P&H(Pneumonia and Heart) Analysis through deep learning

A Major Project Report

in partial fulfillment of the degree

Bachelor of Technology
in
Computer Science & Engineering

By

19K41A0506	Vontela Dhanush
19K41A0590	Amogh Varsh Raju Ambati
19K41A05E3	Sardar Kamaljeeth Singh
19K41A05G3	Mohd Amaan
19K41A05H2	T.Vinay Prakash

Under the Guidance of

Mr.D.RAMESH
Asst.Prof Dept of CSE

Submitted to

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
S. R. ENGINEERING COLLEGE(A), ANANTHASAGAR, WARANGAL
(Affiliated to JNTUH, Accredited by NBA)

November - 2022

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Major Project Report entitled "P&H(Pnemonia and Heart Analysis) through deep learning" is a record of bonafide work carried out by the student(s) Vontela Dhanush ,A. Amogh Varsh Raju ,Sardar Kamaljeeth Singh ,Mohd Amaan ,T. Vinay prakash ,bearing Roll No(s) 19K41A0506, 19K41A0590, 19K41A05E3, 19K41A05G3, 19K41A05H2 during the academic year 2022-23 in partial fulfillment of the award of the degree of ***Bachelor of Technology*** in **Computer Science & Engineering** by the Jawaharlal Nehru Technological University, Hyderabad.

Supervisor **Head of the Department**

External Examiner

ACKNOWLEDGMENT

We owe an enormous debt of gratitude to our project guide **Dr. D. RAMESH, Assistant Professor,** as well as Head of the CSE Department **Dr. Sheshikala, Associate Professor** for guiding us from the beginning through the end of the major project with their intellectual advice and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We express our thanks to project coordinators **Mr. Sallauddin Md, Asst. Prof.,and Y.Chanti , Asst. Prof.** for their encouragement and support.

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved principal, **Dr. V. Mahesh,** for his continuous support and guidance to complete this project in the institute.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

ABSTRACT

Pneumonia is a respiratory infection caused by bacteria or viruses. It affects many people. Developing and underdeveloped countries are particularly affected, where high levels of environmental pollution, unsanitary living conditions, overcrowding and inadequate healthcare infrastructure are relatively common. Pneumonia causes pleural effusion, a condition in which fluid fills the lungs and causes difficulty breathing. Early diagnosis of pneumonia is critical to ensure curative treatment and improve survival. A chest x-ray is the most common method used to diagnose pneumonia. However, examination of chest radiography is a difficult task and subject to subjective variability. In this study, we are developing a computer-aided diagnostic system for automatic detection of pneumonia based on chest X-rays. Addressing the scarcity of available data using deep transfer learning, we designed an ensemble of four of convolutional neural network models:

Custom CNN,VGG-16, ResNet-20, and DenseNet-121. A weighted average ensemble approach was adopted and the weights assigned to the base learners were determined using a new approach. The values of four standard metrics of precision, recall, f1 score, and area under the curve are merged into a weight vector. The heart analysis also plays an important role cause the affects through these kind of diseases can intern lead to heart diseases so monitoring results on the heart also makes it a very useful result to be analyzed ,where the adverse affects are caused due to the deprived oxygen availability due to the infection in lung leading to these consequences.

Table of Contents

Table of figures

1. INTRODUCTION

1.1 Overview:

Pneumonia is acute pneumonia caused by bacteria, viruses, or fungi that infect the lungs, causing inflammation of the air sacs and fluid filling of the lungs, pleural effusion. More than 15% of his deaths in children under the age of 5 are due to this disease. Pneumonia is most common in underdeveloped and developing countries where overpopulation, pollution, and unsanitary environmental conditions exacerbate the situation and lack medical resources. Therefore, early diagnosis and treatment can play an important role in preventing the disease from becoming fatal.Computer tomography (CT), magnetic resonance imaging (MRI), or radiography. Radiological examination of the lungs using (x-rays) is often used for diagnosis. X-ray imaging is a non-invasive and relatively inexpensive examination of the lungs. Figure 1 shows examples of X-rays of a lung and a healthy lung. A white spot (marked by a red arrow) on a chest x-ray called an infiltrate distinguishes pneumonia from a healthy condition. However, chest radiographs for detecting pneumonia are prone to subjective variability. Therefore, we need an automated system to detect pneumonia. In this study, we developed a computer-aided diagnosis (CAD) system that uses an ensemble of deep transfer learning models for accurate classification of chest radiographs.

Deep learning is an important artificial intelligence tool that plays a key role in solving many complex computer vision problems. Deep learning models, especially convolutional neural networks (CNNs), are widely used for various image classification problems. However, such models work best only when presented with large amounts of data. The X-ray image analysis problem requires an experienced physician to classify each image, which is costly and time consuming, making it difficult to collect such large amounts of labeled data. Transfer learning is a workaround to overcome this obstacle. In this technique, a model trained on a large data set is reused and the network weights determined by that model are applied to solve problems associated with small data sets. CNN models trained on large datasets like ImageNet consisting of over 14 million images are widely used for Chest X-Ray image analysis tasks.

(A)

(B)

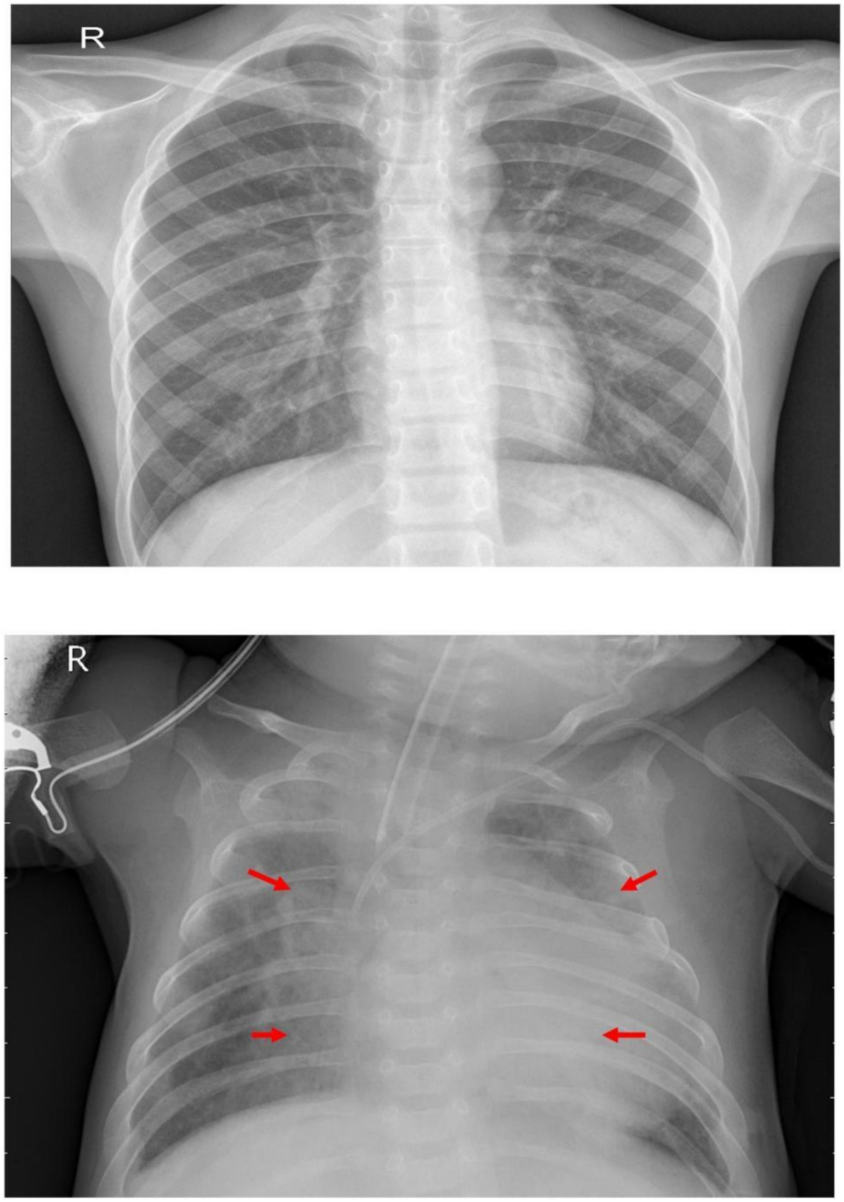


Figure 1: **Examples of two X-ray plates that display (a) a healthy lung and (b) a pneumonia lung.**

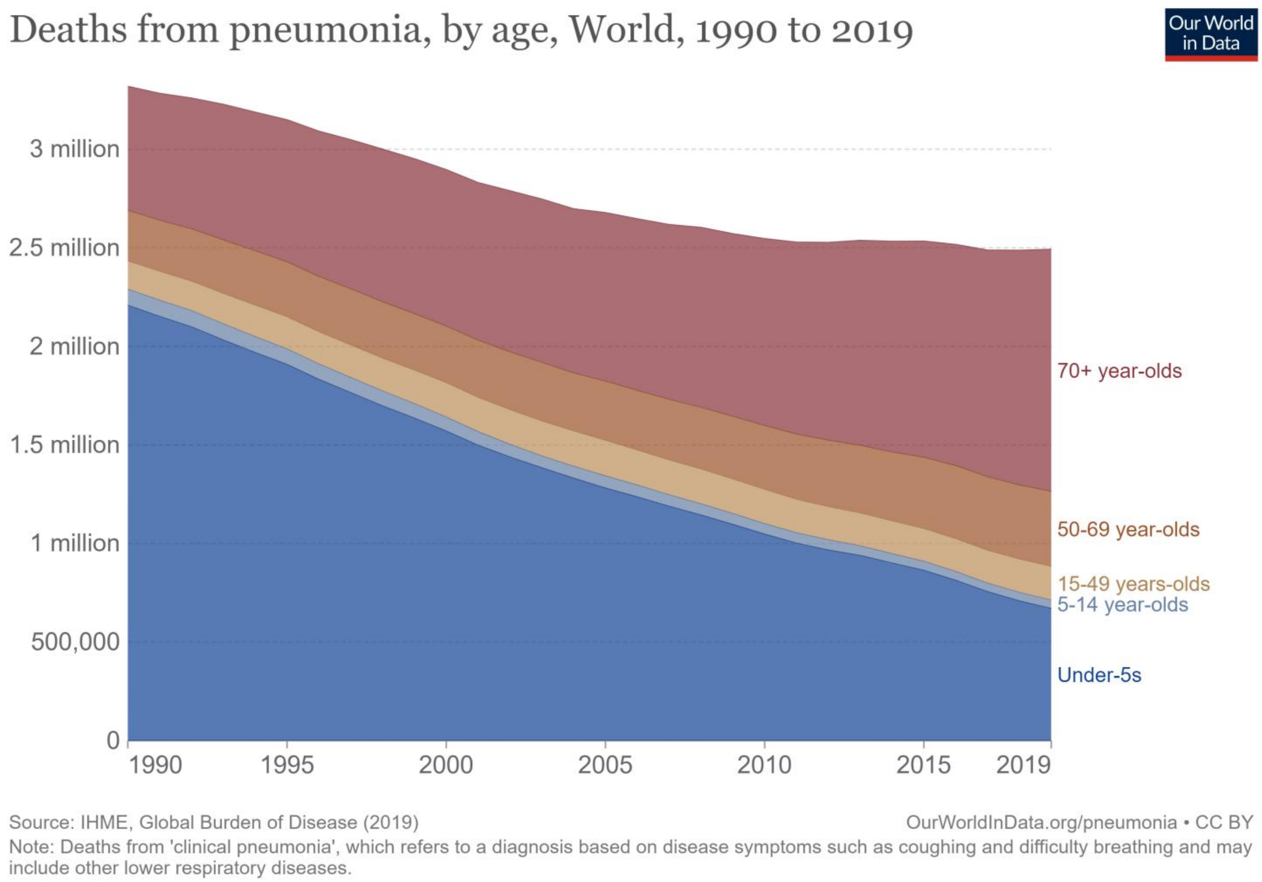


Figure 2:Deaths from pneumonia, by age

1.2: Existing methods

There are several existing methods used in various fields in biomedical field, surgical field, Research sector and much more , the main existing methods involved training through mainly

ResNet-34: First, there is a convolution layer containing 64 filters with kernel size 7×7, this is the first convolution, followed by a max pooling layer. I specified a stride length of 2 in both cases. The following conv2_x has a pooling layer and the following convolutional layers.

ANN,CNN: Various types of neural networks in deep learning, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and artificial neural networks (ANNs), are changing the way we interact with the world.

VGG19+CNN: VGG-19 is a 19-layer deep convolutional neural network. You can load a pretrained version of a network trained with over 1 million images from the ImageNet database. A pretrained network can classify images into 1000 object categories, such as keyboards, mice, pens, and many animals.

GoogLeNet: Google Net (or Inception V1) was proposed in 2014 at Google (in collaboration with various universities) in a research paper "Going Deeper with Convolutions". This architecture won the ILSVRC 2014 Image Classification Challenge. Significantly lower error rate compared to previous winners AlexNet (2012 ILSVRC winner) and his ZF-Net (2013 ILSVRC winner) and significantly lower than VGG (2014 runner-up) was the error rate. This architecture uses techniques such as 1 × 1 convolution and global average pooling in the middle of the architecture.

ResNet-18: ResNet-18 is a convolutional neural network with a depth of 18 layers. You can load a pretrained version of the network trained with over 1 million images from the ImageNet database . A pretrained network can classify images into 1000 object categories, such as keyboards, mice, pens, and many animals.

CAD4TBv6:CAD4TB is a CE-certified AI software that supports cost-effective, fast, easy, and accurate automated tuberculosis detection

DenseNet:A DenseNet is a type of convolutional neural network that uses dense connections between layers through dense blocks. Here, we directly connect all layers (with matching feature map sizes).

ResNet152: ResNet is one of the most powerful deep neural networks which has achieved fantabulous performance results in the ILSVRC 2015 classification challenge.

EfcientNet-B7: It is a convolutional neural network architecture and scaling method that uniformly scales all dimensions of depth/width/resolution using a compound coefficient

DeepLabv3: DeepLabv3 is a semantic segmentation architecture that improves upon DeepLabv2 with several modifications. To handle the problem of segmenting objects at multiple scales, modules are designed which employ atrous convolution in cascade or in parallel to capture multi-scale context by adopting multiple atrous rates.

XGB-linear: XGBoost Linear is an advanced implementation of a gradient boosting algorithm with a linear model as the base model. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier.

2. LITERATURE SURVEY

There are several solutions and approaches designed and are achieved but most of the projects, proposals have bee proven to be either ineffective, only classification based outputs with best score, better analysis but with low score etc... , the main draw back of the following projects was the challenges faced for managing the given data and it is difficult for the image to be completely balanced and is farther more complicated than imagined.

The number of collected papers are 25 and are quality papers which clearly mention whole process of how did they approach the solution and get the best results and challenges faced. The papers also portray different kinds of data-sets being used , the results perspective shows a lot of difference from one to one .

There are certain papers which show only whether the given input x-ray image consists of a disease or not , some other projcts give x-ray image as output but with minimal information for example: Heat map output highlighting region of disease affected, marked locations of tumors etc.

The detection of pneumonia using chest radiographs has been an open problem for many years, with the main limitation being the paucity of published data. Traditional machine learning methods have been extensively studied. Chandra et al. Segmenting lung regions from chest X-rays, from these regions he extracted eight statistical features and used them to classify them. They implemented his five conventional classifiers:

Multilayer Perceptron (MLP), Random Forest, Sequential Minimal Optimization (SMO), Classification by Regression, and Logistic Regression. They evaluated the method on 412 images and achieved a curacy rate of 95.39 using the MLP classifier. Kuo et al. used 11 features to detect pneumonia in 185 schizophrenic patients. We applied these features to various regression and classification models, such as decision trees, support vector machines, and logistic regression, and compared the results of the models. They achieved the highest accuracy of 94.5% using a decision tree classifier. Other models lagged far behind. Similarly, Yue et al. Six features were used to detect pneumonia on chest CT scan images of 52 patients. The highest achieved her AUC value was 97%. However, these methods cannot be generalized and have been evaluated on small data sets.

Title(System)	Model used	Technique(Approach)	Data set	Output metrics
Deep learning for chest X-ray analysis: A survey	CNN	image-level prediction (classification and regression), segmentation, localization, image generation and domain adaptation	Kaggle challenge	Finding the high accuracy using image-level prediction (classification and regression), segmentation, localization, image generation and domain adaptation
Application of deep learning techniques for detection of COVID-19 cases using chest X-ray images: A comprehensive study	ResNet-34	CNN	Kaggle challenge	98.33%
PNEUMONIA DETECTION USING CNN THROUGH CHEST X-RAY	ANN,CNN	CNN	Kaggle challenge	Architecture 5 works better
CHEST X-RAYS IMAGE CLASSIFICATION IN MEDICAL IMAGE ANALYSIS	ResNet-50	CNN	ChestX-Ray14	ResNet-50 achieved state-of-the-art results in four out of fourteen classes.
Deep-Pneumonia Framework Using Deep Learning Models Based on Chest X-Ray Images	ResNet152V2	CNN	Kaggle challenge	99.22%
Deep-chest: Multi-classification deep learning model for diagnosing COVID-19, pneumonia, and lung cancer chest diseases	VGG19+CNN	CNN	Kaggle challenge	98.05%
Pneumonia detection in chest X-ray images using an ensemble of	GoogLeNet, ResNet-18,	convolutional neural network	Kaggle challenge	98.81%

deep learning models	and DenseNet-121			
Chest x-ray analysis with deep learning-based software as a triage test for pulmonary tuberculosis: a prospective study of diagnostic accuracy for culture-confirmed disease	CAD4TBv6	Deep learning based software , regression model	Kaggle challenge	High accuracy than qXRv2
Identifying Pneumonia in Chest XRays: A Deep Learning A	(ResNet50 + ResNet101)	Mask-RCNN	Kaggle challenge	0.218051
Chest Radiograph Interpretation with Deep Learning Models: Assessment with Radiologist-adjudicated Reference Standards and Population-adjusted Evaluation	CNN	convolutional neural networks	Google cloud, kaggle	The model demonstrated population-adjusted areas under the receiver operating characteristic curve of 0.95 (pneumothorax), 0.72 (nodule or mass), 0.91 (opacity), and 0.86 (fracture)
Detection of tuberculosis patterns in digital photographs of chest X-ray images using Deep Learning: feasibility study	ANN, ViDi classification tool, ViDi detection tool	Deep Learning image analysis software (Suite v2.0; ViDi Systems, Villaz-Saint-Pierre, Switzerland	Kaggle challenge	area under the ROC curve 0.82 & AUC 0.98
Deep learning based detection and analysis of COVID-19 on chest X-ray images	Xception	CNN	Kaggle challenge	97.97%
A machine learning- based framework for diagnosis of COVID- 19 from chest X- ray images	CNN +PCA	CNN	Kaggle challenge	The proposed method achieved high accuracy of 100% using CNN +PCA when variance of 0.99 was used
Machine- learning classifcation of texture features of portable chest X- ray accurately classifes COVID- 19 lung infection	XGB-linear	DCNN	Kaggle challenge	100%
A deep-learning pipeline for the diagnosis and discrimination of	DeepLabv3	Semantic Segmentation	Kaggle challenge	DeepLabv3 outperformed

viral, non-viral and COVID-19 pneumonia from chest X-ray images				both FCN and U-Net
Deep learning for distinguishing normal versus abnormal chest radiographs and generalization to two unseen diseases tuberculosis and COVID- 19	EfcientNet-B7	CNN	NH	EfcientNet-B7 performs better than other advanced networks
Machine learning applied on chest x-ray can aid in the diagnosis of COVID-19: a first experience from Lombardy, Italy	Image analysis for deep learning classifier	CNN	Kagg le challenge	99%
Pneumonia Detection and Classification Using Deep Learning on Chest X-Ray Images	ResNet152	CNN	Kaggle challenge	97%
Predicting COVID-19 Pneumonia Severity on Chest X-ray With Deep Learning	DenseNet	CNN	ChestX-ray8 dataset from the NIH with labels from Google	The use of a score combining geographical extent and degree of opacity allows clinicians to compare CXR images with each other using a quantitative and objective measure.

Table 2.1

2.1 Comparison Table

3. DATA-SET:

The data set we have gathered manually is taken many sources and is arranged into 3 files with each specific to its feature perspective.The data set consists of chest-x-ray images.

The data set consists of 3 folders namely:

1. **Train**
2. **Test**
3. **Val**

1. **Train** : this folder further consists of two sub folders which are named as abnormal and normal which consists of seprate x-ray images of diseased and non-diseased respectively. The total number of images together are 5,218 images respectively.

2. **Test:** this folder also consists of two sub-folders with normal and ab-normal file names and also the number of x-rays in these folders together are 624 images

3. **Val:** This also consists of same sub folders but only 18 images in them we call them agumented data which are completed filtered data for stronger training of data

Fig: 3.1 Main folders

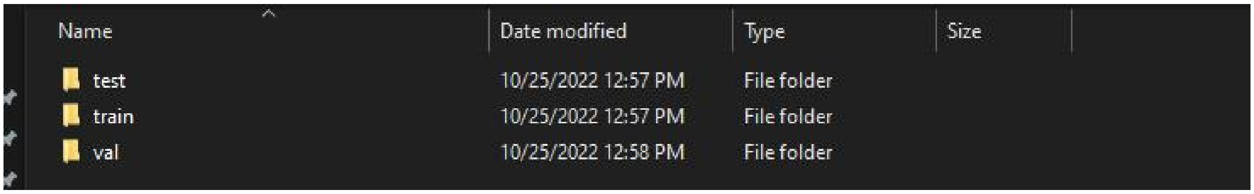


Fig 3.2 Sub folders

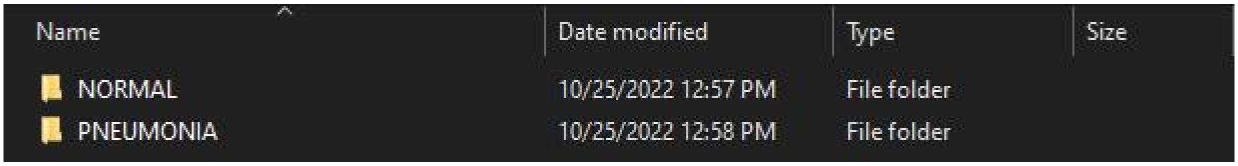
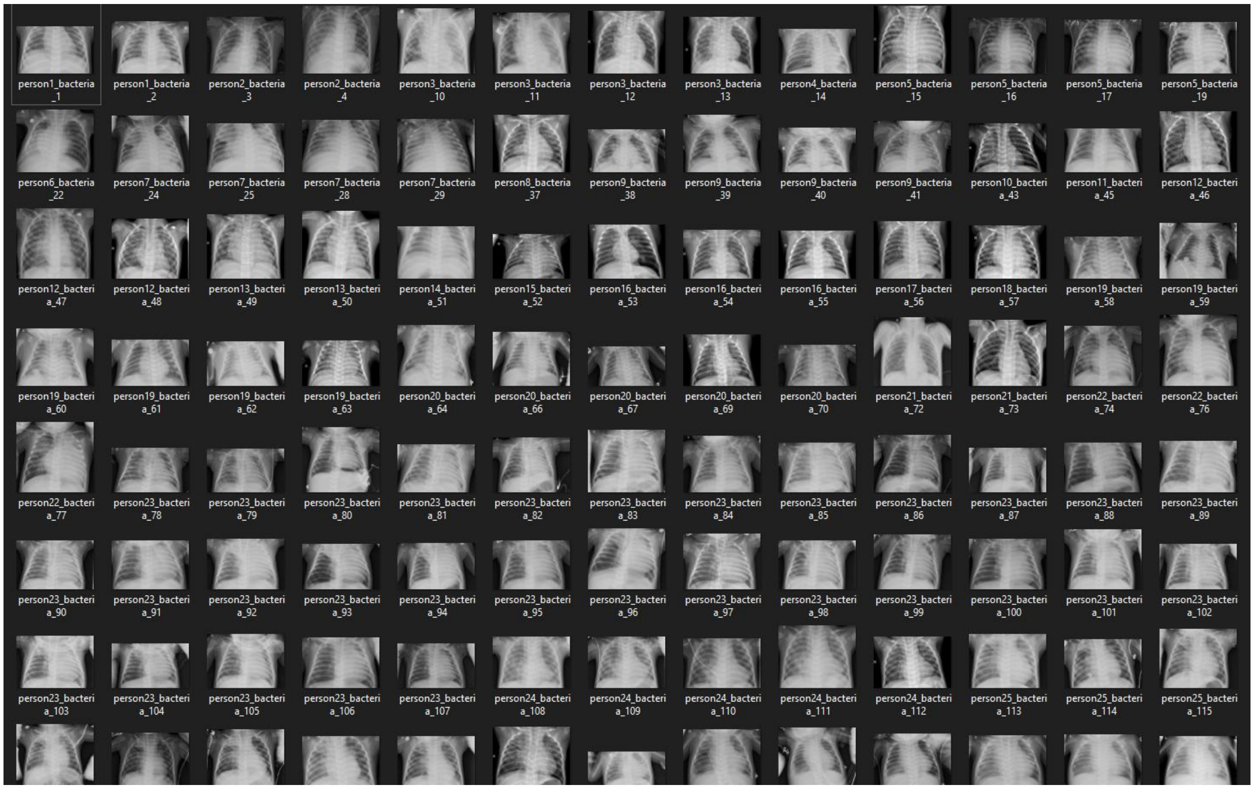


Fig 3.3: X-ray images from data set



4. Data Pre-Proccesing

There are two efficent models to perform the data pre-proccesing for the taken huge amount of data-set and It is worth noting that our dataset contains multiple images for each patient. This could be the case, for example, when a patient has taken multiple X-ray images at different times during their hospital visits. In our data splitting, we have ensured that the split is done on the patient level so that there is no data "leakage" between the train, validation, and test datasets.

The Two methods are:

1. **Checking Data Leakage:**
2. **Preparing images**
   a) **ImageDataGenerator**

1. Checking Data Leakage:

Data leakage refers to a mistake make by the creator of a machine learning model in which they accidentally share information between the test and training data-sets. Typically, when splitting a data-set into testing and training sets, the goal is to ensure that no data is shared between the two. This is because the test set's purpose is to simulate real-world, unseen data. However, when evaluating a model, we do have full access to both our train and test sets, so it is up to us to ensure that no data in the training set is present in the test set.

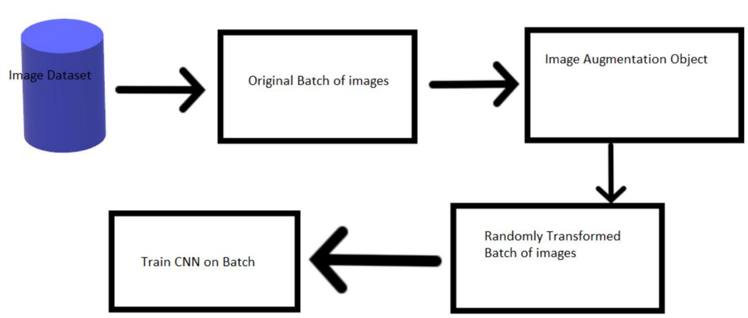
Data leakage often results in unrealistically-high levels of performance on the test set, because the model is being ran on data that it had already seen — in some capacity — in the training set. The model effectively memorizes the training set data, and is easily able to correctly output the labels/values for those test data-set examples. Clearly, this is not ideal, as it misleads the person evaluating the model. When such a model is then used on truly unseen data, performance will be much lower than expected.

2. ImageDataGenerator:

After setting up of the model we need to consume them for this we will use the off-the-shelf ImageDataGenerator class from the Keras framework, which allows us to build a "generator" for images specified in a dataframe.

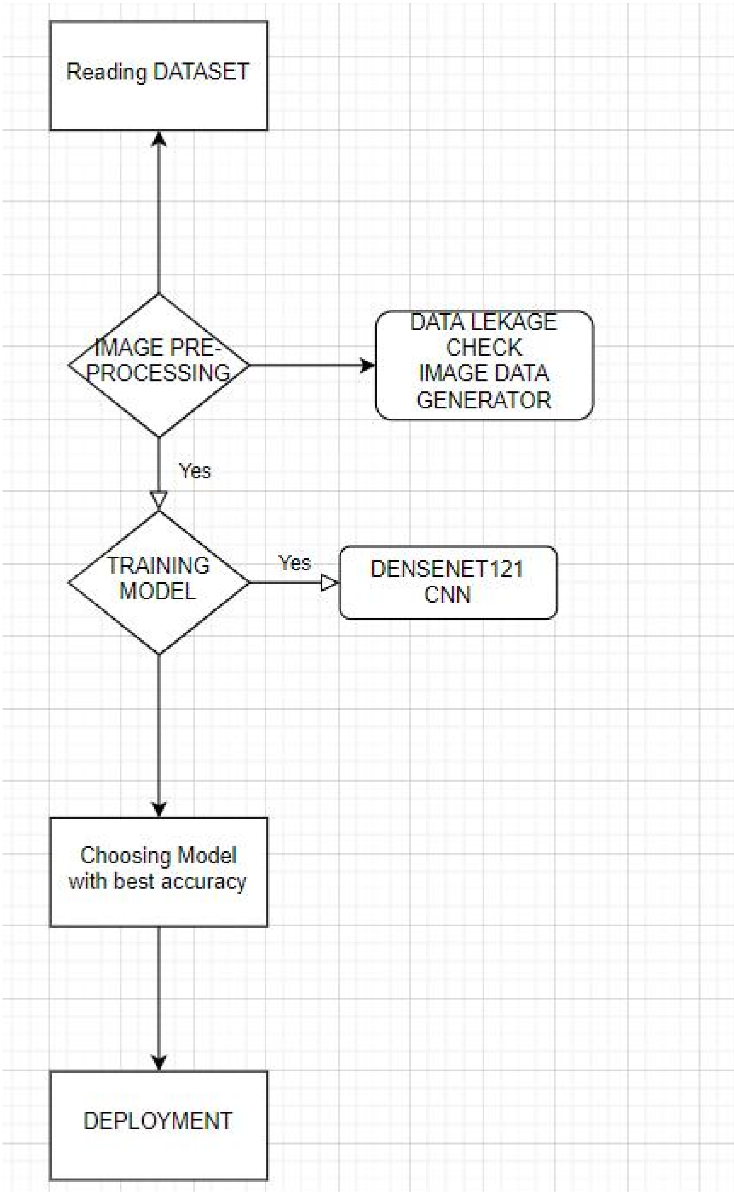
- This class also provides support for basic data augmentation such as random horizontal flipping of images.
- We also use the generator to transform the values in each batch so that their names is 0 and their standard deviation is 1.
  - This will faciltate model training by standardizing the input distribution.
- The generator also converts our single channel X-ray images(gray-scale) to a three-channel format by repeating the values in the image across all channels.
  - We wil want this because the pre-trained model that we'll use requires three-channel inputs
    - Normalize the mean and standard deviation of the data
    - Shuffle the input after each epoch.
    - Set the image size to be 320px by 320px

Fig 4.1: Keras image data generator



4.1 Flow chart(Uml diagram)

Fig: 4.1.1



The flow char consists of consists of 5 stages:

A. Reading Data-set.

A data set is a structured collection of data points related to a particular subject. A collection of related data sets is called a database. Data sets can be tabular or non-tabular. Tabular data sets contain structured data that is organized by rows and columns In our case images.

B. Image Pre-processing

Image preprocessing are the steps taken to format images before they are used by model training and inference. This includes, but is not limited to, resizing, orienting, and color corrections.

a) Data leak check

b) Image Data generator

C. Training model

Model training is the phase in the data science development lifecycle where practitioners try to fit the best combination of weights and bias to a machine learning algorithm to minimize a loss function over the prediction range.

a) CNN

b) DenseNet-121

c) VGG-16

d) Resnet50

D. Choosing model with best accuracy

Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition: Accuracy = Number of correct predictions Total number of predictions

E. Deployment

It is the final stage of project , where all things are put together are taken to usable phase and all the testing, user interface happens are made , this phase helps the project to have wide rang of usage across the users and also helps determine the feedback to correct and update the software, technology, methodologies etc.

5. Methodology

The proposed models to train the given machine by consuming the data-set are:

I. **Custom CNN**
II. **DenseNet-121**
III. **VGG-16**
IV. **ReseNet50**

Now lets look into each of the given models and see how they apply to our dataset:

I. Custom CNN:

When it comes to machine learning, artificial neural networks work very well. Artificial neural networks are used in a variety of classification tasks such as images, sounds, and words. Different types of neural networks are used for different purposes. For example, word order prediction uses recurrent neural networks, or more precisely LSTMs, and similarly image classification uses convolutional neural networks. In this blog, we will create the basic building blocks of CNN.

In a regular Neural Network there are three types of layers:

1. **Input layer**
2. **Hidden Layer.**
3. **Output Layer.**

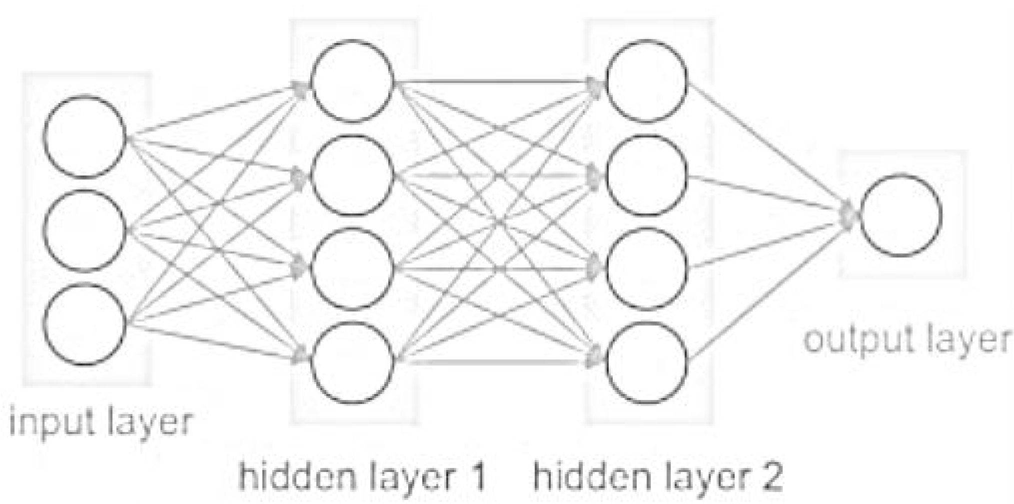
1. **Input Layer:**It's the layer in which we give input to our model. The number of neurons in this layer is equal to the total number of features in our data (number of pixels in the case of an image).

2. **Hidden Layer:** The input from the Input layer is then feed into the hidden layer. There can be many hidden layers depending upon our model and data size. Each hidden layer can have different numbers of neurons which are generally greater than the number of features. The output from each layer is computed by matrix multiplication of output of the previous layer with learnable weights of that layer and then by the addition of learnable biases followed by activation function which makes the network nonlinear.

3.**Output layer:** The output from the hidden layer is then fed into a logistic function like sigmoid or softmax which converts the output of each class into the probability score of each class.

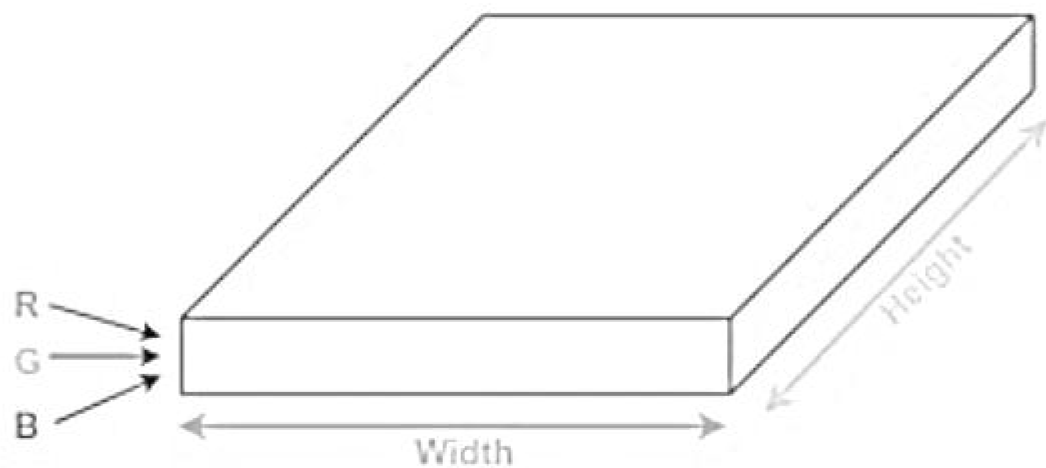
The data is then fed into the model and output from each layer is obtained this step is called feed-forward, we then calculate the error using an error function, some common error functions are cross-entropy, square loss error, etc. After that, we back-propagate into the model by calculating the derivatives. This step is called Back-propagation which basically is used to minimize the loss.

Fig 5.1 Input layer to output layer cnn



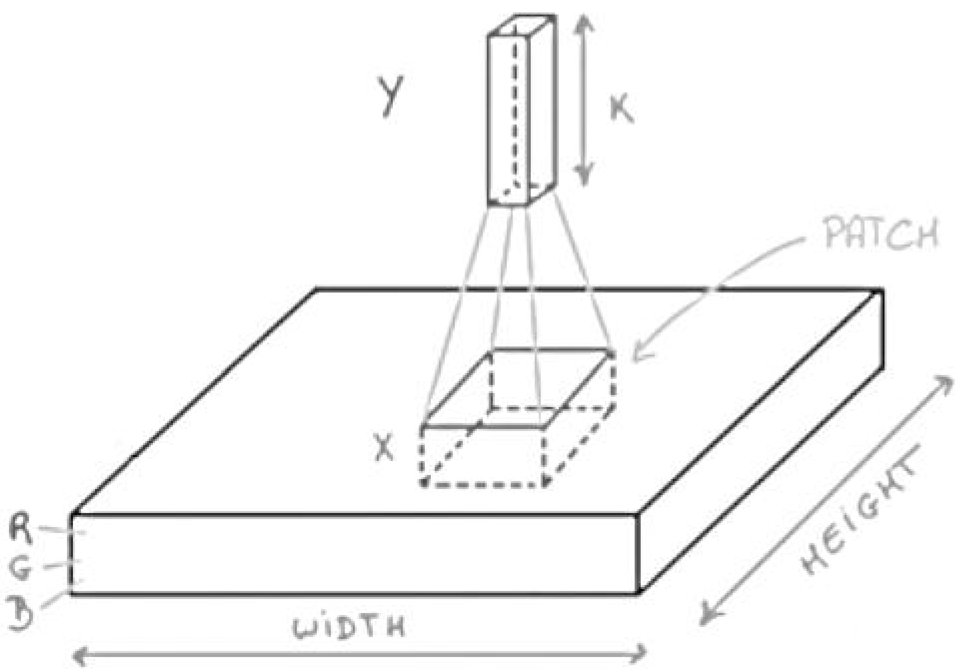
Convolution Neural Networks or covnets are neural networks that share their parameters. Imagine you have an image. It can be represented as a cuboid having its length, width (dimension of the image), and height (as images generally have red, green, and blue channels).

Fig 5.2 :Image pixel cuboid



Now imagine taking a small patch of this image and running a small neural network on it, with say, k outputs and represent them vertically. Now slide that neural network across the whole image, as a result, we will get another image with different width, height, and depth. Instead of just R, G, and B channels now we have more channels but lesser width and height. This operation is called Convolution. If the patch size is the same as that of the image it will be a regular neural network. This small patch gives us less weight.

Fig: 5.3 Cnn taking small patch of image



Now let's talk about a bit of mathematics that is involved in the whole convolution process. :

- Convolution layers consist of a set of learnable filters (a patch in the above image). Every filter has small width and height and the same depth as that of input volume (3 if the input layer is image input).
- For example, if we have to run convolution on an image with dimension 34x34x3. The possible size of filters can be axax3, where 'a' can be 3, 5, 7, etc but small as compared to image dimension.
- During forward pass, we slide each filter across the whole input volume step by step where each step is called stride (which can have value 2 or 3 or even 4 for high dimensional images) and compute the dot product between the weights of filters and patch from input volume.

- As we slide our filters we'll get a 2-D output for each filter and we'll stack them together and as a result, we'll get output volume having a depth equal to the number of filters. The network will learn all the filters.

Layers used to build ConvNets:

A covnets is a sequence of layers, and every layer transforms one volume to another through a differentiable function.

Types of layers:

Let's take an example by running a covnets on of image of dimension 32 x 32 x 3.

1. **Input Layer:** This layer holds the raw input of the image with width 32, height 32, and depth 3.
2. **Convolutional Layer:** his layer computes the output volume by computing the dot product between all filters and image patches. Suppose we use a total of 12 filters for this layer we'll get output volume of dimension 32 x 32 x 12.
3. **Activation Function Layer:** his layer will apply an element-wise activation function to the output of the convolution layer. Some common activation functions are RELU: max(0, x), Sigmoid: 1/(1+e^-x), Tanh, Leaky RELU, etc. The volume remains unchanged hence output volume will have dimension 32 x 32 x 12.
4. **Pool Layer:** This layer is periodically inserted in the covnets and its main function is to reduce the size of volume which makes the computation fast reduces memory and also prevents overfitting. Two common types of pooling layers are max pooling and average pooling. If we use a max pool with 2 x 2 filters and stride 2, the resultant volume will be of dimension 16x16x12.

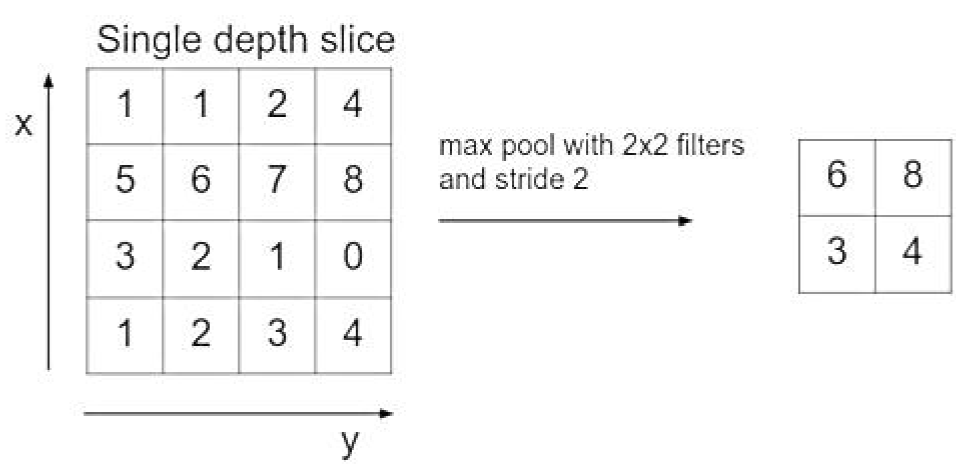


Fig 5.4

1. **Fully-Connected Layer:** This layer is a regular neural network layer that takes input from the previous layer and computes the class scores and outputs the 1-D array of size equal to the number of classes.

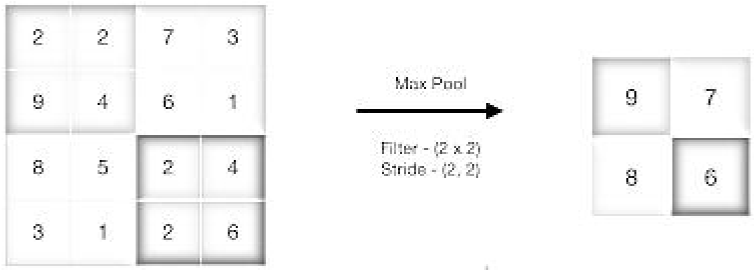


Fig 5.5 Process of pooling

II. DenseNet121:

In a traditional feedforward convolutional neural network (CNN), each convolutional layer except the first convolutional layer (which receives the input) receives the output of the previous convolutional layer and generates an output feature map before the next convolutional layer pass to Therefore, 'L' layers have 'L' direct connections. One between each layer and the next.

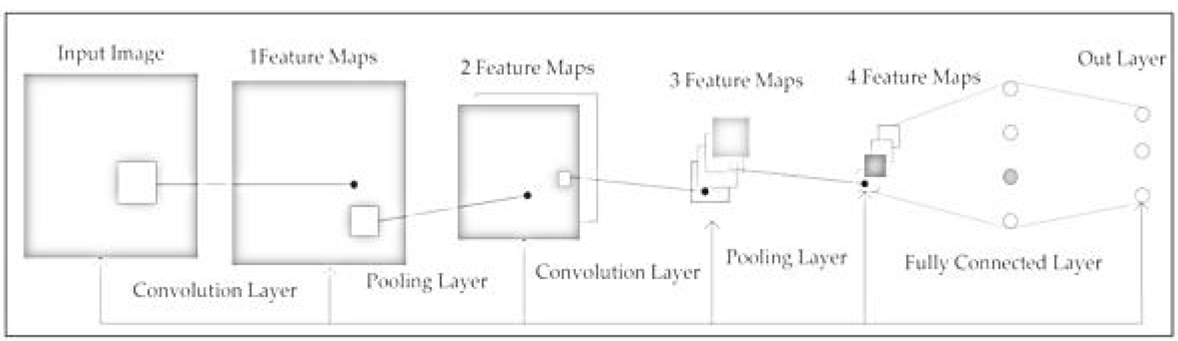


Fig 5.6

However, as the number of layers in the CNN increase, i.e. as they get deeper, the 'vanishing gradient' problem arises. This means that as the path for information from the input to the output layers increases, it can cause certain information to 'vanish' or get lost which reduces the ability of the network to train effectively.

DenseNets resolve this problem by modifying the standard CNN architecture and simplifying the connectivity pattern between layers. In a DenseNet architecture, each layer is connected directly with every other layer, hence the name Densely Connected Convolutional Network. For 'L' layers, there are L(L+1)/2 direct connections.

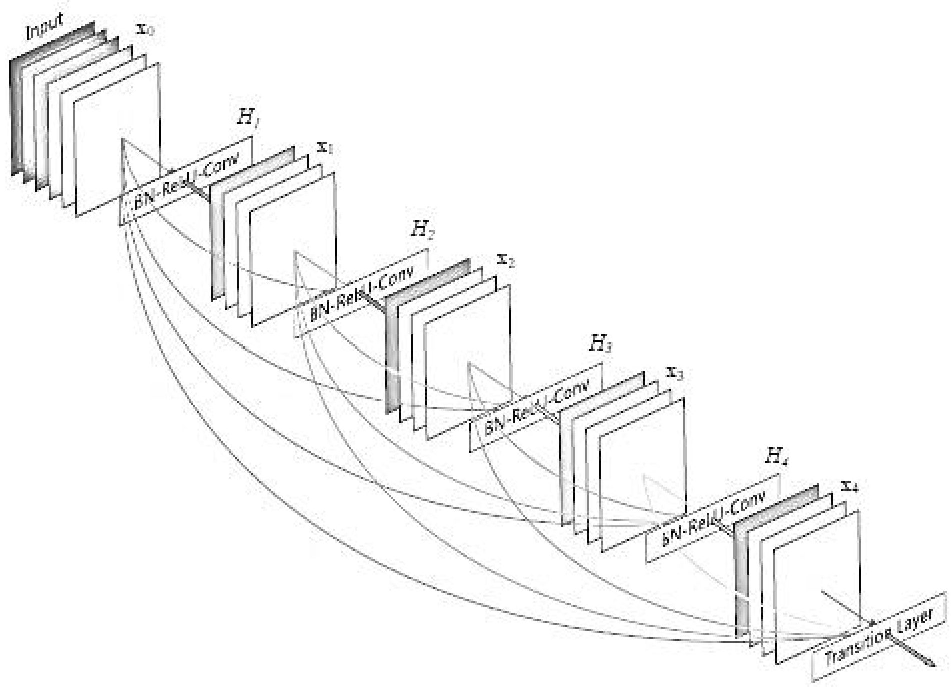


Fig: 5.7

Components of DenseNet inbclude:

- Connectivity
- Dense Blocks
- Growth Rate
- Bottleneck layers

CONNECTIVITY

In each layer, the feature maps of all the previous layers are not summed, but concatenated and used as inputs. Consequently, DenseNets require fewer parameters than an equivalent traditional CNN, and this allows for feature reuse as redundant feature maps are discarded. So, the lth layer receives the feature-maps of all preceding layers, x0,...,xl-1, as input:

$$\mathbf{x}_\ell = H_\ell([\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{\ell-1}]),$$

where [x0,x1,...,xl-1] is the concatenation of the feature-maps, i.e. the output produced in all the layers preceding l (0,...,l-1). The multiple inputs of Hl are concatenated into a single tensor to ease implementation.

DENSE BLOCKS

The use of the concatenation operation is not feasible when the size of feature maps changes. However, an essential part of CNNs is the down-sampling of layers which reduces the size of feature-maps through dimensionality reduction to gain higher computation speeds.

To enable this, DenseNets are divided into DenseBlocks, where the dimensions of the feature maps remains constant within a block, but the number of filters between them is changed. The layers between the blocks are called Transition Layers which reduce the the number of channels to half of that of the existing channels.

For each layer, from the equation above, Hl is defined as a composite function which applies three consecutive operations: batch normalization (BN), a rectified linear unit (ReLU) and a convolution (Conv).

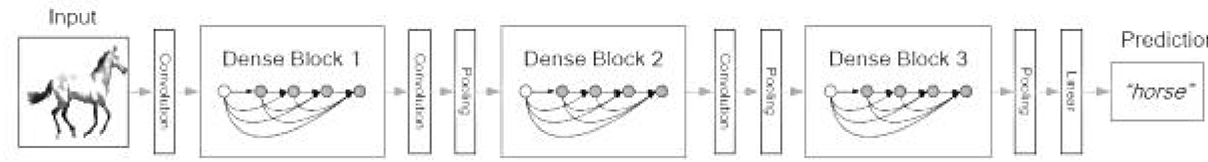


Fig:5.8

In the above image, a deep DenseNet with three dense blocks is shown. The layers between two adjacent blocks are the transition layers which perform down sampling (i.e. change the size of the feature-maps) via convolution and pooling operations, whilst within the dense block the size of the feature maps is the same to enable feature concatenation.

GROWTH RATE

One can think of the features as a global state of the network. The size of the feature map grows after a pass through each dense layer with each layer adding 'K' features on top of the global state (existing features). This parameter 'K' is referred to as the growth rate of the network, which regulates the amount of information added in each layer of the network. If each function H l produces k feature maps, then the lth layer has

$$k_l = k_0 + k * (l - 1)$$

input feature-maps, where k0 is the number of channels in the input layer. Unlike existing network architectures, DenseNets can have very narrow layers.

BOTTLENECK LAYERS

Although each layer only produces k output feature-maps, the number of inputs can be quite high, especially for further layers. Thus, a 1x1 convolution layer can be introduced as a bottleneck layer before each 3x3 convolution to improve the efficiency and speed of computations.

As DenseNets require fewer parameters and allow feature reuse, they result in more compact models and have achieved state-of-the-art performances and better results across competitive datasets, as compared to their standard CNN or ResNet counterparts.

III. VGG-16:

A convolutional neural network is also known as a ConvNet, which is a kind of artificial neural network. A convolutional neural network has an input layer, an output layer, and various hidden layers. VGG16 is a type of CNN (Convolutional Neural Network) that is considered to be one of the best computer vision models to date. The creators of this model evaluated the networks and increased the depth using an architecture with very small (3 × 3) convolution filters, which showed a significant improvement on the prior-art configurations. They pushed the depth to 16–19 weight layers making it approx — 138 trainable parameters.

What is VGG16 used for

VGG16 is object detection and classification algorithm which is able to classify 1000 images of 1000 different categories with 92.7% accuracy. It is one of the popular algorithms for image classification and is easy to use with transfer learning.

VGG16 Architecture

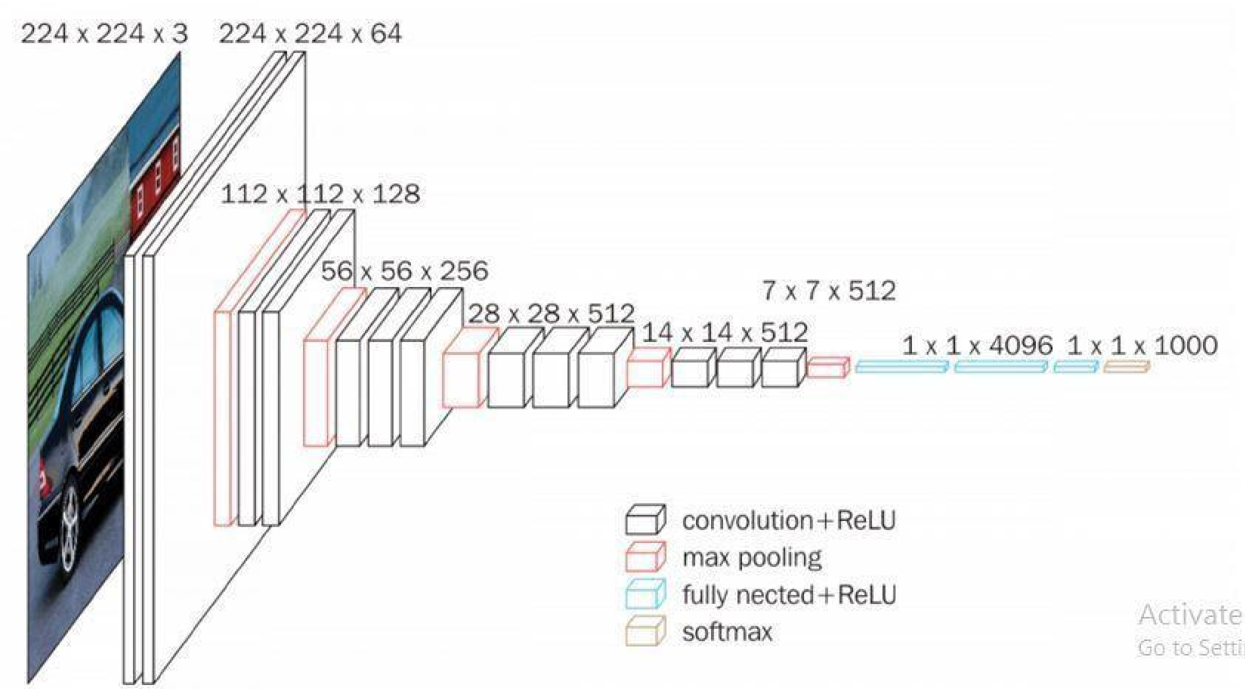


Fig:5.9 VGG architecture

- The 16 in VGG16 refers to 16 layers that have weights. In VGG16 there are thirteen convolutional layers, five Max Pooling layers, and three Dense layers which sum up to 21 layers but it has only sixteen weight layers i.e., learnable parameters layer.
- VGG16 takes input tensor size as 224, 244 with 3 RGB channel
- Most unique thing about VGG16 is that instead of having a large number of hyper-parameters they focused on having convolution layers of 3x3 filter with stride 1 and always used the same padding and maxpool layer of 2x2 filter of stride 2.
- The convolution and max pool layers are consistently arranged throughout the whole architecture

- Conv-1 Layer has 64 number of filters, Conv-2 has 128 filters, Conv-3 has 256 filters, Conv 4 and Conv 5 has 512 filters.
- Three Fully-Connected (FC) layers follow a stack of convolutional layers: the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer.

IV **ResNet50:**

After the first CNN-based architecture (AlexNet) that win the ImageNet 2012 competition, Every subsequent winning architecture uses more layers in a deep neural network to reduce the error rate. This works for less number of layers, but when we increase the number of layers, there is a common problem in deep learning associated with that called the Vanishing/Exploding gradient. This causes the gradient to become 0 or too large. Thus when we increases number of layers, the training and test error rate also increases.

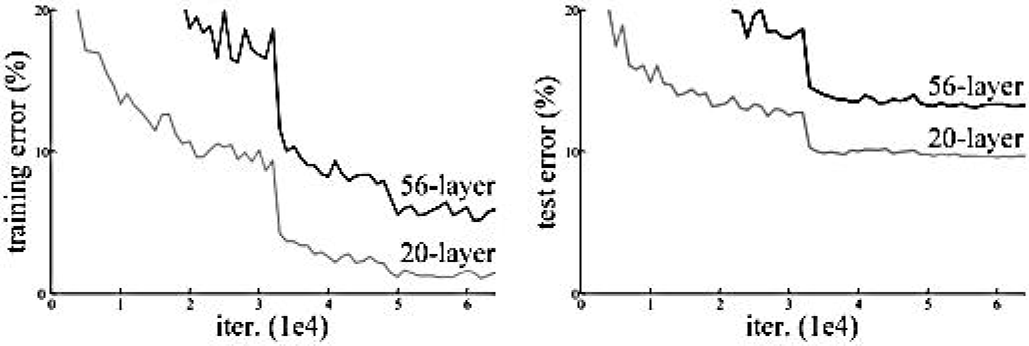


Fig:5.10 Comparison of 20 vs 56 layers architecture.

In the above plot, we can observe that a 56-layer CNN gives more error rate on both training and testing dataset than a 20-layer CNN architecture. After analyzing more on error rate the authors were able to reach conclusion that it is caused by vanishing/exploding gradient.

ResNet, which was proposed in 2015 by researchers at Microsoft Research introduced a new architecture called Residual Network.

Residual Network: In order to solve the problem of the vanishing/exploding gradient, this architecture introduced the concept called Residual Blocks. In this network, we use a technique called skip connections. The skip connection connects activations of a layer to further layers by skipping some layers in between. This forms a residual block. Resnets are made by stacking these residual blocks together.

The approach behind this network is instead of layers learning the underlying mapping, we allow the network to fit the residual mapping. So, instead of say H(x), initial mapping, let the network fit,

F(x) := H(x) - x which gives H(x) := F(x) + x.

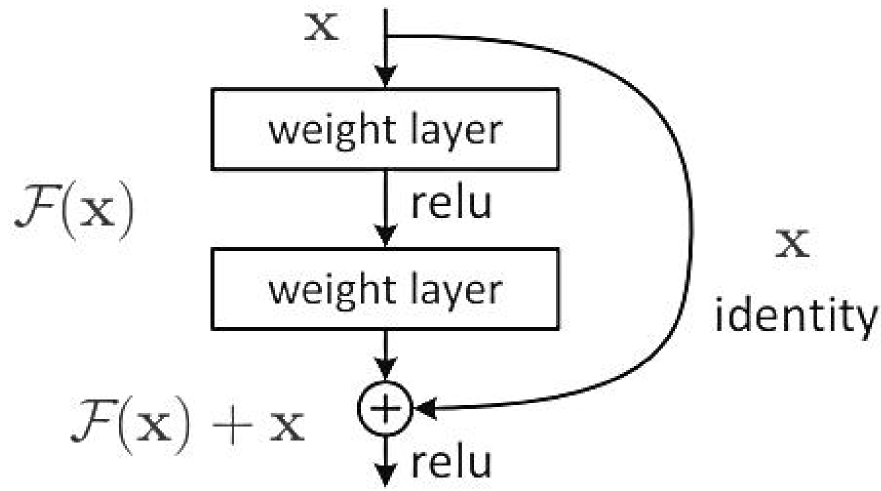


Fig 5.11 Skip connection

The advantage of adding this type of skip connection is that if any layer hurt the performance of architecture then it will be skipped by regularization. So, this results in training a very deep neural network without the problems caused by vanishing/exploding gradient. The authors of the paper experimented on 100-1000 layers of the CIFAR-10 dataset.

There is a similar approach called "highway networks", these networks also use skip connection. Similar to LSTM these skip connections also use parametric gates. These gates determine how much information passes through the skip connection. This architecture however has not provided accuracy better than ResNet architecture.

Network Architecture: This network uses a 34-layer plain network architecture inspired by VGG-19 in which then the shortcut connection is added. These shortcut connections then convert the architecture into a residual network.

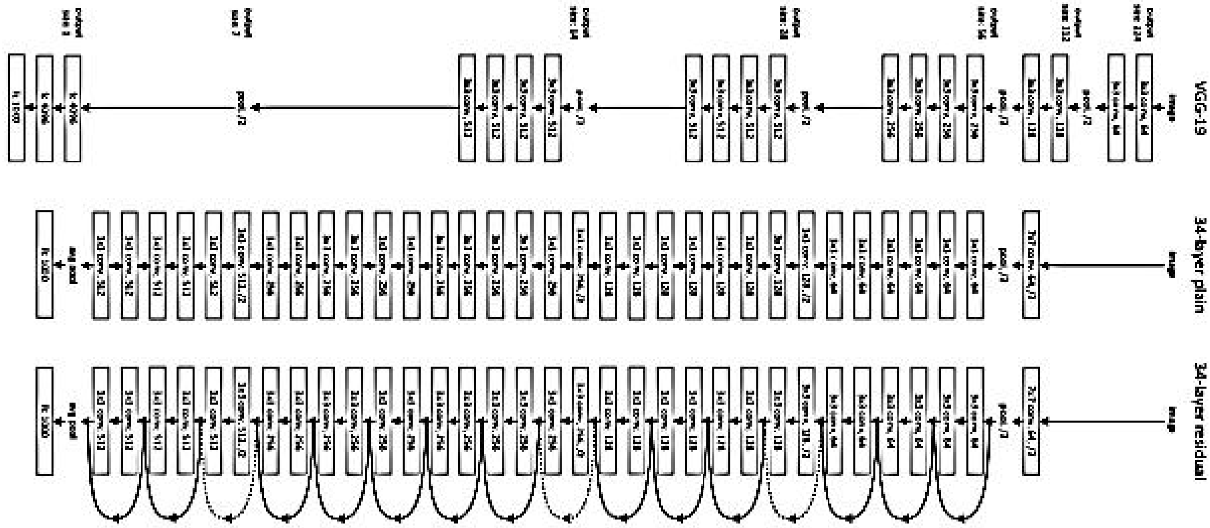


Fig 5.12: Resnet architecture

5. CONCLUSION

In this work, we have presented our approach for identifying pneumonia and understanding how the lung image size plays an important role for the model performance. We found that the dis- tinction is quite subtle for images among presence or absence of pneumonia, large image can be more beneficial for deeper informa- tion. However, the computation cost also burden exponentially when dealing with large image. Our proposed architecture with regional context, such as Custom CNN,Densenet121, Resnet50, VGG-16 supplied extra context for generating accurate results. Also, using thresholds in background while training tuned our network to perform well in the this task.

With the usage of ImagedataGenerator, Data-leakcheck We can prevent data overfit, rotating and other abnormal training possibilities making the te4chnique more possible to train easily. With compared to other research and approaches from other others this approach is much more efficient and reliable to get accurate results.

6. Bibliography.

1. Erdi Call1, Ecem sogancioglu, Bram van ginneken, kicky G.van Leeuwen, keelin Murphy Medical Image analysis

2. Soumya Ranjan Nayak, Deepeak Ranjan Nayak Utkarsh Sinha, vaibhav Arora, Ram Bilas Pachori Biomedical signal processing and control application of deep learning technique for detection of covid-19 cases using chest-x-0ray images: A comprehesnive study

3. Harshvardhan GM, Mahendra kumar gourisaria, Siddharth swarup rautary, Manjusha pandy Pneumonia Detection Usinmg Cnn through chest x-ray school of computer engineering., kiit Deemed to be university

4. Joseph Paul cohen, Lan Dao, Karsten Roth, Paul Morrison, Yoshua Bnggio, AlmasF. Abbasi, Beliyi Shen, Hoshmnand Kochi Masha, Marzyeh Ghassemi, Haifanhg Li, Tim Q. Duong Predicting COVID-19 Pneumonia serverity on chest x-ray with deep learning

5. Taufik Rahamat Azian Ismail, Sharifah Aliman University Teknologi MARA Chest X-ray image calssification in medical image analysis .

6. Ecem Sogancioglu, Erdi calli, Bram van Ginneken, Kicky G. van Leeuwen, Keelin Murphy Deep learning chest X-ray Anlaysis: A survey

7. Nada M. Elshennawy and Dina M.Ibrahim Deep-Pneumonia Framework Using deep Learning Models Based on Chest X-ray Images

8. Karim Hammoudi, Halim Benhabiles Mahmoud Melkemi Fadi Dorkanaika Ignacio Arganda - Carreras. Daminique Collard, Arnaud Scherepereel

9. Rohith kundu, Ritacheta Das, Zong Woo Geem, Gi-Tae Han, Ram Sarkar Penumonia detection in chest x-ray images using an ensemble of deep learning models

10. Faiz Azhmad Khan, Arman majidulla, Gamuchirai Tavazia, Ahsana Nazish, syed Kumail abidi, Chest x-rya analysis with deep learning based-software as atriage test for pulmonary tuberculosis: a prospective study of diagnosis accuracy for culture-confirmeed disease.

11. Amith Kumar Jaiswal Identifying Pneumonia in chest X-rays: A deep learning Approach

12. Anna Majkowska, Phd, Sid Mittal , BS , David F. Steiner,MD,PHD. Jousua J. Reicher, MD . Scott Mayer Mckinney, Ms Chest Radiograph Interpretation with deep learning Models.

13. S. Becker , C. Bulthgen, V. D. Phi van, C.Sekaggya-Wiltshire, B.Castelnuovo, A. kambugu j.pher, T. Frauenfelder Detection of tuberculosis patterns in digital photgraphs of chest X-ray images using Deep learning : feasibility study

14. Rachna Jain, Meena Gupta, Soham Taneja, D. Jude Hemanth Deep learning based detection and anlaysis of Covid-19

15. Jawad Rasheed , Alaa Ali Hameed, Chawik Djeedi, Akhtar Jamil, Fadi Al-Turjman A machine learning-based framework for diagnosis of Covid-19 from chest X-ray images

16. Zaid Nabulsi[1], Andrew Sellergren[1,5], Shahar Jamshy[1,5], Charles Lau[2], Edward Santos[1], Atilla P. Kiraly[1], Wenxing Ye[1], Jie Yang[1], Rory Pilgrim[1], Sahar Kazemzadeh[1], Jin Yu[1], Sreenivasa Raju Kalidindi[3], Mozziyar Etemadi[4], Florencia Garcia-Vicente[4], David Melnick[4], Greg S. Corrado[1], Lily Peng[1], Krish Eswaran[1], Daniel Tse[1] , Neeral Beladia[1], Yun Liu[1], Po-Hsuan Cameron Chen[1] & Shravya Shetty[1]

Deep learning for distinguishing normal versus abnormal chest radiographs an generalizationto two unseen diseases tuberculosis and COVID-19

17. Isabella Castiglioni[1,2†], Davide Ippolito[3†], Matteo Interlenghi[2], Caterina Beatrice Monti[4], Christian Salvatore[5,6*] , Simone Schiaffino[7], Annalisa Polidori[6], Davide Gandola[3], Cristina Messa[8,9] and Francesco Sardanelli[4,7]

Machine learning applied on chest x-raycan aid in the diagnosis of COVID-19: a firstexperience from Lombardy, Italy

18. Guangyu Wang[1,25], Xiaohong Liu[2,25], Jun Shen[3,25], ChengdWang[4,25],Zhihuan Li[5,25], Linsen Ye[6,25], Xingwang Wu[7,25], Ting Chen[2] , Kai Wang[2], Xuan Zhang[2], Zhongguo Zhou[8], Jian Yang[9], Ye Sang[9], Ruiyun Deng[10], Wenhua Liang[11], Tao Yu[3], Ming Gao[3], Jin Wang[6], Zehong Yang[3], Huimin Cai[10], Guangming Lu[12], Lingyan Zhang[13], Lei Yang[14], Wenqin Xu[5], Winston Wang[5], Andrea Olvera[5],Ian Ziyar[5], Charlotte Zhang[10], Oulan Li[10], Weihua Liao[15], Jun Liu[16], Wen Chen[17], Wei Chen[18], Jichan Shi[19], Lianghong Zheng[5], Longjiang Zhang[12], Zhihan Yan[18], Xiaoguang Zou[20], Guiping Lin[3], Guiqun Cao[4], Laurance L. Lau[5], Long Mo[15], Yong Liang[5], Michael Roberts[1,22], Evis Sala[23], Carola-Bibiane Schönlieb[22],Manson Fok[5], Johnson Yiu-Nam Lau[24], Tao Xu[10], Jianxing He[11]Kang Zhang[5,10] , Weimin Li[4] and Tianxin Lin[3]

A deep-learning pipeline for the diagnosis and discrimination of viral, non-viral and COVID-19 pneumonia from chest X-ray images

19. Lal Hussain[1,2*] ,Tony Nguyen[3], Haifang Li[3], Adeel A. Abbasi[1], Kashif J. Lone[1], Zirun Zhao[3], Mahnoor Zaib[2], Anne Chen[3] and Tim Q. Duong

Machine-learning classification of texture features of portable chest X-ray accurately classifies COVID-19lunginfection

20. Ecem Sogancioglu*[a], Erdi Çallı*[a], Bram van Ginneken[a], Kicky G. van Leeuwen[a], Keelin Murphy[a] [a]*Radboud University Medical Center, Institute for Health Sciences, Department of Medical Imaging, Nijmegen, The Netherlands*

Deep Learning for Chest X-ray Analysis: A Survey