Comparative Analysis on Neural Networks based on their performance in Pneumonia Detection

Naman Puri*(naman.puri@s.amity.edu), Dr. Geetika Munjal**(gmunjal@amity.edu)
*Student, Department of Computer Science Engineering, Amity School of Engineering and Technology, Noida
**Associate Professor, Department of Computer Science Engineering, Amity School of Engineering and Technology, Noida

***Abstract*—Pneumonia has been a concerning issue worldwide. This infectious disease has a higher mortality rate than Covid-19. More than two million individuals lost their lives in 2019 out of which almost 600,000 were infants less than 5 years of age. Globally, identification of the disease is done manually by radiologists, but this method is highly unreliable as its accuracy is not sufficiently good. With the evolution of computational resources, especially the computing power of GPUs, it has become possible to train very deep CNNs. This study involves a comparative analysis of neural networks for pneumonia recognition. The goal is to do binary image classification for pneumonia recognition on each of the three models, namely, a Sequential model using TensorFlow (built from scratch), ResNet50 and InceptionV3 and comparing their efficiency, to discover which model suits best for smaller datasets and which suits best for larger datasets. Dataset consists of 5856 anterior and posterior Chest X-Ray images labeled as either Normal or Pneumonic.**

Keywords—CNN, Pneumonia Recognition, Chest X-Ray, Deep Learning, Binary Classification

I. Introduction

Pneumonia is an infection which causes inflammation in either of the lungs or both. Virus, bacteria or fungi trigger Pneumonia disease. It can adversely affect the patient's health and could even prove to be fatal if timely detection and treatment is not done in early stages. The conventional method for Pneumonia detection currently in use is manual diagnosis using Chest X-Ray Images, but this method as discussed is not sufficiently good as it depends on clinical experience of the medical professional. Result is subjective, leading to high variations in accuracy. Deep learning assisted healthcare through computer aided medical diagnosis to make the task of pneumonia detection way easier and more accurate.

The study involves recognizing a *suitable* dataset. For this problem statement I used a dataset consisting of 5856 CXR images (front-rear), which are records of timely clinical checkup of affected infants of 1-5 years of age from a Medical Center for Children and Women in GuangZhou, China. All chest radiographs are obtained through patients' periodic medical examination. All unclear CXR images and those which had compromised quality were removed for quality control at the source itself. To distinguish how well the models perform as opposed to practicing medical personnels, all the chest radiographs in the dataset were evaluated by two expert radiologists (or physicians) before being employed for training to our 3 deep learning models. To verify any errors in evaluation, each image in the dataset was also evaluated by a third radiologist. The next step is to build three pneumonia recognition deep learning models, Sequential Model using TensorFlow, ResNet50, InceptionV3. Lastly, we do comparative analysis on these models to filter out which model out of the three yields the best results on the test set.

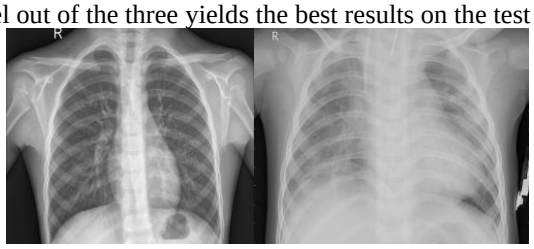

Fig. 1(a) *Fig. 1(b)*

Figure 1(a) shows an X-Ray image of a healthy person while Figure 1(b) shows an X-Ray image of a pneumonia patient.

II. Literature Review

Lately expert systems have been getting outpouring recognition in every field we can think of. The snowball effect of deep-learning has struck the field of computer aided medical diagnosis. Many studies were published on and around Pneumonia Recognition over the decade.

For Example, *Rajpukar et al., 2017*[1] successfully aimed at designing a model which would give considerably better predictions for Pneumonia Recognition as compared to practicing radiologists. They built a 121 layer deep neural network, which they call "ChexNet". CheXNet takes a CXR image as input and outputs a heatmap of the recognized disease. CheXNet is capable of recognizing 14 lung related diseases like Pneumonia, Atelectasis, Edema, Hernia and Pneumothorax among others. The model essentially uses DenseNet121(which was introduced by *Huang et al., 2016*[2]) along with batch normalization for optimization. Some of the limitations of the model as stated in the paper were; not employing the model on rear chest radiographs and no information about the patients' history was given either to the model or the radiologists.

Jaiswal et al., 2019[3], uses a model inspired from a Mask-R CNN,and employs both global and local/texture features to build a promising image segmentation model. Moreover, large attention was given to the image post processing phase which merges bounding boxes from numerous models.

TieNet proposed by *Wang et al., 2018*[4] is a Text-Image Embedding Network. TieNet is based on a CNN-RNN architecture and essentially simulates the reporting process. The model extricates features from Chest X-Ray and embeds them with the text attributes derived from the patients' reports. Once the annotations relating to a CXR image have been done, the mentioned areas in the text are marked as

bounding boxes in the image. This way the study successfully solves the matter in question, the scantiness of a large labeled dataset of chest X-Ray images.

Zhou et al., 2016[5], enables deep learning models or CNNs to have exceptional results in their localization ability, which further helped us to generate class-activation maps or CAMs and bounding boxes accurately in the chest radiographs. The study underrates the use of fully connected layers as it hinders performance because of the large number of parameters to learn from and gives remarkable results otherwise. In the near future, many expert systems were devised which would keep fully-connected layers while keeping the performance intact.

Lakhani et al., 2017[6], proposed a deep learning model to detect TuberCulosis(TB) on chest X-Ray images. The proposed model is a combination of GoogLeNet and AlexNet which yielded an AUC of 0.99. It was also stated that transfer learning leads to better results as compared to previously untrained models.

Yao et al., 2017[7], uses ChestX-Ray14 dataset from NIH with over 112,000 frontal Chest radiographs to train but the approach it adopts is inclined towards using classical architectures to make multilabel classification of upto 14 thoracic diseases. They use LSTMs to make predictions based on the interdependencies among different labels. It also states that a very basic baseline convolutional network which is not pretrained and without any label dependencies can very well beat pre-trained advanced CNNs.

Wang et al., 2017[8], introduced a new chest X-Ray dataset, known as 'ChestX-ray8' which includes 108,948 frontal chest radiographs of 32,717 patients(collected over the years 1992-2015) with a total of 8 labels for 8 different thoracic diseases namely Atelectasis, Cardiomegaly, Effusion, Pneumonia, Pneumothorax,Infiltration, Mass and Nodule. Every image is either labeled as one or more of the pathology keywords or "Normal" otherwise. Fabrication of the dataset was achieved using various Natural Language Processing techniques. For mining any pathological disease in the radiographs, 2 ML techniques were used namely DNorm and MetaMap.

Esteva et al., 2017[9], depicts binary classification of skin lesions as either keratinocyte carcinomas or benign seborrheic keratoses; and malignant melanomas or benign nevi using a deep learning model. The dataset used has 129,450 images with 2,032 distinct diseases. Transfer Learning technique is used as InceptionV3 is employed for performing this task.

Gabruseva et al., 2020[10], does binary classification of CXR images for Pneumonia Detection based on lung opacity. The model employed for this purpose is an ensemble/combination of RetinaNet made using PyTorch and SE-ResNext101. The encoder i.e. ResNext CNN is pretrained on the ImageNet dataset. Though the method used does not involve test v/s time augmentation and doesn't provide a good balance between the resources and the accuracy.

Hussain et al., 2021[11], detects Cocid-19 by doing binary classification or multiple label classification on X-Ray images and CT scans. A model was proposed, CoroDet which is a 22 layer deep convoluted network. The dataset employed in this study is Covid-R dataset with 7390 images. Accuracy for 2 label classification is 99.1% which is better than the state-of-the-art.

III. Methodology

A. Overview

This paper involves comparing different deep learning models like a Sequential Model using TensorFlow, ResNet50, Inception V3 based on their performance in pneumonia detection and eventually give suitable suggestions on which model to go for in actual pneumonia diagnosis situations. First task is importing the dataset consisting of 5856 Chest X-Ray images each of which is labelled into 2 classes either Normal or Pneumonic. The dataset is divided into 3 sets; the train set, the validation test and the test set. We need not care much about enhancing the image quality as the unclear scans have already gone through a round of screening and have been removed. This is followed by data pre-processing which involves many tasks such as grayscale normalization(if a specific model demands for one input channel), resizing images to match the input dimensions of a model. This is followed by Data Augmentation which restrains overfitting and would help to produce more features to train from. Moreover, data augmentation would help to train the model in a more natural way, practically radiographs have variations in terms of rotation and scale. So we would want to mimic that and take those intricacies into account for real scenarios. Third, we will build the deep learning model, train them and check their accuracy on the test set. Followed by a comparative analysis among all the three models. Figure 2 below shows the flowchart of the algorithm to be followed.

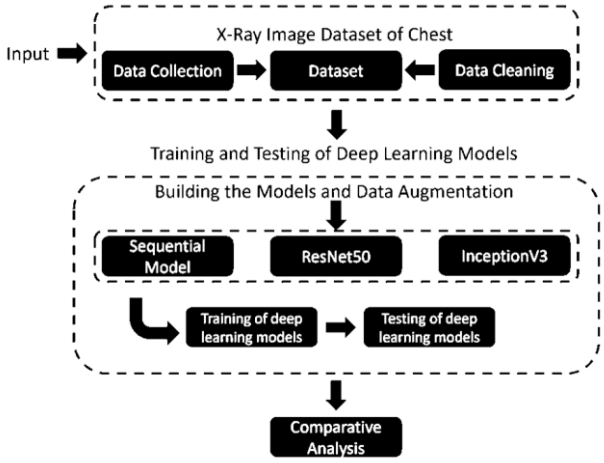

Fig. 2

B. Algorithm

1) Data Collection and Data Preprocessing: Involves recognizing and acquiring a suitable dataset for the problem statement. This study uses a dataset with 5856 images, from a Women and Children Medical Center in GuangZhou, China, each one of which is labeled as either Pneumonic or Normal. The website link to the dataset provided at the end of this section would provide you access to the dataset already segregated into 3 different directories, train set directory, validation set directory and test set directory, each one of which has two subdirectories: Normal consisting of images labeled as Normal and Pneumonia consisting of images labeled as Pneumonic. One may manually filter the

dataset further to remove the more inconsistent and duplicate data, and only keep high resolution pneumonia radiographs but we need not care much about enhancing the quality of images as the unclear scans have already gone through a round of screening and have been removed at the source itself.

Dataset Breakdown:

Test Set = 234 Normal images + 390 Pneumonic images = 10.65%

Train Set = 1341 Normal images + 3875 Pneumonic images = 89.07%

Validation Set = 8 Normal images + 8 Pneumonic images = 0.273%

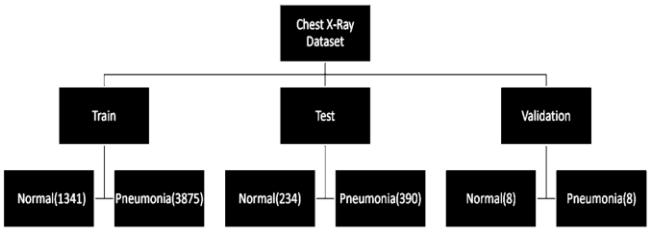

Fig. 3

Data Preprocessing involves resizing images to the specified input size of the respective models. Moreover, we may do grayscale normalization if the model allows only the L channel as input.

The dataset can be fetched from the following source:

Dataset: Chest Radiographs (anterior and posterior)

URL:https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

2) Augmenting the Data: Because of less features to train from, the model might not predict accurate results. In order to increase the number of images in the dataset significantly we adopt the data augmentation technique. Data Augmentation is a method to proliferate the dataset by generating new images which is made possible through applying a series of transformations on the existing images. Applying data augmentation can considerably enhance generalization ability of our model and prevents overfitting to a great extent.

The transformations applied on the images generally involve rotating the image, scaling the image, flipping the image, cropping the image, adding hue and adding noise filters to the images among others. Augmenting data in Python can be very easy through ImageDataGenerator in the Keras image processing library.

3) Building the Deep Learning Model: This study involves building 3 expert systems/deep learning models for binary image classification. These three transfer learning models are extensively used in image classification, namely a Sequential Model using TensorFlow, Resnet50 and InceptionV3. We will augment these models to do binary image classification of CXR images.

1. Sequential Model using TensorFlow built from scratch
2. Resnet50
3. InceptionV3

The details of each model are as follows:

1. ResNet50: ResNet or Residual Network is a NN extensively used for image classification The ResNet50 is a variant in the ResNet architecture with 50 convolutional layers that has been trained on the ImageNet dataset. It was introduced back in 2015 by Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun in their white paper “Deep Residual Learning for Image Recognition”. ReLU is mostly used as the non-linearity function, batch-normalization is employed and the model skips one or more than one layers at a time(a concept which they call “identity-shortcut connection”). For this reason alone, a model called HighwayNet used an additional weight to learn these skips or skip weights. These are the few features that are often encompassed within ResNet. ResNet50 architecture comprises a series of convolutional blocks with average pooling layers. Softmax is employed as the final layer for classification. The input dimensions for ResNet are (224, 224, 3).

2. InceptionV3: Launched in 2015 InceptionV3 by Google is also an extensively used convolutional network for Image Classification. It has made successful claims on yielding higher accuracy on a considerably lower number of epochs(computational efficiency) and can prove to be very helpful in making accurate predictions on CXR images with comparatively lesser training. The architecture of InceptionV3 involves factorized convolutions which helps to reduce the number of parameters to learn from, thereby making the model more efficient computationally. The model side stepped from using larger convolutions and resorted to using smaller convolutions instead for feature extraction thereby significantly decreasing the number of parameters. Asymmetric convolutions and Auxiliary classifiers are among some additional features of Inception Networks. The input dimensions for InceptionV3 are (299, 299, 3).

3. Sequential Model on TensorFlow built from Scratch: A Sequential Model can be built out of a completely simple stack of layers where every layer possesses two tensors; one input tensor and one output tensor. It works best for linear topology. ReLU is a conventional activation function for non-linearity. The proposed model in the paper is a 22 layer deep learning model with Conv2D Layers, Batch Normalization Layers, Max Pooling Layers and Dropout Layers. Dropout Layers are used to drop-out randomly certain percentages of data at each stage. Dropout Layers were very helpful as they restrain the overfitting of the training data(i.e. help in generalization). Learning Rate Reduction is done to rapidly decrease the validation loss and increase accuracy.

4) Loading the Model in GPU (optional): The CPU is used by default for training the model but leads to a relatively slower computation as compared to a GPU. If feasible, we can try loading our model in the GPU instead which speeds up the computation exponentially. Loading the model for

GPU computation can be divided into simple steps. We will start off by setting up the device for computation, if a GPU is available, we will designate the task to the GPU else the computation would be done by the CPU. This can be achieved through PyTorch framework.
Once the model is loaded into the GPU, all the input and output data of the model (training and testing) is loaded into the GPU for computation.

5) Comparative Analysis of the Model on CXR Images*:* The goal is to do binary image classification on Chest X-Ray images for Pneumonia Detection on a Sequential Model made using TensorFlow, ResNet50 and InceptionV3 and compare their efficiency, and discover which model suits best for smaller datasets and which suits best for larger datasets. This way we would be able to filter out the best model based on their performance on the test set.

IV. Result

A. *Experimental Data and its Specifications*

Using a dataset with 5856 CXR images, each one of which is labeled as either Pneumonic or Normal. The dataset is already segregated into 3 different directories, train set directory, validation set directory and test set directory, each one of which has two subdirectories: *Normal* consisting of images labeled as Normal and Pneumonia consisting of images labeled as Pneumonic. One may manually filter the dataset further to remove the more inconsistent and duplicate data, and only keep high resolution pneumonia radiographs but we need not care much about further improving the quality of images as the unclear scans have already gone through a round of screening and have been removed at the source itself. Dataset breakdown is given in *Table 1* below. Also, one of our main ideas behind Image Augmentation was because the number of 'Normal' labeled images was way lesser than the number of 'Pneumonia' labeled images in the dataset. *Section 4.2* shows visually through a bar graph the ratio between the two. Before loading the model up for training each image was reshaped according to the required input dimensions of the respective models, images for the Sequential Model were adjusted to (150,150,1), the images for the ResNet50 model were reshaped to (224,224,3), while the images for the InceptionV3 model were adjusted to (299, 299, 3). For visual comparison purposes, *Section 4.2* below shows a bunch of pneumonic and normal CXR images. One could clearly see some fuzziness in affected patients' X-Ray images while there is no or little fuzziness in the X-Ray images of healthy people. Refer to the following Table 1.

Pneumonia Images	4273 Chest X-Ray images	Training set (90.685% randomly selected) Testing set (9.127% randomly selected) Validation Set (0.187% randomly selected)	390 Chest X-Ray images 3875 Chest X-Ray images 8 Chest X-Ray images
Normal Images	1583 Chest X-Ray images	Training set (14.782% randomly selected) Testing set (84.712% randomly selected) Validation Set (0.506% randomly selected)	234 Chest X-Ray images 1341 Chest X-Ray images 8 Chest X-Ray images

B. *Data Visualization*

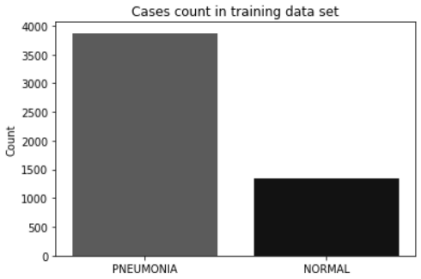

Fig. 4: Pneumonia v/s Normal Case Count in Training Set

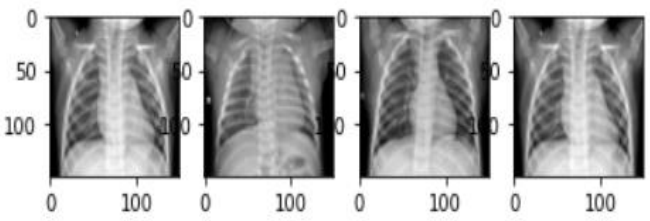

Fig.5: Few Normal Images from the dataset

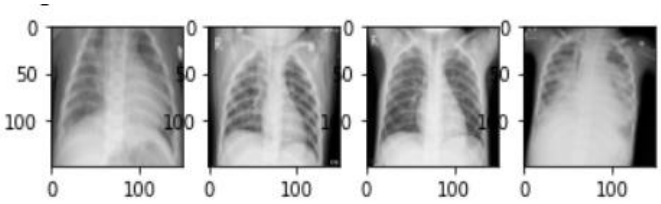

Fig. 6: Few Pneumonia images from the dataset

C. Individual Model Results

*1) **Sequential Model using TensorFlow***

Table 2

Epochs	Loss	Accuracy	Validation Loss	Validation Accuracy
1	0.5078	0.8466	57.8534	0.5000
2	0.2803	0.9032	55.7605	0.5000
3	0.2279	0.9179	60.9766	0.5000
4	0.1599	0.9417	1.5219	0.6875
5	0.1385	0.9502	0.6316	0.6875
6	0.1333	0.9526	1.0054	0.5000
7	0.1186	0.9616	10.3545	0.5000
8	0.1166	0.9599	9.7449	0.5000
9	0.1039	0.9647	1.2314	0.5625
10	0.1070	0.9649	1.8770	0.5625

Model Accuracy= 92.46794581413269% = 92.46% (approx.)
Loss of the model = 0.23139269649982452 = 0.23 (approx.)

In Figure 7, the curves are such because the training and validation data are highly unrepresentative of one another. The model is working well with the training data but not as much on the validation data. This means that our model is not generalized enough to work on different types of data. One way to regularize the data is to add dropout.

1. One can change the random seed of the train and validation set and fit the model again to get satisfactory results.
2. One other way to seek better results is to try to change the dropout rates at each stage, start by testing at 0 dropout rate initially and go higher if needed.
3. Changing the batch_size for the model can also help significantly (by default it is 32 for Keras) to alter the training and validation curves for better results.
4. Last but not the least, one can put early stopping (stop at a certain epoch) as part of their code because perhaps the validation loss and training loss may start diverging from that epoch onwards.

Figure 8 as compared to Figure 7 displays a much more ideal trend of training and validation loss converging in the end.

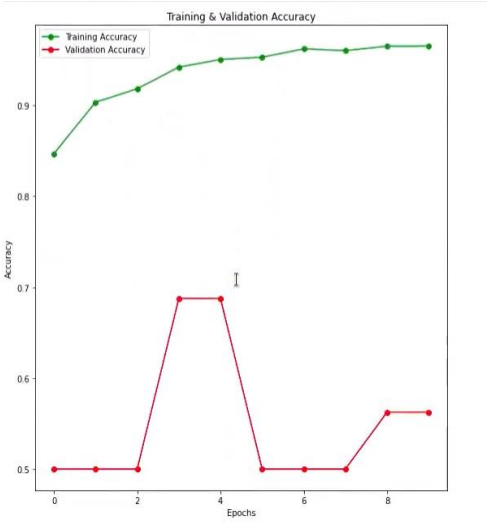

Fig. 7: Training and Validation Accuracy

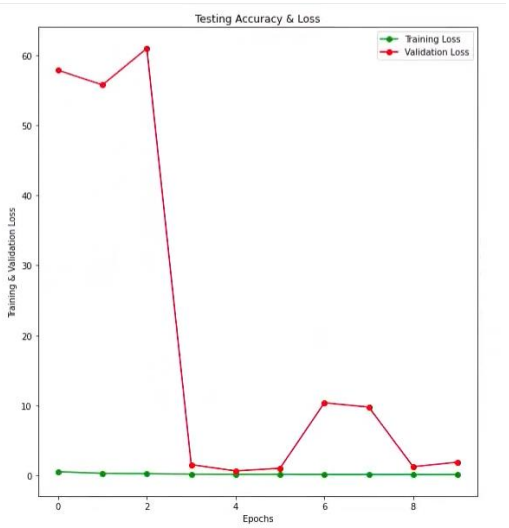

Fig. 8: Training and Validation Loss

Table 3- Confusion Matrix for Sequential Model

	Predicted Normal	Predicted Pneumonic
Normal	371	19
Pneumonic	28	206

*2) **ResNet50 Model***

Table 4

Epochs	Loss	Accuracy	Validation Loss	Validation Accuracy
1	0.2060	0.9206	0.5090	0.7500
2	0.1501	0.9459	0.3579	0.7500
3	0.1164	0.9576	0.3120	0.8750
4	0.1088	0.9597	0.1450	0.9375
5	0.0994	0.9624	0.0822	1.0000
6	0.0906	0.9663	0.0841	1.0000
7	0.0778	0.9697	0.0658	1.0000
8	0.0774	0.9749	0.0758	0.9375
9	0.0651	0.9760	0.0354	1.0000
10	0.0646	0.9762	0.0368	1.0000

Accuracy of the model= 96.093%
Loss of the Model= 0.10562

ResNet50 seems to work really well on as both curves(Fig. 9 and Fig. 10) converge in an ideal fashion at higher epochs but again the points mentioned above for Sequential model would largely help fine-tuning the training and validation accuracy curve for the ResNet Model too.

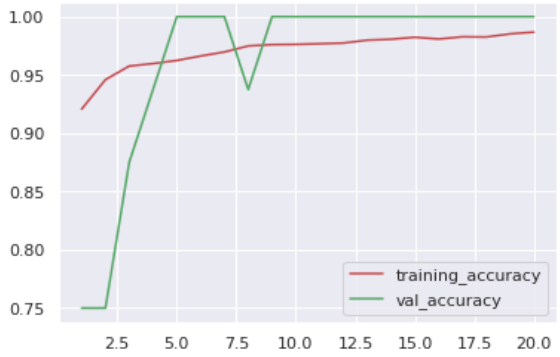

Fig. 9: Training and Validation Accuracy

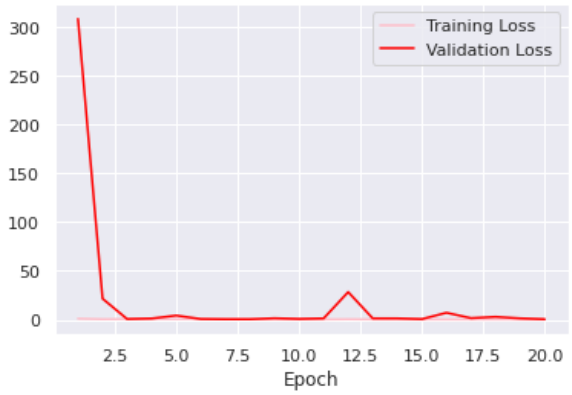

Fig. 10: Training and Validation Loss

Table 5- Confusion Matrix for ResNet Model

	Predicted Normal	Predicted Pneumonic
Normal	379	11
Pneumonic	23	211

3) *InceptionV3 Model*

Table 6

Epochs	Loss	Accuracy	Validation Loss	Validation Accuracy
1	0.6241	0.8384	2.0741	0.4375
2	0.2238	0.9124	1.3485	0.6250
3	0.1881	0.9317	0.8375	0.6875
4	0.1471	0.9471	0.8639	0.6875
5	0.1268	0.9538	0.3246	0.8125
6	0.1273	0.9540	0.7919	0.7500
7	0.1039	0.9613	0.7045	0.7500
8	0.1026	0.9661	0.4520	0.8125
9	0.0962	0.9657	0.4160	0.7500
10	0.0918	0.9661	0.2477	0.9375

Accuracy of the Model= 93.966%= 94%(approx.)
Loss of the Model= 0.18317

One inference we can directly draw from Figure 11 and Figure 12 is that the model is overfitted because it works well on the training data but not as much on the validation data. The training curve (loss and accuracy) in both the Figures 11 and 12 seems to jump up and down in zig-zag fashion because it's working with limited data. Though the curves technically do follow the conventional trend, we could perhaps get more false positives as compared to the previous models because the data is highly unregularized. Fine-tuning is necessary, adding a dropout of 0.2 or 0.3 for a dataset of this size should help.

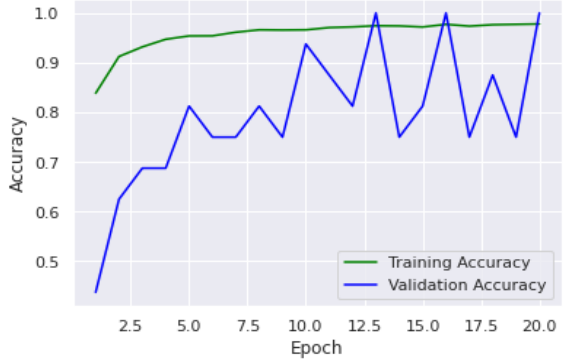

Fig. 11: Training and Validation Accuracy

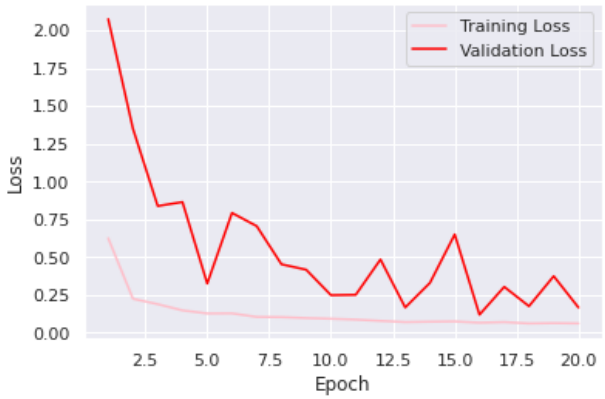

Fig. 12: Training and Validation Loss

Table 7- Confusion Matrix for InceptionV3 Model

	Predicted Normal	Predicted Pneumonic
Normal	362	28
Pneumonic	36	198

D. *Comparitive Analysis*

Table 8

Epochs	Sequential	ResNet50	InceptionV3
1	0.8466	0.9206	0.8384
2	0.9032	0.9459	0.9124
3	0.9179	0.9576	0.9317
4	0.9417	0.9597	0.9471
5	0.9502	0.9624	0.9538
6	0.9526	0.9663	0.9540
7	0.9616	0.9697	0.9613
8	0.9599	0.9749	0.9661
9	0.9647	0.9760	0.9657
10	0.9649	0.9762	0.9661

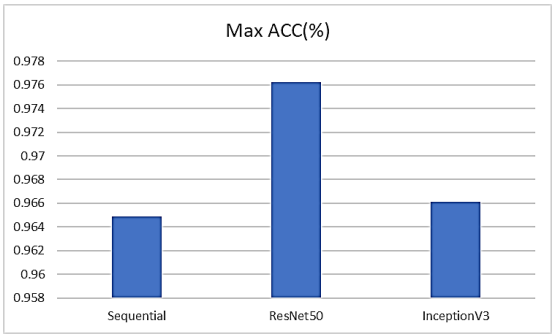

Fig. 13: Maximum Accuracy Percentage of 3 Models

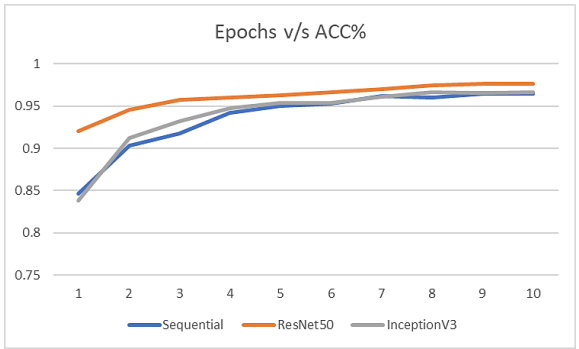

Fig. 14: Epochs v/s Accuracy Percentage Plot

V. Conclusion and Future Work

This study involved building three expert systems for Pneumonia recognition namely, a Sequential Model using TensorFlow, ResNet50 and InceptionV3, their performances were compared and analyzed to identify the optimal model for the problem statement. These three expert systems were employed on the same set of CXR images and the results were studied to find out which model might be the best fit in real-life scenarios where various factors might hinder the output, for example, if not that powerful computer is available onsite which might be true for a remote area, we might go for a model that technically gives better results as compared to other models in lower to medium number of epochs. The comparison establishes the premise for the practical application of neural networks in pneumonia detection to maintain people's health and lives.

The main conclusions which can be drawn are: 1) For smaller datasets any of the three models, ResNet50, InceptionV3 and Sequential Model will provide great results.

2) Though, for very small datasets, data augmentation did not seem to improve the accuracy of the models, they are too insensitive to changes in such a small dataset. 3) GPU Loading is always recommended as it considerably increases the computational efficiency of the model and thereby, increasing the model's accuracy.

Forthcoming research should involve rich, high-scale datasets to improve the accuracy of the model(For instance, ImageNet Dataset and how it has lead deep-learning models to perform Image Recognition in a way that it surpasses human ability) and further refine our pneumonia recognitions models for better practical results and we might see significant performance results as a whole in the interdisciplinary field of machine learning as machines keep getting more powerful and computationally efficient.

References

[1] Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., ... & Ng, A. Y. (2017). Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. arXiv preprint arXiv:1711.05225.

[2] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4700-4708

[3] Jaiswal, A. K., Tiwari, P., Kumar, S., Gupta, D., Khanna, A., & Rodrigues, J. J. (2019). Identifying pneumonia in chest X-rays: A deep learning approach. Measurement, 145, 511-518.

[4] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Ronald M. Summers; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 9049-9058

[5] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva and A. Torralba, "Learning Deep Features for Discriminative Localization," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2921-2929, doi: 10.1109/CVPR.2016.319.

[6] Lakhani P, Sundaram B. Deep Learning at Chest Radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks. Radiology. 2017 Aug;284(2):574-582. doi: 10.1148/radiol.2017162326. Epub 2017 Apr 24. PMID: 28436741.

[7] arXiv:1710.10501 [cs.CV]

[8] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, Ronald M. Summers; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2097-2106

[9] Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, Thrun S. Dermatologist-level classification of skin cancer with deep neural networks. Nature. 2017 Feb 2;542(7639):115-118. doi: 10.1038/nature21056. Epub 2017 Jan 25. Erratum in: Nature. 2017 Jun 28;546(7660):686. PMID: 28117445; PMCID: PMC8382232.

[10] Tatiana Gabruseva, Dmytro Poplavskiy, Alexandr Kalinin; Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2020, pp. 350-351

[11] Hussain E, Hasan M, Rahman MA, Lee I, Tamanna T, Parvez MZ. CoroDet: A deep learning based classification for COVID-19 detection using chest X-ray images. Chaos Solitons Fractals. 2021 Jan;142:110495. doi: 10.1016/j.chaos.2020.110495. Epub 2020 Nov 23. PMID: 33250589; PMCID: PMC7682527.