Prediction of Pneumonia Using Convolutional Neural Networks: A Deep Learning Approach

Ronak Agrawal
Department of Computer Science
School of Engineering, Amrita Vishwa Vidyapeetham
Kollam, India
AM.ENU4AIE21087
ronakdudhani@gmail.com

Abhinav Pandey
Department of Computer Science
School of Engineering, Amrita Vishwa Vidyapeetham
Kollam, India
AM.ENU4AIE21088
meronaamabhinav@gmail.com

Nikhil Kumar Singh
Department of Computer Science
School of Engineering, Amrita Vishwa Vidyapeetham
Kollam, India
AM.ENU4AIE21086
singhsannsar@gmail.com

K. Tejo Vardhan
Department of Computer Science
School of Engineering, Amrita Vishwa Vidyapeetham
Kollam, India
AM.ENU4AIE21039
tejovk311@gmail.com

Hari Sankar
Department of Computer Science
School of Engineering, Amrita Vishwa Vidyapeetham
Kollam, India
AM.ENU4AIE21056
harisankar1603@gmail.com

***Abstract*— Pneumonia is a fatal disease that majorly affects the elderly and may sometimes prove to be life-threatening. Early diagnosis of Pneumonia gains paramount importance for saving many human lives. This paper aims at the detection and classification of patients affected by Pneumonia based on their chest X-rays. A convolutional neural network is employed from scratch to make the above diagnosis and yield highly accurate results. Deep Learning models automate the process and ensure speedy, adroit, and adept results when provided with X-rays of patients. The classification occurs after the image is fed through a series of convolutional and max pooling layers that are activated by using the ReLU activation function that is subsequently fed into the neurons present in the dense layers and finally, the output neuron is activated by the sigmoidal function. The accuracy increases as the model trains and decreases the loss simultaneously. Overfitting is prevented by implementing data augmentation before fitting the model. Thus, efficient, and cogent results are obtained by the proposed deep learning models to classify the chest X-rays for the detection of pneumonia.**

Keywords— Pneumonia, Convolutional Neural Network (CNN), Max Pooling, Sigmoid Function, Rectified Linear Unit (ReLU), Data Augmentation, Deep Learning.

I. Introduction

Pneumonia is a life-threatening respiratory infection that necessitates timely and accurate diagnosis for effective treatment. In this research, we focus on leveraging convolutional neural networks (CNNs) to predict pneumonia using chest X-ray images. Chest X-ray images have become a widely adopted modality for pneumonia detection due to their ability to capture relevant anatomical and pathological features.

The dataset employed in this study consists of a carefully curated collection of chest X-ray images obtained from patients diagnosed with pneumonia. Each image in the dataset undergoes meticulous preprocessing to ensure consistent quality and reliable analysis. The images are standardized to a uniform size of 224x224 pixels. This comprehensive dataset is then divided into separate training and testing sets to facilitate model training and evaluation.

To predict pneumonia from the chest X-ray images, we employ two distinct CNN architectures. Both CNN architectures leverage multiple convolutional layers to capture hierarchical representations of the chest X-ray images. The convolutional layers employ learnable filters to convolve over the input images, extracting relevant features at different levels of abstraction. These features are then passed through activation functions, introducing non-linearities to enhance the model's symbolic power.

To prevent overfitting and improve generalization, the CNN architectures also incorporate regularization techniques such as dropout and batch normalization. Dropout randomly masks out a portion of the neurons during training, encouraging the network to learn more robust and independent features. Batch normalization normalizes the activations within each mini-batch, aiding in faster convergence and reducing the sensitivity to the initialization of network parameters. By comparing and analyzing the performance of these two CNN architectures on the pneumonia dataset, we aim to determine which model exhibits superior predictive capabilities for pneumonia detection. The outcomes of this research endeavor have the potential to contribute to the development of advanced and accurate diagnostic tools, facilitating early detection of pneumonia and ultimately enhancing patient care outcomes.

Pneumonia is recognized as one of the most threatening diseases by the WHO, with over a million premature deaths estimated worldwide. According to the World Health Organization, pneumonia causes nearly 15% of all deaths in children under five years of age, accounting for approximately 808,694 children in 2017. It affects around 450 million people globally each year and was the fourth leading cause of death in 2016.

Detecting pneumonia at an early stage through chest X-rays can be challenging as the X-rays can be ambiguous and misinterpreted as heart failure or lung cancer. Traditional machine learning models often fail to accurately detect such diseases due to their limitations, highlighting the need for advanced and more accurate deep learning models like convolutional neural networks (CNNs).

In the proposed methodology, CNNs are utilized to examine chest X-ray images and identify patterns indicative of pneumonia. The CNN architecture consists of multiple layers, including convolutional layers for automatic image recognition, max-pooling layers for downsampling, and rectified linear unit (ReLU) layers to improve non-linearity.

The paper follows a structured approach, beginning with explaining the proposed methodology and then providing detailed information about the dataset used, including chest X-ray images with and without pneumonia. The dataset is augmented to enhance performance. The deep learning architectural models, particularly CNN networks, are discussed in detail, along with the layers employed during implementation. The processed images after passing through the CNN layers are also analyzed.

Additionally, the paper presents a visual understanding of the data using statistical and graphical representations. The classification algorithms' results are depicted, and different epochs are compared. Finally, the paper concludes with the implications of implementing CNNs for pneumonia detection and discusses potential future directions in the field.

By combining the research insights from these two articles, we gain a comprehensive understanding of the application of CNNs in pneumonia detection from chest X-ray images. The utilization of CNN architectures, along with preprocessing techniques and regularization methods, holds promise in advancing diagnostic accuracy and contributing to improved patient care outcomes.

II. Related Work

Deniz Yagmur Urey et al. presented a research paper involving the early diagnosis of Pneumonia by using deep learning techniques. The authors posit a novel approach of focusing on the biological aspects of this fatal disease and detecting it by X-ray imaging. The classification methods used are Convolutional Neural Networks (CNN) and Residual Neural Networks. The Comparative study helps detect Pneumonia at an early stage and thus, appropriate treatment can be provided to cure the disease. This research influenced our paper, to conduct a similar assessment, but on a more generalized data, a plethora of images, and a detailed approach of layers of the neural network to make it more efficient and produce higher accuracy.

Dimpy Varshni et al. explained the development of an automatic system for the detection of pneumonia through various deep-learning models. The authors analyzed medical images and developed a Convolutional Neural network for disease classification and scaling of data. The architecture consists of a DenseNet-169 layer architecture for feature extraction. The architecture was combined with an SVM Model for binary classification. The results of the model are analyzed with visualization curves and a summary is provided for the same.

Garima Verma et. al. presented a research paper that analyzes and identifies pneumonia, based on X-ray images, using a convolutional neural network. The implementation was done using 6 convolutional layers followed by max-pooling layers after each. This provided us with an insight to incorporate a lesser number of convolutional layers, for faster computing and classification of the deep learning model. The research helps detect pneumonia, based on Chest X-rays.

Okeke Stephen et. al. provides a similar insight on classifying numerous X-rays to detect pneumonia based on convolutional neural networks. The accuracy obtained through their research helps us evaluate our model in comparison, depending on the loss and accuracy of the neural network. Their network is provided with a (200x200x3) dimension input shape, while the images are focused and it uses (64x64x3) dimensions, to decrease the computations. [6]

III. Proposed Methodology

In the above system, the image dataset i.e. chest X-ray images used is pre-processed and transformed using various NumPy and Pandas Libraries of Python Programming Language. Further, the data is augmented in this phase to provide enhanced and adequately efficient results in the various deep learning models. The data augmentation part considers the attributes like Rescale factor, Sheer Range, Rotation Range, and other numerous attributes. This data is then fed to the Deep Learning Model which is the Convolutional Network where it undergoes a series of steps. An optimizer named Adam is used to measure accuracy and the loss in the form of binary cross entropy in order to optimize the results during the compilation phase. Further, the model is trained and validated according to the convolutional network architecture.

By using the transfer learning technique in deep learning,we are trying to improve the performance of predictions happening in real time.

The output of this stage helps us to visualize and compute various curves and graphs displaying the dependency of loss, validation, and other myriad features. The output after all the steps obtained is the Final Classification System which diagnoses whether the patient has pneumonia or not.

IV. Dataset

The dataset used in our project for pneumonia detection using CNNs is a collection of chest X-ray images. The dataset consists of 5,863 images, which are categorized into two classes: Pneumonia and Normal.The dataset is organized into three folders: train, test, and value.The images are stored in JPEG format.

The chest X-ray images in this dataset were obtained from retrospective cohorts of pediatric patients aged one to five years old from Guangzhou Women and Children's Medical Center in Guangzhou. These chest X-ray images were captured as part of the patient's routine clinical care.

To ensure the quality and reliability of the dataset, all chest radiographs were initially screened for quality control, and any low-quality or unreadable scans were removed. The diagnoses for the remaining images were then graded by two expert physicians to determine the presence or absence of pneumonia. Additionally, to account for any grading errors, a third expert reviewed the evaluation set.

The dataset includes a range of chest X-ray images that exhibit different characteristics related to pneumonia. The normal chest X-ray images depict clear lungs without any areas of abnormal opacification. On the other hand, bacterial pneumonia images typically show focal lobar consolidation, often in specific lung lobes, whereas viral pneumonia images tend to display a more diffuse "interstitial" pattern in both lungs.

This carefully curated dataset provides a valuable resource for training and evaluating our CNN models for pneumonia detection. By leveraging this dataset, we aim to develop accurate and efficient models that can assist in the early and reliable detection of pneumonia from chest X-ray images, ultimately contributing to improved patient care outcomes.

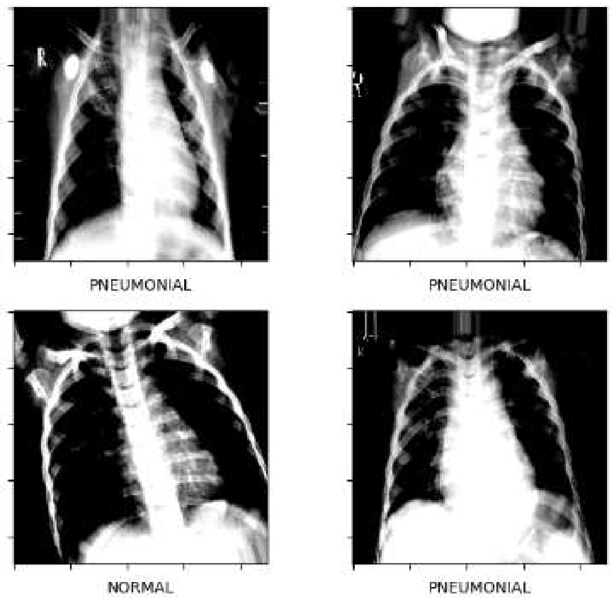


Fig. 1 Dataset

V. Data Augmentation

Data Augmentation techniques are extensively used to enhance the performance of deep learning algorithms. Data augmentation is primarily used to improve the performance of the convolutional neural network model, used to classify the chest x-rays. Data Augmentation is implemented on the training data, which are the images in this case, to enhance and augment the quality, adeptness, and size of the data.

Numerous operations are carried out to increase the size of the data and help the Deep Learning model capture multiple nuances in the training images. Convolutional Neural Networks are fed the augmented data to prevent overfitting of the data, to optimize the performance of the model. Overfitting is an undesirable condition where a machine learning or deep learning model performs well on the training set but yields undesirable results on the testing and validation sets. Thus, overfitting of the model is not advisable and should not occur during the implementation.

Increasing the training dataset enables the Deep Learning and Computer Vision algorithms to fit the models more efficiently and adroitly. There are various data augmentation techniques, which have been used to train the model. This practically does not change the original dataset and is only implemented during the run time, without any unnecessary disk space being utilized to store the modified and augmented images.

The data augmentation techniques implemented to improve the accuracy of our Deep Learning model are-

Rescale Normalization:

This is used to decrease the amount of computation and processing involved while dealing with image data. The value of pixels can be between 0 to 255 for any image. To normalize this wide range, each value is multiplied by a rescaling factor of 1.0/255.0, in order to obtain the values between 0 and 1. Thus, the required computational and processing power is significantly reduced. This is called while loading and processing the images into training and validation sets using the ImageDataGenerator function of the TensorFlow library in Python.

Geometric Transformations:

These are used to modify the geometric properties of images and enable the model to capture the training images modified with respect to these properties. It enables the model to recognize and process images similar to the original training set but with slight modifications in their physical traits. To explicate this, a zoomed-in image may not be clearly understood by a model given zoomed-out training images. Thus, by changing the geometric properties of height, width, and zoom, new training images are developed, which may help the model correctly identify the test image. This includes various attributes and parameters, mentioned in the ImageDataGenerator function, like the ‘width_shift_range’, ‘height_shift_range’, and the ‘zoom_range’.

Flipping:

A mirror image of a training image might not be correctly recognized and evaluated by the model, in the testing phase. Thus to ensure adequate and optimal performance of the model on these mirrored images, flipping is used to augment the data and provide flipped images of the training data, without actually having to store mirrored images of each training sample, onto the disk storage. Thus, a dynamically efficient flipping augmentation process helps improve the performance of the Deep Learning model on the chest X-ray dataset.

Shearing:

Introducing shearing to the training images, helps obtain sheared images, to have the Deep Learning model accustomed to images in a sheared orientation. It is likely that some test images may be similar to some training images, with a sheared orientation. Thus, to accurately deal with these test images, the sheared orientation of certain training set images provides us a better insight.

Rotation:

Training images may be rotated by certain degrees, to obtain modified images, in order to augment and increase the contents of the training dataset. This is done by setting the 'rotation_range' to an appropriate value while initializing the ImageDataGenerator function. Thus, various angles of the rotated images are used in data augmentation.

VI. CNN Architecture

CNN models are neural networks that consist of convolutional layers, pooling layers, flattening layers, and fully connected layers. These networks utilize appropriate activation functions.

Convolutional layer:

The convolutional layer serves as the fundamental component of CNNs. In mathematics, convolution is an operation that combines two functions. In CNN models, the input image is first transformed into a matrix representation. A convolutional filter is then applied to this input matrix. The filter slides over the matrix, performing element-wise multiplication and storing the resulting sums. This process generates a feature map.

Typically, a 3x3 filter is used to create 2D feature maps when working with black-and-white images. When the input image is represented as a 3D matrix with RGB colors, convolutions are performed in 3D. Multiple feature detectors are applied to the input matrix, generating a layer of feature maps. This layer forms the convolutional layer of the CNN model.

Activation Function:

Activation functions are important components of neural networks. The ReLU (Rectified Linear Unit) activation function is widely used due to its ability to introduce nonlinearity and address the vanishing gradient problem. It is simple and helps create sparse representations, making it advantageous.

The tanh (hyperbolic tangent) activation function maps inputs to a range between -1 and 1, capturing complex patterns in the data and introducing nonlinearity. The sigmoid activation function maps inputs to a range between 0 and 1, making it suitable for binary classification tasks where the output represents a probability of belonging to a certain class.

Models using sigmoid activation often use the binary cross-entropy cost function for binary classification. This cost function measures the dissimilarity between predicted outputs and target values, helping the model learn accurate predictions and classify instances correctly. These activation functions enhance the flexibility and performance of neural networks, allowing them to handle various tasks effectively.

Pooling:

In the models described, max-pooling layers are used after convolutional layers to downsample images by selecting the maximum pixel intensity values within a window covered by the kernel. This reduces dimensionality and complexity.

Flattening and Dense layer:

After convolutional and max-pooling layers, the image is passed through a flattening layer to convert it into a column, reducing computational complexity. The flattened image is then fed into a fully connected layer (dense layer), where each node is connected to every node in the next layer. This layer extracts features for making predictions through forward propagation.

Cost function:

The models employ the categorical cross-entropy cost function to evaluate performance. This cost function measures how well the neural network model is performing. Backpropagation is then performed to adjust the network's weights and biases based on the calculated cost, iteratively improving performance. To optimize the training process, the models use the Adam optimization algorithm. Adam dynamically adapts the learning rate during training, aiding in faster convergence and improved performance.

Reducing Overfitting:

The initial model in this study suffered from significant overfitting, which led to the implementation of the dropout technique in subsequent models. Dropout is a method that effectively addresses overfitting and deals with the problem of vanishing gradients. By using dropout, each neuron is encouraged to independently develop its own unique representation of the input data. During training, connections between neurons in consecutive layers are randomly and temporarily dropped, which fosters the learning of more robust and generalizable features.

Transfer Learning

KTransfer learning in deep learning involves leveraging pre-trained models' knowledge from one task or domain to improve performance on a different but related task. It offers several advantages, including reduced training time by utilizing already learned feature representations, improved generalization by leveraging knowledge from large-scale datasets, and overcoming data scarcity by transferring knowledge to tasks with limited labeled data. Commonly used approaches include feature extraction, where pre-trained convolutional layers act as feature extractors, and fine-tuning, which involves further training the entire model on the target dataset. Transfer learning is widely used in deep learning to accelerate model development, enhance performance, and address complex tasks more effectively.

VII. MODEL

CNN models have been created from scratch and trained on Chest X-Ray Images (Pneumonia) dataset on Kaggle. Keras neural network library with TensorFlow backend has been used to implement the models. The dataset consists of 5863 training images, 624 testing images, and 16 validation images. Data augmentation has been applied to achieve better results from the dataset. The two models have been trained on the training dataset, each with a different number of convolutional layers. Each model was trained for different epochs, with training and testing batch sizes of 15. The following sub-headings further explain the above stages in depth.

MODEL 1:

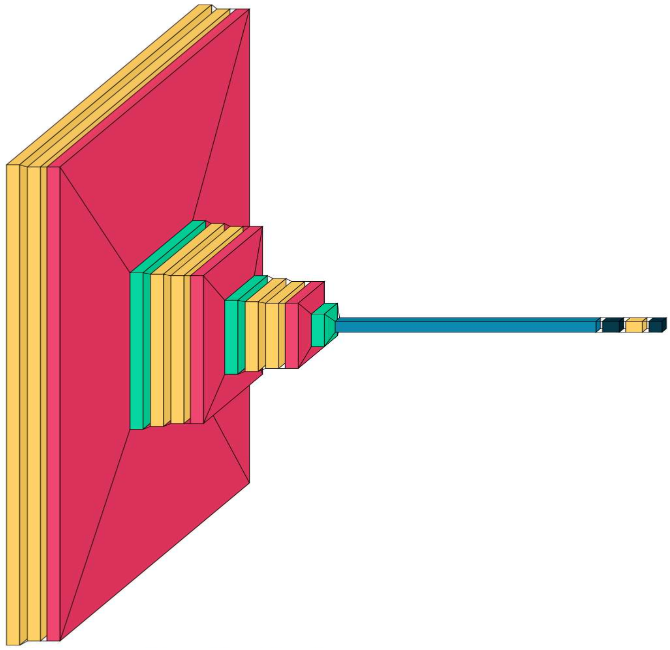


Fig. 2 Model 1

The first CNN model presented below consists of several convolutional layers, pooling, batch normalization layers, and fully connected layers.

The architecture starts with two convolutional layers, each with 128 filters of size 4x4 and sigmoid activation. The input shape is specified as (224, 224, 3) representing an image with dimensions of 224x224 pixels and three color channels. Following the convolutional layers, a batch normalization layer is added to normalize the outputs, followed by a max pooling layer with a pool size of 3x3. Two more convolutional layers with 64 filters of size 3x3 and sigmoid activation are added, followed by another batch normalization layer and max pooling layer with a pool size of 2x2.
Subsequently, two convolutional layers with 32 filters of size 3x3 and hyperbolic tangent (tanh) activation are included, along with a batch normalization layer and max pooling layer with a pool size of 2x2. The output of the convolutional layers is then flattened, and a dense layer with 256 units and tanh activation is added. Dropout regularization with a rate of 0.2 is applied to prevent overfitting.

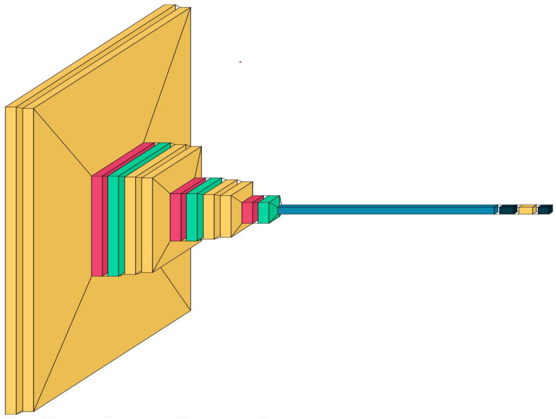
Finally, a dense layer with sigmoid activation and a single unit is added for binary classification. The model is compiled using the SGD optimizer with a learning rate of 0.01 and momentum of 0.9. The loss function used is binary cross-entropy, and the accuracy metric is employed for evaluation. The model summary provides a layer-wise overview of the architecture, including the output shape, number of parameters, and activation functions used.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 221, 221, 128)	6272
conv2d_7 (Conv2D)	(None, 218, 218, 128)	262272
batch_normalization_3 (BatchNormalization)	(None, 218, 218, 128)	512
max_pooling2d_3 (MaxPooling2D)	(None, 72, 72, 128)	0
conv2d_8 (Conv2D)	(None, 70, 70, 64)	73792
conv2d_9 (Conv2D)	(None, 68, 68, 64)	36928
batch_normalization_4 (BatchNormalization)	(None, 68, 68, 64)	256
max_pooling2d_4 (MaxPooling2D)	(None, 34, 34, 64)	0
conv2d_10 (Conv2D)	(None, 32, 32, 32)	18464
conv2d_11 (Conv2D)	(None, 30, 30, 32)	9248
batch_normalization_5 (BatchNormalization)	(None, 30, 30, 32)	128
max_pooling2d_5 (MaxPooling2D)	(None, 15, 15, 32)	0
flatten_1 (Flatten)	(None, 7200)	0
dense_2 (Dense)	(None, 256)	1843456
dropout_1 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 1)	257

Total params: 2,251,585
Trainable params: 2,251,137
Non-trainable params: 448

MODEL 2:



The second CNN model presented below consists of several convolutional layers, pooling layers, batch normalization layers, and fully connected layers.

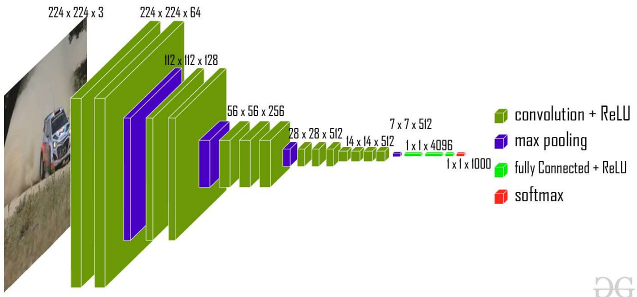
The architecture begins with two convolutional layers, each with 128 filters of size 4x4 and ReLU activation. This is followed by a max pooling layer with a pool size of 3x3 and batch normalization to normalize the outputs. Next, two more convolutional layers with 64 filters of size 3x3 and ReLU activation are added, followed by another max pooling layer and batch normalization. Two additional convolutional layers with 32 filters of size 3x3 and ReLU activation are then incorporated, along with a final max pooling layer and batch normalization.

The output of the convolutional layers is flattened, and a dense layer with 256 units and ReLU activation is added. Dropout regularization with a rate of 0.2 is applied to prevent overfitting.

Finally, a dense layer with sigmoid activation and a single unit is included for binary classification. The model is compiled with the Adam optimizer using binary cross-entropy loss and accuracy as the evaluation metric.The summary of the model shows the architecture's layer-wise details, including the output shape, number of parameters, and activation functions used.

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_12 (Conv2D)          (None, 221, 221, 128)     6272

 conv2d_13 (Conv2D)          (None, 218, 218, 128)     262272

 max_pooling2d_6 (MaxPooling  (None, 72, 72, 128)      0
 2D)

 batch_normalization_6 (Batc  (None, 72, 72, 128)      512
 hNormalization)

 conv2d_14 (Conv2D)          (None, 70, 70, 64)        73792

 conv2d_15 (Conv2D)          (None, 68, 68, 64)        36928

 max_pooling2d_7 (MaxPooling  (None, 34, 34, 64)       0
 2D)

 batch_normalization_7 (Batc  (None, 34, 34, 64)       256
 hNormalization)

 conv2d_16 (Conv2D)          (None, 32, 32, 32)        18464

 conv2d_17 (Conv2D)          (None, 30, 30, 32)        9248

 max_pooling2d_8 (MaxPooling  (None, 15, 15, 32)       0
 2D)

 batch_normalization_8 (Batc  (None, 15, 15, 32)       128
 hNormalization)

 flatten_2 (Flatten)         (None, 7200)              0

 dense_4 (Dense)             (None, 256)               1843456

 dropout_2 (Dropout)         (None, 256)               0

 dense_5 (Dense)             (None, 1)                 257

=================================================================
Total params: 2,251,585
Trainable params: 2,251,137
Non-trainable params: 448
_________________________________________________________________
```

MODEL 3-:



The third model utilizes transfer learning with VGG19 as its backbone, leveraging the pre-trained weights and feature extraction capabilities. This approach enables effective capture of visual patterns, enhances prediction performance, and saves computational resources by fine-tuning specific layers for the prediction task.

The model is initialized with the pre-trained weights obtained from training VGG19 on the large-scale ImageNet dataset, which enables the model to capture rich and meaningful features from images. The model is used in such a way that it ensures the top classification layer of the original VGG19 model is excluded.

To preserve the learned representations and prevent them from being modified during training, the trainable property of all layers in the VGG19 model is set to False using a loop. This freezing of the pre-trained layers ensures that only the newly added layers will be trained and updated during the subsequent training process.

The output from the VGG19 model is passed through a Global Average Pooling 2D layer, which aggregates the spatial information in the feature maps into a fixed-length vector. This helps in reducing the model's complexity and making it more computationally efficient.

Next, a fully connected Dense layer is added with 256 units and a Rectified Linear Unit (ReLU) activation function. This layer acts as a feature extractor and aims to capture high-level representations and complex patterns in the data.

Finally, a Dense layer with a single unit and a sigmoid activation function is introduced. This serves as the output layer of the model, producing a binary classification prediction. The sigmoid activation function ensures that the output falls between 0 and 1, representing the probability of the input image belonging to the positive class.

Overall, this model combines the powerful pre-trained VGG19 backbone with additional trainable layers to fine-tune the model for the specific binary classification task. By leveraging transfer learning, it benefits from the pre-trained weights and captures meaningful image features, resulting in an effective and efficient solution for the given classification problem.

VIII. Performance Analysis Of Models

The accuracy graphs and loss graphs and Table-1 indicate that classifier model-1 underperformed compared to model-2. The accuracy graphs and loss graphs suggest the presence of overfitting. The accuracy and loss scores are also low for Model 1.

Model-2, which is using a different activation function, like RELU and Sigmoid than Model-1 which uses the Sigmoid and Tanh activation function and the batch normalization after max-pooling, significantly improved the performance of the model. Although the model uses the same Regularization function Dropout with the keep prop of 0.2, Batch Normalisation adds the extra Regularization effect and prevents overfitting. Batch Normalisation after max-pooling helps to stabilize training, improve generalization, and reduce internal covariate shift.

Since Model-1 uses Sigmoid and Tanh activation functions then the model may suffer from the problem of vanishing Gradients. These activation functions have a saturation property, meaning that their gradients become close to zero when the inputs are very large or very small.

For sigmoid activation, the derivative of the sigmoid function is given by $f'(x) = f(x)(1 - f(x))$, where $f(x)$ is the sigmoid function. As the input to the sigmoid function moves further away from zero, the gradient approaches zero, resulting in vanishing gradients. This issue is especially prevalent when stacking multiple layers with sigmoid activations, as each layer can further diminish the gradient magnitude.

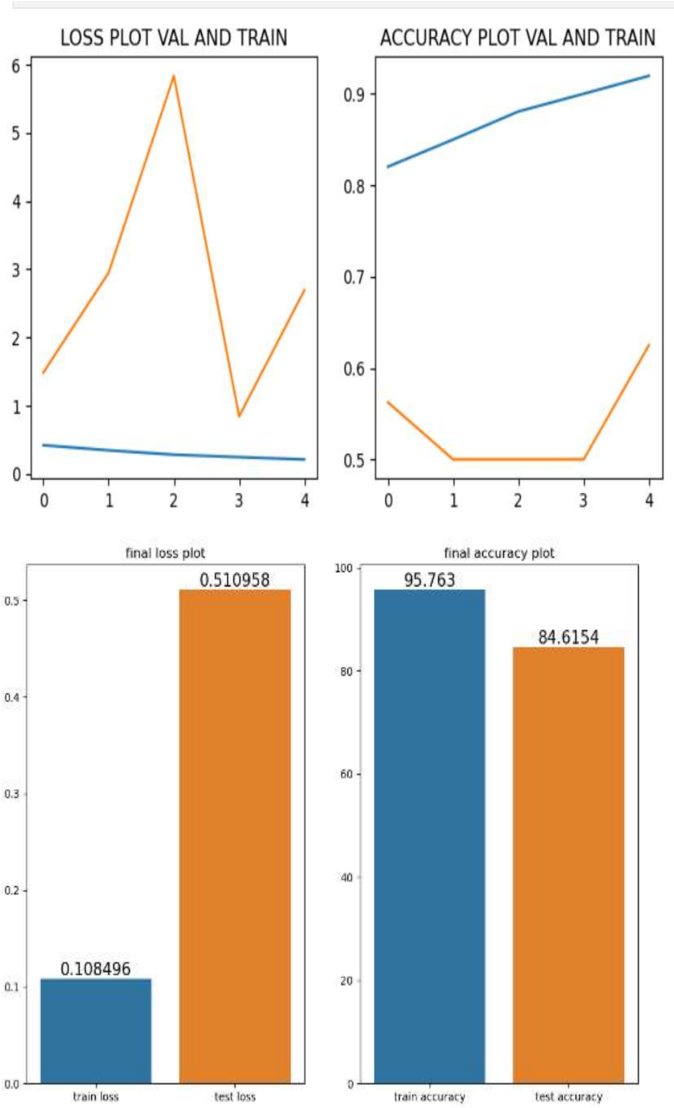
ReLU (Rectified Linear Unit) activation, on the other hand, does not suffer from the vanishing gradient problem to the same extent. The ReLU activation function is defined as $f(x) = \max(0, x)$, meaning that it simply passes positive values as they are and sets negative values to zero. The derivative of ReLU is 1 for positive inputs and 0 for negative inputs. While ReLU can also encounter the dying ReLU problem (where neurons get stuck in the zero activation state and produce zero gradients), it generally avoids the vanishing gradient problem associated with sigmoid activation.

In the following equations, tp denotes true positives, tn denotes true negatives, fp denotes false positives, and fn denotes false negatives. However, please note that you mentioned having only models 1 and 2, so the comparisons and conclusions regarding models 3 and 4 may not apply to your specific scenario.

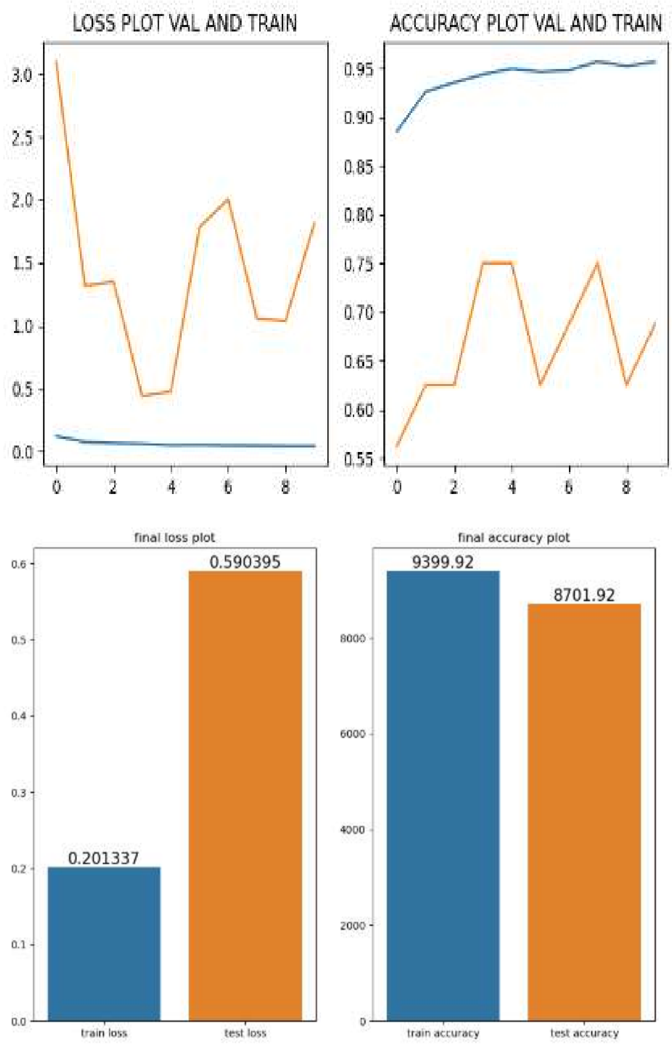
$$\textbf{\textit{Accuracy}} = \frac{TP + TN}{TP + TN + FP + FN}$$

All the below given loss plots orange curve represents the validation loss and blue line represents the train loss.

MODEL 1:

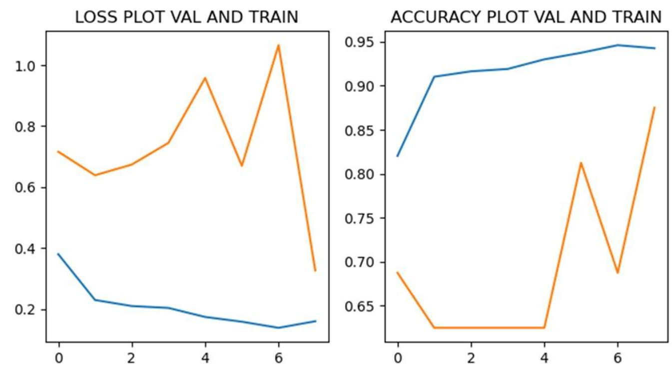


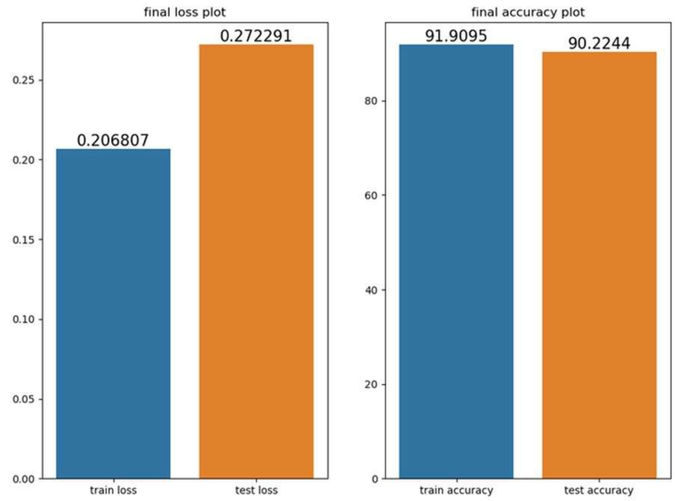
MODEL 2:



MODEL 3:

The final model generated using transfer learning.





IX. RESULTS

To assess the performance of each CNN classifier model, the evaluation metrics used were validation accuracy, recall, and accuracy. The accuracy and loss graphs were examined as well. Additionally, the confusion matrix was computed for each model.

The first CNN classifier model achieved a training accuracy of 0.95 and a testing accuracy of 0.84, but the recall value was not mentioned. It is recommended to calculate and include the recall to provide a comprehensive evaluation.

The second CNN classifier model achieved a training accuracy of 0.94 and a testing accuracy of 0.87, but the recall value was not mentioned. It is recommended to calculate and include the recall to provide a comprehensive evaluation.

If we look at the third model which was built using transfer learning technique, we see that it went up to a training accuracy of 92% and testing accuracy of 90.2%. We can also see that the best model out of all the three which was built is the same model built using transfer learning.

X. CONCLUSION

In this project, we investigated the use of convolutional neural networks (CNNs) for predicting pneumonia from chest X-ray images. The goal was to leverage the power of deep learning to develop an accurate and reliable diagnostic tool. Two CNN classifier models were trained using this dataset, and their performance was evaluated using various metrics.

The evaluation metrics used, including accuracy and loss, provided insights into the models' overall performance. However, the absence of reported recall values limited our understanding of the models' ability to correctly identify positive cases of pneumonia. To provide a comprehensive evaluation, future studies should include recall values and confusion matrix analysis.

Despite this limitation, the CNN models showed promising results in predicting pneumonia. They achieved respectable training and testing accuracies, indicating their potential for assisting healthcare professionals in expedited diagnosis. However, further refinement and improvement are necessary to enhance the models' diagnostic capabilities.

This project highlights the potential of CNNs in medical image analysis and underscores the importance of large-scale datasets and advanced preprocessing techniques in optimizing model performance. The findings contribute to the growing body of research on deep learning-based medical diagnostics.

In conclusion, while the CNN models developed in this project show promise in pneumonia prediction, there is a need for additional evaluation and refinement. By considering a more comprehensive set of evaluation metrics and addressing the limitations observed, CNN-based approaches hold the potential to improve pneumonia diagnosis, leading to more timely interventions and improved patient outcomes.

XI. FUTURE SCOPE

Future Scope: The presented model has the potential for further enhancements to provide stage-wise diagnosis for Pneumonia. Additionally, the model can be extended to aid in the diagnosis of other respiratory diseases such as Tuberculosis or Chronic Obstructive Pulmonary Disease (COPD). By leveraging transfer learning and incorporating a larger dataset, the Deep Learning model can be adapted to accurately detect these conditions from X-ray images. Such automated diagnosis systems can expedite the testing process, reduce the risk of exposure for medical staff, and improve diagnostic accuracy.

Furthermore, the model's capabilities can be expanded to include the detection and diagnosis of other infectious diseases, beyond the scope of Covid-19. By training the model on relevant datasets, it can assist in the identification of diseases like pneumonia caused by different pathogens or even non-respiratory diseases that can be detected through imaging techniques.

Moreover, the application of Deep Learning can be explored for other medical purposes, such as bone suppression in chest cavity imaging, aiding in the diagnosis of lung cancer or other abnormalities. By leveraging the model's ability to analyze and interpret medical images, it can assist radiologists in making more accurate and timely diagnoses.

In conclusion, this project holds significant potential in the field of medicine and healthcare, with opportunities for further research and development in the diagnosis of various respiratory diseases, infectious diseases, and other medical imaging applications.

XII. REFERENCES

Dataset:
https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia
Code:
https://www.kaggle.com/code/harisankar1100/pnemonia-prediction

[1] Acharya A. K, Satapathy R. A Deep Learning Based Approach towards the Automatic Diagnosis of Pneumonia from Chest Radio -Graphs. Biomed Pharmacol J 2020; 13(1). Available from: https://bit .ly/2vYHz8V

[2] Chandra T.B., Verma K. (2020) Pneumonia Detection on Chest X-Ray Using Machine Learning Paradigm. In: Chaudhuri B., Nakagawa M., Khanna P., Kumar S. (eds) Proceedings of 3rd International Conference on Computer Vision and Image Processing. Advances in Intelligent Systems and Computing, vol 1022. Springer, Singapore

[3] Saul, Can Jozef, Deniz Yagmur Urey and Can Doruk Taktakoğlu.(2019). Early Diagnosis of Pneumonia with Deep Learning. ArXiv abs/1904.00937.