PNEUMONIA DETECTION USING CNN ALGORITHM

Vadapalli Akarsh(18MIS1006)

Vellore Institue of Technology, Chennai

vadapalli.akarsh2018@vitstudent.ac.in

Setty Rashmika(18MIS1049)

Vellore Institue of Technology, Chennai

setty.rashmika2018@vitstudent.ac.in

Kuraparthi Thanmayee(18MIS1076)

Vellore Institue of Technology, Chennai

Kurparthi.thanmayee2018@vitstudent.ac.in

Gangalapudi Samyuktha REDDY(19MIA1058)

Vellore Institue of Technology, Chennai

gangalapudi.samyuktha2019@vitstudent.ac.in

ABSTRACT

(i) Title of the project:-

PNEUMONIA DETECTION USING CNN ALGORITHM

(ii) Brief explanation about project:-

Pneumonia is a life-threatening infectious disease affecting one or both lungs in humans commonly caused by bacteria called Streptococcus pneumoniae. One in three deaths in India is caused due to pneumonia as reported by World Health Organization (WHO). Chest X-Rays which are used to diagnose pneumonia need expert radiotherapists for evaluation. Thus, developing an automatic system for detecting pneumonia would be beneficial for treating the disease without any delay particularly in remote areas. Due to the success of deep learning algorithms in analyzing medical images, Convolutional Neural Networks (CNNs) have gained much attention for disease classification. In addition, features learned by pre-trained CNN models on large-scale datasets are much useful in image classification tasks. In this work, we appraise the functionality of pre-trained CNN models utilized as feature-extractors followed by different classifiers for the classification of abnormal and normal chest X-Rays. We analytically determine the optimal CNN model for the purpose. Statistical results obtained demonstrates that pretrained CNN models employed along with supervised classifier algorithms can be very beneficial in analyzing chest X-ray images, specifically to detect Pneumonia.

LITERATURE SURVEY:

A. Intelligent Pneumonia Identification from Chest X-Rays: A Systematic Literature Review:

This paper provides the modern literature on pneumonia classification in CXR images. Specifically, it analyzes the usability, goodness factors, and computational complexity of the algorithms that carry out those methods. It additionally discusses the quality, usability, size, and stability volume of the accessible datasets.Thus, many algorithms the usage of numerous methods are accessible from the research community. However, there's a loss of convenient literature that summarizes all of the present day convenient practices, in order to visualize what techniques to pick out as a real-time perspective, that are the to be accessible datasets, and what are the presently performed outcomes on this domain.

B. Pneumonia Detection Using CNN based Feature Extraction:

This paper basically pursuits to enhance the clinical adeptness in regions in which the supply of radiotherapists continues to be limited. Our look at facilitate the early analysis of Pneumonia to stop unfavourable consequences (which include death) in such faraway regions. So far, now no longer a lot work has been contributed to particularly to discover Pneumonia from the dataset. The improvement of algorithms on this area may be especially useful for offering higher health-care services.

C.) Pneumonia Detection Using X-Ray Image Processing Using CNN:

With the appearing computer technology, development on an automated device to hit upon pneumonia and treating the disorder is now viable particularly if the affected person is in a far off region and medical aids is limited. This examine intends to contain deep learning techniques to alleviate the problem. Convolutional Neural Network is optimized to carry out the complex challenge of detecting sicknesses like pneumonia to help health workers in analysis and feasible remedy of the disorder. The authors advanced numerous models to decide the great feasible model in detecting pneumonia with the maximum specific results.This examine has educated unique models of CNN, namely ResNet and VGGNet the usage of 1024 through 1024 decision of 26,684 dataset pictures. The end result accomplished a 97 percentaccuracy rate for VGGNet and the bottom rate is 74 percent executed through the ResNet version. The end result of data shows that the educated model turned into capable of discover Pneumonia through tested photos of chest X-ray.

D.) Pneumonia Detection Using Deep Learning Based on Convolutional Neural Network:

This paper describes the usage of deep learning so as to classify digital photos of chest X-rays in keeping with presence or absence of modifications constant with pneumonia. The implementation became primarily based totally on CNN model the use of Python programming and medical tools. Initial experiments show promising results, however greater studies is needed. Even so the version accuracy is incredibly high, nearly 90%, there's a chance of overfitting because of the scale of the dataset. Also, the accuracy approach that the prediction model should probably be used as a choice aid tool,however there may be nonetheless a lot of tasks to be done. The right diagnosis of any sort of disorder nevertheless calls for the involvement and presence of medical specialists. In order to construct an excellent and dependable disorder class model, it's miles very crucial to acquire as plenty information as possible. Further studies steps will consist of experimenting with numerous preprocessing and CNN configurations, information augmentation techniques, in addition to the usage of extra X-ray datasets with extra X-ray datasets with extra information labels displaying other pathologies.

E.) X-Ray Image based Pneumonia Classification using Convolutional Neural Networks

In this research, unique CNN primarily based totally architectures have been investigated and designed to deal with the Pneumonia category problem. The proposed system became meant to understand pneumonia cases primarily based totally on X-ray images. Moreover, information augmentation became used to conquer the incredibly low area of the facts and keep away from overfitting. ResNet-50 and DenseNet161 models have been adopted, custom designed and applied to hit upon tremendous pneumonia cases. The wellknown datasets and overall performance measures used in our experiments proved that DenseNet-161 primarily based totally models outperformed ResNet-50 primarily based totally once.One have to be aware that the principle issue of theproposed system lasts at its incapacity to discriminate among bacterial Pneumonia and viral Pneumonia.Particularly, this will assist the medical network in diagnosing COVID-19. Thus, using applicable X-ray datasets consisting of bacterial and viral Pneumonia could be taken into consideration as thereafter works.Also, deep ensemble models may be designed using diverse pre-educated models in an effort to refine the pneumonia detection overall performance via way of means of exploiting the range of the base models.

PROPOSED WORK:

(i) List the Modules in your project.

- ✓ Import libraries
- ✓ Process the images and resize them to the preferred size
- ✓ Preparing the training and testing data
- ✓ Visualize training images
- ✓ incoprating the validation data into the training data
- ✓ Data augmentation
- ✓ Implementation of Convolutional Neural Network
- ✓ Visualizing our training progress
- ✓ Prepare data for precision vs. recall and ROC
- ✓ Set thresholds for our model for results of accuracy,precision and recall and generating confusion matrix

(ii) List the concepts to be applied in your project.

1. CNN Algorithm:

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The benefit of using CNNs is their ability to develop an internal representation of a two-dimensional image. This allows the model to learn position and scale in variant structures in the data, which is important when working with images.

2. RELU Activation function:

ReLU helps to prevent the exponential growth in the computation required to operate the neural network. If the CNN scales in size, the computational cost of adding extra ReLUs increases linearly. We use ReLUs for the same reason we use any other non-linear activation function: To achieve a non-linear transformation of the data.

3. Sigmoid Activation function:

The sigmoid activation function, also called the logistic function, is traditionally a very popular activation function for neural networks. The input to the function is transformed into a value between 0.0 and 1.0.

4. MAXPOOLNG 2D:

Max pooling operation for 2D spatial data. Downsamples the input along its spatial dimensions (height and width) by taking the maximum value over an input window (of size defined by pool_size) for each channel of the input. The window is shifted by strides along each dimension.

5. CONV2D:

Conv2D parameter is the numbers of filters that convolutional layers will learn from. It is an integer value and also determines the number of output filters in the convolution.

(iii) Platform & Language to be used for Implementation.

Language: Python

Platform: Google Colab

DATASET:

The dataset used here is chest xray dataset which is taken from Kaggle and preprocessed The dataset consists of :

- 5216 training images of which 3815 are of Pneumonia and 1341 are normal images.
- 624 testing images of which 390 are of Pneumonia and 234 are normal.

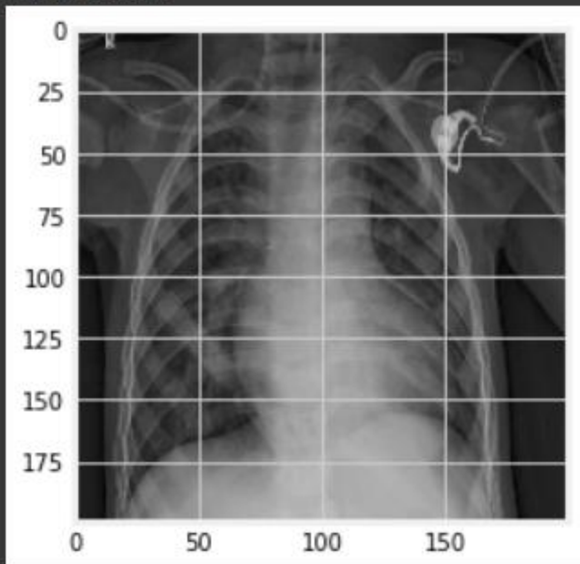
RESULTS AND DISCUSSION:

```
pnenumonia = 0
normal = 0

for i, j in train:
    if j == 0:
        pnenumonia+=1
    else:
        normal+=1

print('Pneumonia:', pnenumonia)
print('Normal:', normal)
print('Pneumonia - Normal:', pnenumonia-normal)
```

```
Pneumonia: 3875
Normal: 1369
Pneumonia - Normal: 2506
```

```
[ ]  plt.imshow(train[1][0], cmap='gray')
     print(labels[train[1][1]])
```

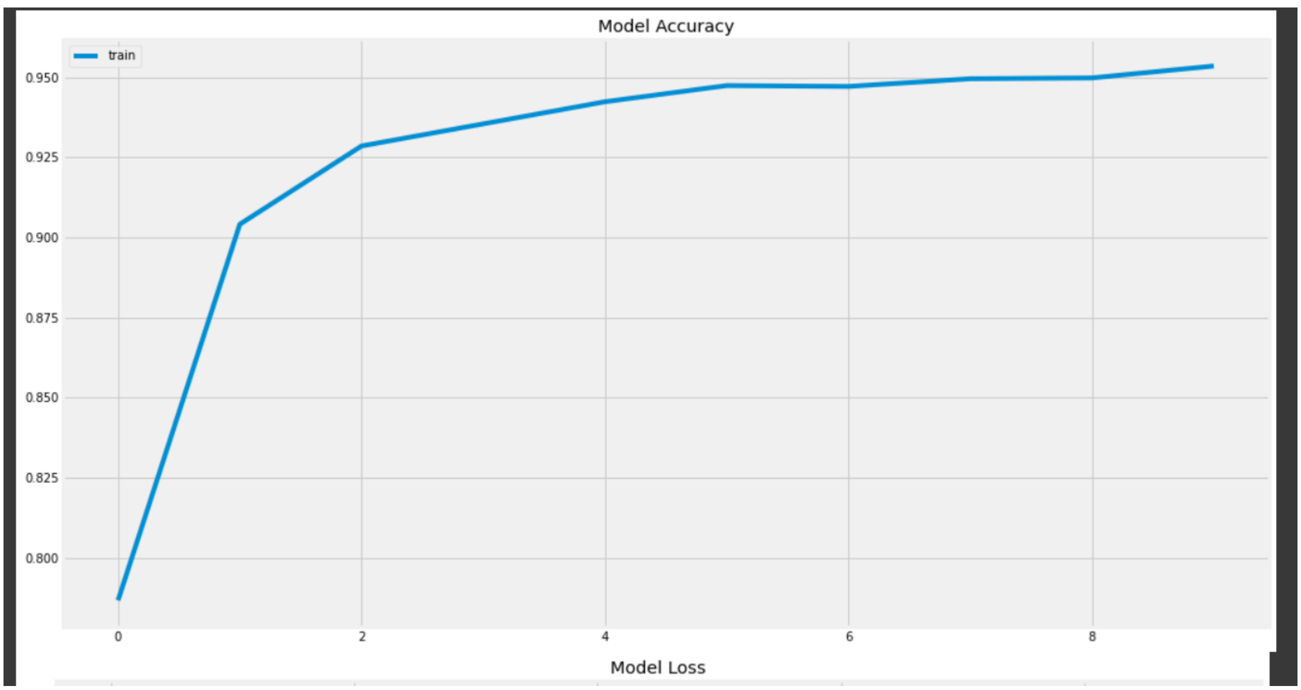
```
PNEUMONIA
```

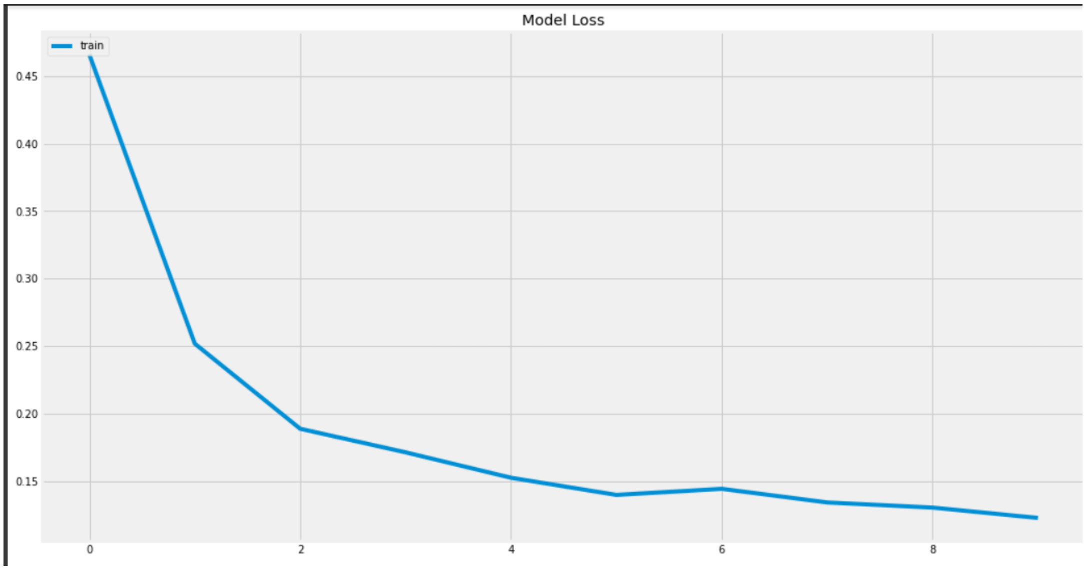


```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_3 (Conv2D)           (None, 200, 200, 256)     2560

 activation_5 (Activation)   (None, 200, 200, 256)     0

 max_pooling2d_3 (MaxPooling  (None, 100, 100, 256)    0
 2D)

 batch_normalization_3 (Batc  (None, 100, 100, 256)    400
 hNormalization)

 conv2d_4 (Conv2D)           (None, 100, 100, 64)      147520

 activation_6 (Activation)   (None, 100, 100, 64)      0

 max_pooling2d_4 (MaxPooling  (None, 50, 50, 64)       0
 2D)

 batch_normalization_4 (Batc  (None, 50, 50, 64)       200
 hNormalization)

 conv2d_5 (Conv2D)           (None, 50, 50, 16)        9232

 activation_7 (Activation)   (None, 50, 50, 16)        0

 max_pooling2d_5 (MaxPooling  (None, 25, 25, 16)       0
 2D)

 batch_normalization_5 (Batc  (None, 25, 25, 16)       100
 hNormalization)

 flatten_1 (Flatten)         (None, 10000)             0

 dropout_2 (Dropout)         (None, 10000)             0

 dense_2 (Dense)             (None, 64)                640064

 activation_8 (Activation)   (None, 64)                0

 dropout_3 (Dropout)         (None, 64)                0

 dense_3 (Dense)             (None, 1)                 65

 activation_9 (Activation)   (None, 1)                 0

=================================================================
Total params: 800,141
Trainable params: 799,791
Non-trainable params: 350
_________________________________________________________________
```

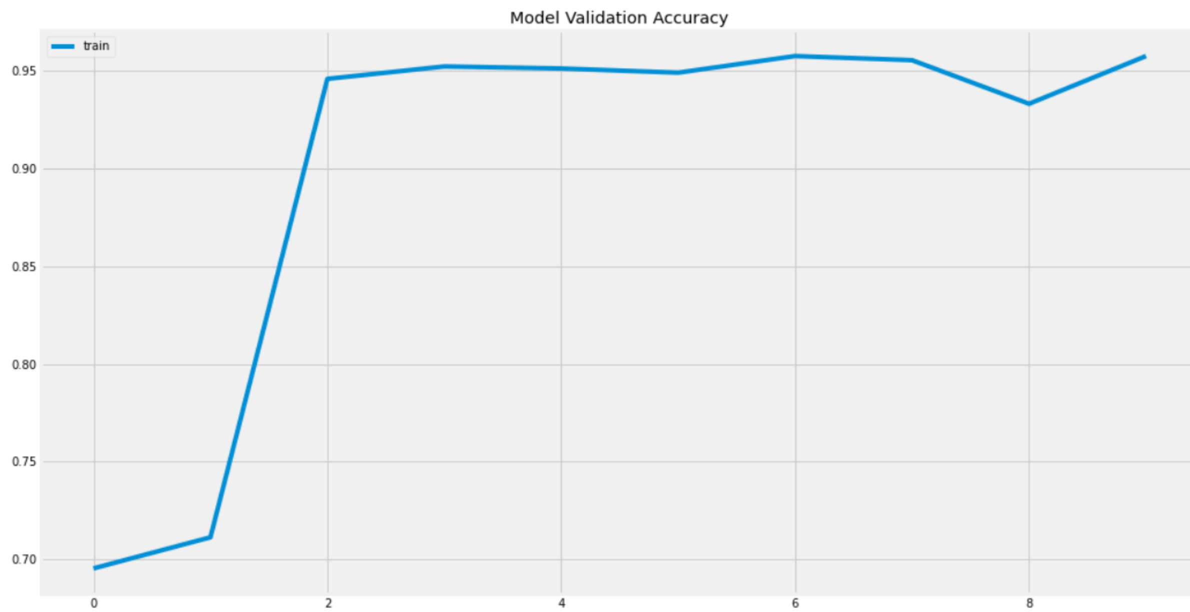
```
[ ] history = model.fit(X_train, y_train, batch_size=15, epochs=10, validation_split=0.20, callbacks=[early_stop])

    Epoch 1/10
    251/251 [==============================] - 1553s 6s/step - loss: 0.4658 - acc: 0.7867 - val_loss: 1.7246 - val_acc: 0.6953
    Epoch 2/10
    251/251 [==============================] - 1577s 6s/step - loss: 0.2518 - acc: 0.9041 - val_loss: 0.5132 - val_acc: 0.7113
    Epoch 3/10
    251/251 [==============================] - 1576s 6s/step - loss: 0.1888 - acc: 0.9286 - val_loss: 0.1430 - val_acc: 0.9459
    Epoch 4/10
    251/251 [==============================] - 1575s 6s/step - loss: 0.1713 - acc: 0.9355 - val_loss: 0.1396 - val_acc: 0.9522
    Epoch 5/10
    251/251 [==============================] - 1623s 6s/step - loss: 0.1525 - acc: 0.9424 - val_loss: 0.1352 - val_acc: 0.9512
    Epoch 6/10
    251/251 [==============================] - 1581s 6s/step - loss: 0.1398 - acc: 0.9474 - val_loss: 0.1345 - val_acc: 0.9490
    Epoch 7/10
    251/251 [==============================] - 1578s 6s/step - loss: 0.1443 - acc: 0.9471 - val_loss: 0.1358 - val_acc: 0.9575
    Epoch 8/10
    251/251 [==============================] - 1578s 6s/step - loss: 0.1342 - acc: 0.9495 - val_loss: 0.1321 - val_acc: 0.9554
    Epoch 9/10
    251/251 [==============================] - 1576s 6s/step - loss: 0.1304 - acc: 0.9498 - val_loss: 0.2019 - val_acc: 0.9331
    Epoch 10/10
    251/251 [==============================] - 1578s 6s/step - loss: 0.1227 - acc: 0.9535 - val_loss: 0.1373 - val_acc: 0.9575
```

```
[ ] model.evaluate(X_test, y_test)

    37/37 [==============================] - 129s 3s/step - loss: 0.6912 - acc: 0.6347
    [0.6912070512771606, 0.6346644163131714]
```

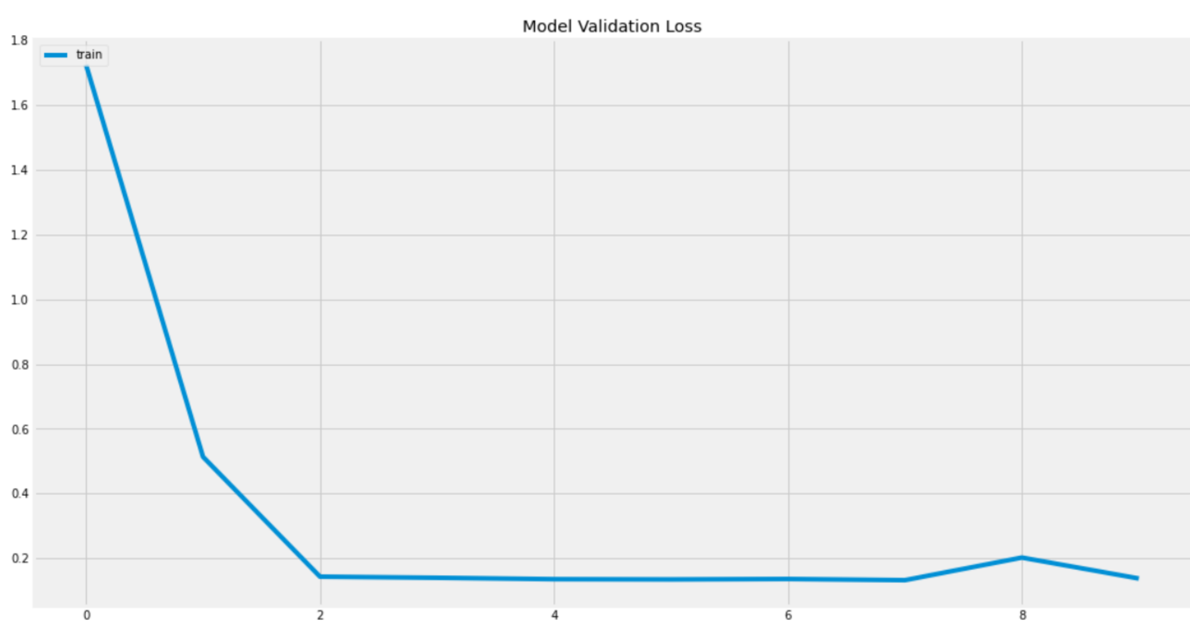




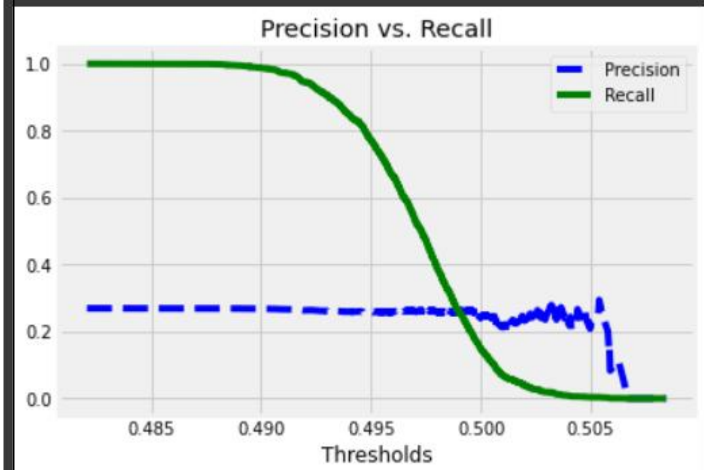
Model Loss
train
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0
2
4
6
8

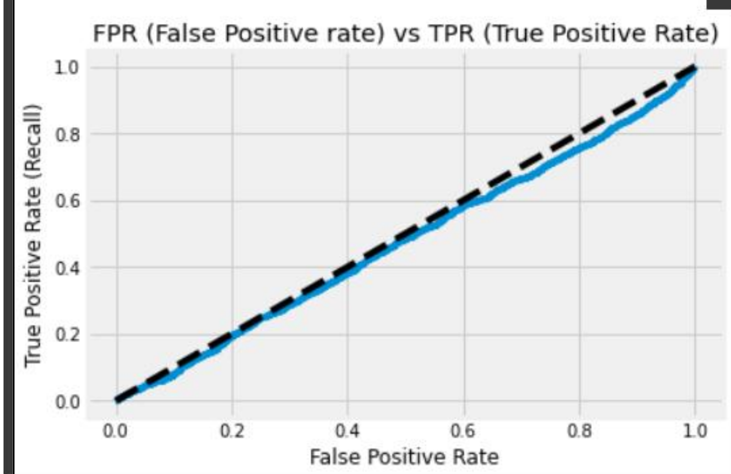


Model Validation Accuracy
train
0.95
0.90
0.85
0.80
0.75
0.70
0
2
4
6
8



Model Validation Loss
train
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0
2
4
6
8

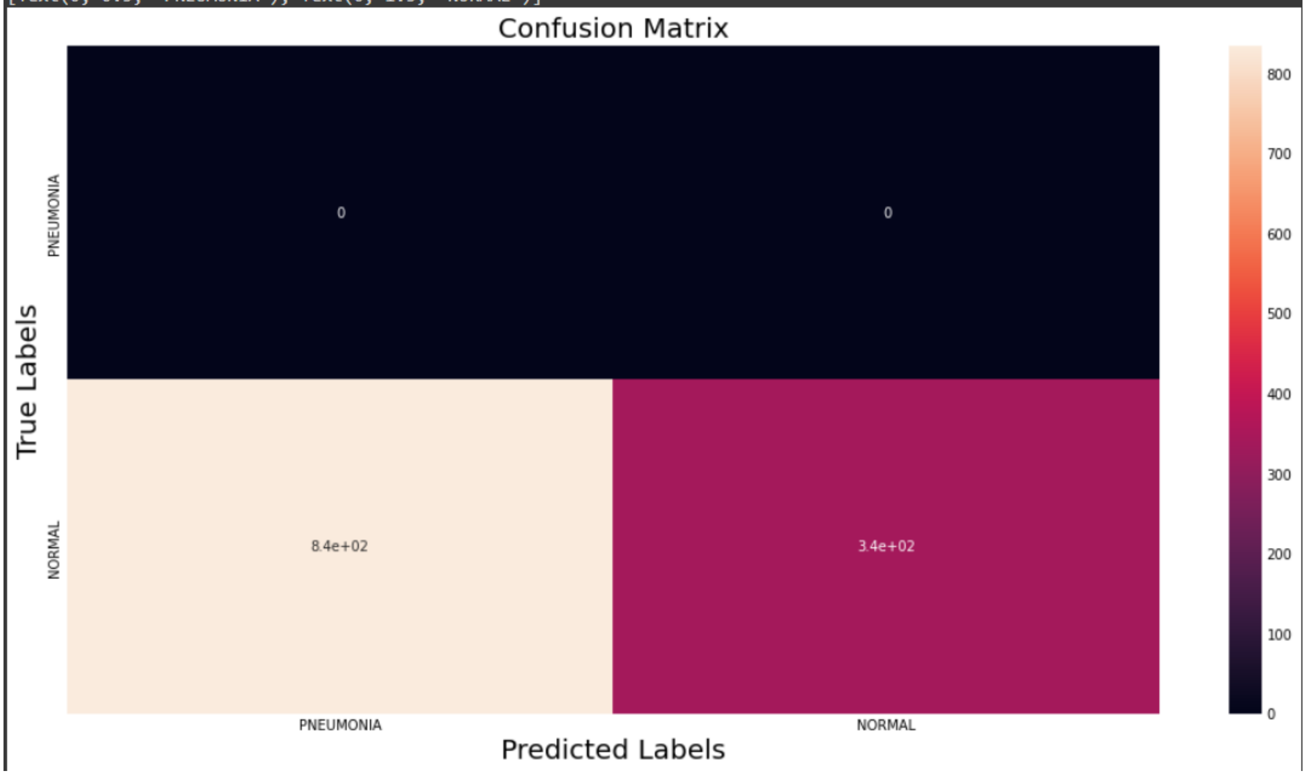


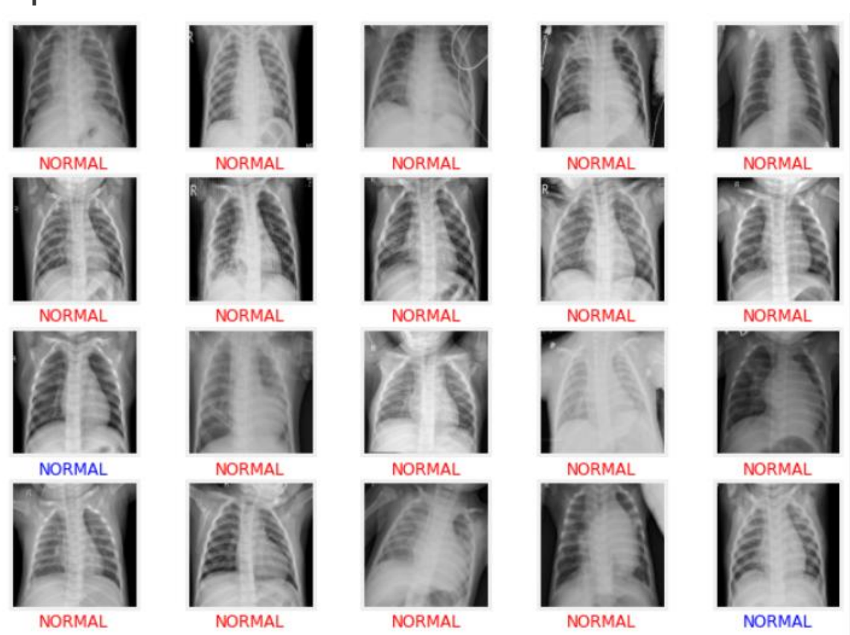



```
print('Accuracy on testing set:', accuracy_score(binary_predictions, y_test))
print('Precision on testing set:', precision_score(binary_predictions, y_test))
print('Recall on testing set:', recall_score(binary_predictions, y_test))
```

```
Accuracy on testing set: 0.2905692438402719
Precision on testing set: 1.0
Recall on testing set: 0.2905692438402719
```

```
[Text(0, 0.5, 'PNEUMONIA'), Text(0, 1.5, 'NORMAL')]
```





Conclusion:-

This research explains the usage of the algorithm of TF-IDF in an automatic textual content summarization software. Through this examinaton, it could be visible that the TF-IDF algorithm may be used as the powerful technique to supply an extractive summary. It generates the summary with 67% of accuracy, that is a higher result of the precis than other online summarizers.From the assessment end result among software summarizer and online summarizers via way of means of the usage of the statistical approach, it may be concluded that this system produces the higher summary. By the usage of the extractive technique, TF-IDF is validated as a effective technique to generate the cost which determines how vital a word in the file is. The value allows this system to determine which sentence for use withinside the a part of the summary. There are a few enhancements that may be implemented to this application to supply a extra accurate summary. First, it's far through making the summary biased at the name of the file. A identify is a sentence or word that describes the main event or what the article is. Therefore, a excessive value of TF-IDF may be given to the word that looks withinside the title in order that the program can produce a higher end result of the summary. Second, it's far via way of means of growing the wide variety of test with a numerous form of sample file to growth the accuracy to calculate precision, recall, and f-measure value. It is due to the fact the more files are summarized, the greater valid the end result of the average f-measure value becomes. Third, it must contain greater respondents to evaluate the system through determining the range of correct, wrong, or missed sentences withinside the summary. This technique will increase the validity of the test due to the fact the decision whether or not the sentence is the a part of the summary is decided among the respondents.

REFERENCES:

[1] Intelligent Pneumonia Identification from Chest X-Rays: A Systematic Literature Review, WASIF KHANNAZAR ZAKI and LUQMAN ALI,2020

[2] Pneumonia Detection Using CNN based Feature Extraction, Dimpy Varshni, Kartik Thakral, Lucky Agarwal, Rahul Nijhawan, Ankush Mittal,2019

[3] Pneumonia Detection Using X-Ray Image Processing Using CNN, MD.Tanvir Rahman,2021

[4] Pneumonia Detection Using Deep Learning Based on Convolutional Neural Network,Luka Račić, Tomo Popović, Senior Member, IEEE, Stevan Čakić, Stevan Šandi,2021

[5] X-Ray Image based Pneumonia Classification using Convolutional Neural Networks,2020, Sarah Badr AlSumairi and Mohamed Maher Ben Ismail