Multi-class Chest X-ray classification of Pneumonia, Tuberculosis and Normal X-ray images using ConvNets

Rachita Mogaveera[1], *Roshan* Maur[1] *, *Zeba* Qureshi[1] and *Yogita* Mane[2]

[1]Student, Dept. of Information Technology, Universal College of Engineering, Maharashtra, India
[2]Professor, Dept. of Information Technology, Universal College of Engineering, Maharashtra, India

Abstract. Pneumonia and Tuberculosis (TB) are two serious and life-threatening diseases that are caused by a bacterial or viral infection of the lungs and have the potential to result in severe consequences within a short period of time. Therefore, early diagnosis is a significant factor in terms of a successful treatment process. Chest X-Rays which are used to diagnose Pneumonia and/or Tuberculosis need expert radiologists for evaluation. Thus, there is a need for an intelligent and automatic system that has the capability of diagnosing chest X-rays, and to simplify the disease detection process for experts and novices. This study aims to develop a model that will help with the classification of chest X-ray medical images into normal vs Pneumonia or Tuberculosis. Medical organizations take a minimum of one day to classify the diagnosis, while our model could perform the same classification within a few seconds. Also, it will display a prediction probability about the predicted class. The model had an accuracy, precision and recall score over 90% which indicates that the model was able to identify patterns. Users can upload their respective chest X-ray image and the model will classify the uploaded image into normal vs abnormal.

1 Introduction

Pneumonia and Tuberculosis are caused by viruses or bacteria, and less often, other microorganisms[1]. Nowadays, chest X-ray (CXR) imaging is commonly used for health intensive care and analysis of many lung diseases such as Pneumonia and Cancer because of the relatively low costs[2]. Therefore, Chest X-rays are the best tool for diagnosing diseases related to which has played a significant role in clinical care and epidemiological research. However, detecting Pneumonia and Tuberculosis in chest X-rays is a challenging task that relies on the availability of expert radiologists. However, not every doctor has high quality medical tools to diagnose patients . As a result, sometimes, their diagnosis is not very accurate. It is also much harder to judge Pneumonia and Tuberculosis just by looking at chest X-rays images.

With the advancements of information technology, Statistical analysis, Machine Learning, and Deep Learning algorithms have been successfully applied to many healthcare problems and have explained complex relationships and improved clinical predictions[3]. They can solve computer vision problems in the medical domain.. In recent years, researchers have proposed different Machine Learning based solutions for medical problems. The objective of this paper is to create a custom CNN model and detect the diseases from their chest X-rays with higher accuracy, recall, precision and F1-Score. Classification methods are among the most commonly used techniques in medical imaging, where the goal is building classifiers capable of predicting whether X-ray images are normal or abnormal (i.e whether it is Pneumonia or Tuberculosis).

Identifying and analyzing an X-ray image might take a minimum of one day by a medical organization or a doctor and that could be troublesome for patients and therefore, to benefit the doctors as well as decrease the waiting period for patients, this project would truly have an adverse effect on the world. With the help of Deep learning we would be able to build a model and create a website which would return you the result of your Chest X-rays in a few seconds thus, classifying if you are diagnosed with Tuberculosis or Pneumonia or a healthy chest X-ray.

2 Literature survey

Different classic Machine learning algorithms like SVM (Support Vector Machines) or Random Forest Classifiers can achieve a decent F1-score and accuracy score but not as good as a CNN model as discussed in [4].

In [5], the authors used a ResNet-50 model to classify the top ten most common chest diseases through X-ray images but the dataset was highly imbalanced which led to bias and thus, affected the model’s performance in inference mode.

Similar approach was adopted in [6], where a DenseNet model was able to classify two pathologies but dataset imbalance affected the model’s AUC score.

* Corresponding author:maurroshan17@gmail.com

3 Dataset and methods

3.1 Dataset

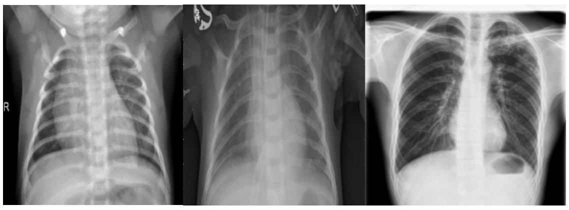


Fig. 1. X-ray images of Normal, Pneumonia and Tuberculosis.

The dataset used for this research is acquired through Kaggle. There were few individual datasets available for each of the three classes and we merged the datasets from all the sources to have an all-in-one dataset for our research. Kaggle is an open source platform for Data Science/Machine learning researchers and practitioners where they can find datasets or participate in Data Science hackathons. The sample images of each class from the dataset is shown in Fig 1.

The dataset contains three sub categories which are Normal, Pneumonia and Tuberculosis. In this dataset there were 9288 images of Normal chest X-rays, 4245 images of Pneumonia chest X-rays and 1888 images of Tuberculosis chest X-rays. All these images are in JPEG format. [7]

3.2 Balancing and data preprocessing

Preparing the data in other words, extracting information from the raw data, taking care of the class imbalance, eliminating duplicate images, preprocessing the data by normalizing the pixel values, resizing the images and adding some data augmentation.

After eliminating the duplicate images, and segregating the images into their respective classes a huge class imbalance occurred between the classes. There were over 9100 images for Normal chest X-rays, 4145 for Pneumonia chest X-rays and 1788 images for Tuberculosis chest X-rays. There are two types of Pneumonia infections: Pneumonia through virus and Pneumonia through bacteria. To avoid bias, a 50-50 ratio for both viral and bacterial infection was maintained in the Pneumonia class. In order to eliminate the class imbalance, we used the downsampling approach. The dataset was balanced by downsampling Pneumonia and Normal chest X-rays images to 1700 thereby maintaining balance between all the classes.

There were 5100 total images in the training set which included 1700 images for each class. There were 150 total images in the validation set which included 50 images for each class. There were 238 images in the test set which included 100 images for Pneumonia and Normal chest X-rays respectively and 38 images for Tuberculosis.

Once the dataset was prepared it was divided into training set, validation set and testing set. A general split ratio of 80:10:10 was used.

All the classes in the train folder have the same number of images. Since the images inside the training set, validation set and test set are of different sizes and ConvNets do not accept images of different sizes as inputs we had to perform data preprocessing.

3.2.1 Data preprocessing

For data preprocessing, we will resize the image, preprocess (normalize/rescale) the images and add some data augmentation techniques.

The images were resized to a certain height x width value. We used 256 x 256 px as the size of the image. The images were also converted into grayscale mode if any of the images were in RGB mode. That results in the image size as (256,256,1). The 1 in the end indicates that the image is in grayscale mode. We used grayscale mode because we are dealing with X-ray images.

Data augmentation techniques can not be applied to validation and test dataset. Data augmentation techniques add some variations to the training set. We used three data augmentation techniques: Random Zoom, Random Contrast and Random Rotation. We have selected a value of 0.3 (30%) for each parameter.

3.3 Creating optimized input pipeline

3.3.1 Prefetching

Rather than passing the whole data i.e the training set, validation set and the test set directly to the model, we created small batches of data. The size of the batches was 32. Normal batches of data will be converted to a prefetch dataset. The process carried out in Fig. 2 is the training part where our model receives the input and learns the patterns from the batch.

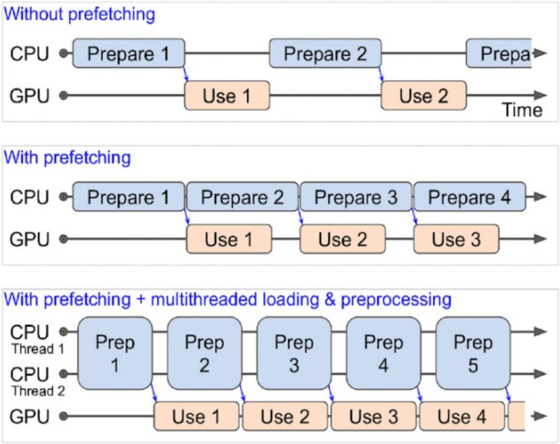


Fig. 2. Prefetch Dataset [8]

So, the first method is without a prefetch dataset, where the CPU first prepares a dataset and then passes it to the GPU where the model learns the patterns and updates its weights. Until the GPU has not completed training the first batch, the CPU cannot start the preparation for the second batch and thus, CPU and GPU work dependently, meaning they will wait for each other to finish their respective process in order to start working with the new batch.

Prefetch dataset helps us avoid this anomaly, meaning after the CPU passes on the first batch to the GPU for the model to train and learn the patterns, the CPU can start preparing the second batch, the third batch and so on, resulting in an optimized input pipeline and faster training time. Here, both the CPU and GPU are working independently. In the third process, which is prefetch dataset + multithreading processing and loading, is the optimal input pipeline. We have used this optimized input pipeline for our project. This works exactly like the prefetch dataset except it has more than two GPUs, so if the first batch is being used by GPU - 1, the second batch after preparation by the CPU would be used by GPU - 2 and so on. This speeds up the training process by 3x.

3.4 ConvNets

We built a custom model using the functional API. The reason for using the functional API was the flexibility it offers over the Sequential API. Fig. 4. shows the model architecture.

The layers used in this model are Convolutional layers, Max Pooling layers, a Global Max Pooling layer, Dense layers, a Batch Normalization layer and a Dropout layer. Categorical Cross Entropy is used to calculate the loss with a label smoothing of 0.2 (20%) to refrain the model from being too confident on any prediction. Adam optimizer was used as the learning rate optimizer with a default learning rate of 0.001. The three metrics, accuracy, precision and recall were used to calculate the performance of the model along with the loss. The model was trained for 80 epochs and we used two callbacks. The Model checkpoint callback and the Early Stopping callback.

Batch Normalization layer helps to normalize the output from activation layers. It is entirely possible that the output could have a very small value or a very large value and that may affect the weights of the following hidden layers accordingly. Thus, to avoid both the exploding gradient problem and the vanishing gradient problem, the Batch Normalization layer is used to normalize the output from the activation functions.

When defining the dense layer we set the units i.e. the output size of the next layer and the activation function. For our project we used 64 units and the ReLU activation function for our first dense layer. Another dense layer is applied with smaller units i.e. 32 and lastly, another dense layer of units 3 with activation function softmax is applied to the final dense layer from which the output can be decided. Each neuron of the output layer will perform matrix-vector multiplication to decide its value. The output of the neurons is calculated by calculating the dot product of the vector and matrix.

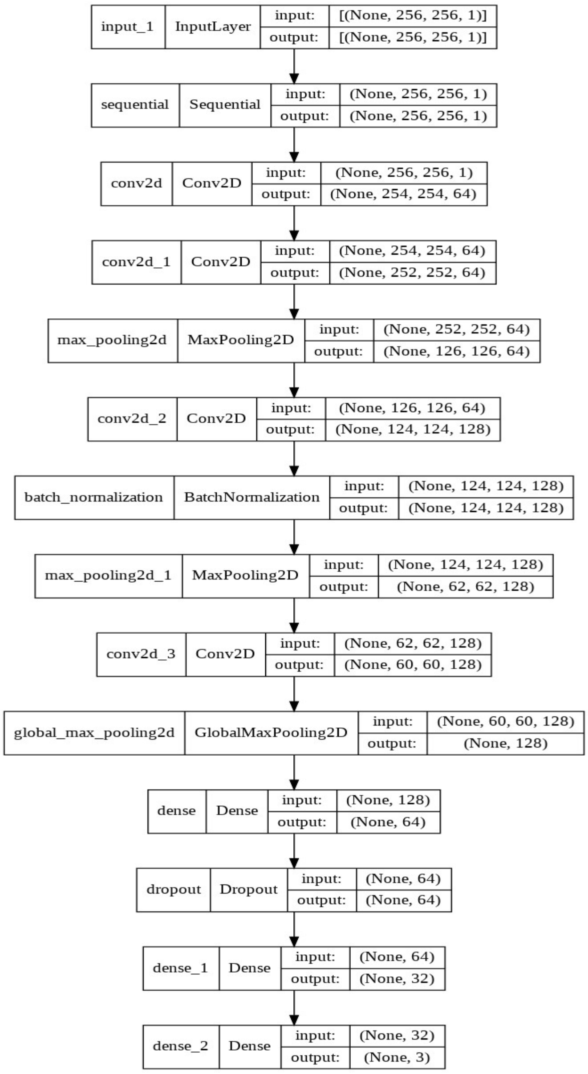

input_1 | InputLayer | input: [(None, 256, 256, 1)] | output: [(None, 256, 256, 1)]
sequential | Sequential | input: (None, 256, 256, 1) | output: (None, 256, 256, 1)
conv2d | Conv2D | input: (None, 256, 256, 1) | output: (None, 254, 254, 64)
conv2d_1 | Conv2D | input: (None, 254, 254, 64) | output: (None, 252, 252, 64)
max_pooling2d | MaxPooling2D | input: (None, 252, 252, 64) | output: (None, 126, 126, 64)
conv2d_2 | Conv2D | input: (None, 126, 126, 64) | output: (None, 124, 124, 128)
batch_normalization | BatchNormalization | input: (None, 124, 124, 128) | output: (None, 124, 124, 128)
max_pooling2d_1 | MaxPooling2D | input: (None, 124, 124, 128) | output: (None, 62, 62, 128)
conv2d_3 | Conv2D | input: (None, 62, 62, 128) | output: (None, 60, 60, 128)
global_max_pooling2d | GlobalMaxPooling2D | input: (None, 60, 60, 128) | output: (None, 128)
dense | Dense | input: (None, 128) | output: (None, 64)
dropout | Dropout | input: (None, 64) | output: (None, 64)
dense_1 | Dense | input: (None, 64) | output: (None, 32)
dense_2 | Dense | input: (None, 32) | output: (None, 3)

Fig. 3. Model architecture

Model checkpoint callback is a vital callback as it saves the best model after every epoch. By setting the monitor parameter it can monitor that variable for example, we set the monitor to validation loss so every time it had a lowest validation loss recorded, it will save the model in the path defined.

Early Stopping callback similarly monitors the validation loss and it stops training the model after there is no improvement in the model's validation loss. To stop the model from training we have to set the patience parameter meaning how many epochs it should consider before stopping the model from training further. The patience parameter was set to 10.

The Relu activation function formula is

$$f(x) = \max(0, z)$$

If a negative value is inputted the output is 0 else it is z. Due to this formula the output value is always a positive value. The purpose of the activation function is to provide non-linearity to our function and to set boundary condition rules.

The last step is to apply the categorical cross entropy function. It is a loss function and it uses the formula.

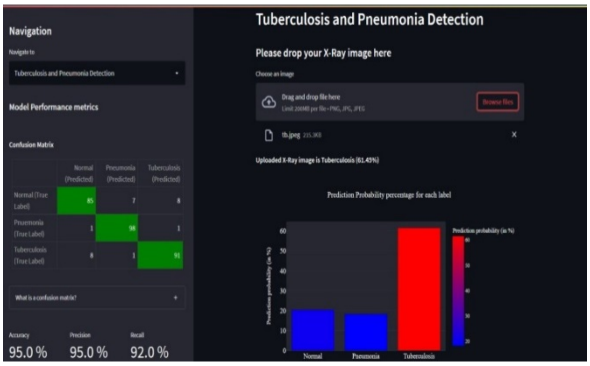
$$Loss = -\sum_{i=1}^{output\ size} yi \cdot \log yi$$

This function is used for multi-level categorical models. The sum of all the values of the neurons to which cross entropy is applied is 1. It is used to give the highest probability to the correct class and lowest to the other class.

4 Streamlit web-app

In this research work, we used Streamlit for deploying the model. Streamlit is one of the best Python frameworks to create a quick web-app with an attractive user-interface. Our Streamlit web-app consists of three pages:

The first page as shown in Fig. 4. as the name suggests is where the user can upload his/her chest X-ray image and the model will classify the image. It will also create a simple bar graph to represent the confidence for each class. The sidebar on the first page will also have the confusion matrix for the universal test dataset and the other metrics of the model.


Navigation
Navigate to
Tuberculosis and Pneumonia Detection
Choose an image
Drag and drop file here
Limit 200MB per file • PNG, JPG, JPEG
Browse files
tb.jpeg 215.3KB
What is a confusion matrix?
Prediction Probability percentage for each label
Prediction probability (in %)
0
10
20
30
40
50
60
Normal
Pneumonia
Tuberculosis
Prediction probability (in %)
20
30
40
50
60

Tuberculosis and Pneumonia Detection

Please drop your X-Ray image here

Uploaded X-Ray image is Tuberculosis (61.45%)

Model Performance metrics

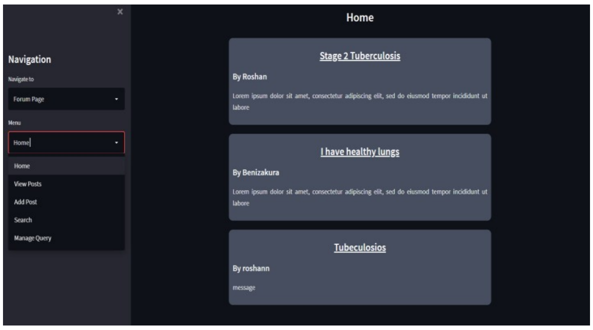
Confusion Matrix

	Normal (Predicted)	Pneumonia (Predicted)	Tuberculosis (Predicted)
Normal (True Label)	85	7	8
Pnuemonia (True Label)	1	98	1
Tuberculosis (True Label)	8	1	91

Accuracy	Precision	Recall
95.0 %	95.0 %	92.0 %

Fig. 4. Tuberculosis and Pneumonia Detection page

The second page will consist of a forum page shown in Fig.5. where users can post their queries, search queries and read about other people's queries. It will help the users to interact with each other, recommend methods that worked for them and even recommend hospitals as to where they received the best treatment for any particular disease.



Home

Stage 2 Tuberculosis

By Roshan

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore

I have healthy lungs

By Benizakura

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore

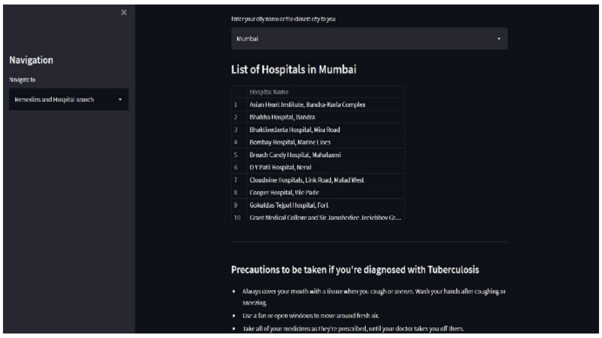
Tubeculosios

By roshann

message

Fig. 5. Forum Page

The third page Fig. 6. is where users can search about different hospitals. The dataset for the following hospitals filtered out by their respective locations was scraped from Wikipedia[9] using Beautiful Soup library. After scraping, it was converted into a Pandas dataframe. There is also a remedies note if you're diagnosed with Pneumonia or Tuberculosis which will automatically get displayed on the screen after you've uploaded your image on the first page.



Enter your city name or the closest city to you

Mumbai

List of Hospitals in Mumbai

	Hospital Name
1	Asian Heart Institute, Bandra-Kurla Complex
2	Bhabha Hospital, Bandra
3	Bhaktivedanta Hospital, Mira Road
4	Bombay Hospital, Marine Lines
5	Breach Candy Hospital, Mahalaxmi
6	D Y Patil Hospital, Nerul
7	Cloudnine Hospitals, Link Road, Malad West
8	Cooper Hospital, Vile Parle
9	Gokuldas Tejpal Hospital, Fort
10	Grant Medical College and Sir Jamshedjee Jeejebhoy Gr...

Precautions to be taken if you're diagnosed with Tuberculosis

- Always cover your mouth with a tissue when you cough or sneeze. Wash your hands after coughing or sneezing.
- Use a fan or open windows to move around fresh air.
- Take all of your medicines as they're prescribed, until your doctor takes you off them.

Fig. 6. Hospital search and remedies page

5 Performance Analysis

5.1 Comparision with previous works

There were some research papers in this domain where different diseases were identified from chest X-rays using Deep learning techniques. Few of them built their own custom ConvNets while others used pretrained models like AlexNet or ResNet or VGG networks and thus, leveraging the power of transfer learning. The research works we used for references as shown in Table.1 were some of the classical papers in this field.

Our dataset was created from different open source contributions on Kaggle. Many research papers used one specific dataset but we merged many different datasets to create an all-in-one dataset for Normal, Pneumonia and Tuberculosis. Rather than relying on transfer learning techniques we created our own custom ConvNet model that performed well on two different datasets i.e the balanced dataset and the unbalanced dataset. Also, our model was able to achieve an F1-score over 90% on both the datasets.

Table 1. Comparision with previous works

Name	Metrics used	Dataset imbalance	Pneumonia And Tuberculosis
[4]	Accuracy and F1-score	Chest X-ray images with Pnuemonia were over 4200 whereas total number of Normal Chest X-ray images were around 1200.	Only Pneumonia
[5]	Specificity and Sensitivity	High class imbalance.	10 Chest X-ray images but none included Pnuemonia or Tuberculosis
[6]	AUC score	High class imbalance led to a good AUC score on pulmonary modules but a poor AUC score of 73% on cardiomegaly.	Pathologies like pulmonary modules and cardiomegaly.

5.2 Balanced dataset model vs Unbalanced dataset model

We used the same model architecture to train on the balanced dataset and the unbalanced dataset. To analyze the model's performance metrics on unseen data, we created an universal test dataset. The universal test dataset was created after eliminating duplicate images in the later stages of the data collection process. 100 images from each class were selected at random to create a universal test data to further analyze the model's performance metrics. The model trained on the balanced dataset for 80 epochs and the model trained on the unbalanced dataset for 44/80 epochs. We used the Early Stopping callback for the model trained on the unbalanced dataset to refrain overfitting.

The confusion matrices for the universal test dataset by the model trained on the balanced dataset and the model trained on the unbalanced dataset are in table 2 and table 3 respectively.

Table 2. Confusion matrix of model trained on balanced dataset

True/Pred	Normal (pred)	Pneumonia (pred)	TB (pred)
Normal (True)	85	7	8
Pneumonia (True)	1	98	1
TB (True)	9	.0	91

In table 2 the important takeaway is that the model trained on the balanced dataset was better able to classify both Tuberculosis and Pneumonia compared to Normal. It was able to classify 85 Normal chest X-ray images correctly compared to Pneumonia chest X-rays 98 correctly classified images and Tuberculosis chest X-rays 91 correctly classified images.

Table 3. Confusion matrix of model trained on unbalanced dataset

True/Pred	Normal (pred)	Pneumonia (pred)	TB (pred)
Normal (True)	95	4	1
Pneumonia (True)	9	90	1
TB (True)	14	2	84

In table 3 the important takeaway is that the model trained on the unbalanced dataset had a bias while classifying Tuberculosis. Since the images in the Normal class were 5x that of Tuberculosis, it had a bias factor which is clearly visible in the confusion matrix. 14 chest X-ray images with Tuberculosis were misclassified as Normal which indicates the bias factor that the model trained on the unbalanced dataset.

In the medical image classification domain, a False Negative is as bad as a False Positive. Comparatively, the model trained on the balanced dataset beats the model trained on the unbalanced dataset model slightly.

The performance metrics of the model trained on the balanced dataset and the unbalanced dataset on their respective training set, validation set and test set are in Fig. 7 and Fig. 8 respectively.

In Fig. 7, we have the metrics for the model trained on the balanced dataset, the F1 scores for the training set and test set are above 0.90 which indicates that the model is able to generalize well. Though the loss is over 0.60 for the training and validation set it plummeted to 0.58 which is the lowest recorded loss.

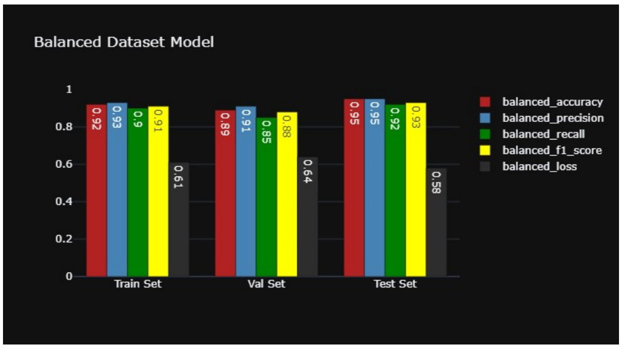


Fig. 7. Balanced dataset model performance metrics

In Fig. 8, we have the metrics for the model trained on the unbalanced dataset, the F1 scores for all the sets is around 0.90 though it was trained for only 44/80 epochs training because of the Early Stopping callback.

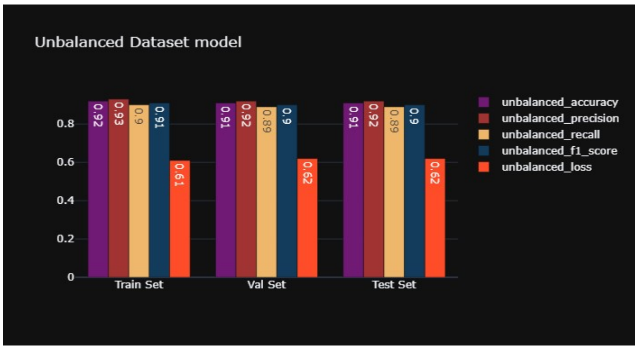


Fig. 8. Unbalanced dataset model performance metrics

6 Future Scope

This research paper has a few limitations for example, the dataset is not diverse enough. Since, the dataset was collected from open source platforms like Kaggle, the model is accustomed to inference bias like the patches of Tuberculosis or Pneumonia will differ for a child to that of an adult, different stages of patches in Tuberculosis and Pneumonia. Another limitation is the hospital feature search, since the list of hospitals were scraped from Wikipedia, there were only a limited number of hospitals added to the dataset, so we would like to add more hospitals from many different locations and add more information about different hospitals in the dataset so the user can feel satisfied. In future, we look forward to work along with medical organizations to create a robust model using a large and diverse dataset that will be able to classify different chest X-ray diseases.

7 Conclusion

This Deep Learning model is not created to replace the jobs of doctors, but rather help them and save more time for the patients. By adding different lung diseases with more images of different classes, medical organizations can identify the type of disease the patient may be diagnosed with as early as possible so as to initiate the medical procedure to cure the disease and on a larger scale, it would benefit both the medical organization as well as the patients. Our proposed model achieved an accuracy, precision and recall score of over 90%.

References

[1] R. T. Sousa, O. Marques, F. A. A. Soares, I. I. Sene Jr, L. L. de Oliveira, and E. S. Spoto, “Comparative performance analysis of machine learning classifiers in detection of childhood pneumonia using chest radiographs,” Procedia Computer Science, vol. 18, pp. 2579–2582(2013).

[2] W. H. Organization et al., “Standardization of interpretation of chest radiographs for the diagnosis of pneumonia in children,” World Health Organization, Tech. Rep.(2001).

[3] J. De Fauw, J. R. Ledsam, B. Romera-Paredes, S. Nikolov, N. Tomasev, S. Blackwell, H. Askham, X. Glorot, B. O’Donoghue, D. Visentin et al., “Clinically applicable deep learning for diagnosis and referral in retinal disease,” Nature medicine, vol. 24, no. 9, pp. 1342–1350,(2018).

[4] R. E. Al Mamlook, S. Chen and H. F. Bzizi, "Investigation of the performance of Machine Learning Classifiers for Pneumonia Detection in Chest X-ray Images," 2020 IEEE International Conference on Electro Information Technology (EIT),pp.098-104,doi:10.1109/EIT48999.2020.9208232(2022).

[5] Y. Dong, Y. Pan, J. Zhang and W. Xu, "Learning to Read Chest X-Ray Images from 16000+ Examples Using CNN," 2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE), 2017, pp. 51-57, doi: 10.1109/CHASE.2017.59.

[6] M. S. Majdi, K. N. Salman, M. F. Morris, N. C. Merchant and J. J. Rodriguez, "Deep Learning Classification of Chest X-Ray Images," 2020 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI), 2020, pp. 116-119, doi: 10.1109/SSIAI49293.2020.9094612.

[7] Roshan Maur, “Imbalanced Tuberculosis and Pneumonia dataset,” Kaggle.com, 2022. [Online]. Available:https://www.kaggle.com/roshanmaur/imbalanced-tuberculosis-and-pnuemonia-dataset. [Accessed: 27-Feb-2022]

[8] Géron, A. Learning with Scikit-Learn, Keras & TensorFlow. 2nd ed. O'Reilly Media, Inc., p.422.(2019)

[9] Wikipedia Contributors, “List of hospitals in India,” Wikipedia, 17-Feb-2022. [Online]. Available: https://en.wikipedia.org/wiki/List_of_hospitals_in_India. [Accessed: 27-Feb-2022]