Predicting Pneumonia using CNN

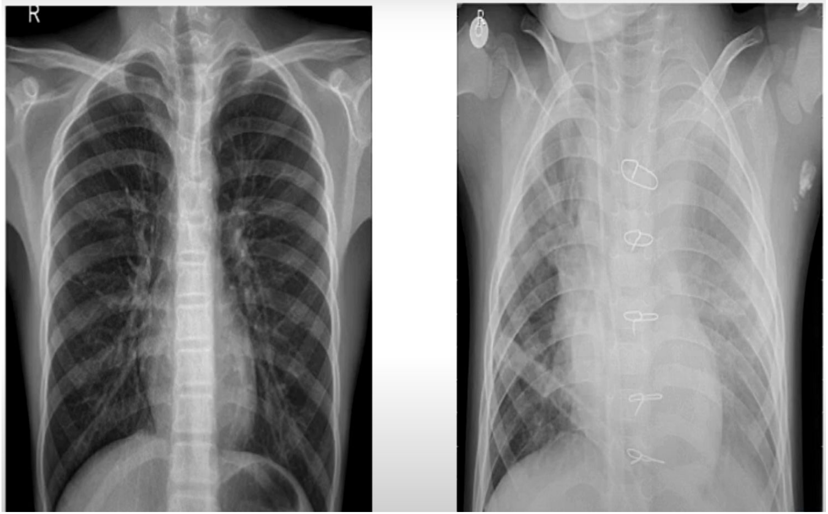
Guide: Dr. Pooja Mishra

Group Members:
Soumyadeep Laskar (IEC2019031)
Hritik Chauhan (IEC2019033)
Mayur Kumar (IEC2019039)

Motivation

- Pneumonia is a respiratory infection / lung illness caused by a number of bacteria, fungi, or viruses; it is more frequent in developing and impoverished countries due to high levels of pollution, unsanitary living conditions, and overpopulation, as well as limited medical infrastructure.
- The main aim of this project is to identify Pneumonia just by using the X-Ray images of the patients. As doctors must do a lot of certain tests to identify if the patient has Pneumonia or not because there might be some cases in which identifying infection in lungs by naked eye is difficult so to achieve remarkable efficiency and accuracy deep learning models is developed.
- Pneumonia detection on CXR is hectic, to improve the efficiency and accuracy and to make the work of the doctors simpler by making the diagnosis process quicker and cheaper with a reduced reliance on the intervention of human experts.
- Automated methods to detect and classify human diseases from medical images.

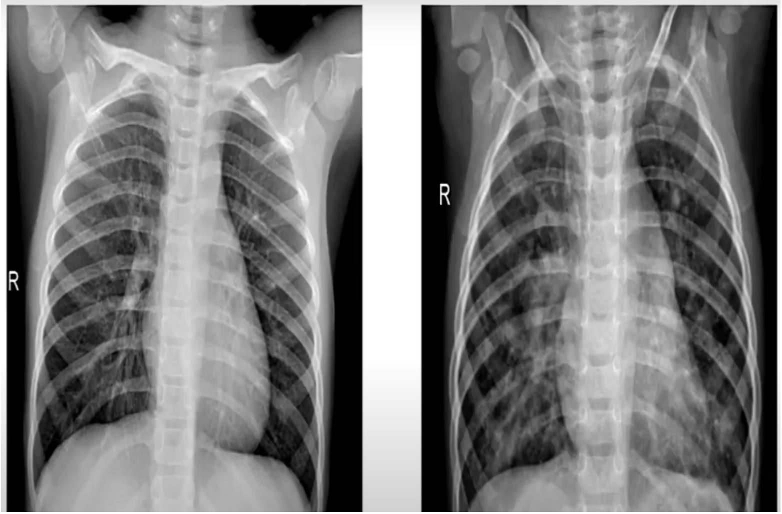
Case 1



Normal **Pneumonia**

Demarcation between normal lung and pneumatic lung (Easy to differentiate the infection)

Case 2



Normal **Pneumonia**

Demarcation between normal and pneumatic lung (Differentiating is not easy as the infection is not visible by naked eyes)

Literature Survey

- Rajpurkar & other. In reference [2] (2017) used a 121 layered convolutional neural network and chest X-ray 14 dataset to develop an algorithm that can detect pneumonia from chest X rays at a level exceeding practicing radiologist. The patient history and lateral radiographs could be taken into consideration to improve their model.

- Parveen & other in reference[3] reports an unsupervised fuzzy c-means classification learning algorithm to detect pneumonia infected X-ray images. This approach improves classification accuracy as fuzzy c-means allocate weights to all the pixels of the input X-ray images.

- Rahman, et al. [4] (2020) aimed to construct a system that could detect bacterial and viral pneumonia using X-ray images. They worked on 5247 X-ray images using convolutional neural networks giving an accuracy of 85-90%. Stephen, et al. [6] (2019) used a collection of 5856 chest X-ray images to train a convolutional neural network made from scratch to classify and detect the presence of pneumonia.

- Chapman et al [5]demonstrated three computerized methods using a rule base, a probabilistic Bayesian network, and a decision tree to diagnose the chest X-ray report associated with acute bacterial pneumonia

Methodology

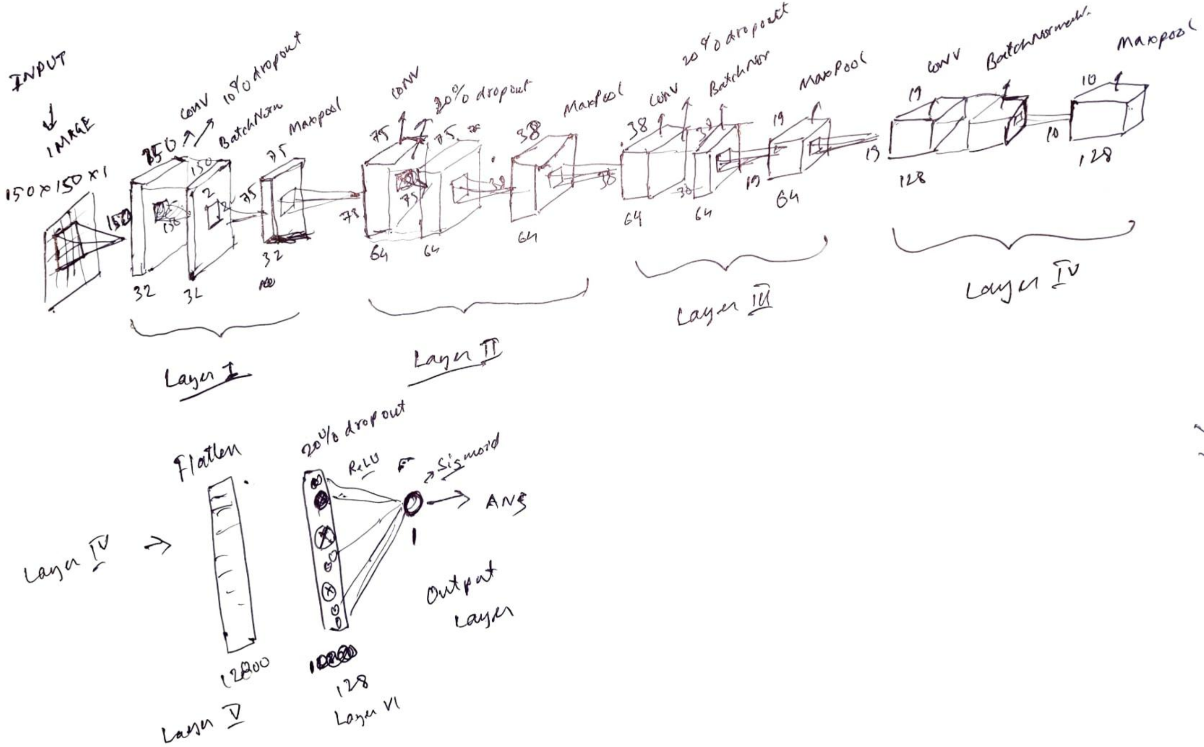
- We will be using Convolutional Neural Networks in our project since we are trying to a solve an image classification problem where we need to select features and CNN is good in feature extraction .
- The convolutional neural network is formed by building a layered neural network called convolutional layers. Convolutional layers use a filter of defined size to extract various features.

What is CNN ?

- A convolutional neural network (CNN) is a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data.
- CNN does not require much pre-processing. The CNN image classifier takes an input, evaluates it, and categorizes it.
- Images are fed into convolutional neural networks (CNNs), which prioritize different parts of the image.
- CNN’s are equipped with an input layer, an output layer, and hidden layers, all of which help process and classify images. The hidden layers comprise convolutional layers, ReLU layers, pooling layers, and fully connected layers

How CNN is used in our Project

- In our input layer we are taking images of dimension 150x150x1 (grayscale) image. The filter (3x3) convolutes over the image and gives a feature map after that a activation map is produced . Now the different activation function is used and hence we extract features that is required in our project.
- Similarly many layer are added to extract features of the image



Methodology

- We have a training dataset of 5216 x-ray images in jpg format divided into Pneumonic and Normal images.
- Before we fit our dataset into our model we have normalized our dataset and have used data augmentation to diversify our dataset and enhance the performance of the model. The techniques used are:
  - Rotating image by upto 30 degrees
  - A zoom of 20%
  - Horizontal and vertical shifting by 10%

Thus by doing this we can ensure randomness in the data in our dataset so that our model does not over fit and is not vulnerable to real world data.

- We have also used a variable reduce_lr. It reduces the loss when we find cases where the gradient is constant at the increase of epochs in the loss v/s epoch graph

Methodology

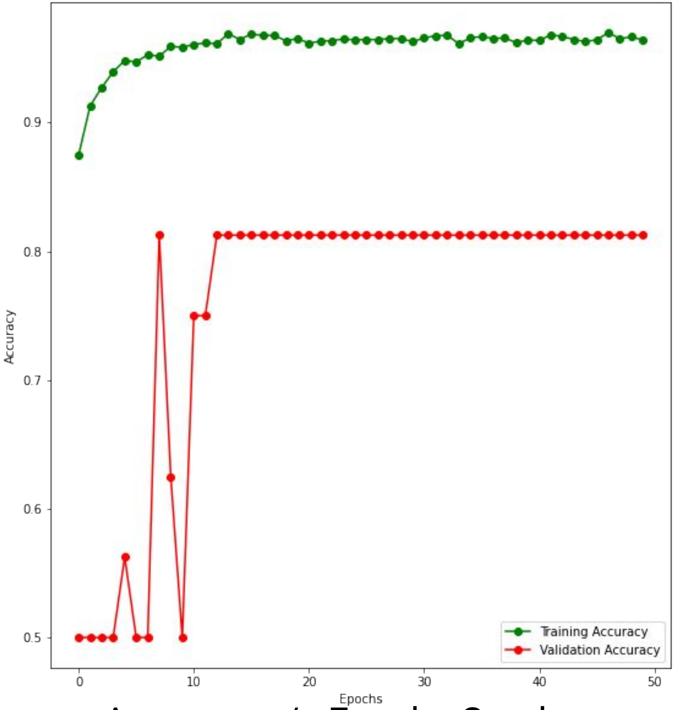
Model Summary:

- Implementation of 4 layer CNN with 2 dense layers.
- Activation function used is ReLU and in the output layer Sigmoid activation function is used
- We have used 50 epochs
- Optimizer used is Adam

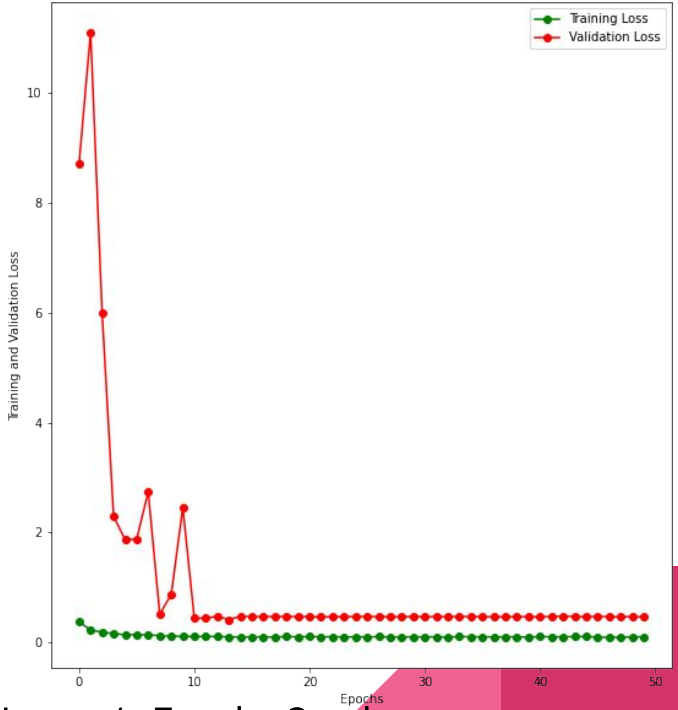
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 150, 150, 32)      320
_________________________________________________________________
batch_normalization (BatchNo (None, 150, 150, 32)      128
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 75, 75, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 75, 75, 64)        18496
_________________________________________________________________
dropout (Dropout)            (None, 75, 75, 64)        0
_________________________________________________________________
batch_normalization_1 (Batch (None, 75, 75, 64)        256
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 38, 38, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 38, 38, 64)        36928
_________________________________________________________________
batch_normalization_2 (Batch (None, 38, 38, 64)        256
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 19, 19, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 19, 19, 128)       73856
_________________________________________________________________
dropout_1 (Dropout)          (None, 19, 19, 128)       0
_________________________________________________________________
batch_normalization_3 (Batch (None, 19, 19, 128)       512
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 10, 10, 128)       0
_________________________________________________________________
flatten (Flatten)            (None, 12800)             0
_________________________________________________________________
dense (Dense)                (None, 128)               1638528
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 129
=================================================================
Total params: 1,769,409
Trainable params: 1,768,833
Non-trainable params: 576
```

Results

- We have seen a gradual increase in our validation accuracy and a gradual decrease in validation loss with the increase in number of epochs.



Accuracy v/s Epochs Graph



Loss v/s Epochs Graph

Results

- We have achieved an accuracy of 91.66% and loss of 0.24 however it is not the only criteria to check the validity of our model
- We also have explored different evaluation metrics. Like precision, recall, F1-score
- Formula:

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{F1} = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

True Positive (TP) refers to a sample that is appropriately categorized as positive by the model and belongs to the positive class.
False Positive (FP) is a case when the model mistakenly classifies a sample from the negative class as belonging to the positive class.
True Negative (TN) is a case when the model accurately classifies a sample as belonging to the negative class.
False Negative (FN) is a case when a sample belonging to the positive class is wrongly categorized as belonging to the negative class by the model.

- We have obtained the following results

```
Accuracy: 0.9166666666666666
Precision: 0.8823529411764706
Recall: 0.8974358974358975
F1-score: 0.8898305084745762
```

Results

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model.

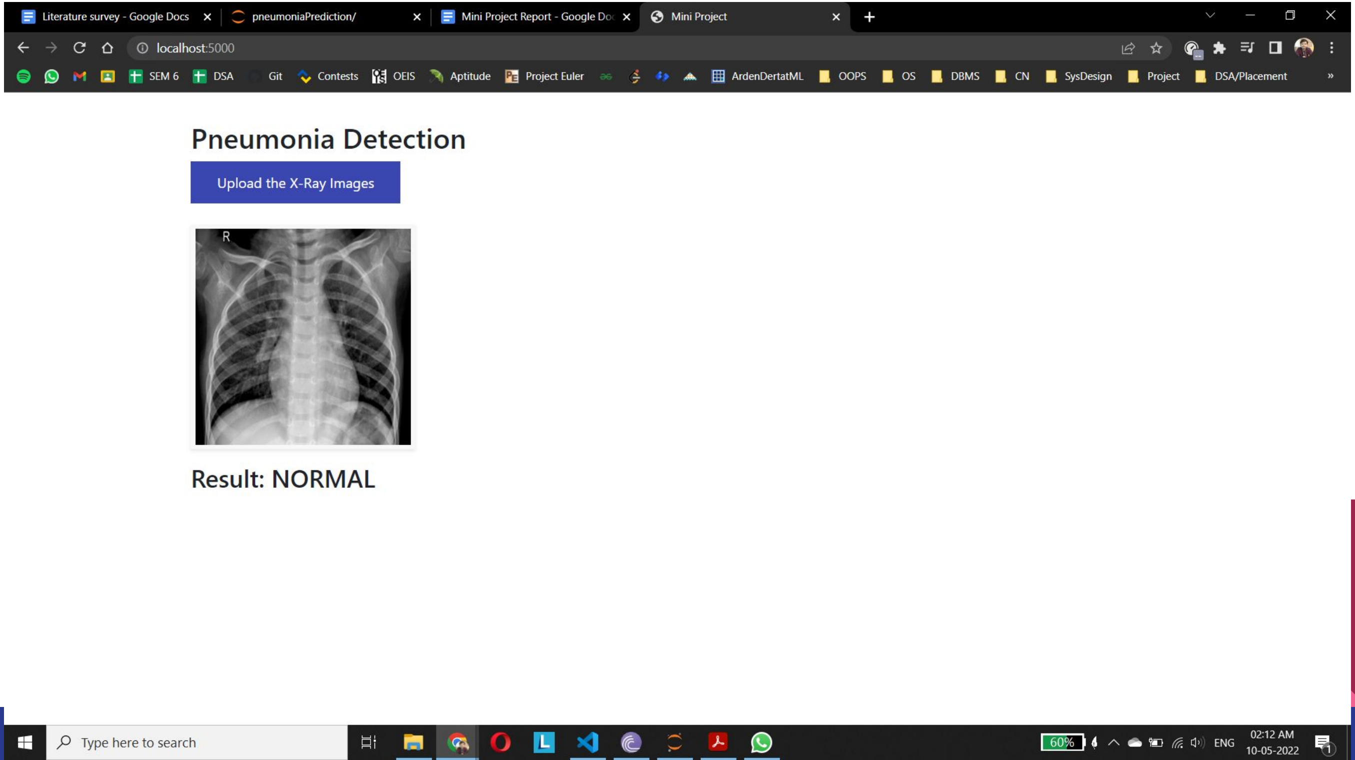
Since in our project we are predicting an image is pneumonic or normal, or we are doing binary classification we will be having a 2 x 2 matrix

The Confusion Matrix obtained is given beside:

Conclusion

We have created a web-based application to help medical practitioners to systematize chest X-ray images into “Pneumonia” and “Normal” two different classes using categorization based on deep learning.

Screenshot of the web based app that we have built is shown beside:



Dataset

- The data is organized into three folders: training, testing, and validation, with subfolders for each image category (Pneumonia/Normal). There are 5,856 JPEG X-Ray images divided into two groups (Pneumonia/Normal).

  Link to the dataset

References

[1] Kermany, Daniel; Zhang, Kang; Goldbaum, Michael (2018), “Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification”, Mendeley Data, V2, doi: 10.17632/rscbjbr9sj.2 [Dataset]

[2] Rajpurkar, et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. arXiv preprint. 2017. arXiv:1711.05225.

[3] N. Parveen and M. M. Sathik, "Detection of pneumonia in chest X-ray images", Journal of X-ray Science and Technology, vol. 19, no. 4, pp. 423-428, 2011.

[4] Rahman, et al. Transfer Learning with Deep Convolutional Neural Network (CNN) for Pneumonia Detection using Chest X-ray. Applied Sciences. 2020. 10.9: 3233.

[5] 9.W. W. Chapman, M. Fizman, B. E. Chapman and P. J. Haug, "A comparison of classification algorithms to automatically identify chest x-ray reports that support pneumonia", Journal of Biomedical Informatics, vol. 34, no. 1, pp. 4-14, 2001.

[6] Stephen, et al. An efficient deep learning approach to pneumonia classification in healthcare. Journal of healthcare engineering 2019.

[7] Liang G. & Zheng L. A transfer learning method with deep residual network for pediatric pneumonia diagnosis. Computer Methods And Programs In Biomedicine. 187 pp. 104964 (2020) pmid:31262537

Thank You