Pneumonia detection using Neural Networks in Machine Learning with Python programming

Pratheeksha Nath Narikkadan
Computer Science
University of Massachusetts, Lowell
Lowell, United States
Pratheekshanath_narikkadan@student.uml.edu

Sahithi Sallaram
Computer Science
University of Massachusetts, Lowell
Lowell, United States
sahithi_sallaram@student.uml.edu

Sruthi Atluri
Computer Science
University of Massachusetts, Lowell
Lowell, United States
Sruthi_Atluri@student.uml.edu

Supriya Dhamapurkar
Computer Science)
University of Massachusetts, Lowell
Lowell, United States
Supriya_dhamapurkar@student.uml.edu

***Abstract*—Pneumonia is a bacterial or viral respiratory infection that affects a large number of people, particularly in developing and impoverished countries where pollution, unsanitary living conditions, and overcrowding are all too common, as well as a lack of medical infrastructure. Globally, pneumonia causes the highest number of deaths in children under the age of five. In the US, it is the eighth leading cause of death in children under the age of five. Humans and technology must work together in order to save lives by detecting pneumonia in earlier stages, so the relevant treatment can be given. Pneumonia produces plural effusion, which is a condition in which fluids fill the lung and create breathing problems. It is critical to diagnose pneumonia early in order to receive curative treatment and boost survival chances. Several deep learning algorithms have been proposed for diagnosing pneumonia from chest X-ray pictures. Finding an acceptable and efficient model that meets all performance metrics has been one of the most difficult tasks. The major goal of this research is to propose efficient and strong deep learning models for detecting and categorizing pneumonia. By varying the deep learning method utilized, two distinct algorithms which is described in this paper: A Convolutional Neural Network (CNN), and a Multilayer Perceptron (MLP). The proposed algorithms are implemented and assessed in Python, and they are compared to another recent research in the field. Kaggle used a chest X-ray data-set to create a custom Convolutional neural network and multi-layer perceptron. A user interface was developed that accepts a chest X-ray and forecasts the occurrence of pneumonia as well as the percentage of congestion. The accuracy of the convolutional neural network and multilayer perceptron models was 91.02 percent and 76.40 percent, respectively. When compared to the multi-layer perceptron, the convolutional neural network produced better results. As a result, the GUI was created using a custom convolutional neural network.**

***Index Terms*—Convolution Neural Network, Pneumonia Detection, Multilayer Perceptron**

I. Introduction

Pneumonia is a lung infection that accounts for 18 of 100 deaths in children under the age of five. Furthermore, pneumonia affects about two billion people worldwide each year, with death a possibility if no action is taken. The importance of early pneumonia diagnosis cannot be overstated. To avoid misdiagnosis, a prompt diagnosis by an expert radiologist using chest X-rays is necessary. The most common and least expensive technique to identify pneumonia is via chest X-rays. Similarly, there is a scarcity of radiologist professionals, particularly in low-resource countries and rural areas, resulting in lengthier delays for diagnoses and a higher death rate. Because of the nature of chest X-ray image processing, pneumonia diagnoses are frequently ambiguous and can be mistaken for other diseases with similar symptoms, such as opacity, cavities, and pleural effusions. As a result, chest X-rays aren't as effective at detecting illnesses. Even for seasoned specialists, detecting pneumonia on chest x-ray can be difficult since other lung abnormalities, such as lung cancer and excess fluid, might exhibit comparable opacities. The expert's judgment can be influenced by the patient's position and level of inspiration at the time of the examination, in addition to such lung anomalies. Other factors at the time of analysis and interpretation, in addition to the factors inherent in the examination, can contribute to an incorrect diagnosis: (1) the professional's subjectivity and experience, (2) work fatigue due to repetitive actions, and (3) lighting levels in the consultation room. In this setting, detecting pneumonia in children using traditional chest x-ray image analysis is time-consuming and subjective. Both the diagnosis and therapy are delayed as a result of these issues. As a result, a dependable system capable of overcoming such challenges is required so that the expert can diagnose and refer the patient for treatment in real time and with more confidence. As a result, numerous computer-aided diagnosis (CAD) systems and computer algorithm diagnostic tools for X-ray image processing have been presented by researchers; these proposed systems assist radiologists in quickly detecting chest X-ray pneumonia. Recently, solutions based on Artificial Intelligence (AI) approaches such as handcrafted techniques, deep learning, and machine learning techniques have been used to solve numerous biological challenges such as skin cancer detection, brain tumor detection, and breast cancer detection. Convolutional neural network and Multilayer per-

ceptron models are shown in this paper to accurately diagnose pneumonic lungs from chest X-rays, which can be used in the real world by medical practitioners to treat pneumonia. These models have been taught to classify chest X-ray pictures into normal and pneumonia in a matter of seconds, allowing for early pneumonia detection. To assist control this dangerous infection in youngsters and other age groups, CNN and MLP to detect pneumonia from chest X-ray pictures. The model's accuracy is directly connected with the size of the data-set; that is, using large data-sets improves the model's accuracy; however, there is no direct association between the number of convolutional layers and the model's accuracy. A specified number of permutations of convolution layers, dense layers, dropouts, and learning rates must be trained by evaluating the models after each execution to achieve the best outcomes. Simple models were trained on the data-set at first, and then the complexity were increased until a model was found that not only achieved the desired accuracy but also surpassed other models in terms of recall and F1 scores. The goal of this study is to create CNN and MLP models from the ground up that can accurately identify and hence detect pneumonic patients from chest X-rays with high validation accuracy, recall, and F1 scores. In medical imaging cases, recall is frequently preferred above other performance assessing measures since it provides a measure of false negatives in the data. The number of false negatives in a result is critical in determining model performance in the actual world. When a model has high accuracy but low recall values, it is said to be under-performing, inefficient, and even dangerous, because higher false-negative values imply a higher number of cases where the model predicts a patient as normal when the individual is sick. As a result, the patient's life would be jeopardized. To avoid this, only models with high recall values, good accuracies, and F1 scores would be considered. The following sections make up the paper: Section 1: introduces the topic of this research study, discusses its significance and relevance, the reason and motivation for conducting this research, and the paper's goal. Section 2: delves into the work that has been done in this sector thus far. Section 3: explains the paper's approach, including the model architecture, flowchart, and data-set used to train and test the two models. Section 4: examines the performance of CNN models and MLP model utilizing accuracy and loss graphs and confusion matrices to display the findings obtained. Section 5: concludes the paper with a brief conclusion and the best-suited model. Furthermore, the research work's future scope has been considered. All the references which are cited in the paper have been listed in the end

II. Related Work

Many researchers have tackled the problem of classifying images with high accuracy.Here are some citations related to our paper:

Rubin and colleagues created a CNN model to detect common thorax disorders from frontal and lateral chest X-ray pictures. The MIMIC-CXR data-set was utilized to accomplish large-scale automatic picture recognition. The data-set was divided into 70 percent training, 20 percent testing, and 10 percent validation sets, accordingly. To increase overall performance, data-augmentation and pixel normalization were applied. For PA and AP, their DualNet CNN model had an AUC of 0.72 and 0.688, respectively. Lakhani et al. created a deep convolutional neural network to classify pulmonary TB. To categorize chest X-ray pictures, transfer learning models like AlexNet and GoogleNet were utilized. Training, testing, and validation sets accounted for 68 percent, 14.9 percent, and 17.1 percent of the data-set, respectively. To produce the top performing model with an AUC of 0.99, data augmentation and pre-processing approaches were used. The model's precision and recall were 100 percent and 97.3 percent, respectively. Guan et al. built an AG-CNN model to detect thoracic disease. To detect thorax disease from chest X-ray images, the ChestX-ray14 data-set was employed. For categorization, a global and local branch attention-guided CNN was utilized. With an AUC of 0.868, their model outperformed the other models stated in their study report. Rajpurkar et al. created a deep convolutional neural network model to categorize chest X-ray pictures into pneumonia and 14 other diseases. The model was trained using the ChestX-ray14 dataset. They compared their ChXNet model (121 layered model) to academic radiologists in practice.Their ChXNet model achieved an F1 score (95 percent CI) of 0.435outperforming radiologists which achieved an F1 score (95 percent CI) of 0.387. Krizhevsky et al. developed a deep convolutional neural network model with five convolutional layers, some followed by max-pooling layers, and three fully connected layers. There were 60 million different parameters in this network. This model earned a top-five error rate of 17 percent by using dropout. Simonyan et al. created a highly accurate model that used numerous small kernel-sized filters to reach top-five test accuracy of 92.7 percent. The ImageNet data-set was used to train this model, which was then entered into the ILSVRC 2014 competition. Xu et al. created a convolution neural network for the classification and segmentation of brain tumor MRIs. This model used a variety of strategies, including data augmentation, feature selection, and pooling techniques. This model achieved a validation accuracy of 97.5 percent for classification and 84 percent for segmentation on 256 256 pixels sized frontal chest radio graphs that were fed to a deep convolution neural network to detect anomalies.

III. Problem Definition

In order to detect pneumonia, a radiologist has to examine the chest X-ray and update the doctor correctly. This includes locating the infected area of the lungs. The main objective of this deep learning model is to identify if a person has Pneumonia, identify the infected area in the lung. In order to train this model, a Convolutional Neural Network and MLP is used. This network can train the model with the images of chest x-rays and predict with high accuracy. This model can be used in the health care industry, especially in the radiology department. We can also deploy this model for x-ray machine for precise prediction of pneumonia. This model can

help radiologists predict the condition of the patient based on chest x-rays easily and accurately. This paper also Compare the accuracy and precision of both the models and conclude efficient model to detect pneumonia using chest x-ray images.

IV. Methodology/Proposed Method

We created an ensemble framework using two models: convolutional neural networks and multilayer perceptron, in this study. The subheadings below go over the models in greater detail.

A. CNN Architecture

CNN models are feed-forward networks with convolutional, pooling, flattening, and fully connected layers that use appropriate activation functions.

- Convolution layer - It is the foundation of the CNNs. In mathematics, the convolution process is used to combine two functions. The input image is initially transformed to matrix form in CNN models. The input matrix is sent through a convolution filter, which performs element-wise multiplication and saves the sum. A feature map is created as a result of this. When photos are black and white, the 3 3 filter is commonly used to create 2Dfeature maps. When the input image is represented as a 3D matrix with the RGB color representing the third dimension, convolutions are done in 3D. The input matrix is used to operate several feature detectors, resulting in a layer of feature maps that constitutes the convolutional layer.
- Activation functions - ReLU activation function are used for this model. The term ”rectified linear function” refers to the activation function of a rectified linear function. It’s a nonlinear function that produces zero when the input is negative and one when it’s positive. This sort of activation function is widely utilized in CNNs since it solves the problem of vanishing gradients and increases layer non-linearity. There are many different types of ReLU activation functions, such as Noisy ReLUs, Leaky ReLUs, and Parametric ReLUs. Computational simplicity and representational sparsity are two advantages of ReLU over other activation functions. These models use this widely used activation function in the last dense layer. This activation function converts inputs to a probability distribution by normalizing them. With this sort of activation function, the categorical cross-entropy cost function is commonly utilized.
- Pooling layer - Pooling layers come after convolutional layers. Max-pooling layers are the type of pooling layer employed in this model. The maximum pixel intensity values from the window of the image now covered by the kernel are selected by the max-pooling layer, which has a dimension of 2x2. Down sampling photos via max-pooling reduces the dimensionality and complexity of the image. General pooling and overlapping pooling are two different types of pooling layers that can be used. The max-pooling technique is used in the models described in this paper because it aids in the recognition of important elements in the image.

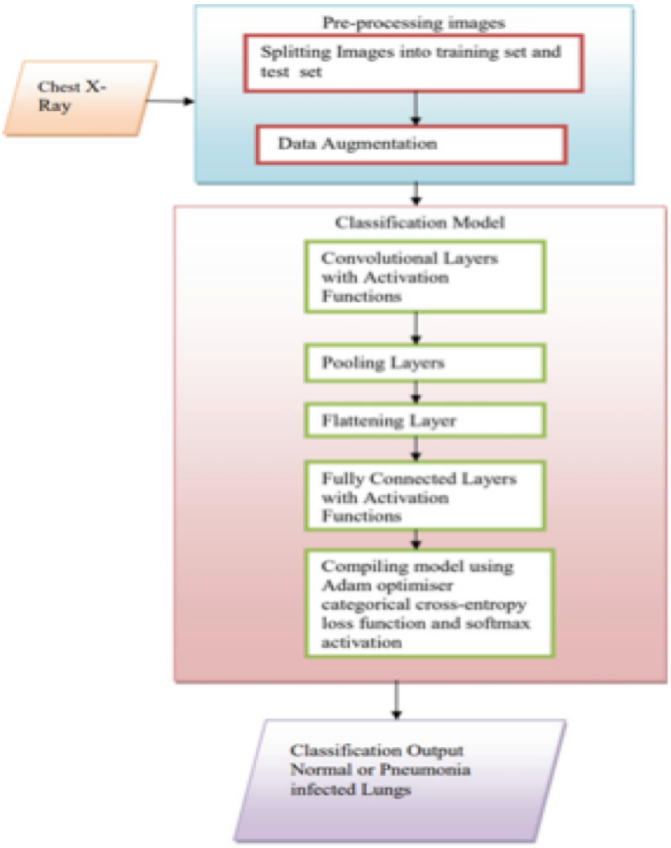

Fig. 1. CNN Architecture

- Flattening layer and fully connected layers - The input image is delivered into the flattening layer after passing through the convolutional and pooling layers. This layer reduces the computational cost of the incoming image by flattening it into a column. This is subsequently fed into the thick layer/fully linked layer. Every node in the first layer is connected to every node in the second layer in the completely connected layer, which has numerous levels. Each layer in the fully connected layer extracts features, and the network provides a prediction based on these features. Forward propagation is the term for this phenomenon. A cost function is calculated after forward propagation. It’s a metric measuring how well a neural network model performs. The categorical cross-entropy cost function is utilized in all four models. Back propagation occurs after the cost function has been determined. This process is repeated until the network achieves optimum performance.

B. CNN Model

There are four layers in a CNN model: an input layer, a hidden layer, a dense layer, and an output layer.

- the first layer we’ll make is an input layer with 64 nodes and the activation function relu.

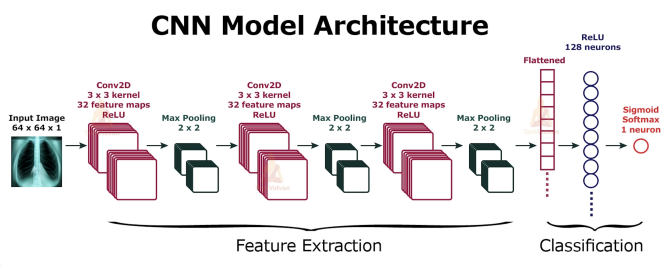

Fig. 2. CNN Model

- We'll next add two more Conv2D layers, each with 64 nodes, and relu as an activation function.
- Then, employing relu as an activation function, we'll make a Dense layer with 128 nodes.
- Finally, we'll create a single-node output layer with sigmoid as the activation function.

C. Multilayer Perceptron Architecture

- Input and output layers, as well as one or more hidden layers with numerous neurons layered together, make up a Multilayer Perceptron.
- While neurons in a Perceptron must have an activation function that enforces a threshold, such as ReLU or sigmoid, neurons in a Multilayer Perceptron can have any activation function they like.
- Because inputs are integrated with initial weights in a weighted sum and applied to the activation function, the Multilayer Perceptron falls under the category of feedforward algorithms. Each linear combination, on the other hand, gets propagated to the next layer.
- Each layer feeds the output of its computation, or internal representation of the data, to the next. This goes all the way through the hidden layers to the output layer..
- But it has more to it, the algorithm would be unable to learn the weights that optimize the cost function if it merely computed the weighted sums in each neuron, propagated the results to the output layer, and then stopped there. There would be no learning if the algorithm simply computed one iteration. This is where Backpropagation comes into play.

a) Backpropagation:

- Backpropagation is a learning process that allows the Multilayer Perceptron to iteratively alter the network's weights in order to minimize the cost function.
- Backpropagation must meet one stipulation in order to function effectively. The threshold function, such as ReLU, and the function that mixes inputs and weights in a neuron, such as the weighted sum, must both be differentiable. Gradient Descent is often the optimization function used in Multilayer Perceptron hence these functions must have a bounded derivative.
- The gradient of the Mean Squared Error is computed over all input and output pairs in each iteration, after the weighted sums have been sent through all layers.

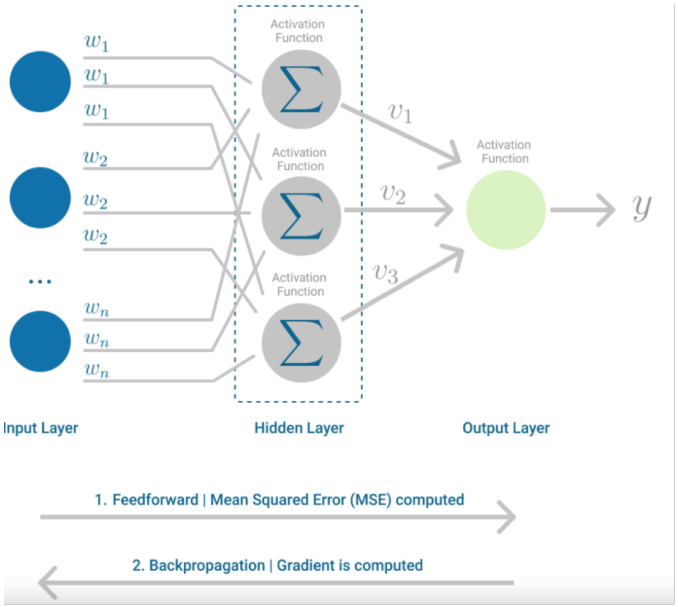

Fig. 3. MLP Architecture

- The weights of the first hidden layer are then modified with the gradient value to propagate it back.
- The weights are propagated back to the neural network's starting point in this manner. This process continues until each input-output pair's gradient has converged, which means the freshly computed gradient hasn't changed more than a set convergence threshold since the previous iteration.

D. Dataset

The dataset is divided into three folders (train, test, and val) with subfolders for each image type (Pneumonia/Normal). There are 5,863 X-Ray images (JPEG) in total, divided into two groups (Pneumonia/Normal). Anterior-posterior chest X-ray images were chosen from retrospective cohorts of children patients aged one to five years old at Guangzhou Women and Children's Medical Center in Guangzhou. All chest X-ray imaging was done as part of the patients' regular medical treatment. To analyze chest x-ray images, all chest radiographs were first reviewed for quality control, with any scans that were low quality or unreadable being removed. After that, the diagnoses for the photographs were rated by two experts before being approved for use in the AI system. The evaluation set was also examined to account for any grading issues.

V. Experimental Results

Validation accuracy, recall, and F1 score were used as performance indicators to assess the performance of the MLP and CNN classifier models. Accuracy and loss graphs were examined as well. In addition, the confusion matrix was calculated. We ran the trials 5 times each for three hours each to evaluate and validate the effectiveness of the proposed technique. Parameters and hyper parameters were heavily tweaked to improve the model's performance. Different outcomes were

obtained,however only the most reliable were reported in this study.

A. MLP Output

Training loss = 0.2153, training accuracy = 0.7640 are the final results.

Total count of train , test and validation images available for the model

```
[23] data,labels0=zip(*trainset)
     test,testlabels0=zip(*testset)
```

```
[25] print('Total number of images in training dataset :',len(data))
     print('Number of images of NORMAL:',sum(labels0))
     print('Number of images of PNEUMONIA :',len(data)-sum(labels0))

     Total number of images in training dataset : 5216
     Number of images of NORMAL: 1341
     Number of images of PNEUMONIA : 3875
```

```
[26] print('Total number of images in testing dataset :',len(test))
     print('Number of images of NORMAL:',sum(testlabels0))
     print('Number of images of PNEUMONIA :',len(test)-sum(testlabels0))

     Total number of images in testing dataset : 624
     Number of images of NORMAL: 234
     Number of images of PNEUMONIA : 390
```

Fig. 4.

```
[36] Iteration 64, loss = 0.09132124
     Iteration 65, loss = 0.09928209
     Iteration 66, loss = 0.08162910
     Iteration 67, loss = 0.21827532
     Iteration 68, loss = 0.25790310
     Iteration 69, loss = 0.11608467
     Iteration 70, loss = 0.07224194
     Iteration 71, loss = 0.08881358
     Iteration 72, loss = 0.07511794
     Iteration 73, loss = 0.08537008
     Iteration 74, loss = 0.09135324
     Iteration 75, loss = 0.07125026
     Iteration 76, loss = 0.06199566
     Iteration 77, loss = 0.05655142
     Iteration 78, loss = 0.09676908
     Iteration 79, loss = 0.07606305
     Iteration 80, loss = 0.06779951
     Iteration 81, loss = 0.05773968
     Iteration 82, loss = 0.07015361
     Iteration 83, loss = 0.11464375
     Iteration 84, loss = 0.12454708
     Iteration 85, loss = 0.08111726
     Iteration 86, loss = 0.06007177
     Iteration 87, loss = 0.07168772
     Iteration 88, loss = 0.11724914
     Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
     0.7640449438202247
```

Accuracy of the model = 76.404%

Fig. 5. MLP Accuracy

```
[38] from sklearn.metrics import classification_report
     target_names = ['Pneumonia', 'Normal']
     print(classification_report(testy.argmax(axis=1),predy.argmax(axis=1), target_names=target_names))
```

	precision	recall	f1-score	support
Pneumonia	0.73	0.99	0.84	389
Normal	0.95	0.40	0.56	234
accuracy			0.77	623
macro avg	0.84	0.69	0.70	623
weighted avg	0.81	0.77	0.74	623

Fig. 6. Performance of MLP

```
[39] prediction='/content/drive/MyDrive/chest_xray/train/NORMAL/IM-0125-0001.jpeg'
     image=load_img(prediction, grayscale=False, color_mode='grayscale', target_size=(150,150))
     image=img_to_array(image)
     image=image/255.0
     x=model.predict(image.reshape(1,22500)).argmax(axis=1)[0]
     if(x==0):
         print('Pneumonia')
     else:
         print('Normal')
     plt.imshow(load_img(prediction))
```

```
Normal
<matplotlib.image.AxesImage at 0x7f640a217350>
```

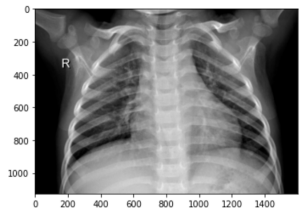

Fig. 7. Normal Chest X-ray

```
prediction ='/content/drive/MyDrive/chest_xray/train/PNEUMONIA/person22_bacteria_77.jpeg'
image=load_img(prediction, grayscale=False, color_mode='grayscale', target_size=(150,150))
image=img_to_array(image)
image=image/255.0
x=model.predict(image.reshape(1,22500)).argmax(axis=1)[0]
if(x==0):
    print('Pneumonia')
else:
    print('Normal')
plt.imshow(load_img(prediction))
```

```
Pneumonia
<matplotlib.image.AxesImage at 0x7f640a17f790>
```

Fig. 8. Pneumonia Chest X-ray

```
[37] predy = model.predict(testx)
     from sklearn.metrics import confusion_matrix,ConfusionMatrixDisplay
     cm=confusion_matrix(testy.argmax(axis=1),predy.argmax(axis=1))
     disp = ConfusionMatrixDisplay(confusion_matrix=cm,display_labels=['Pneumonia','Normal'])
     disp.plot()
     print("Accuracy of the model :",100*model.score(testx,testy))
```

```
Accuracy of the model : 76.40449438202246
```

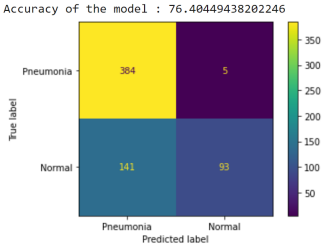

Fig. 9. Confusion Matrix

B. CNN Output for 20 epochs

Training loss = 0.1249, training accuracy = 0.9505, validation loss: 0.3216, and validation accuracy: 0.8750 are the final results

C. CNN Output for 35 epochs

Training loss = 0.1202, training accuracy = 0.9538, validation loss: 0.4962, and validation accuracy: 0.7500 are the final results.

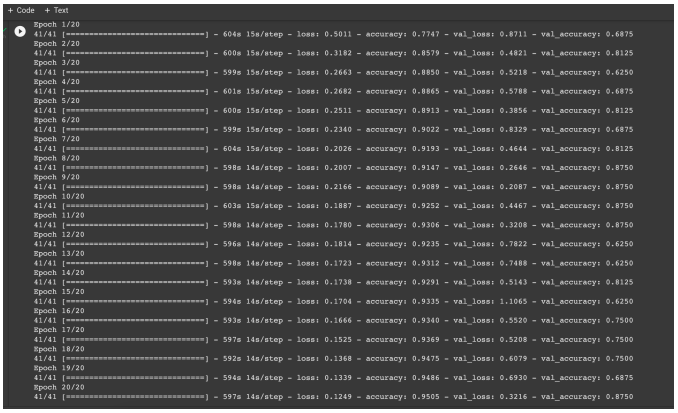

Fig. 10. CNN Execution for 20 epochs

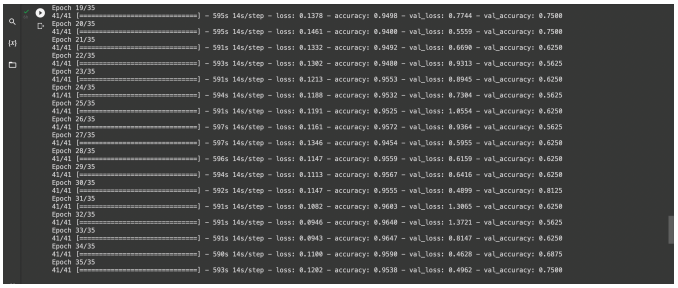

Fig. 11. CNN Execution for 35 epochs

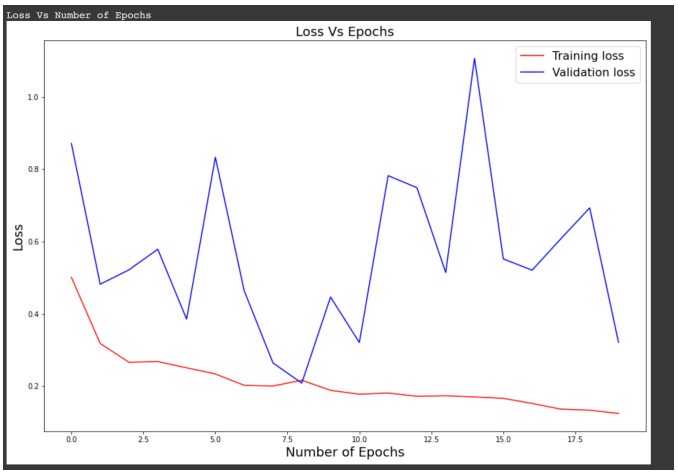

Fig. 12. Loss vs No.of Epochs

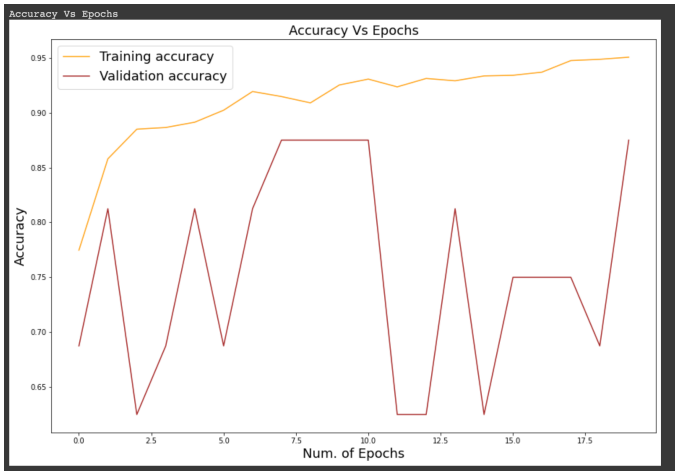

Fig. 13. Accuracy vs Epochs

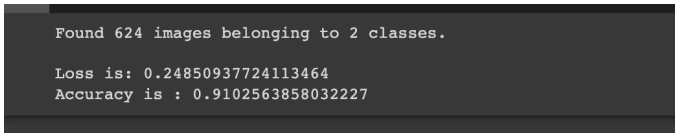

Fig. 14. CNN Accuracy

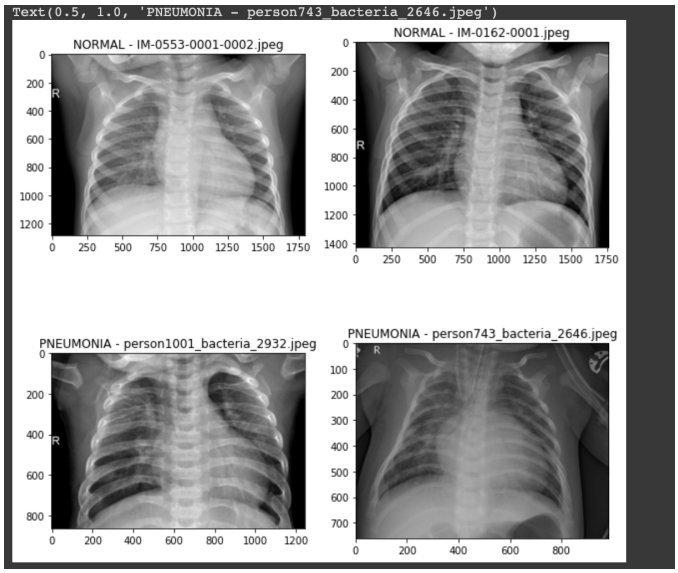

Fig. 15. Chest X-ray identification using CNN

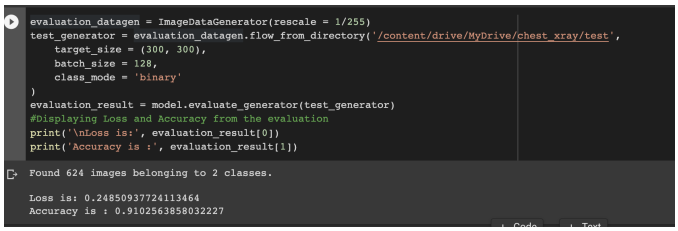

Fig. 16. Result of CNN with 20 epochs

Fig. 17. Result of CNN with 35 epochs

VI. Conclusion

The Accuracy of both the model CNN and MLP are 95.05 percent and 76.40 percent respectively. For CNN model, we have implemented the model for both 20 epochs and 35 epochs. The CNN model was over trained when the epochs was set to 35. Over fitting leads to high variance and the final validation accuracy drastically reduced. Hence, we do not want to over-train the model, so we are choosing the peak accuracy of our CNN model which is at 20 epochs. Now comparing CNN and MLP accuracy, CNN gives more accuracy then MLP, as a result, it can be stated that CNN classifier may be employed efficiently by medical officers for diagnostic purposes in the early detection of pneumonia in both children and adults. A huge number of X-ray images can be processed fast to produce extremely precise diagnostic results, allowing healthcare organizations to deliver more efficient patient care and lower mortality rates. Various parameter tuning strategies were used to construct these convolutional neural network model, including adding dropout, changing learning rates, changing batch size, number of epochs, adding more sophisticated fully connected layers, and adjusting various stochastic gradient optimizers.

VII. Future Work

Future research will concentrate on segmenting the areas of the lungs when and where pneumonia is prevalent. A historical progressive monitoring of the patient and affected region, supporting the pulmonologist in a visual examination of the patient's on-going condition, is another area of future research. On this data-set, it is hoped that transfer learning models will be developed that will outperform the CNN models. The models given in the research are also intended to be used to train larger data-sets. Neural network models based on GAN, or generative adversarial networks, are also expected to be taught and compared to current models.

References

[1] Lakhani, P., Sundaram, B.: Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. Radiology 284(2), 574–582(2017).

[2] Sirish Kaushik, Anand Nayyar, Gaurav Kataria, Rachna Jain: Pneumonia Detection Using Convolutional Neural Networks (CNNs), April 2020.

[3] Rohit Kundu, Ritacheta Das, Zong Woo Geem, Gi-Tae Han, Ram Sarkar :Pneumonia detection in chest X-ray images using an ensemble of deep learning models, September 2021. https://doi.org/10.1371/journal.pone.0256630.

[4] Chagas, J.V.S.d., de A. Rodrigues, D., Ivo, R.F. et al. A new approach for the detection of pneumonia in children using CXR images based on an real-time IoT system. J Real-Time Image Proc 18, 1099–1114 (2021). https://doi.org/10.1007/s11554-021-01086-y.

[5] Nada M. Elshennawy, Dina M. Ibrahim: Deep-Pneumonia Framework Using Deep Learning Models Based on Chest X-Ray Images, 2020 Aug 28. doi: 10.3390/diagnostics10090649.

[6] L. Račić, T. Popović, S. čakić and S. Šandi, "Pneumonia Detection Using Deep Learning Based on Convolutional Neural Network," 2021 25th International Conference on Information Technology (IT), 2021, pp. 1-4, doi: 10.1109/IT51528.2021.9390137.

[7] S. Singh, "Pneumonia Detection using Deep Learning," 2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE), 2021, pp. 1-6, doi: 10.1109/ICNTE51185.2021.9487731.

[8] Shangjie Yao, Yaowu Chen, Xiang Tian and Rongxin Jiang: Pneumonia Detection Using an Improved Algorithm Based on Faster R-CNN, 2021. https://doi.org/10.1155/2021/8854892

[9] Alzubaidi, L., Zhang, J., Humaidi, A.J. et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data 8, 53 (2021). https://doi.org/10.1186/s40537-021-00444-8

[10] Yamashita, R., Nishio, M., Do, R.K.G. et al. Convolutional neural networks: an overview and application in radiology. Insights Imaging 9, 611–629 (2018). https://doi.org/10.1007/s13244-018-0639-9

[11] Okeke Stephen, Mangal Sain, Uchenna Joseph Maduh, Do-Un Jeong, "An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare", Journal of Healthcare Engineering, vol. 2019, Article ID 4180949, 7 pages, 2019.https://doi.org/10.1155/2019/4180949

[12] Li, Y., Zhang, Z., Dai, C., Dong, Q., Badrigilan, S.: Accuracy of deep learning for automated detection of pneumonia using chest X-ray images: a systematic review and meta-analysis. Comput. Biol. Med. 123, 103898 (2020). https://doi.org/10.1016/j.compbiomed.2020.103898

[13] Labhane G., Pansare R., Maheshwari S., Tiwari R., Shukla A. Detection of Pediatric Pneumonia from Chest X-Ray Images using CNN and Transfer Learning; Proceedings of the 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE); Jaipur, India. 7–8 February 2020; pp. 85–92.

[14] Rahman T., Chowdhury M.E., Khandakar A., Islam K.R., Islam K.F., Mahbub Z.B., Kadir M.A., Kashem S. Transfer Learning with Deep Convolutional Neural Network (CNN) for Pneumonia Detection using Chest X-ray. Appl. Sci. 2020;10:3233. doi: 10.3390/app10093233.

[15] Stephen O., Sain M., Maduh U.J., Jeong D.U. An efficient deep learning approach to pneumonia classification in healthcare. J. Healthc. Eng. 2019;2019:4180949. doi: 10.1155/2019/4180949.

[16] Rajpurkar P., Irvin J., Zhu K., Yang B., Mehta H., Duan T., Ding D., Bagul A., Langlotz C., Shpanskaya K., et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. arXiv. 20171711.05225.

[17] Eckle, K., Schmidt-Hieber, J.: A comparison of deep networks with ReLU activation functionand linear spline-type methods. Neural Netw. 110, 232–242 (2019).

[18] Maghdid et al., "Diagnosing COVID-19 pneumonia from X-ray and CT images using deep learning and transfer learning algorithms", arXiv preprint, 2020.

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

VIII. Task distribution

- Approach strategy - Pratheeksha
- Literature review - Supriya
- Data acquisition - Sruthi
- Algorithm and code - Everyone
- Accuracy and Result analysis - Sahithi
- Progress PPT Data Collection - Sahithi, Pratheeksha
- Progress PPT Preparation - Sruthi, Supriya
- Final PPT Data Collection- Supriya, Pratheeksha
- Final PPT Preparation- Sruthi, Sahithi
- Final Report data - Everyone
- Final Report - Everyone