to a feature map, with shape (number of images) x (feature map width) x (feature map height) x (feature map channels). A convolutional layer within a neural network should have the following attributes:

* Convolutional kernels defined by a width and height (hyper-parameters).
* The number of input channels and output channels (hyper-parameter).
* The depth of the Convolution filter (the input channels) must be equal to the number channels (depth) of the input feature map.

Convolutional layers convolve the input and pass its result to the next layer. This is similar to the response of a neuron in the visual cortex to a specific stimulus. Each convolutional neuron processes data only for its [receptive field](https://en.wikipedia.org/wiki/Receptive_field). Although [fully connected feed forward neural networks](https://en.wikipedia.org/wiki/Multilayer_perceptron) can be used to learn features as well as classify data, it is not practical to apply this architecture to images. A very high number of neurons would be necessary, even in shallow architecture, due to the very large input sizes associated with images, where each pixel is a relevant variable. For instance, a fully connected layer for a (small) image of size 100 x 100 has 10,000 weights for each neuron in the second layer.

The convolution operation brings a solution to this problem as it reduces the number of free parameters, allowing the network to be deeper with fewer parameters. For instance, regardless of image size, tiling regions of size 5 x 5, each with the same shared weights, requires only 25 learnable parameters. In this way, it resolves the vanishing or exploding gradients problem in training traditional multi-layer neural networks with many layers by using [back propagation](https://en.wikipedia.org/wiki/Backpropagation).

### Pooling

Convolutional networks may include local or global pooling layers to streamline the underlying computation. Pooling layers reduce the dimensions of the data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Local pooling combines small clusters, typically 2 x 2. Global pooling acts on all the neurons of the convolutional layer. In addition, pooling may compute a max or an average. Max pooling uses the maximum value from each of a cluster of neurons at the prior layer.Average pooling uses the average value from each of a cluster of neurons at the prior layer.

### Fully connected

Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional [multi-layer perceptron](https://en.wikipedia.org/wiki/Multi-layer_perceptron) neural network (MLP). The flattened matrix goes through a fully connected layer to classify the images.

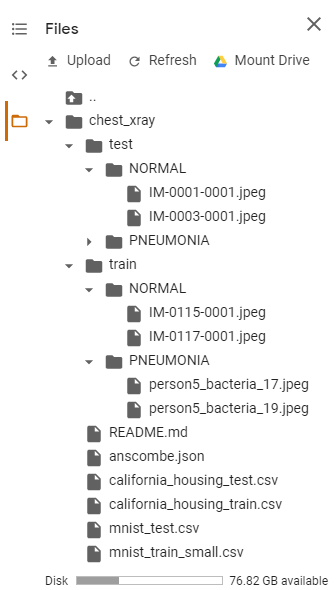
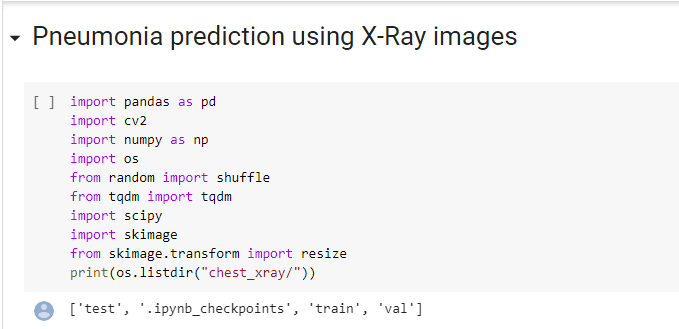
### Receptive field

In neural networks, each neuron receives input from some number of locations in the previous layer. In a fully connected layer, each neuron receives input from every element of the previous layer. In a convolutional layer, neurons receive input from only a restricted subarea of the previous layer. Typically the subarea is of a square shape (e.g., size 5 by 5). The input area of a neuron is called its receptivefield. So, in a fully connected layer, the receptive field is the entire previous layer. In a convolutional layer, the receptive area is smaller than the entire previous layer.

**Coding: Pneumonia Prediction Using X-Ray Images through CNN**

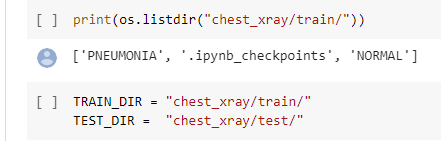
We will be using to create a convolutional neural network with two convolutional layers and one fully connected layer to predict pneumonia, given a chest X-ray image as the input.

**Steps:**

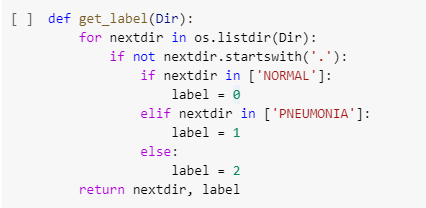
1. Upload the dataset in googlecolab. The dataset has been split into train and test folders. Additionally, the images within these folders have been split into Normal and Pneumonia subfolders.
2. **Starting by importing the different libraries**.

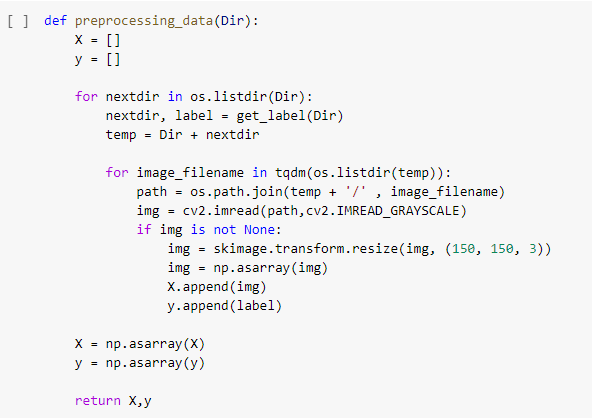
# Visualize our image data and split it into training and testing sets

The dataset has been split into 'train' and 'test'' folders. Additionally, the images within these folders have been split into 'NORMAL' and 'PNEUMONIA' subfolders. First, we will go through the folders of the dataset and append each image into an array. Both the training and testing set images will be placed into the same matrices for now later to be randomly split into our training and test sets.

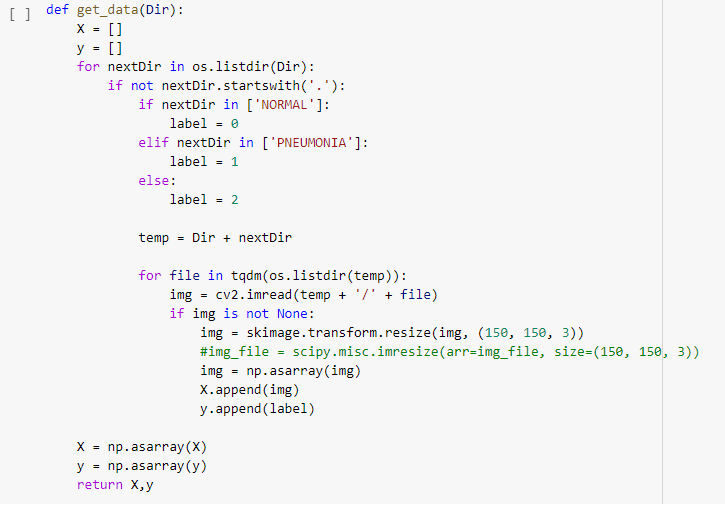


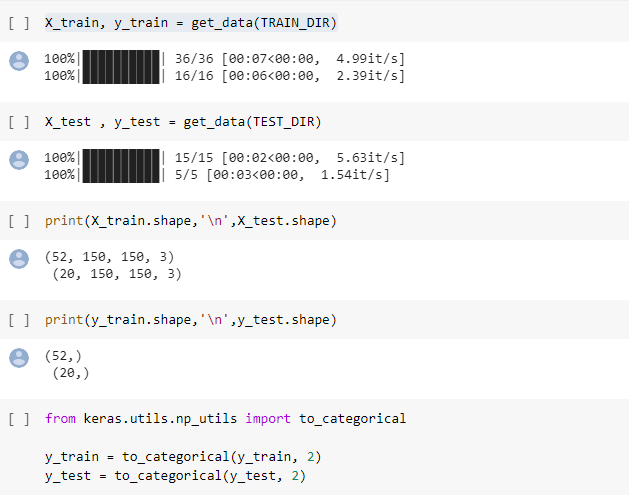
1. **Preprocessing steps**



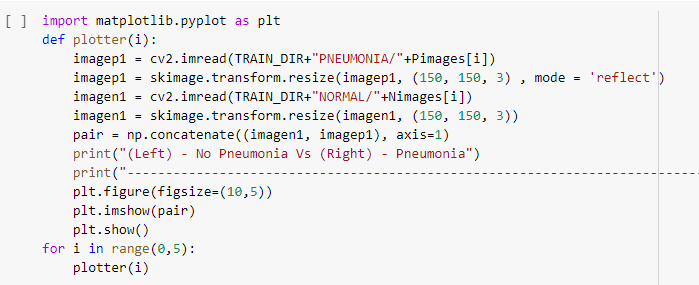


1. **X\_train , y\_train = preprocessing\_data**

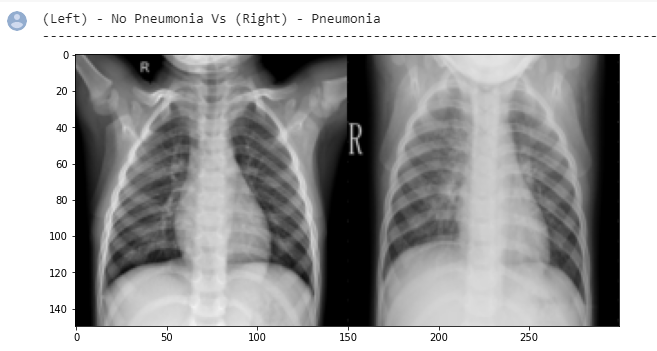
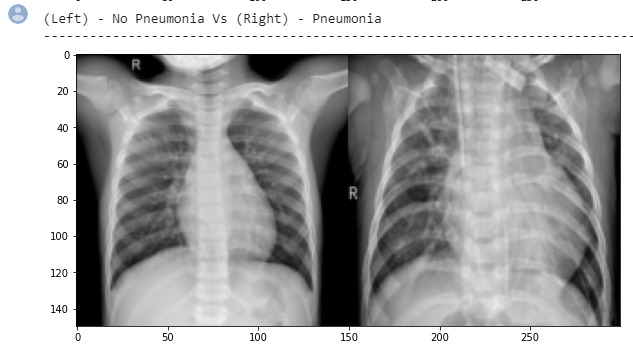


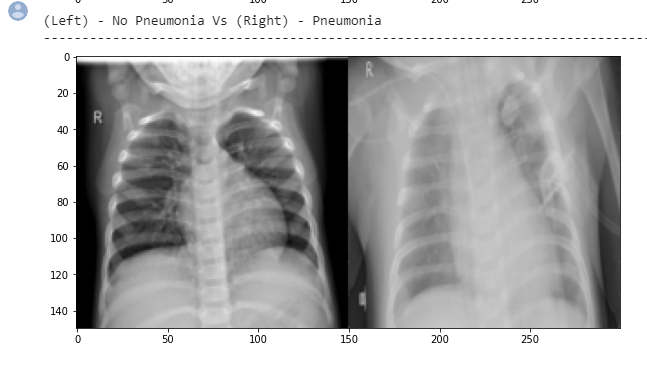


1. **Implemantion in matlab:**

In this coding firstly display the images in train data,than concatenate and resize both images and automatic detect what image is pneumonia or not.

**Output**: Image should be display x-axis and y-axis because both images concatenate and resize and divide both image in 150-150 in x-axis.

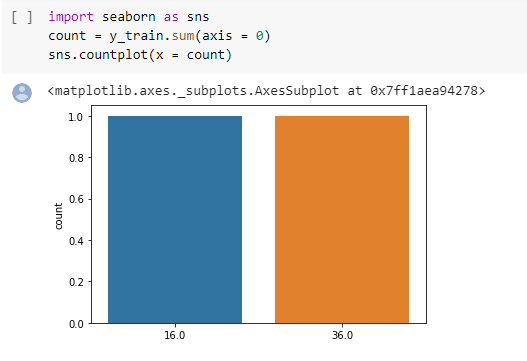
Left side image have detect No Pneumonia and right side image have detect pneumonia.



1. **Data visualization**

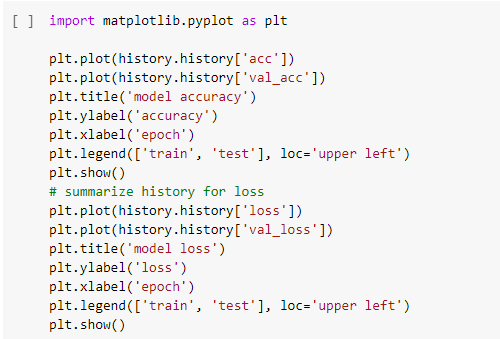
Let's visualize our data before we split it into our training and test sets in the following steps:

* Observe an example of a positive and negative pneumonia diagnosed X-ray to see what our algorithm is trying to detect
* Count the number of pneumonia positive and negative results in our data set (i.e. our data's distribution)
* Create a bar graph of the positive and negative count as a simple visualization of our distribution
* This is essentially what we would like our algorithm to detect.



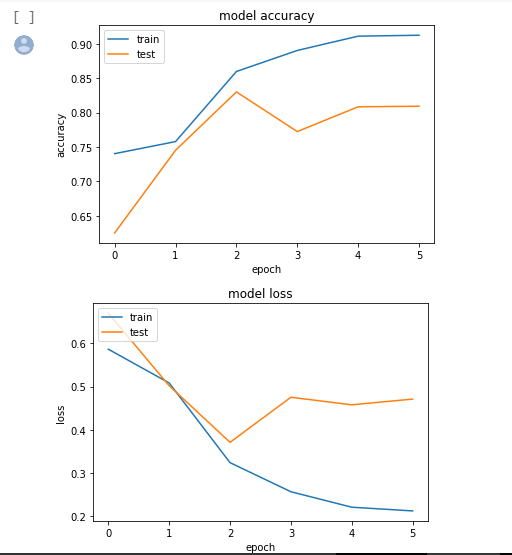
# Define our model:

In this coding training and testing images plot the graph and detect the accuracy and loss.



**Output**: According to this graph accuracy show both training and testing images

* 1. Training 0.74% accuracy start and time to time accuracy will be increased
  2. Testing accuracy start in 0% , but according to different condition accuracy should be flip increasing and decreasing



# 

# Analyzing our results

Our results are looking good. In addition to our overall excellent training and test prediction accuracy, both our positive and negative predictions independently have beyond acceptable accuracy. We have created a convolutional neural network that can be used to predict pneumonia from X-ray images.