Classification of Pneumonia and Covid-19 using Convolutional Neural Network

A. K. Mishra [1], Aanchal Ghimire [2], Mamata Joshi [3], Srijana Khatiwada [4], Naw Raj Joshi [5], P. B. Jha [6] & & P. S. Aithal [7]

[1] Research Professor, Srinivas University, India, and Apex College, Kathmandu, Nepal, OrcidID: 0000-0003-2803-4918; Email: anjaymishra2000@gmail.com

[2, 3, 4] B.Sc. CSIT Scholars, Madan Bhandari Memorial College, Kathmandu, Nepal: pbjha2039315@gmail.com

[5, 6] Assistant Professor, School of Science and Technology, Madan Bhandari Memorial College, Kathmandu, Nepal: head_schoolsot@mbmc.edu.np

[7] Professor, Institute of Management & Commerce, Srinivas University, Mangalore, India, OrcidID: 0000-0002-4691-8736; E-mail: psaithal@gmail.com

Area/Section: Health Management.
Type of the Paper: Research Analysis.
Type of Review: Peer Reviewed as per |C|O|P|E| guidance.
Indexed in: OpenAIRE.
DOI: https://doi.org/10.5281/zenodo.10001745
Google Scholar Citation: IJHSP

How to Cite this Paper:
Mishra, A. K., Ghimire, A., Joshi, M., Khatiwada, S., Joshi, N. R., Jha, P. B., & & Aithal, P. S., (2023). Classification of Pneumonia and Covid-19 using Convolutional Neural Network. *International Journal of Health Sciences and Pharmacy (IJHSP)*, *7*(2), 65-88. DOI: https://doi.org/10.5281/zenodo.10001745

International Journal of Health Sciences and Pharmacy (IJHSP)
A Refereed International Journal of Srinivas University, India.

Crossref DOI: https://doi.org/10.47992/IJHSP.2581.6411.0110

Received on: 17/04/2023
Published on: 16/10/2023

Classification of Pneumonia and Covid-19 using Convolutional Neural Network

A. K. Mishra [1], Aanchal Ghimire [2], Mamata Joshi [3], Srijana Khatiwada [4], Naw Raj Joshi [5], P. B. Jha [6] & & P. S. Aithal [7]

[1] Research Professor, Srinivas University, India, and Apex College, Kathmandu, Nepal, OrcidID: 0000-0003-2803-4918; Email: anjaymishra2000@gmail.com

[2, 3, 4] B.Sc. CSIT Scholars, Madan Bhandari Memorial College, Kathmandu, Nepal: pbjha2039315@gmail.com

[5, 6] Assistant Professor, School of Science and Technology, Madan Bhandari Memorial College, Kathmandu, Nepal: head_schoolsot@mbmc.edu.np

[7] Professor, Institute of Management & Commerce, Srinivas University, Mangalore, India, OrcidID: 0000-0002-4691-8736; E-mail: psaithal@gmail.com

ABSTRACT

Purpose: *The early and exact classification and identification is necessary for proper treatment which needs excessive time and effort of professionals. This examination is meant to foster a task to recognize Pneumonia and Coronavirus utilizing the idea of the Convolutional Neural Network (CNN) for picture grouping and is centered on building the profound learning model that aids in the characterization utilizing chest X-beam pictures in one of the quick and financially savvy ways.*

Design/Methodology/Approach: *This study uses a wide dataset comprising of chest X-beam pictures accumulated from the Mendeley dataset. Include extraction strategies like picture pre-handling and data augmentation are applied to improve the arrangement execution. The framework utilizes the ResNet-18, which is a sort of CNN model for order. The examination includes assessing the exactness, accuracy, review, F1 score, and area under the receiver working trademark bend (AUC-ROC) for every classification model.*

Findings/Result: *The dataset is separated into preparing and testing subsets to ensure unbiased performance evaluation. For the development and deployment of an accurate and reliable system, factors like data quality, model interpretability, and ethical considerations are considered. We successfully used the pre-trained ResNet-18 CNN model with chest X-ray image data that helped to build a robust classification system with a learning rate of 0.0001 and epoch size 10 having approx. 98.12% train accuracy and 97.70% test accuracy.*

Since the start of the project, we researched several methodologies to build the system. The other models (e.g., ResNet-50) were too big algorithms for our problem which created a problem of overfitting. Hence performance was not very accurate. So, we planned to go with the ResNet-18 model. As per our plan, we developed a system that operates as expected.

Originality/Value: *It helps medical professionals in diagnosing and managing these diseases.*

Paper Type: *Research paper*

Keywords: Classification, dataset, X-ray image, CNN, ResNet, Pneumonia, Covid.

1. INTRODUCTION :

COVID illness (Coronavirus) is an irresistible infection brought about by the SARS-CoV-2 infection that influences our lungs [1]. The most common symptom of COVID-19 is inflammation of the lungs where patients experience mild to moderate respiratory illness while some people become seriously ill and can die at any age. In total till August 13, 2023, there have been 769,369,823 confirmed cases out of which 6,907,141 died all around the world [2-7]. Pneumonia is the most complicated stage of COVID-19 [1, 5]. Pneumonia is a type of intense respiratory contamination that influences little sacs in lungs called alveoli where the tainted individual's alveoli are loaded up with discharge and liquid, making breathing excruciating. It causes the biggest number of passings in kids. It killed 740180 youngsters younger than 5 years in 2019, 14% of all deaths under age 5 years and 22% of all deaths under age 1-5 yrs. It mostly affects the children with low immune power [8].

Coronavirus pneumonia spreads across our lungs gradually, utilizing our own safe framework to spread, which will in general endure longer and cause harm in additional spots. Different pneumonias cause intense sickness side effects that come on at the same time however don't keep going as lengthy [1,9].

The other differentiating characteristics of Covid-19 and Pneumonia are:

Contagiousness: Pneumonia without the infection of coronavirus is not contagious. COVID-19 is highly contagious and spreads from one person to another. Infection Category: Pneumonia infects one part of the lung. But in COVID-19, the whole lung gets infected which leads to other complications such as rapid heartbeat, and difficulty in breathing [9]. Characterization of Pneumonia and Coronavirus utilizing CNN is centered on building the profound learning model that will help in the grouping of Coronavirus and Pneumonia utilizing chest X-beam pictures, in one of the quick and financially savvy ways. Classification of Covid-19 and Pneumonia has progressed with an abundance of data and research going on. The accuracy of this classification has been significantly boosted since the evolution of deep neural network (DNN) based hybrid modeling was adopted. Various research activities have been done on the classification of Pneumonia and Covid-19. According to a journal article [10], the classification is done using deep transfer learning. The created strategy focuses on supporting accuracy and utilizes an exchange learning method as well as a model that is uniquely crafted. Different pre-prepared profound convolutional brain organization (CNN) models were utilized to separate profound elements. As per the discoveries of this review, profound exchange gaining can recognize Coronavirus and pneumonia from CXR pictures. Pre-prepared tweaked models, for example, MobileNetV2 had a 98% exactness, InceptionV3 had a 96.92% exactness, EffNet edge had a 94.95% precision, and VGG19 had a 92.82% exactness. MobileNetV2 has the best exactness of these models.

One more examination article [11] has been distributed in this order where they proposed two CNN models, which were prepared utilizing two distinct datasets. The main model was prepared for a double arrangement with one of the datasets that mainly included pneumonia cases and typical chest X-beam pictures. The subsequent model utilized the information advanced by the primary model utilizing move learning and prepared for 3 class arrangements on Coronavirus, pneumonia, and ordinary cases in light of the second dataset that included chest X-beam (CXR) pictures. The model gave 98.3% precision on the test information.

As indicated by the Diary of clinical Physical Science, 2022 Jan-Blemish [12], a model for mechanized recognition and grouping of Coronavirus and viral pneumonia infections by applying profound CNN models utilizing chest X-beam pictures was based on top of VGG16 engineering with pre-prepared ImageNet loads. A sum of 15,153 examples were utilized in this work. These examples incorporate chest X-beam pictures of Coronavirus, viral pneumonia, and typical cases. The whole dataset was divided into train and test sets, with a proportion of 80:20 prior to preparing the model. The model had accomplished a characterization precision of 98%.

As found in the article [13], the review incorporates eleven pre-prepared CNN models, for example, VGG organization and ResNet. They have been effectively tried and assessed on a public X-beam picture dataset for ordinary and three unhealthy cases. The outcomes showed that VGG16, ResNet50V2, and DenseNet169 models accomplished the best identification exactness of Coronavirus, viral pneumonia, and bacterial pneumonia pictures separately.

Similar research papers and articles have been published by various sources regarding the classification of COVID-19 and Pneumonia using CNN. Different CNN models have given various classification accuracies above 90%. Using the various models increases the use of heavy computing resources which may not be economically feasible. This is the field of study which can be widely researched using different machine learning and deep learning CNN models to gain the best classification accuracy. Therefore, we will be following the ResNet-18 model of CNN with various numbers of datasets for better classification accuracy. In the event of ordinary X-beam picture input, it will arrange the picture as typical. In this way, there will be 3 classes in our framework. The CNN model proposes channel-based extraction highlights, which might be actually arranged.

The fundamental objective of our proposed framework is to give a stage to the well-being specialist co-ops where the doctors can analyze between Coronavirus, pneumonia and sound individuals all the more precisely utilizing the chest X-beam pictures; stay away from misdiagnosis and further spreading of the sicknesses.

2. PROBLEM STATEMENT :

The very common method of diagnosis of pneumonia is chest X-beam pictures and that of COVID-19 is a sputum/ mucus test. Since the symptoms of pneumonia and COVID-19 are very similar, there may be confusion while diagnosing via X-beam pictures whether the person is infected with either of the diseases. There can also be some people who are suffering from pneumonia but are not infected by COVID-19. It is vital to comprehend the dainty line that separates the two of them. COVID-19's spread rate is higher than that of pneumonia, correct diagnosis is vital to keep the sickness from additional spreading. The well-being business is basic, and the work of artificial intelligence instruments can be very risky. At the point when living souls are on the line, a model with low precision is probably not going to be utilized (Mahin et al. 2021)[14]. With all that being said, this project used CNN model (ResNet-18) to classify Pneumonia and COVID-19 in X-beam pictures. It is endeavoring to take care of the characterization issue, which has proactively been settled by various past profound learning models. Since COVID-19 is a new disease, the data is scarce; many models are unlikely to generate reliable outputs. This motivates this project to implement the linear augmentation for generating the synthetic images which will be utilized for the classification model to generate a reliable output. Previous studies have also utilized various convolutional neural network architectures; however, in our project, we are trying to study the CNN ResNet-18 model.

3. OBJECTIVES :

The main goal of this project is to classify the chest X-beam pictures into pneumonia, COVID-19, or normal healthy person with accuracy to ease for doctors, patients, and health workers.

4. METHODOLOGY :

Due to the iterative and incremental nature of the system, the agile approach best describes the development methodology. The model is passed through multiple designing, training, and testing iterations [15].

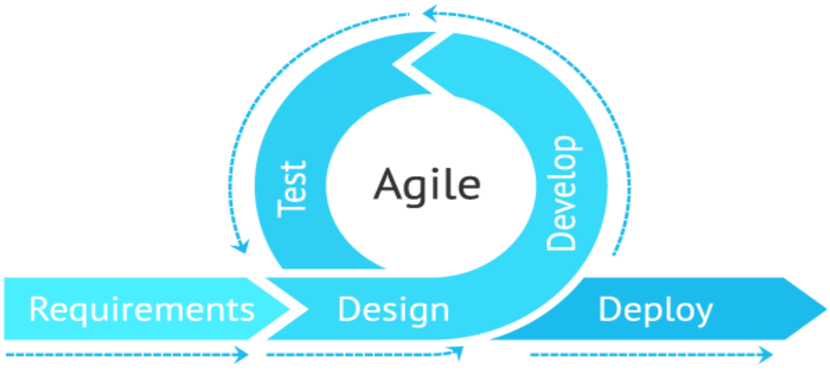


Fig. 1: Agile Development Methodology

4.1 Requirements:

Functional requirement of a system specifies how the system should react to a certain situation that it is put on and how the system comes up with the output to the given input [16, 17, 18, 19, 20]. Following are the functional requirements for our proposed system: Only accept chest X-ray in image format, Extract features from images and Classify images depending on the extracted features. It shows the interaction between the system and the user in a particular environment. The use case model contains actors and the use cases. The actors are the external entities, and the use cases are the system's functions as shown in figure 2.

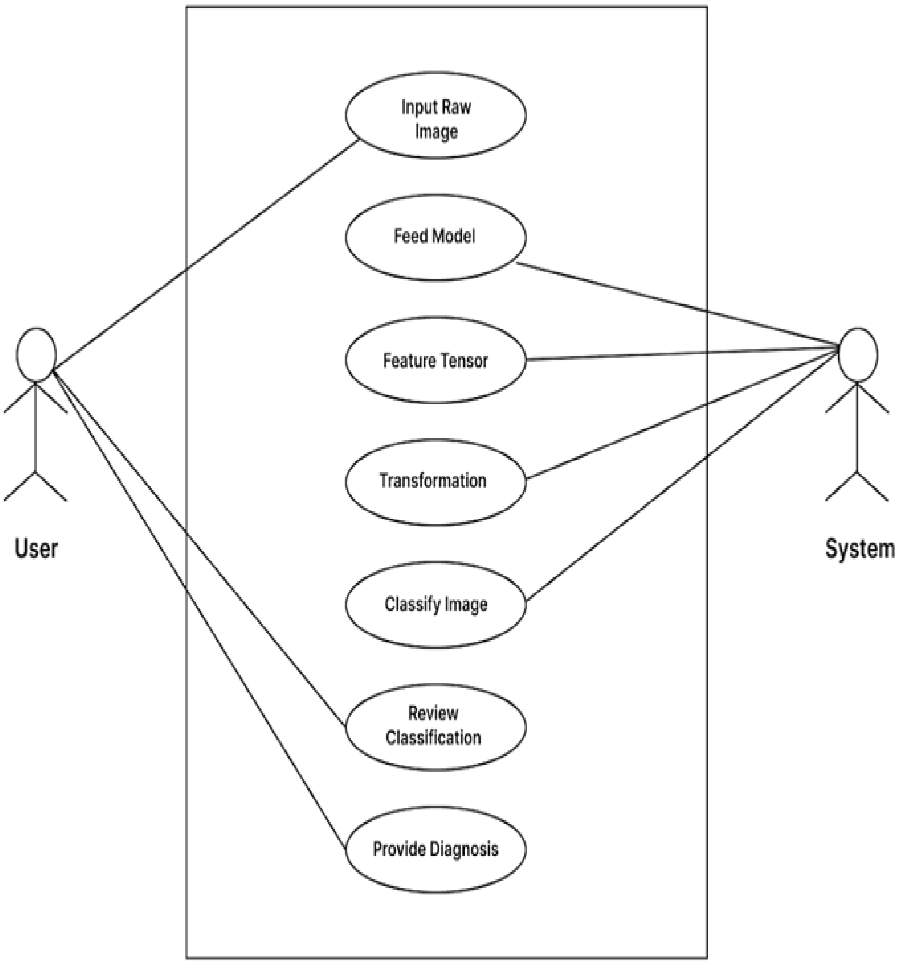


Fig. 2: Use Case Diagram for Classification Module

The non-functional requirements can be expressed as follow.
4.1.1 Accuracy: The model should have high accuracy for clear images without noise. The system's accuracy will depend on how accurately it will be able to classify the image. During inference, the accuracy of the model also depends upon the image from the user.
4.1.2 Availability: The end system will be running with no as much downtime as possible. The web-based system should work on all major web browsers. If certain failures occur exceptionally then it wouldn't take a time to repair.
4.1.3 Efficiency: The end system is accessed through an API and a web-based system, which should have low latency, i.e., inference time.
4.1.4 Reliability: The model should have fewer errors and no drastic significant errors. The system should be active all the time and users can use it whenever they want [16-21].

4.2 Feasibility of the System:
Here, we have studied all the feasibility aspects of the project under consideration to check out if the project is feasible with decided requirements and availability of information, technologies, and budget.
4.2.1 Technical:
Every one of the devices and programming items expected to fabricate this venture are accessible on the web. It doesn't need a unique climate. All the operations can be performed in our daily use laptops. The model will be implemented using web IDE called as Google Collab where we can leverage the GPU for free. For storage, we will be utilizing Google Drive which provides us with 50GB of free storage each, which is a total of 150GB of free storage. Data is also downloaded from the internet as mentioned in the later section.

4.2.2 Operational:

Operational feasibility refers to solving problems and building new systems with the help of a new proposed system. It takes the ideas and opportunities developed during the initial phase and the insights from requirement gathering to build a new system. The proposed system can be used in many different applications.

4.2.3 Economical:

The project will only use the usual laptop specification for building the system and cloud GPUs, which are pay-as-you-go, making this economically feasible. As this system is not tested in the working field, the heavy computing resources will not be needed. The inference can be carried out in smaller computing devices too [16-21].

4.3 System Analysis:

4.3.1 Object modeling using Class and Object Diagrams:

In the object-oriented approach as shown in figure 2 can also be termed as a type of structure diagram which provides a conceptual model and architecture of the system being developed.

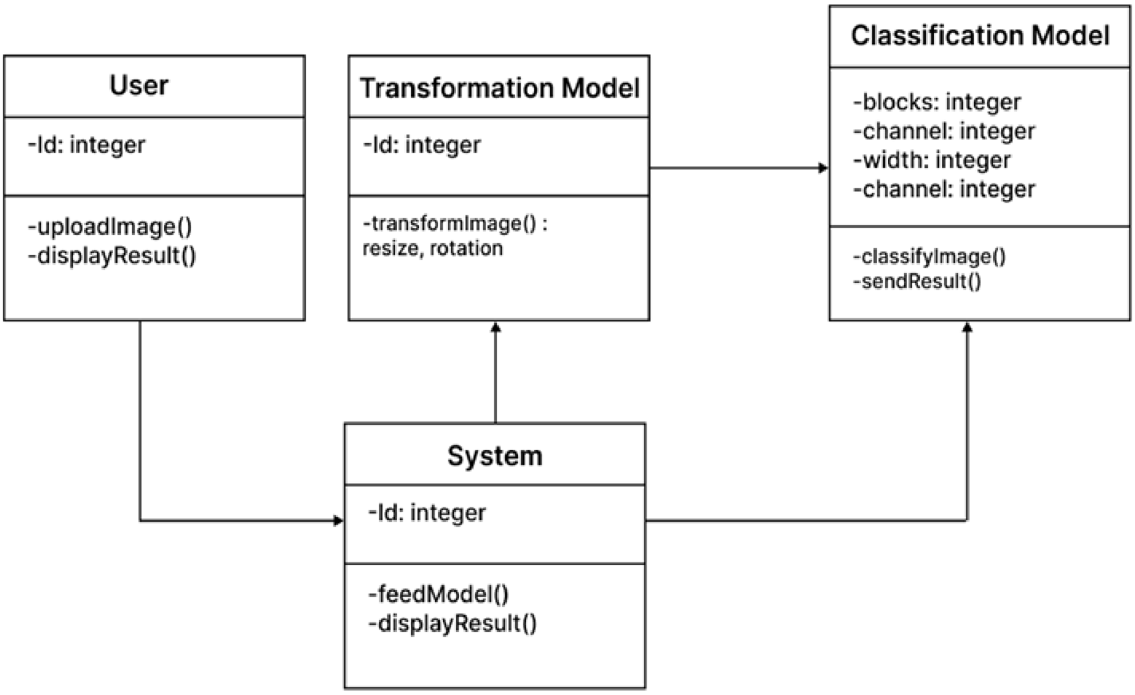


Fig. 3: Class Diagram

Explanation:

- User class represents the user who uploads the image of the patient's chest X-ray to the system using the UploadImage() method.
- The System class represents the deployed system that feeds the classification model using the uploaded image.
- Classification Model class contains the methods to transform and classify the image. The methods here are TransformImage(), ClassifyImage() and SendResult().
- There is a relationship between User and System classes that shows the DisplayResult() method is used by the System to display the result and the User class uses it to ask for the result.

4.3.2 Dynamic Modeling using sequence diagram:

The diagram shows how an object operates with one another and in what order. The following sequence diagram as shown in figure 4 depicts the flow of information in our system. Generated. Firstly, uploadImage() method gives the input by a user to the classification system. The system feeds the model with feedModel() method in sequential order. The Image Transformation Model transforms the images to improve the image quality and increase the number of datasets and passes it to the CNN model, which classifies the images to normal, pneumonia or covid-19 using classifiedLabel() method. Thus classified image data by the CNN model is passed into the classification system which displays the final output to the user using displayResult() method. The sequence diagram above describes the sequential interaction of the System from input to output.

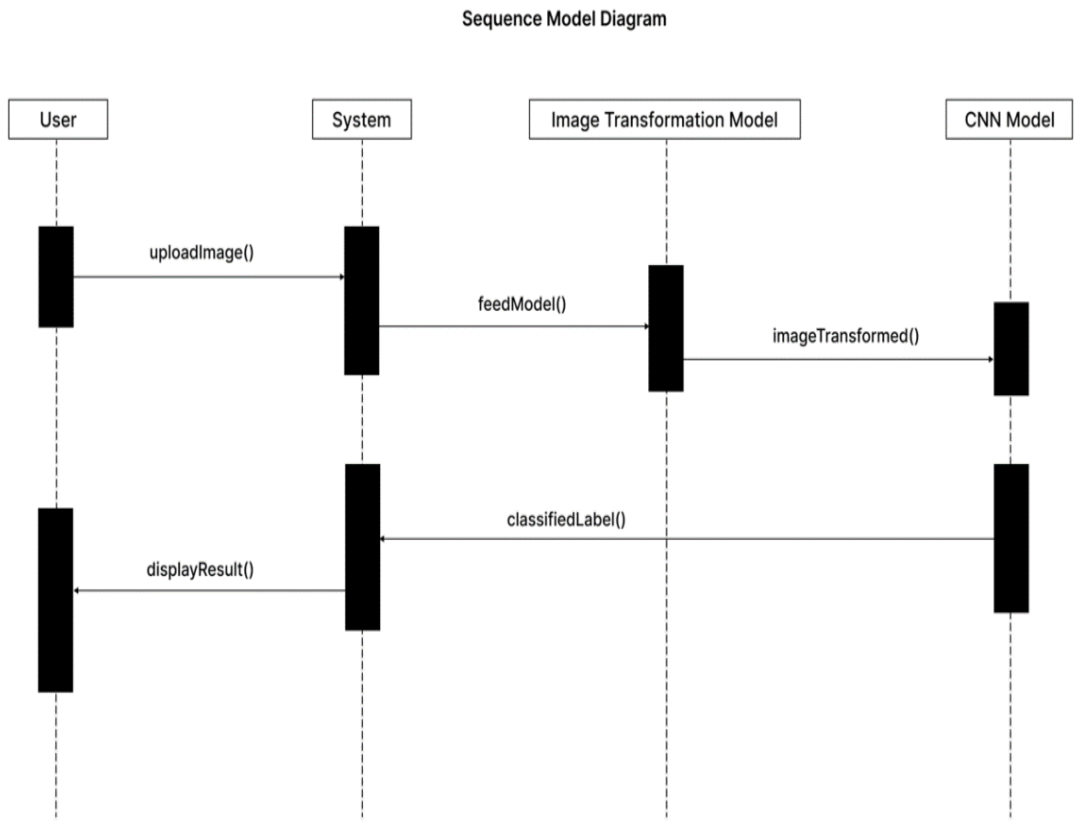


Fig. 4: Sequence Diagram

4.3.3 Process Modelling using Activity Diagram:

An activity diagram is essentially an advanced form of a flow chart that generally describes the model's flow. The activity diagram as shown in figure 5 follows a behavioral approach which shows the flow from one activity to another from start to end. The activity diagram elaborates the flow of the whole system from the starting state to the ending state. The activity starts with the input that the user provides to the system. This system takes raw images as input. This input is then fed to the model of the classification system. XXX type of CNN model is used for classification. The input is firstly transformed by using the PyTorch's transforms class, which transforms the image into a tensor and resizes the image. It then predicts the label of the image, whether the image is of pneumonia, covid-19 or normal chest X-ray. After the label is predicted, the result is displayed to the user on the output screen.

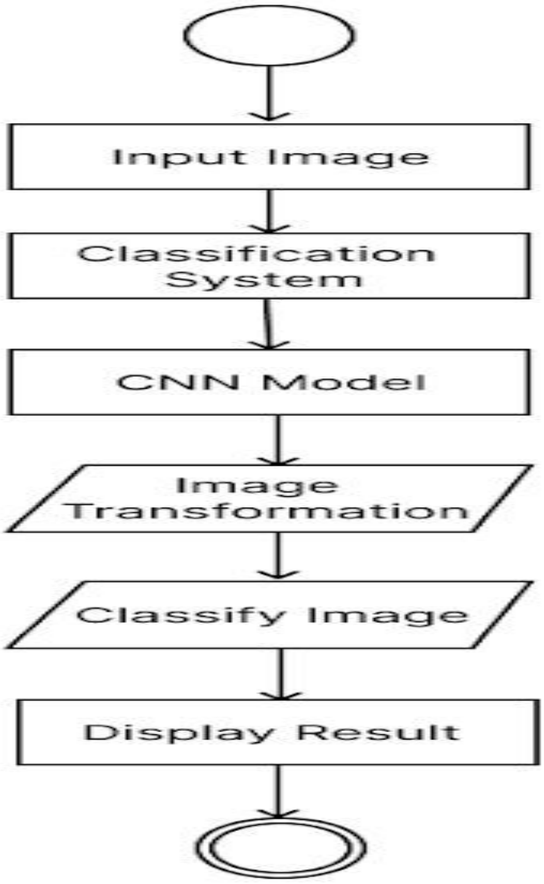


Fig. 5: Activity Diagram

5. SYSTEM DESIGN :

The object-oriented approach is being used for system design. We have developed architecture for the system with a class diagram, sequence diagram, and activity diagram to demonstrate how different models in the system interact to provide collective functionalities.

This project intends to use the CNN, specifically ResNet with various layers for the covid-19 and pneumonia classification. Yet, before that we really want to think about a couple of steps for information readiness. The step including the information readiness and handling along the model execution is displayed in figure 5. Since, we don't have a huge scope dataset for the Coronavirus and pneumonia illness so to make a decent expectation we really want to increase the current information and make new ones. The expanded information is put away in an organizer. The pictures are then provided to the picture handling block where they are, resized to a proper proportion; focus edited to keep away from undesirable lines, convert it to tensors lastly standardize the information. After this we execute our model ResNet model of which the portrayal is given in the underneath segment alongside its figure.

5.1 System Architecture:

The framework begins by setting up the dataset from different sources from the web-based stages. The normal credits from the dataset are converged to shape another dataset. These information are then imported for the further handling. The initial step is to expand the information to make a greater amount of them. Since we will require more information for a superior exhibition assessment, the information is then put away in an envelope for future trial and error which ought to be finished by tuning the hyperparameters. After that the information is handled involving the four stages as follows:

• Resize: The pictures should be resized to have a uniform goal. Since we are bringing in the information from different sources we want to make a predictable aspect while passing to the model.
• Focus Harvest: After the pictures are being resized, we want to trim it from the middle to eliminate the overabundance of the boundary from the pictures.

• Type Change: After the editing system is finished, the pictures are then switched over completely to tensors from the numpy cluster.
• Standardization: At last, the power worth of the pixels in the pictures are standardized for different purposes like, try not to disappear slope issue, make a uniform dissemination of the qualities.

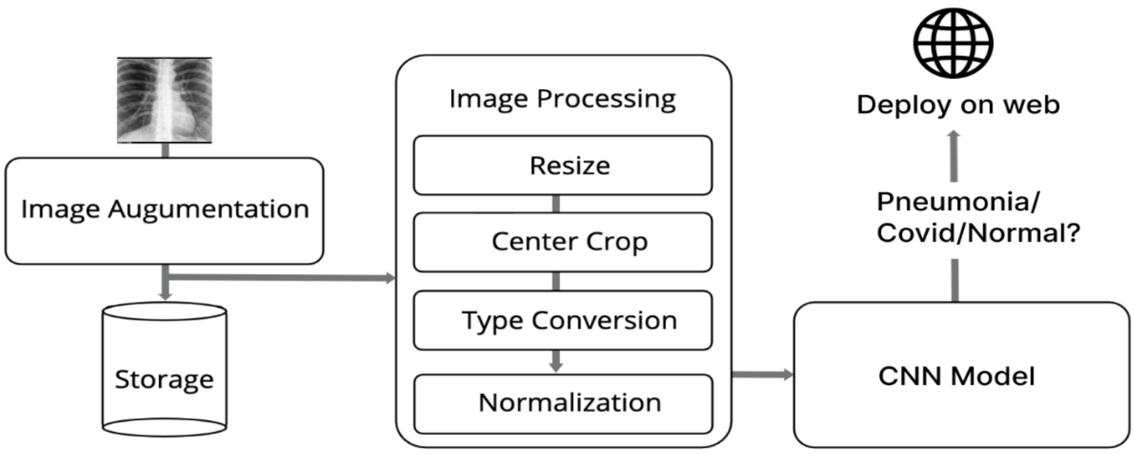


Fig. 6: System Architecture

5.2 Deployment Diagram:

This figure 7 shows the deployment architecture of our system. The user communicates with the system through a web browser where the user has the capability to upload an image file to the web system. The Classification system is deployed using framework known as Gradio. Also, an API is provided to communicate from the external application 15 program. The image files are saved in the file storage for future analysis.

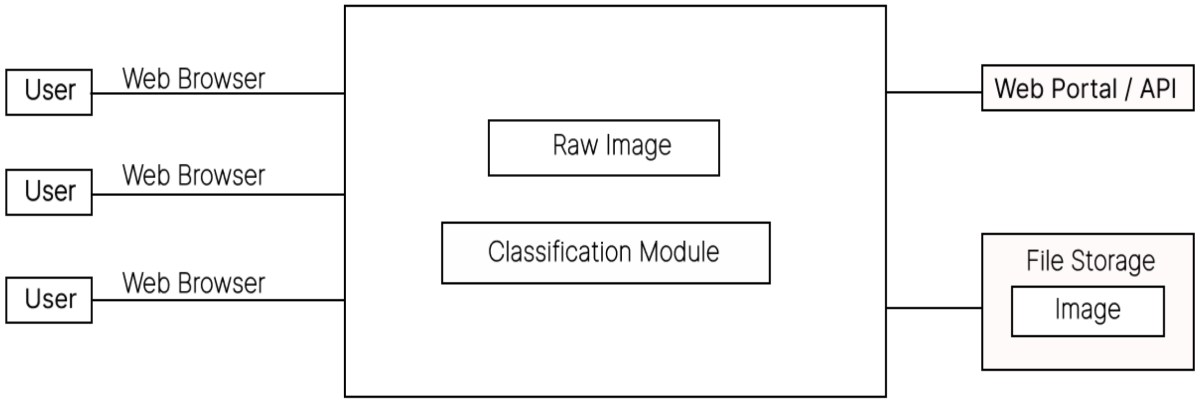


Fig. 7: Deployment Diagram

5.3 Component Diagram:

The input X-ray image is fed into the system which is then converted into the tensor to feed into the model. The model passes the tensor through a series of operations which finally yields the classification as result shown in figure 8.

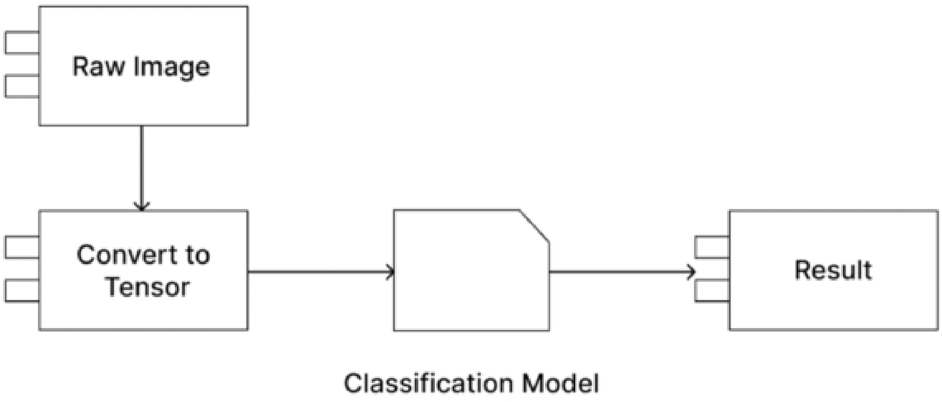


Fig. 8: Component Diagram

5.4 Convolutional Neural Network (CNN):

A convolutional neural network (CNN) can also be used for 1-dimensional data such as speech data or time-series data to extract features from sequences of observations [22].

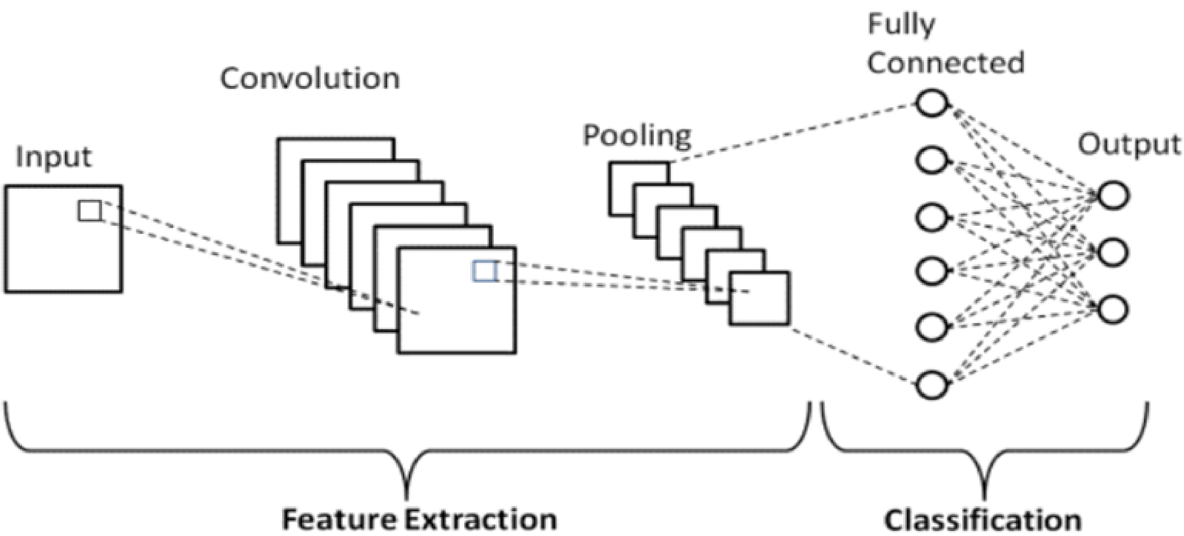


Fig. 9: Architecture of a CNN

i. <u>Input Layer:</u>
The input layer represents the input sequence into the CNN. In the case of image, it would be of 3 dimensions.

ii. <u>Convolution Layer:</u>
The size of portions is a hyper-boundary determined by the originators of the organization engineering. To create the result of the convolutional neuron (initiation map), we should play out an elementwise spot item with the result of the past layer and the extraordinary piece advanced by the organization.

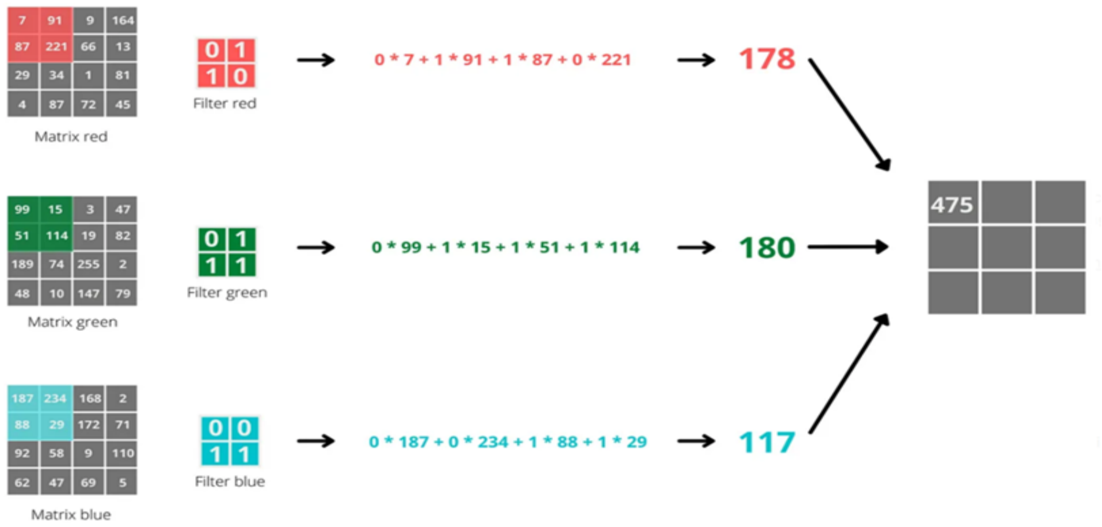


Fig. 10: Convolution Operation

There are multiple hyper parameters in the convolution layer. They are:
Padding is often necessary when the kernel extends beyond the activation map. It conserves data at the borders of activation maps, which leads to better performance, and it can help preserve the input's spatial size, which allows an architecture designer to build deeper higher-performing networks.
The stride indicates how many pixels the kernel should be shifted over at a time.

iii. Activation function:

This is the dynamic place at the neuron yield. The neurons finish direct or nonlinear choices in view of the actuation capability. It forestalls the amplification of neuron yields due to flowing impact due to going through many layers. The three most significant enactment capabilities are sigmoid, Tanh and Redressed Direct Unit (ReLu).
• Sigmoid: It maps the info values inside the scope of 0 to 1.
• Tanh: It maps the info values between - 1 and 1.
• Corrected direct Unit: This capability permits just the positive qualities to course through. The negative qualities are planned at 0.

iv. Pooling Layer:

The pooling layer helps to lessen the spatial size of the portrayal, which diminishes the necessary measure of calculation and loads. The pooling activity is handled on each cut of the portrayal exclusively [14].

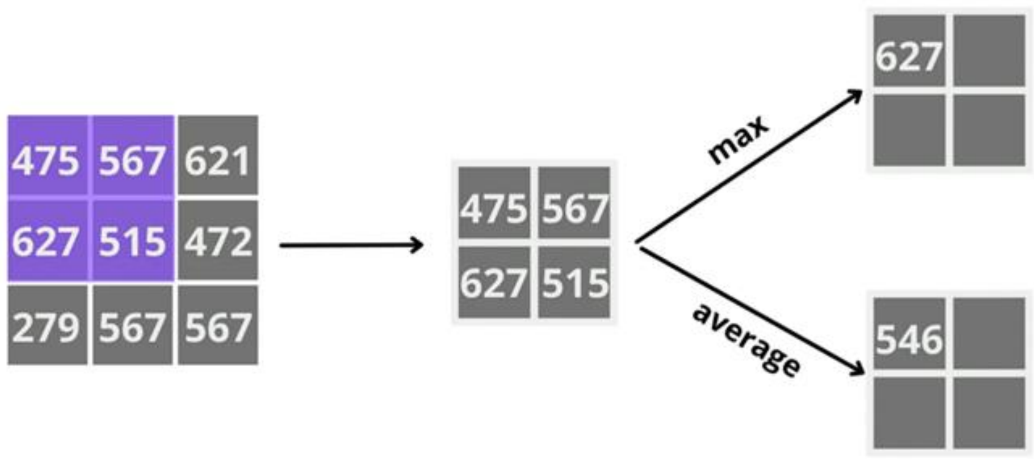


Fig. 11: Pooling Operation

One of the key innovations in the ResNet-18 architecture is the residual block. The residual block allows for deeper neural networks to be trained without encountering the vanishing gradient problem, where the gradient becomes so small that it does not effectively update the weights in the earlier layers of the network. The residual block achieves this by introducing skip connections that allow the input to bypass one or more layers and be added directly to the output of the block.

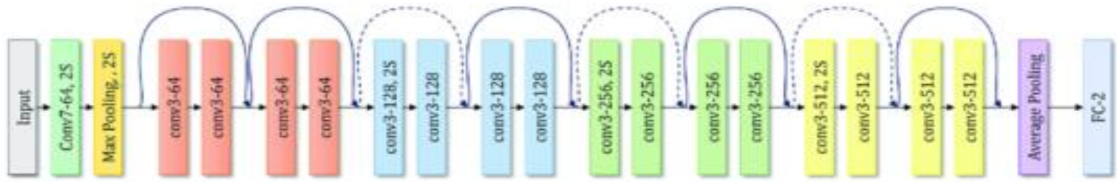


Fig. 12: ResNet-18 Architecture

The ResNet-18 architecture has four stages, each of which consists of multiple residual blocks. The first stage includes a single convolution layer followed by a max pooling layer, while the other stages have multiple residual blocks of varying depths. The final stage includes a global average pooling layer and a fully connected layer that produces the output probabilities. The stages of the ResNet-18 are:

Stage 1: The first stage of ResNet-18 consists of a single convolutional layer with 64 filters and a kernel size of 7x7, followed by a max pooling layer with a pool size of 3x3 and stride 2. The output of this stage is 56x56x64.

Stage 2: The second stage consists of two residual blocks, each with two convolutional layers and 64 filters. The first layer in each block has a kernel size of 3x3 and the second layer has a kernel size of 1x1. The output of this stage is 28x28x128.

Stage 3: The third stage consists of two residual blocks, each with two convolutional layers and 128 filters. The first layer in each block has a kernel size of 3x3 and the second layer has a kernel size of 1x1. The output of this stage is 14x14x256.

Stage 4: The fourth stage consists of two residual blocks, each with two convolutional layers and 256 filters. The first layer in each block has a kernel size of 3x3 and the second layer has a kernel size of 1x1. The output of this stage is 7x7x512.

Global Average Pooling and Fully Connected Layers: The final stage consists of a global average pooling layer that takes the output of the previous stage and produces a 512-dimensional feature vector for each image. This is followed by a fully connected layer that produces the output probabilities for each image [23, 24 & 25].

6. IMPLEMENTING AND TESTING :

The tools used in the project are Python, PyTorch framework as language, Gradio as Deployment Google Colab as IDE, Google Colab's GPU with Google Drive for storage. Data required for this project was freely available online. Data was collected from Mendeley Data [9] which has a collection 5228 images of labeled data that are divided into 3 sub-folders, namely COVID-19, NORMAL and PNEUMONIA. The images contained in each subfolder which we used to train our pre-trained CNN model are as shown in table 1:

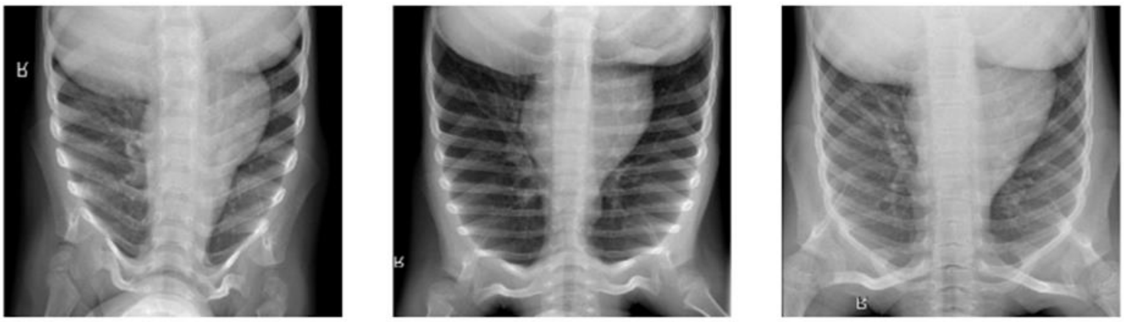
Table 1: No. of images in each folder

Data	No. of data
COVID-19	1626
NORMAL	1802
PNEUMONIA	1800

The features of the collected dataset are all the images are grayscale CXR images, Images of normal CXR consist of images of the heart, lungs, blood vessels, airways, and the bones of the chest and spine, Images of pneumonia and covid-19 infected CXR revealed fluid in/ around lungs, All the images are already pre-processed and resized to 224 x 224 and all the images are in PNG format.
Plates 1 show Sample of COVID-19 infected CXR, followed by plat 2 for Sample of Pneumonia infected CXR and Plat 3 for Sample of Normal CXR are the samples of the images used in our project.

Plate 1: Sample of COVID-19 infected CXR

Plat 2: Sample of Pneumonia infected CXR



Plat 3: Sample of Normal CXR

6.1 Pre-Processing:

The data collected in the data collection section were already preprocessed and resized into 224 x 224. In order to train our model with non-monotonous and a variety of data, some preprocessing operations were performed such as Conversion of image to tensor in order to convert the multi-dimensional image array into a PyTorch tensor, Random rotation, Vertical flip and Horizontal flip.
After converting to tensor, the covid infected CXR image on the left looked like the tensor on the right as shown in figure 13.

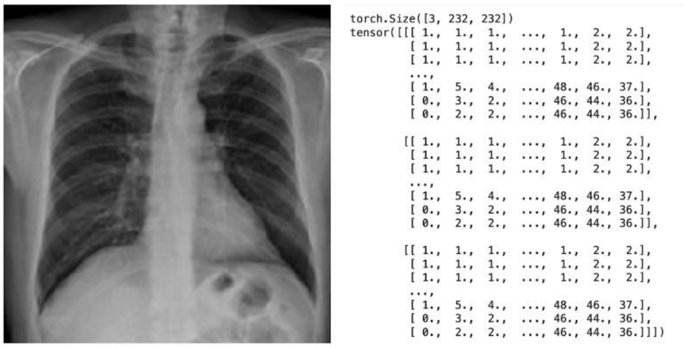


Fig. 13: Covid infected CXR and its corresponding tensor

In the figure 13, the torch Size ([3, 232, 232]) represents the channels, height and width of the image and the tensor represents the input image data. The image dataset before and after all the performing the preprocessing methods mentioned are shown in figure 14 and 15 in before and after condition to compare.

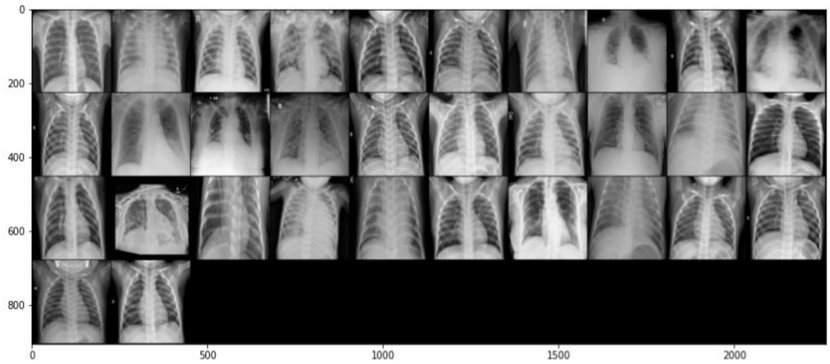


Fig. 14: Image Dataset before preprocessing

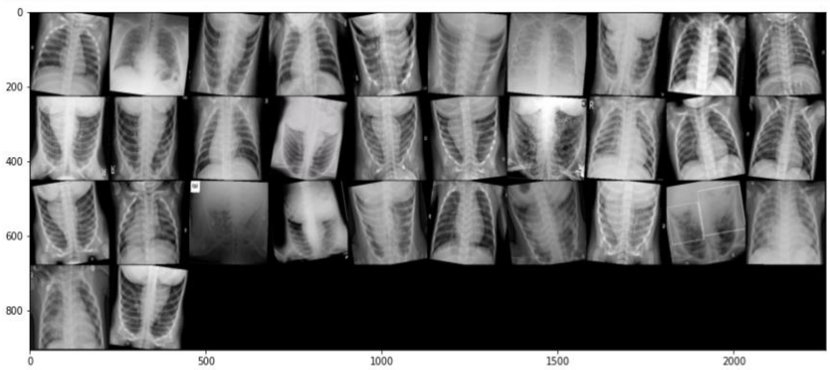


Fig. 15: Image Dataset after preprocessing

6.2 Normalization:

Normalization is a process to bring something to a normal state or condition. Normalization is the process of scaling input data to a specific range or distribution, typically with a mean of zero and a standard deviation of one. In image processing, normalization is performed in order to bring the image pixels into a range that is more familiar or normal to the senses. This will help in improving the convergence of the optimization algorithm during training and prevent the suboptimal performance of features with larger values. In some cases, normalization may not have a significant impact on the performance of the classification model, especially if the features are already on a similar scale. In our project, normalization did not make a better impact since our dataset has similar features. Hence, normalization was abstained from our experiment. Below shown is the image tensor of image before and after normalization as shown in figure 16 and 17.

```
torch.Size([3, 232, 232])
tensor([[[ 1.,  1.,  1.,  ...,  1.,  2.,  2.],
         [ 1.,  1.,  1.,  ...,  1.,  2.,  2.],
         [ 1.,  1.,  1.,  ...,  1.,  2.,  2.],
         ...,
         [ 1.,  5.,  4.,  ..., 48., 46., 37.],
         [ 0.,  3.,  2.,  ..., 46., 44., 36.],
         [ 0.,  2.,  2.,  ..., 46., 44., 36.]],

        [[ 1.,  1.,  1.,  ...,  1.,  2.,  2.],
         [ 1.,  1.,  1.,  ...,  1.,  2.,  2.],
         [ 1.,  1.,  1.,  ...,  1.,  2.,  2.],
         ...,
         [ 1.,  5.,  4.,  ..., 48., 46., 37.],
         [ 0.,  3.,  2.,  ..., 46., 44., 36.],
         [ 0.,  2.,  2.,  ..., 46., 44., 36.]],

        [[ 1.,  1.,  1.,  ...,  1.,  2.,  2.],
         [ 1.,  1.,  1.,  ...,  1.,  2.,  2.],
         [ 1.,  1.,  1.,  ...,  1.,  2.,  2.],
         ...,
         [ 1.,  5.,  4.,  ..., 48., 46., 37.],
         [ 0.,  3.,  2.,  ..., 46., 44., 36.],
         [ 0.,  2.,  2.,  ..., 46., 44., 36.]]])
```

```
torch.Size([3, 232, 232])
tensor([[[0.0039, 0.0039, 0.0039,  ..., 0.0039, 0.0078, 0.0078],
         [0.0039, 0.0039, 0.0039,  ..., 0.0039, 0.0078, 0.0078],
         [0.0039, 0.0039, 0.0039,  ..., 0.0039, 0.0078, 0.0078],
         ...,
         [0.0039, 0.0196, 0.0157,  ..., 0.1882, 0.1804, 0.1451],
         [0.0000, 0.0118, 0.0078,  ..., 0.1804, 0.1725, 0.1412],
         [0.0000, 0.0078, 0.0078,  ..., 0.1804, 0.1725, 0.1412]],

        [[0.0039, 0.0039, 0.0039,  ..., 0.0039, 0.0078, 0.0078],
         [0.0039, 0.0039, 0.0039,  ..., 0.0039, 0.0078, 0.0078],
         [0.0039, 0.0039, 0.0039,  ..., 0.0039, 0.0078, 0.0078],
         ...,
         [0.0039, 0.0196, 0.0157,  ..., 0.1882, 0.1804, 0.1451],
         [0.0000, 0.0118, 0.0078,  ..., 0.1804, 0.1725, 0.1412],
         [0.0000, 0.0078, 0.0078,  ..., 0.1804, 0.1725, 0.1412]],

        [[0.0039, 0.0039, 0.0039,  ..., 0.0039, 0.0078, 0.0078],
         [0.0039, 0.0039, 0.0039,  ..., 0.0039, 0.0078, 0.0078],
         [0.0039, 0.0039, 0.0039,  ..., 0.0039, 0.0078, 0.0078],
         ...,
         [0.0039, 0.0196, 0.0157,  ..., 0.1882, 0.1804, 0.1451],
         [0.0000, 0.0118, 0.0078,  ..., 0.1804, 0.1725, 0.1412],
         [0.0000, 0.0078, 0.0078,  ..., 0.1804, 0.1725, 0.1412]]])
```

Fig. 16: Image tensor before normalization and after normalization

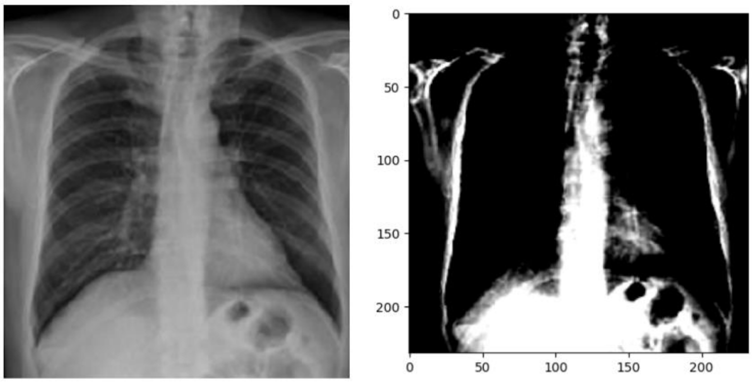


Fig. 17: Covid infected CXR image before normalization (left) and after normalization (right)

6.3 Data Splitting:

After performing necessary operations on the image dataset, the dataset was splitted in the ratio of 9:1 where 90% of the data was used for training purposes and 10% of the data was used for testing purposes based on figure 18 and 19.

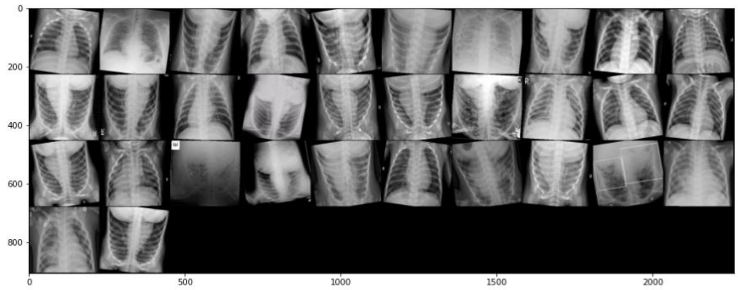


Fig. 18: Sample of Training dataset

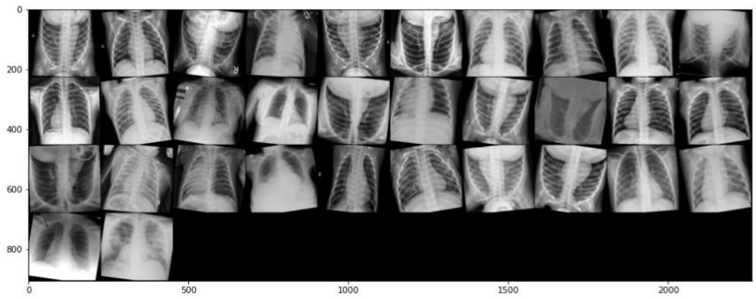


Fig. 19: Sample of testing dataset

6.4 Training and Testing:

6.4.1 Training of Model:

After preparing the data and splitting it into training and testing sets, it was now time to train the CNN model. We used the pre-trained ResNet-18 model and modified the model in order to fit our problem. For that purpose, we added one extra layer after the last layer to make 3 neurons in the output layer to fit our problem of classification. We added a linear classifier layer at the end. Since our problem is a multi-class classification problem, we wanted to add a Softmax layer manually. But, while researching, we came to find out that the Cross Entropy Loss which is a loss function used for multi-class classification tasks, in PyTorch, specifically, adds the softmax layer internally, automatically. Hence, we did not have to add the Softmax layer manually in our code, and it was suitable for a multi-class classification problem like ours. Then, we fed the training data (4705 images) into the pre-trained ResNet-18 model by choosing a random learning rate, epoch size and an appropriate activation function as shown in table 2.

Table 2: Experiments performed on ResNet-18 Model

No. of Epochs	Learning rate (LR)
5	0.1
10	0.1
15	0.1
10	0.01
10	0.001
10	0.0001
20	0.01

In the pre-trained ResNet-18 model implemented in PyTorch, the ReLU (Rectified Linear Unit) Activation Function is used and it is fixed. The ReLU activation function is used throughout the network, both in the residual blocks and in the final fully connected layers. This activation function is applied to the output of each layer before passing it to the next layer as of figure 20**.**

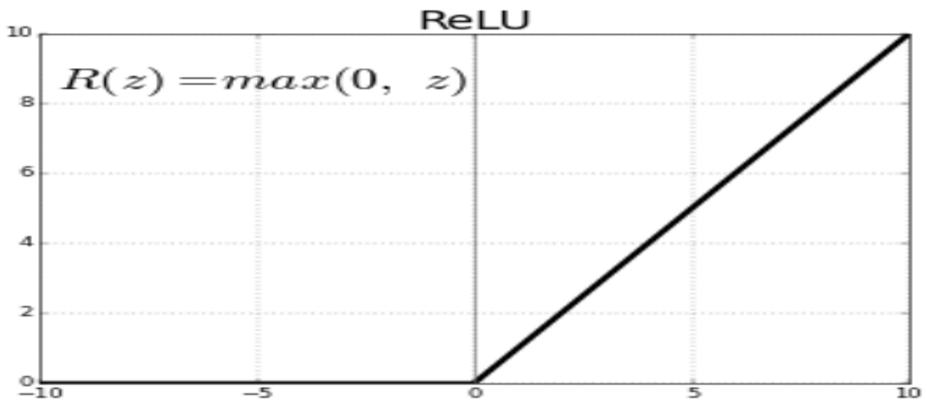


Fig. 20: Hyper parameters used in training ResNet-18 model

The ReLU is half rectified (from bottom). f(z) is zero when z is less than zero and f(z) is equal to z when z is above or equal to zero. Range: [0 to infinity)

Testing of Data:

After training the model with the training data i.e. 90% of the total dataset, the testing data (523 images) was applied to the model. The model was trained with the training data and tested with images from the testing dataset. Various measures were calculated simultaneously such as the train loss, train accuracy, test-loss and test-accuracy. These measures were used to calculate the performance measures like precision, recall, f1-score and confusion matrix in the later phase of the project, which is explained in the upcoming section of the report in detail.

Performance Measure/ Result Analysis:

The experiments were performed as mentioned above. All the experiments did not yield good results. We started experimenting by setting up the learning rate as 0.1, using the Adam optimizer in epoch 10. Then optimization was done by decreasing the learning rate at each step, keeping the optimizer and number of epochs constant. Later on, the number of epochs was also increased from 5 to 10 to 15 then to 20. The results obtained during each experiment with their loss and accuracy in each epoch is given in figure 21, classification in figure 22, Confusion matrix for epoch 5, lr 0.1 in 23, Area under the ROC curve for epoch 5, lr 0.1 in figure 24, Train vs test accuracy for epoch 5, lr 0.1 in figure 25 and Train vs test loss for epoch 5, lr 0.1 in figure 26. Similarly, Experiment 2: Epoch 10, lr 0.1, Experiment 3: Epoch 15, lr 0.1, Experiment 4: Epoch 10, lr 0.01, Experiment 5: Epoch 10, lr 0.001, Experiment 6: Epoch 10, lr 0.0001 and Experiment 7 were done. Let's show experiment 7 results also.

<u>Experiment 1: Epoch=5, lr= 0.1:</u>

```
Epoch: 0 Train-loss: 1.5568104395391167 Train-accuracy: 63.6769394261424 Test-loss: 1.542928397655487 Test-accuracy: 68.83365200764818
Epoch: 1 Train-loss: 0.5196080268738238 Train-accuracy: 82.59298618490966 Test-loss: 1.141748961280374 Test-accuracy: 73.9961759082218
Epoch: 2 Train-loss: 0.3853481047459551 Train-accuracy: 86.86503719447396 Test-loss: 4.933940789278815 Test-accuracy: 35.94646271510516
Epoch: 3 Train-loss: 0.3458092536374524 Train-accuracy: 88.84165781083954 Test-loss: 3.6926644426915786 Test-accuracy: 41.491395793499045
Epoch: 4 Train-loss: 0.31424597761518247 Train-accuracy: 88.90541976620617 Test-loss: 0.7622419911272386 Test-accuracy: 68.45124282982792
Best train accuracy: 82.59298618490966
Best test accuracy: 73.9961759082218
```

Fig. 21: Output result for epoch 5, lr 0.1

Classification report for epoch=5 lr=0.1

	precision	recall	f1-score	support
0	0.94	0.78	0.86	170
1	0.53	0.99	0.69	171
2	0.93	0.30	0.46	182
accuracy			0.68	523
macro avg	0.80	0.69	0.67	523
weighted avg	0.80	0.68	0.66	523

Fig. 22: Classification report for epoch 5, lr 0.1

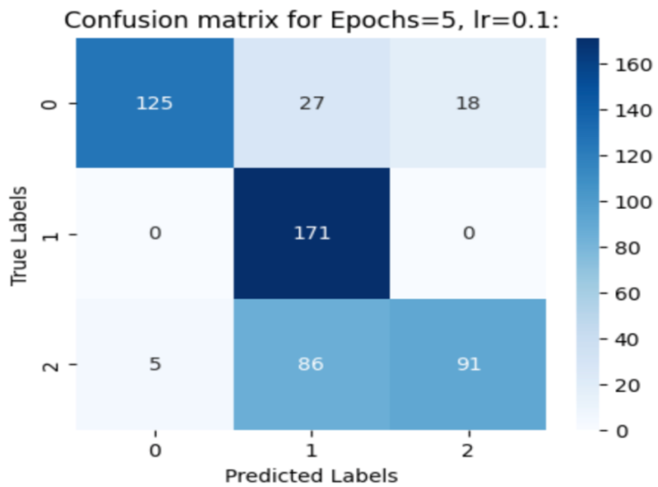


Fig. 23: Confusion matrix for epoch 5, lr 0.1

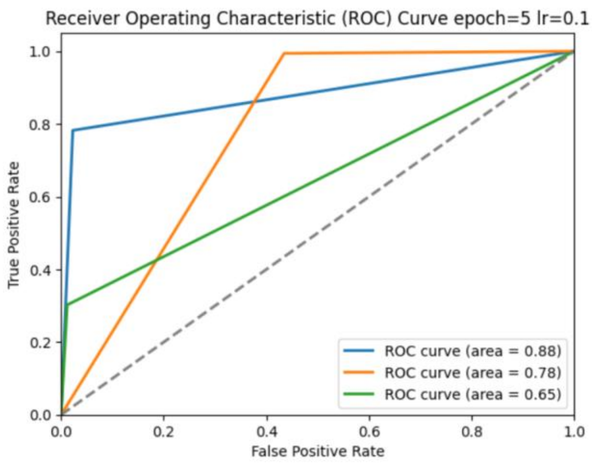


Fig. 24: Area under the ROC curve for epoch 5, lr 0.1

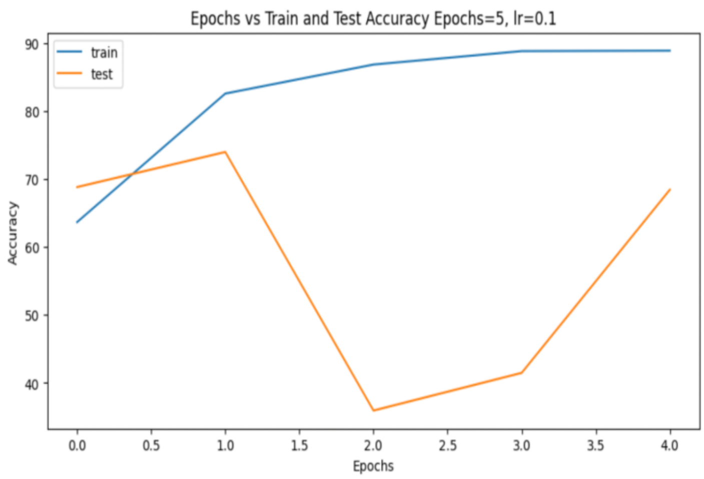


Fig. 25: Train vs test accuracy for epoch 5, lr 0.1

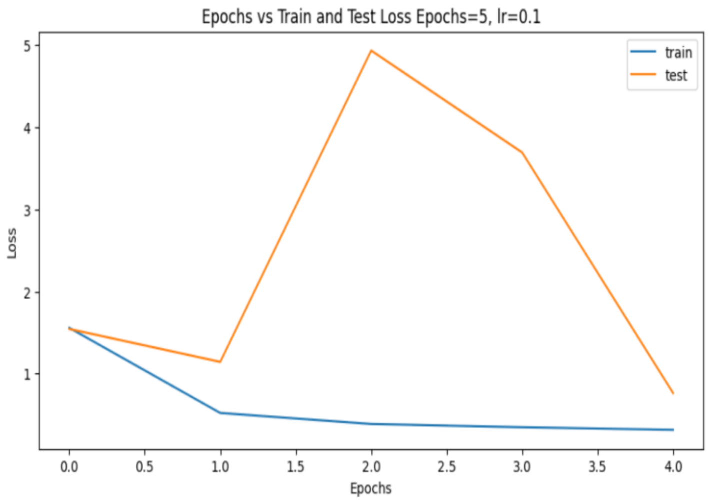


Fig. 26: Train vs test loss for epoch 5, lr 0.1

Experiment 7: Epoch 20, lr 0.01:

Classification report for epoch=20 lr=0.01:

	precision	recall	f1-score	support
0	0.98	0.64	0.78	170
1	0.53	1.00	0.69	171
2	0.93	0.45	0.61	182
accuracy			0.69	523
macro avg	0.81	0.70	0.69	523
weighted avg	0.82	0.69	0.69	523

Fig. 27: Classification report for epoch 20, lr 0.01

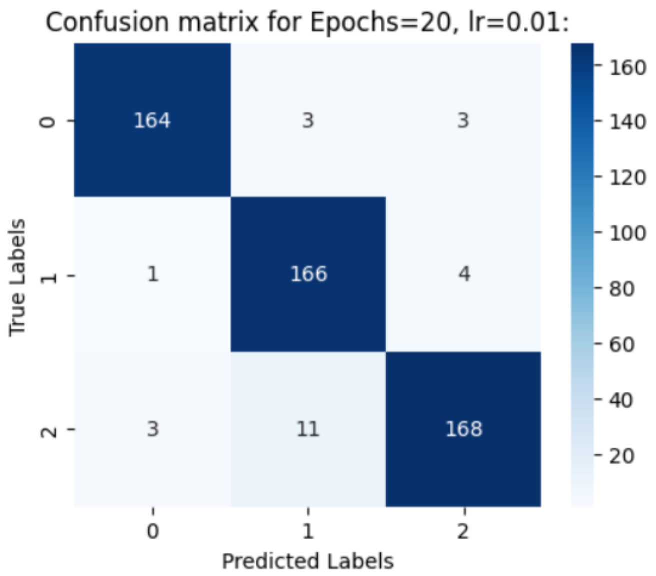


Fig. 28: Confusion matrix for epoch 20, lr 0.01

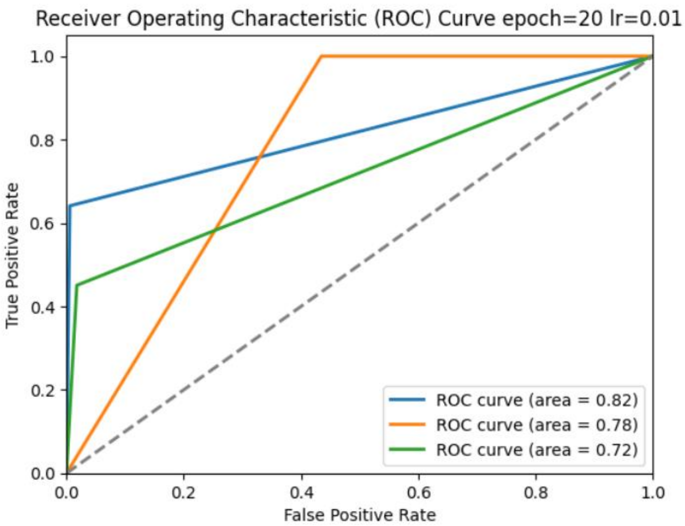


Fig. 29: Area under the ROC curve for epoch 20, lr 0.01

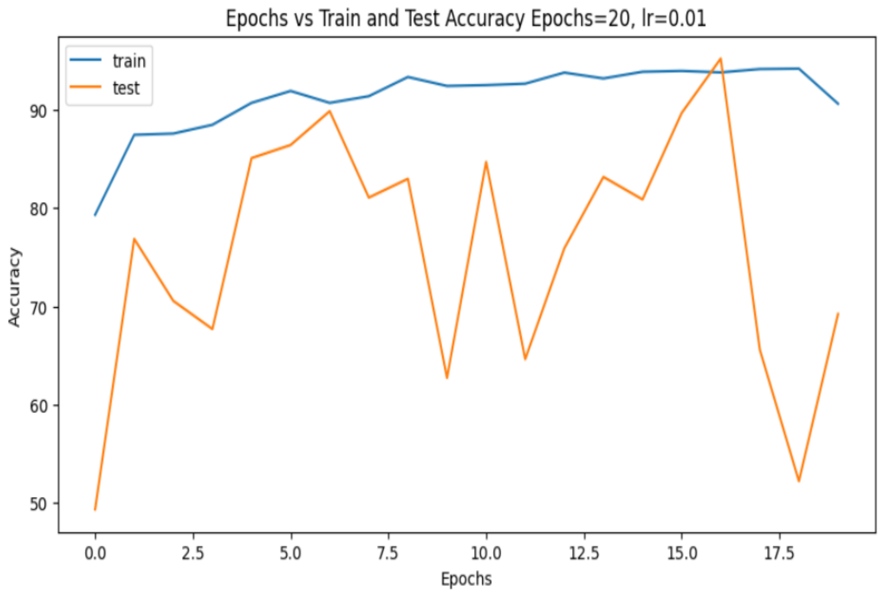


Fig. 30: Train vs. test accuracy for epoch 20, lr 0.01

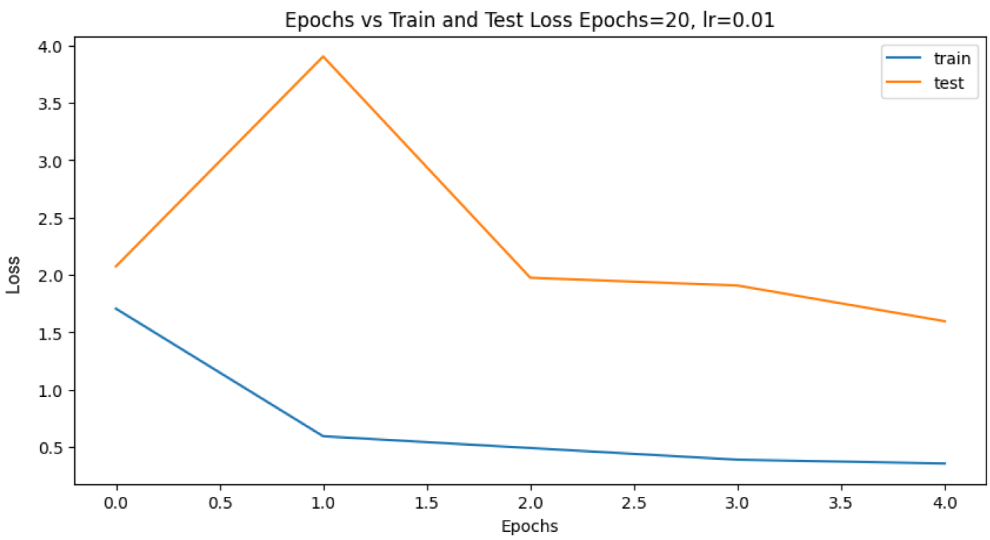


Fig. 31: Train vs. test loss for epoch 20, lr 0.01

From the results of the experiments performed under different hyper-parameters, as shown in above figures, we can observe that on experiment 1 i.e. when epoch=5 and lr=0.1, the train and test accuracies are 82.46% and 64.24% respectively. This suggests that the train accuracy is the only metric that seems to be fine but the other metrics (test-accuracy, f1-score, precision, recall, ROC Curve) are not as convincing as we can observe in figures. On increasing the number of epochs and keeping the learning rate as it is, we notice that the metrics have improved a little bit, if not so much. Analyzing these results, we kept on playing with the hyper parameters and we can notice that on epoch=10 and lr=0.0001, we get convincing performance metrics where:

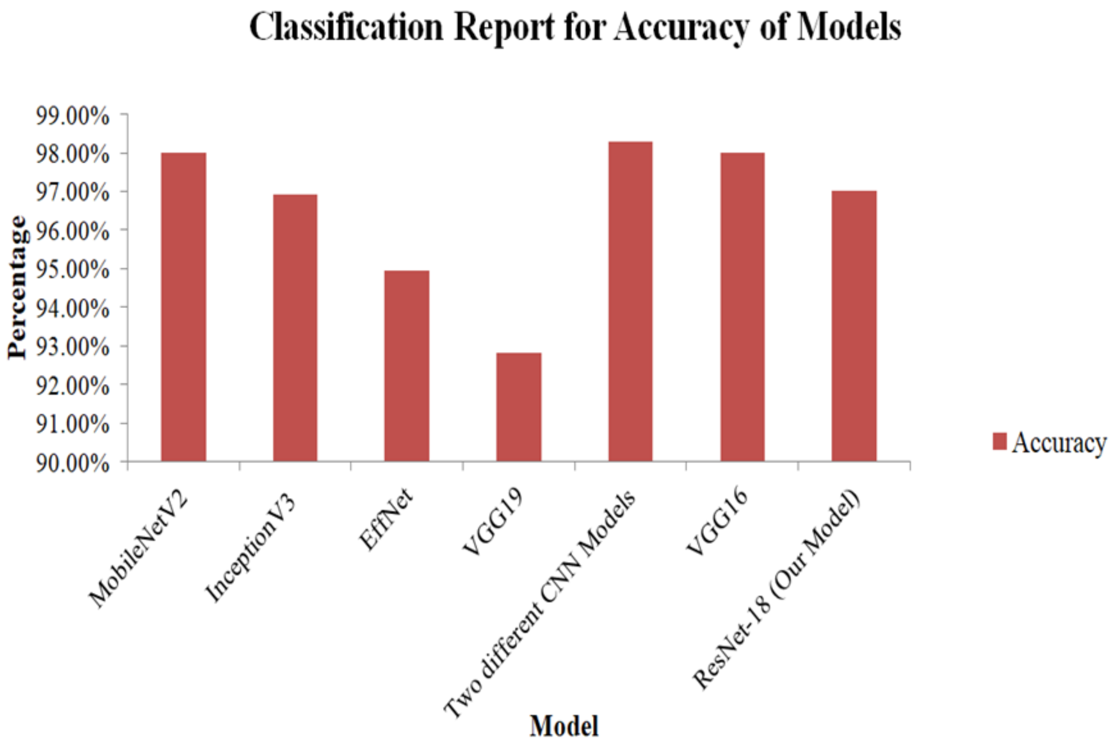


Fig. 32: Classification Report for accuracy of models

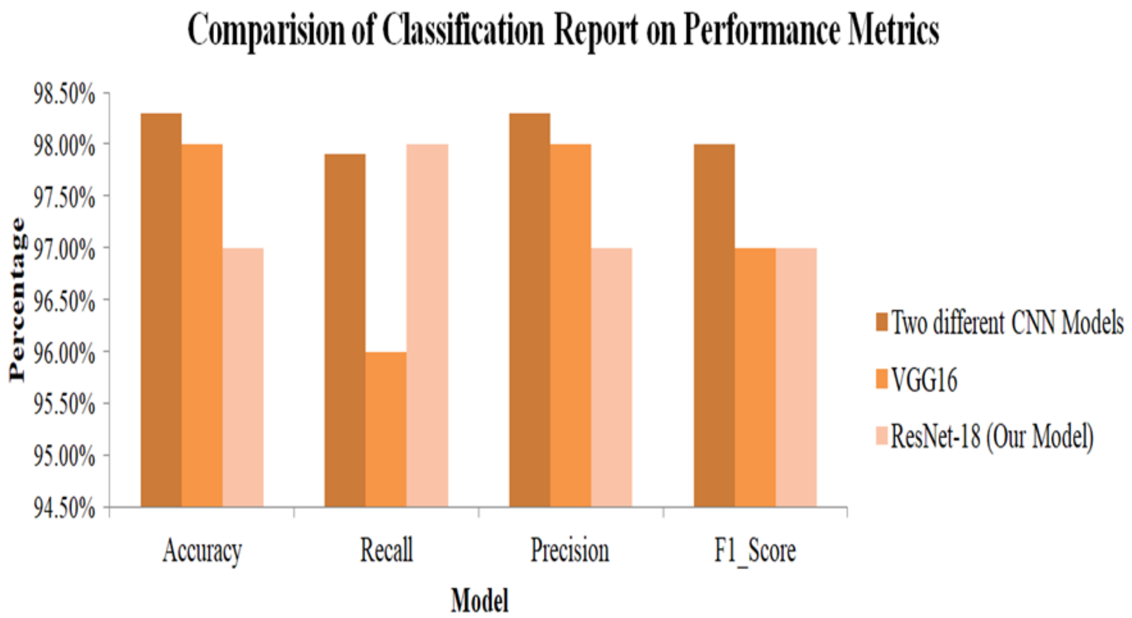


Fig. 33: Comparison of Classification Report on Performance Metrics

Train accuracy = 98.12% Test accuracy = 97.70%
Precision= 0.99, 0.95 and 0.99 for labels 0, 1 and 2 respectively
Recall= 1, 0.99, 0.95 for labels 0, 1 and 2 respectively
F1-score = 0.99, 0.97, 0.97 for labels 0, 1 and 2 respectively
Support= 170, 171, 182 for labels 0, 1 and 2 respectively
ROC Curve= 1, 0.98, 0.97 for labels 0, 1 and 2 respectively
This is the best result that we have observed so far, especially on epoch 9.
On further increasing the number of epochs to 15 and 20 we did not yield as good results. Hence, we decided to stick on this result and deploy the model as it is.

The validation of the model was done by comparing our results with previously done, similar kinds of works by other researchers as in table 3.

Table 3: Comparison Table for validation of the model

Model	Classification Report	Reference	Classification Report of our Model
MobileNetV2 InceptionV3 EffNet VGG19	Accuracy: 98% Accuracy: 96.92% Accuracy: 94.95% Accuracy: 92.82%	[9]	Accuracy: 97% F1-Score: 97% Precision: 97% Recall: 98% Support: 174.33
Two different CNN Models	Accuracy: 98.3% Recall: 97.9% Precision: 98.3% F1_Score: 98.0%	[10]	
VGG16	Accuracy: 98% Recall: 96% Precision: 98% F1_Score: 97%	[11]	

The specialist to populace proportion for the entirety nation is 1:1724. Like generally creating countries, specialists are topographically maldistributed in Nepal. The Kathmandu valley has one specialist for 850 individuals yet in provincial regions the number is one specialist for each 150000 individuals [26, & 27]. The specialist populace thickness in Kathmandu is assessed to be multiple times that in provincial Nepal [26, 27, 28 & 29]**.** In this context technological dependency is a most for countries like Nepal. As we are moving to society 5.0, we must be ready for close interaction with technology [16, 19, and 21].

7. CONCLUSIONS :

The project's primary focus was to build a system capable of classifying the pneumonia, covid19 and normal healthy chest x-ray images with fast interference speed in web systems of any hospitals or health care centers to make chest X-ray diagnosis more accurately and avoid further spreading of Covid-19 and Pneumonia. With addition of augmented data, the classification can be more accurate. The system for classification of Pneumonia, covid-19 and normal chest x-ray images was done successfully using the pre-trained ResNet-18 CNN model with chest X-ray image data that helped to build a robust classification system with learning rate 0.0001 and epoch size 10 having approx 98.12% train accuracy and 97.70% test accuracy.

Since the start of the project, we researched several methodologies to build the system. The other models (e.g. ResNet-50) were too big algorithms for our problem which created a problem of over fitting. Hence performance was not very accurate. So, we planned to go with the ResNet-18 model. As per our plan, we developed a system that operates as expected. The system can continually be improved. We can build a classification model with large datasets, which should be able to give better results. With sufficient compute resources, we can perform further hyper parameter tuning and train the model for longer periods which should increase the system's performance. The study has brought a big solution for developing countries where numbers of doctors are less in comparison to that of nation demands on the basis of population particularly in Nepal.

8. ACKNOWLEDGEMENT :

The author is thankful to all the professionals who took part in discussions. The Author thanks to Saanvi Lavanya (Betkumar) for being with us as source of happiness. It is an independent academic exercise conducted at conducted among authors without involvement of any organizations though authors are associated with different organizations, No institutions have any rights to claim on it as it is outcome

authors hard work on their own expenses. 1 author has documented, 2, 3 and 4 have collected data and lead the experiment under guidance of 5, 6 author. Author 7 has guided the overall process. All authors have performed experiments repeatedly.

REFERENCES :

[1] Vijay Kumar Jain, Karthikeyan P. Iyengar, Pranav Ish (2021). Elucidating causes of COVID-19 infection and related deaths after vaccination. *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, *15*(5), https://doi.org/10.1016/j.dsx.2021.102212.

[2] Global COVID-19 Tracker - Kaiser Family Foundation [https://www.kff.org/coronavirus-covid-19/issue-brief/global-covid-19-tracker/]

[3] WHO Coronavirus (COVID-19) Dashboard [https://covid19.who.int]

[4] Coronavirus Death Toll and Trends - Worldometer [https://www.worldometers.info/coronavirus/coronavirus-death-toll/]

[5] WHO Coronavirus (COVID-19) Dashboard With Vaccination Data [https://covid19.who.int/?mapFilter=vaccinations]

[6] Cumulative confirmed COVID-19 deaths by world region [https://ourworldindata.org/grapher/cumulative-covid-deaths-region]

[7] COVID - Coronavirus Statistics - Worldometer [https://www.worldometers.info/coronavirus/]

[8] World Health Organization, (2022). "Pneumonia in children," https://www.who.int/news-room/fact-sheets/detail/pneumonia

[9] Lv, D. F., Ying, Q. M., He, Y. W., Liang, J., Zhang, J. H., Lu, B. B., Qian, G. Q., Chu, J. G., Weng, X. B., Chen, X. Q., & Mu, Q. T. (2021). Differential diagnosis of coronavirus disease 2019 pneumonia or influenza A pneumonia by clinical characteristics and laboratory findings. *Journal of clinical laboratory analysis*, *35*(2), e23685. https://doi.org/10.1002/jcla.23685.

[10] Abiyev, A. I. R. H. (2021). COVID-19 and pneumonia diagnosis in X-ray images using convolutional neural networks. Hindawi. https://doi.org/10.1155/2021/6653823.

[11] Verma, D. K., Saxena, G., Paraye, A., Rajan, A., Rawat, A., & Verma, R. K. (2022). Classifying COVID-19 and Viral Pneumonia Lung Infections through Deep Convolutional Neural Network Model using Chest X-Ray Images. *Journal of medical physics*, *47*(1), 57–64. https://doi.org/10.4103/jmp.jmp_100_21

[12] Karar, M. E., Hemdan, E. ED. & Shouman, M. A. (2021). Cascaded deep learning classifiers for computer-aided diagnosis of COVID-19 and pneumonia diseases in X-ray scans. *Complex Intell. Syst. 7*(1), 235–247 (2021). https://doi.org/10.1007/s40747-020-00199-4.

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

[14] Mahin, Mainuzzaman, Sajid Tonmoy, Rufaed Islam, Tahia Tazin, Mohammad Monirujjaman Khan, and Sami Bourouis. (2021). Classification of COVID-19 and Pneumonia Using Deep Transfer Learning. edited by K. Rajakani. Journal of Healthcare Engineering, 3514821. DOI: https://doi.org/10.1155/2021/3514821.

[15] Marnada, P., Raharjo, T., Hardian, B., & Prasetyo, A. (2021). Agile project management challenge in handling scope and change: A systematic literature review. *Procedia Computer Science*, 197(1), 290-300. https://doi.org/10.1016/j.procs.2021.12.143

[16] Mishra, A. K., Nepal Ananda, P. S. Aithal, (August 2022). Industry 4.0 Concept for Nepal - Operating Virtual Farming Industry. PP- 31-35, Proceedings on Future Trends in ICCT and its Applications in IT, Management and Education, Editors: Dr. Krishna Prasad, K., Dr. P. S. Aithal, & Dr. A. Jayanthiladevi, ISBN: 978-81-949961-8-7, DOI: https://doi.org/10.5281/zenodo.7215189

[17] Mishra, A. K. (2020). Project management: theory and practice from different countries. Project management: theory and practice from different countries. Tamilnadu: DK International Research Foundation. http://doi.org/10.5281/zenodo.4817542

[18] Mishra, A. K., Jha, P. B., & Aithal, P. S., (2023). Business Operation Using Identification of Product in Context of Developing Countries Emphasizing Nepal. *International Journal of Applied Engineering and Management Letters (IJAEML), 7*(1), 112-126. DOI: https://doi.org/10.5281/zenodo.7798162

[19] Rajat Pokharel, Mishra, A. K., & Aithal, P. S., (2021). Practicability Assessment of Smart VillageProject: A Case of Sandakpur Rural Municipality, Ilam Nepal. *International Journal of Management, Technology, and Social Sciences (IJMTS), 6*(2), 265-281. DOI: https://doi.org/10.5281/zenodo.5799392.

[20] Jha, P. B., Mishra, A. K., & Aithal, P. S., (2023). Operation of Vehicle Maintenance Systemin Context of Developing Countries Emphasizing Nepal. *International Journal of Case Studies in Business, IT, and Education (IJCSBE), 7*(1), 267-279. DOI: https://doi.org/10.5281/zenodo.7790868.

[21] Bhagat, C., Mishra, A. K., & Aithal, P. S. (2022). Model for Implementation of e-Government Services in Developing Countries like Nepal. *International Journal of Case Studies in Business, IT, and Education (IJCSBE), 6*(2), 320-333. DOI: https://doi.org/10.5281/zenodo.7139657.

[22] Abhishek, A. V. S. (2022). Resnet18 Model with Sequential Layer for Computing Accuracy on Image Classification. *International Journal of Creative Research Thoughts (IJCRT), 10*(5), 176-181. https://ijcrt.org/papers/IJCRT2205235.pdf.

[23] Kumar, Sachin (2022), "Covid19-Pneumonia-Normal Chest X-Ray Images", Mendeley Data, V1, DOI: https://doi.org/10.17632/dvntn9yhd2.1

[24] Shastri, S., Kansal, I., Kumar, S. et al. (2022). CheXImageNet: a novel architecture for accurate classification of Covid-19 with chest x-ray digital images using deep convolutional neural networks. Health Technol. 12, 193–204. https://doi.org/10.1007/s12553-021-00630-x

[25] Kumar S, Shastri S, Mahajan S, et al. (2022). Lite CovidNet: A lightweight deep neural network model for detection of COVID-19 using X-ray images. *Int J Imaging Syst Technol*, 1-17. DOI: https://doi.org/10.1002/ima.22770

[26] Graph Nepal. Number of doctors in Nepal (1960-2015). Available from http://graphnepal.com/number-of-doctors-in-nepal/.

[27] Patan Academy of Health Sciences. Nepal at a glance. Available from: http://www.pahs.edu.np/about/about-nepal/.

[28] Phuyel, B. D. Doctor's brain drain in Nepal: exploring the patterns, causes, consequences and solutions. July 2013. Available from: http://www.pp.utokyo.ac.jp/grasppold/courses/2013/documents/5140143_2a.pdf.

[29] Shankar, P. R. (2018). Brain drain and practice locations of Nepalese medical students. *Janaki Medical College Journal of Medical Science, 5*(2), 1–4. https://doi.org/10.3126/jmcjms.v5i2.19010

****