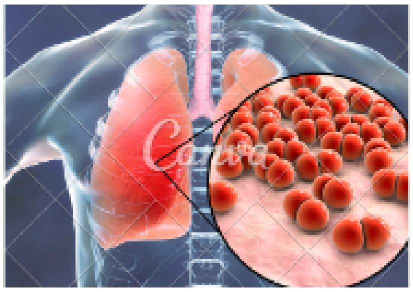
CHEST X-RAY PNEUMONIA DETECTION

Couse code: CS354N

Course instructor: DR.Aruna Tiwari



TEAM MEMBERS:
PRANAY D KUMAR 210001053
SANSKAR 210001063
VELPURU NAVYA 210001075

Introduction

- **TITLE OF THE PROJECT:**

Convolutional Neural Network Models for Pneumonia Detection in Chest X-ray Images

- **PROBLEM DEFINITION:**

Pneumonia remains a significant global health concern, particularly among children under the age of 5, with alarming mortality rates reported annually. In 2016 alone, an estimated 1.2 million cases of pneumonia were documented in this age group, resulting in approximately 880,000 deaths worldwide. Despite advancements in healthcare, pneumonia persists as a leading cause of mortality, especially prevalent in regions such as South Asia and Sub-Saharan Africa. Even in developed nations like the United States, pneumonia ranks among the top 10 causes of death. Timely detection and treatment are critical in mitigating the impact of pneumonia, particularly in regions with high prevalence rates among vulnerable populations.

This project focuses on leveraging Convolutional Neural Network (CNN) models to detect pneumonia from chest X-ray images. The objective is to develop and compare various CNN architectures trained on pediatric chest X-ray datasets, distinguishing between pneumonia and non-pneumonia cases. The study explores different model configurations, including varying numbers of convolutional layers and utilizing pre-trained models such as Inception-v3, ResNet50, VGG16, and VGG19. By evaluating the performance of these models, this project aims to contribute to the early detection and intervention strategies for pneumonia, potentially reducing mortality rates among children, particularly in regions with high prevalence rates.

Analysis & Design:

- **DATA COLLECTION:**

The dataset used is available on Kaggle under the name "Chest X-Ray Images (Pneumonia)." This 2 GB dataset contains 5416 images for training,300 images for validation and 624 images for testing. Images in this dataset are grayscale with the dimension of 64 * 64. The dataset consists of three types of images - Normal, Bacterial Pneumonia, and Viral Pneumonia. The dataset is available on the following weblink: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

- **DATA PREPROCESSING:**

The preprocessing pipeline begins with loading the chest X-ray images from the provided dataset directories, namely 'train', 'test', and 'val'. Each image is loaded and converted to grayscale using to standardize the color channels. Next, the images are resized to a uniform size of imagesize x imagesize pixels . Resizing ensures consistency in the dimensions of the input images, facilitating efficient processing by the Convolutional Neural Network (CNN) models. The dataset is then split into training, validation, and test sets, and the corresponding labels are assigned ('PNEUMONIA' or 'NORMAL'). This facilitates the supervised learning process, where the CNN models learn to classify images based on their assigned labels.

To prepare the data for training, the pixel values of the images are normalized by dividing them by 255. This normalization step scales the pixel values to the range [0, 1], aiding istabilizing the training process and improving convergence. Furthermore, data augmentation techniques are applied. Augmentation introduces variations in the training data by performing random transformations such as rotation, zooming, shifting, and flipping. These transformations simulate real-world scenarios and help the model generalize better to unseen data, thus reducing the risk of overfitting. The `datagen.fit(x_train)` function is then utilized to compute relevant statistics for data normalization and configure the data generator accordingly. This step ensures consistent data augmentation parameters are applied during training, enhancing the model's robustness and generalization capability.

Overall, the data preprocessing pipeline prepares the chest X-ray images for input into the CNN models by standardizing their size, normalizing pixel values, and augmenting the training data to improve model performance in detecting pneumonia. These preprocessing steps are essential for enhancing the model's accuracy and reliability in real-world applications.

- **FLOW CHART OF THE PROJECT :**

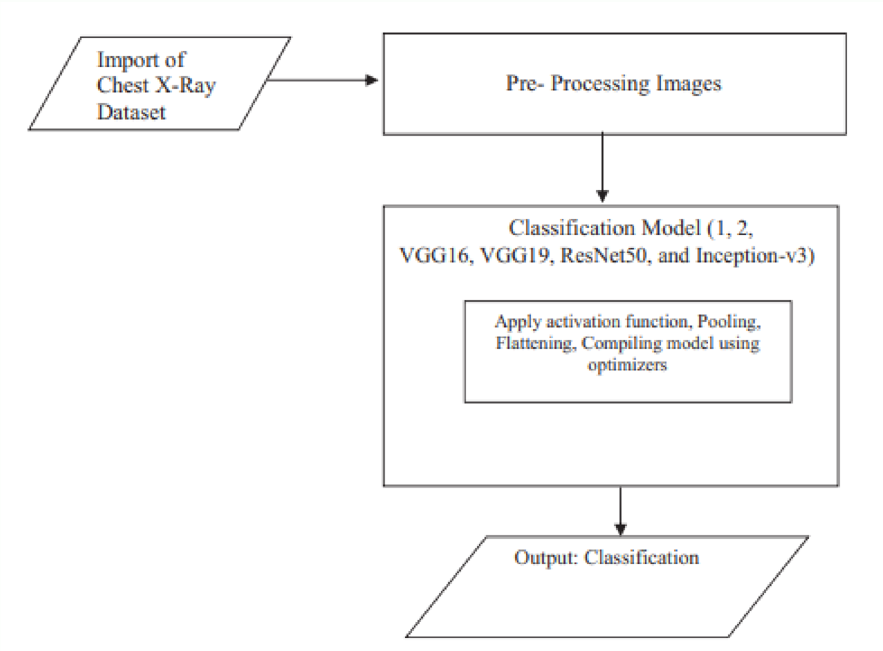


Fig 1: stages of the pneumonia detection project

- # MODEL ARCHITECTURE :

Five models in total were trained and tested on the "Chest X-Ray Images (Pneumonia)" dataset. A detailed description of each model presented in the report below .

Model 1 :

The model consists of 3 convolutional layers; first convolutional layer has 32 feature maps employing ReLU, the second convolutional layer has 64 feature maps employing ReLU and third convolutional layer has 128 feature maps employing ReLU. Max▯pooling layers of 2X2 dimensions are used after each convolutional layer. 2 dense layers have been used, the first dense layer has 256 output perceptrons employing ReLU and second dense layer with two output perceptron using softmax function. Dropout layer is also added. Learning rate of the model is reduced to 0.0001. Adam optimizer has been used with categorical cross-entropy as the cost function.

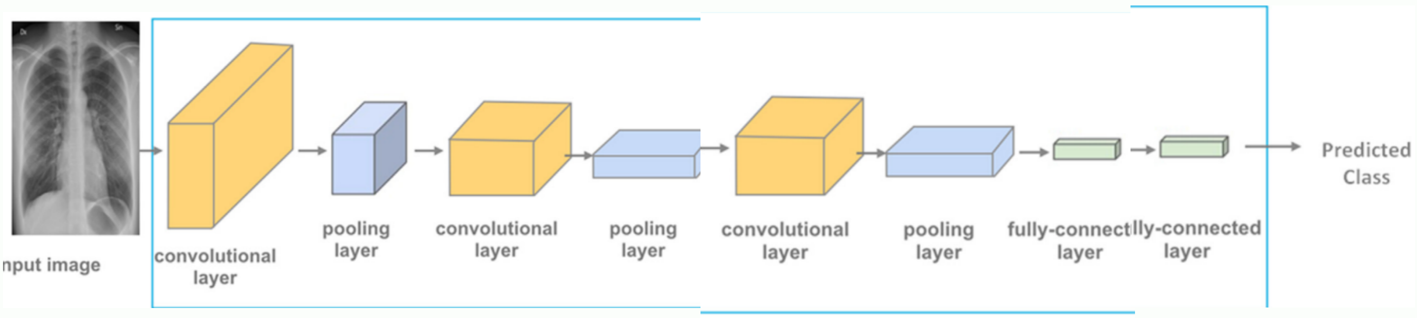


Fig 2: CNN Architecture (model 1)

Model 2 Inception V3 :

Inception v3 depicted in Fig. 3 is a convolutional neural network used for image classification. Inception v3 is a CNN with 42 layers. It has multiple variants such as inceptionv1/google net, inception v2, and inception v4. Inception v1 was the first runner up at the ILSVRC 2015 competition. GoogleNet/ inception v1 was introduced in 2015, later with each new version; some new features were introduced. Auxiliary classifiers were introduced in Inception v1.

Auxiliary classifiers were added to avoid or prevent the activation of each layer to converge to zero. Batch normalization was introduced in Inception v2. This is a technique which rectifies the problem of vanishing gradients and zero activations by reducing the internal covariate shift. Additional factorization was first used in Inception v3, to reduce the number of connections/parameters of the network without decreasing the network efficiency. Learning rate of the network is 0.000001. Adam optimizer has been used with cat egorical cross-entropy as the cost function. Fig. 3 shows the basic architecture of the inception-v3 network.

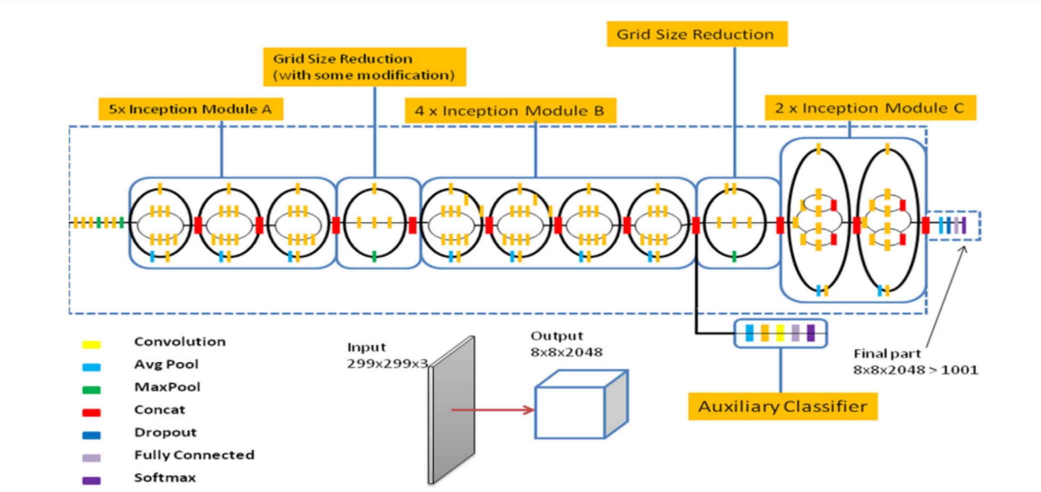


Fig 3: Inceptionv3 Architecture (model 2)

Model 3 ResNet 50 :

ResNet stands for residual network and is primarily used for image classification. Microsoft's ResNet achieved a 3.57% top 5 error on the ImageNet dataset and won the ILSVRC classification contest in 2015 [13]. The network's convolutional layers have 3X3 filters, and downsampling is done directly by the convolutional layers having a stride of 2. The last layer of the network is a fully-connected layer with 256 and two channels employing ReLU and softmax activation functions, respectively. Learning rate of the network is 0.000001. Adam optimizer has been used with categorical cross-entropy as the cost function. Shortcut connections are used in ResNet to rectify the problems of degrading accuracy and vanishing gradient, which occur in deep neural networks. These connections allow the network to skip through layers which it feels are irrelevant for training. This reduces the training error and helps the network to converge faster in comparison to other networks. Fig. 4 depicts the working of shortcut connections in ResNet50 model.

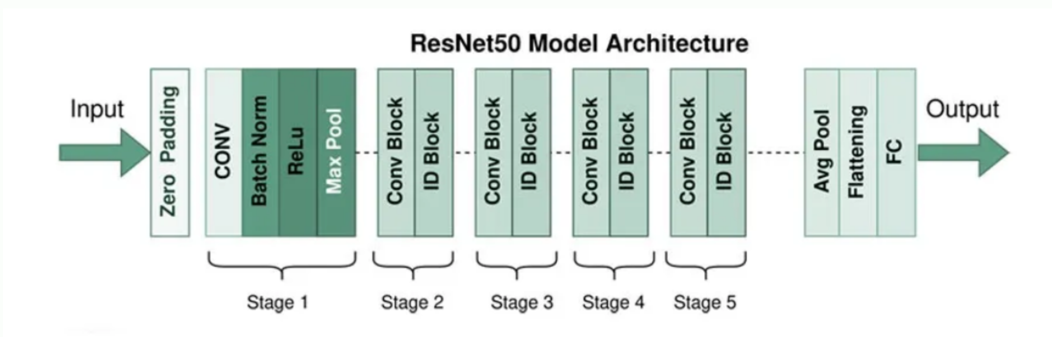


Fig 4: ResNet 50 Architecture (model 3)

Model 4 VGG16 :

VGG16 is a CNN model which was developed by Simonyan and Zisserman [6]. It was one of the most notable models submitted to the ILSVRC 2014 competition. In total, the network has 16 layers [6]. VGG16 introduced multiple 3X3 kernel-sized filters one after the other replacing large kernel sized filters which were used in earlier models. Multiple layers of kernels result in increased depth of the neural network. This enables the neural network to understand and recognize more complex features and patterns. Vgg16 contains convolutional layers of 3x3 dimensions, average-pooling layers of 2x2 dimensions, and fully connected layers. The initial width of the neural network is 64. The width of the neural network doubles after each pooling layer. The first two fully connected layers, each have 256 channels, and the third layer has two channels. The first two hidden layers employ ReLU activation function, and the final layer employs a softmax activation function. Dropout was applied after each 256 channel dense layer. Learning rate of the network is 0.0001. Adam optimizer has been used with categorical cross-entropy as the cost function. The representational depth of VGG16 is beneficial for classification accuracy

Model 5 VGG19 :

Vgg19 which is a variant of VGG16 is a 19-layer convolutional neural network which is used mainly for image classification. Its basic architecture is similar to that of VGG16 . The only difference in VGG19 is the use of 2 dense layers having 256 and two channels, and the learning rate being reduced to 0.00001.

- **ALGORITHM :**

Step 1: Pass 64x64 images through a convolutional layer with 32 feature maps and ReLU activation function.
Step 2: Output of the previous layer is passed through a 2D max pooling layer with dimensions 2x2.
Step 3: The input image size is set to 64x64 and passed through another convolutional layer with 64 feature maps and ReLU activation function.
Step 4: Output of the previous layer is passed through another 2D max pooling layer with dimensions 2x2.

Step 5 :

a. For **Model 1**:Pass the input image size of 64x64 through a convolutional layer with 128 feature maps and ReLU activation function. Output of this layer is passed through a 2D max pooling layer with dimensions 2x2. Flatten the output of the pooling layer.
For InceptionV3, ResNet50, VGG16, and VGG19:
b. For **InceptionV3**: Utilize the pre-trained InceptionV3 model with weights trained on ImageNet. The architecture consists of convolutional layers followed by global average pooling. Output of the global average pooling layer is directly used for classification.
c. For **ResNet50**: Utilize the pre-trained ResNet50 model with weights trained on ImageNet. The architecture consists of residual blocks followed by global average pooling. Output of the global average pooling layer is directly used for classification.
d. For **VGG16 and VGG19**: Utilize the pre-trained VGG16 or VGG19 model with weights trained on ImageNet. The architecture consists of convolutional layers followed by fully connected layers. Output of the last fully connected layer is directly used for classification.

Step 6: Output of the previous layer (flattened output for Model 2, and output of global average pooling or last fully connected layer for pre-trained models) is passed through a fully connected dense layer with 256 perceptrons and ReLU activation.

Step 7: Compile the model using the Adam optimizer with a learning rate of 0.001, categorical cross-entropy loss function, and softmax activation function for binary classification. The cross-entropy loss function is defined CrossEntropyLoss=(ylog(p)+(1−y)log(1−p)) Where y is the binary indicator (0 or 1) and p is the predicted probability.

Performance Measures :

In the context of the project described above, the performance measures used to evaluate the effectiveness of the Convolutional Neural Network (CNN) models in detecting pneumonia from chest X-ray images typically include:

1. **Accuracy**: Accuracy measures the proportion of correctly classified samples out of the total number of samples.
It provides an overall assessment of the model's correctness in predicting both pneumonia and non-pneumonia cases.

2. **Precision**: Precision quantifies the model's ability to correctly identify positive cases (pneumonia) out of all predicted positive cases. It indicates the proportion of true positive predictions among all instances classified as positive by the model. Precision is particularly relevant in scenarios where the cost of false positives is high.

3. **Recall (Sensitivity)**: Recall measures the model's ability to correctly identify positive cases (pneumonia) out of all actual positive cases. It quantifies the proportion of true positive predictions among all instances that are actually positive. Recall is crucial in situations where missing positive cases (false negatives) can have severe consequences.

4. **F1-score**: The F1-score is the harmonic mean of precision and recall. It provides a single metric that balances both precision and recall, making it useful for scenarios where there is an imbalance between the classes or where both precision and recall are equally important. The F1-score ranges from 0 to 1, with higher values indicating better model performance.

These performance measures collectively provide a comprehensive evaluation of the CNN models' performance in detecting pneumonia from chest X-ray images. They help assess the models' accuracy, reliability, and ability to generalize to unseen data, guiding further improvements and decision-making in deploying the models for real-world applications.

Experimentation and Results:

For the experimental result evaluations out of the three classes normal patients, bacterial pneumonia and viral pneumonia for the sake of simplicity, the bacterial and viral pneumonia classes have been merged into one class as infected. The results have thus been evaluated as pneumonia predicted and normal.

For Model 1, the training, validation, and testing accuracy and loss are as follows:

- Training Loss: 0.1736
- Training Accuracy: 93.76%
- Validation Loss: 0.2105
- Validation Accuracy: 91.02%
- Testing Loss: 0.3103
- Testing Accuracy: 90.22%

For Model 2 Inception v3, the training, validation, and testing accuracy and loss are as follows:

- Training Loss: 0.2266
- Training Accuracy: 90.85%
- Validation Loss: 0.2347
- Validation Accuracy: 91.31%
- Testing Loss: 0.4064
- Testing Accuracy: 82.21%

The training, validation, and testing accuracy and loss for Model 3 (ResNet50). Here are the metrics:

Training Loss: 0.3686
Training Accuracy: 85.85%

Validation Loss: 0.2451
Validation Accuracy: 82.68%

Testing Loss: 0.3142
Testing Accuracy: 73.23%

For Model 4 (VGG16), the provided metrics are as follows:

Training Loss: 0.0979
Training Accuracy: 96.30%

Validation Loss: 0.2674
Validation Accuracy: 89.42%

Testing Loss: 0.3674
Testing Accuracy: 89.42%

For Model 5 (VGG19), the provided metrics are as follows:

Training Loss: 0.1025
Training Accuracy: 95.97%

Validation Loss: 0.2777
Validation Accuracy: 92.31%

Testing Loss: 0.2777
Testing Accuracy: 92.31%

These metrics indicate the performance of Model 1,inceptionv3,resnet50,vgg16,vgg19 in terms of its ability to learn from the training data, generalize to unseen data (validation set), and perform on completely new data (testing set). The model achieved high accuracy on both the validation and testing sets, suggesting that it can effectively classify the images into their respective classes.

The below table shows the evaluation metrics of the testing data of all the applied models :

Model name	Accuracy	Precision	F1 score	Recall
Model 1	0.90224	0.92869	0.92263	0.912232
Inception v3	0.82211	0.82869	0.832232	0.84263
Resnet 50	0.73237	0.73458	0.81289	0.73659
VGG 16	0.89423	0.91035	0.95698	0.92365
VGG 19	0.92307	0.93248	0.91556	0.90644

The below are the plots of validation and training loss with respect to the epochs used in the models :

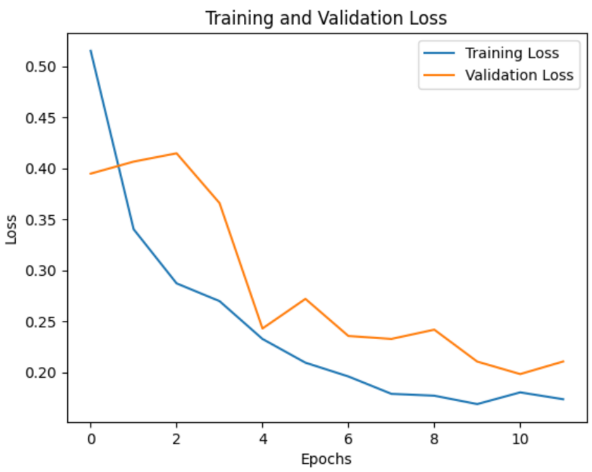


Fig 5: plot of model 1

Fig 5: plot of Inceptionv3

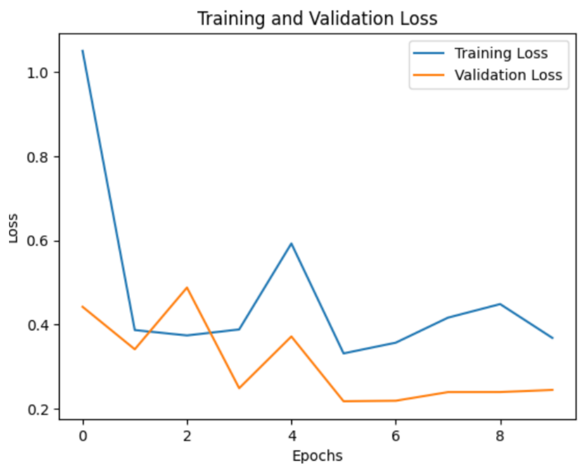


Fig 5: plot of RESNET50

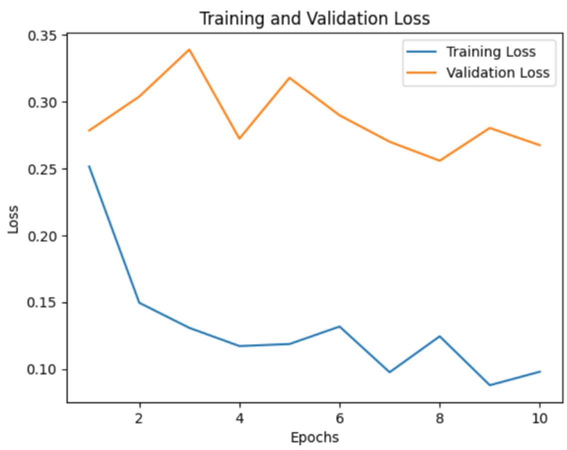


Fig 5: plot of VGG16

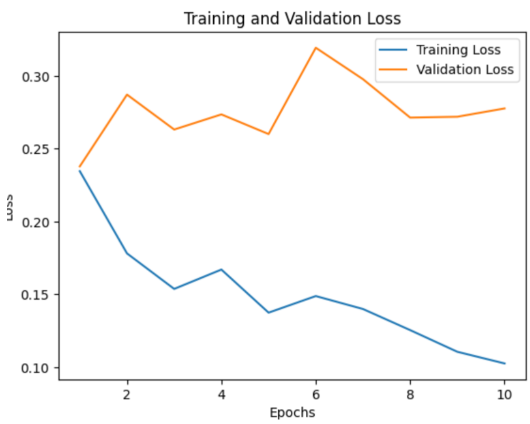


Fig 5: plot of VGG19

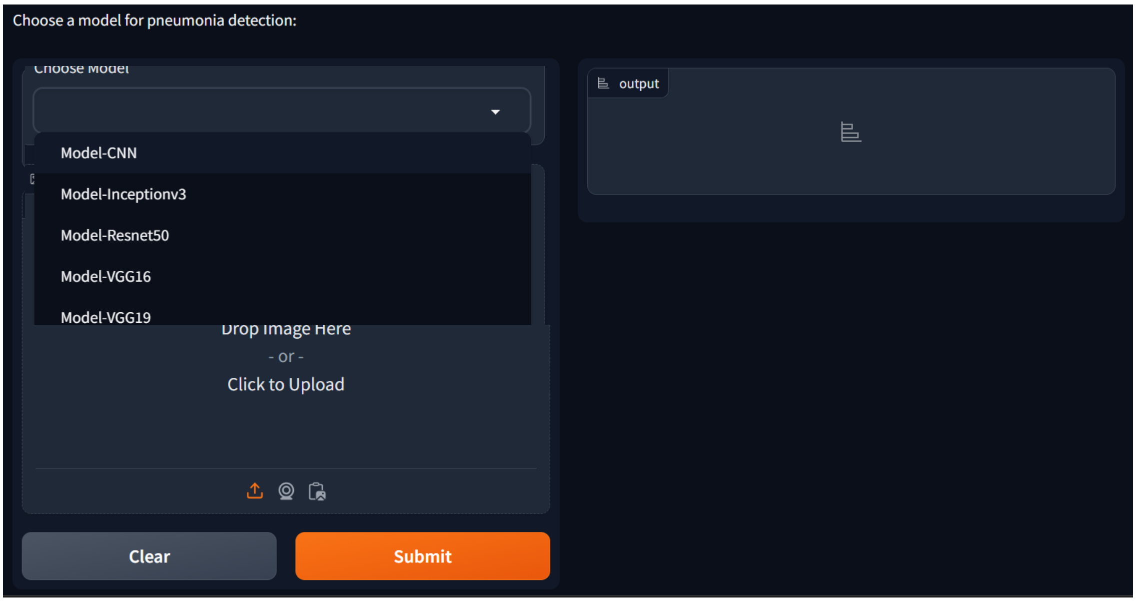
Conclusion:

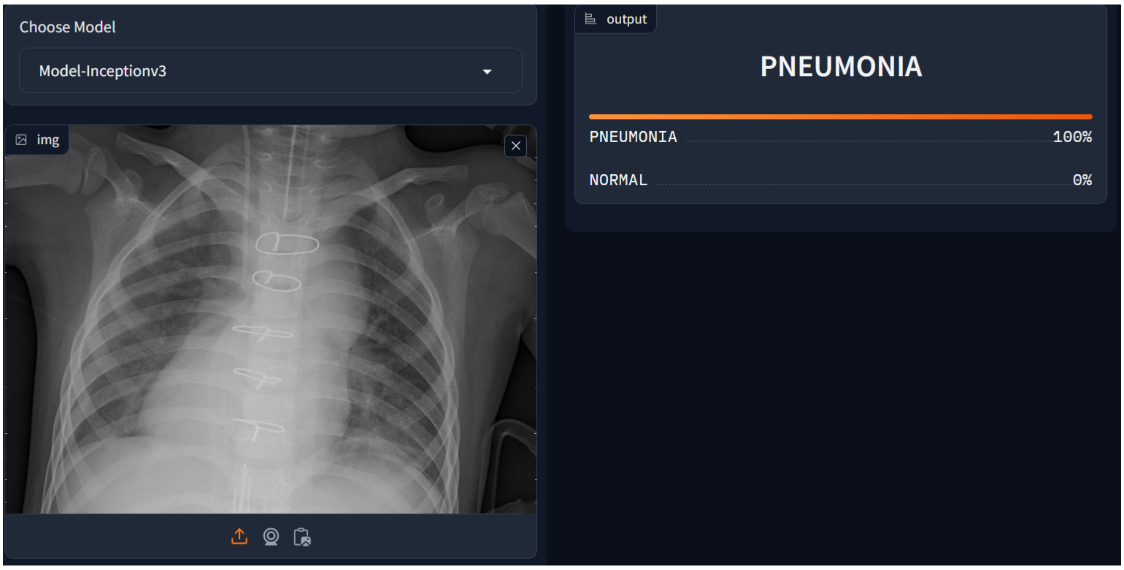
In this project, we developed and evaluated multiple deep learning models for the detection of pneumonia using chest X-ray images. We experimented with various architectures including Convolutional Neural Networks (CNN), InceptionV3, ResNet50, VGG16, and VGG19. Each model was trained, validated, and tested on a dataset of chest X-ray images containing pneumonia and normal cases. The models achieved varying levels of accuracy on the test set, with the best-performing model achieving an accuracy of approximately 92.31%. Precision, recall, and F1-score metrics were also calculated to evaluate the models' performance in classifying pneumonia cases. Additionally, to provide a user-friendly interface for utilizing the trained models, we deployed a web-based application using Gradio. This interface allows users to upload chest X-ray images and obtain predictions for pneumonia detection from the trained models.

For accessing the codebase, please visit our GitHub repository: [GitHub Repository]-(**https://github.com/Sanskar6877/ChestXRay_Pneumonia_Detection**)

- ## user interface :

We have also deployed a multimodel pneumonia detection web application, where users can access all models in one place. The users can compare the results using multiple models, and analyse the results and classify the pneumonia detetction images.
The web application can be accessed at this link: [Pneumonia Detection Web Application](https://huggingface.co/spaces/vnavya2004/PNEUMONIA_DETECTION_MULTIMODEL).





Overall, the project demonstrates the potential of deep learning models in aiding the detection of pneumonia from chest X-ray images, with the deployed web application offering a convenient tool for real-world usage.

- **Reference :**

- https://www.sciencedirect.com/science/article/pii/S0263224120305844