Pattern Recognition and Machine Learning Project: COVID-19 Detection using chest X-rays

Mukul Shingwani, Saurabh Modi, Mitarth Arora
B20AI023, B20EE035, B20EE096

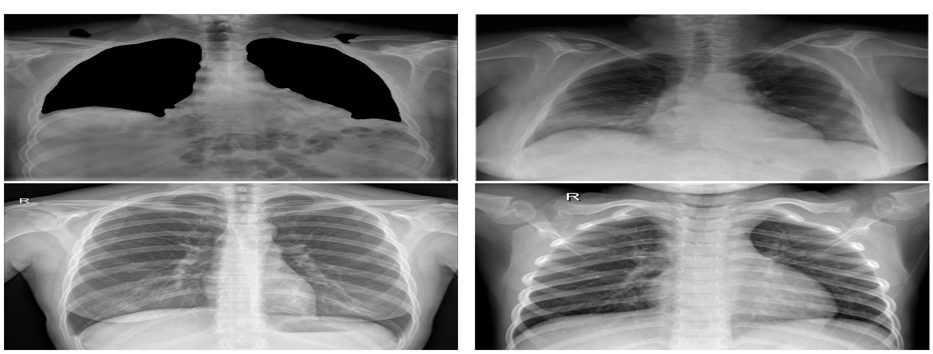
May 2, 2022

1 Abstract

This paper reports our experience with detection of COVID-19 using chest X-ray images(CXR). We here try to implement **Un-Supervised, Supervised and Deep learning methods** for binary image(COVID or NON-COVID) classification on CXR images. Implementation of PCA followed by application of K-means for the unsupervised techniques. In supervised learning we explored models such as Random Forest classifier, Decision Tree classifier, Gaussian-Bayes. For the Deep Learning approach we developed a custom pipeline which involves usage of segmentation and SOTA(State of the art)CNN models.

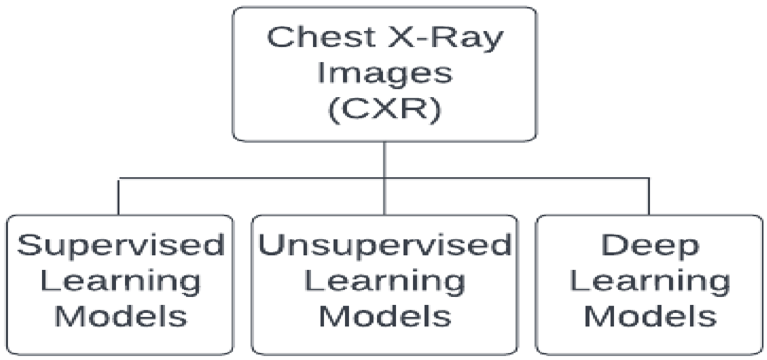
2 Introduction

COVID-19 is an infectious disease which has led to a dramatic loss of human life worldwide, its early detection is critical to control cases' spreading and mortality. The actual leader diagnosis test is the Reverse transcription Polymerase chain reaction (RT-PCR)which is based on Nasopharyngeal swabs (NPSs). some images to show examples of lungs affected and not affected from COVID :-



3 Overview

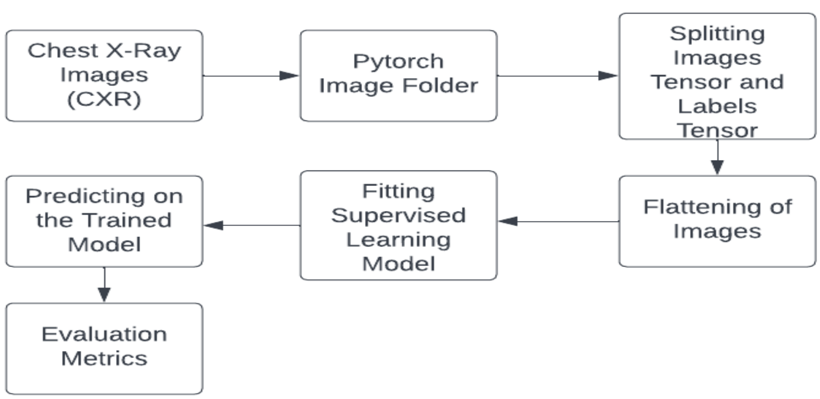
The report is segregated into 3 subsections :-



4 Supervised Learning Techniques

4.1 Pipeline

The Imagefolder function of Pytorch creates a labeled stack of images which finally gets converted from RGB to Grayscale following the pipeline as shown in the figure.



4.2 Evaluation Metrices and Models

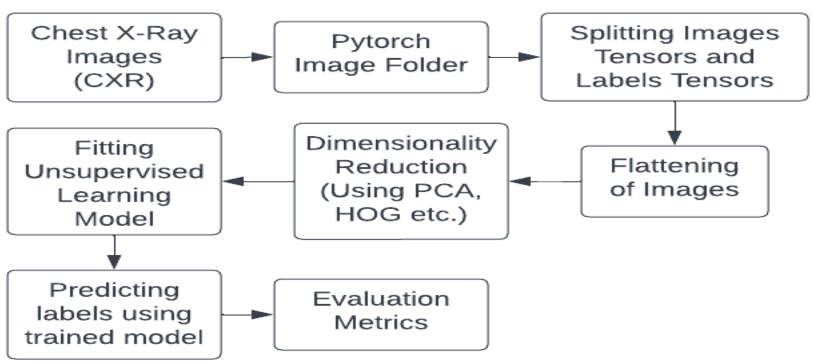
The ROC-AUC score for DTC, RFC,and GNB models are as follows : **0.7843, 0.9347, 0.6878**

Models report				
Model	**F1 score**	**Sensitivity**	**Specificity**	**Accuracy**
Decision Tree classifier	0.7505	0.7811	0.8006	0.7928
Random Forest Classifier	0.8357	0.8550	0.8634	0.8585
Gaussian Naive Bayes	0.6362	0.6538	0.7222	0.6943

5 Unsupervised Learning Techniques

5.1 Pipeline

The Imagefolder function of Pytorch creates a labeled stack of images which finally gets converted from RGB to Grayscale, then we apply **pca_lowrank()** which is 3 times faster than the PCA() of *sklearn* as it works on tensors, following which, we applied **K-means clustering and Agglomerative clustering** algorithm as follows:-



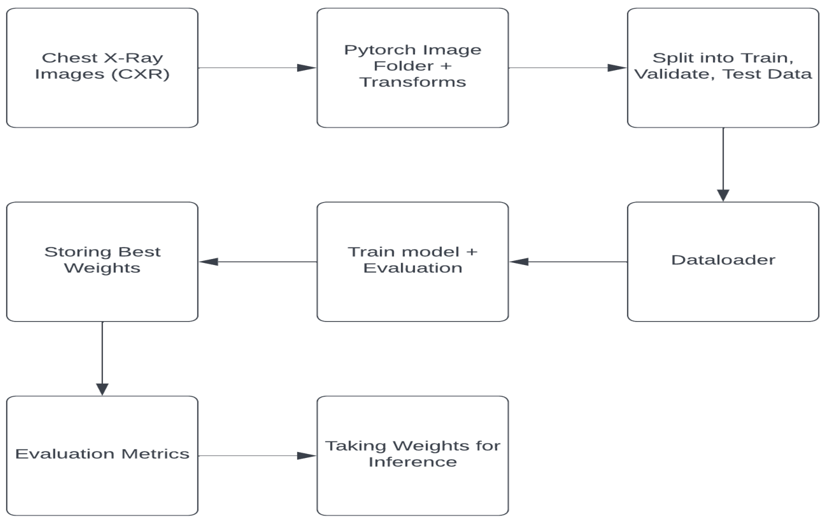
5.2 Evaluation Metrices and Models

Models report				
Model	**F1 score**	**Sensitivity**	**Specificity**	**Accuracy**
K-means	0.5058	0.3904	0.4563	0.4048
Agglomerative	0.3800	0.3593	0.3199	0.3420

6 Deep Learning Techniques

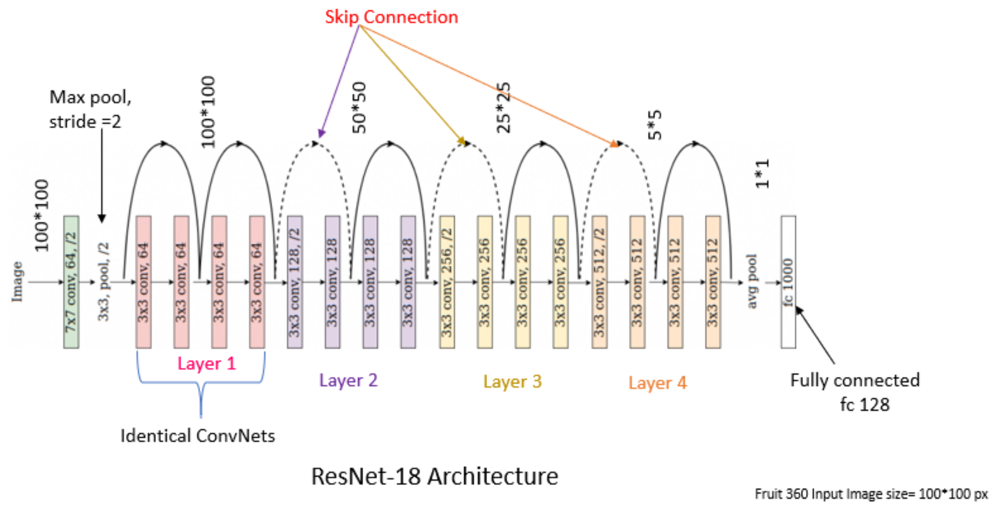
6.1 Pipeline

The images being of unequal sizes were transformed into a common size of 224 x 224 and a function get_mean_and_std() was made to calculate the mean and standard deviation of the images brought together in a batch for the purpose of normalization of the pixels.After this transformations of Resizing, RandomHorizontalFlip, RandomRotations, Conversion to tensors and normalization of the pixels were applied on the dataset.



6.2 Models

- ResNet 18
  - ResNet18 is a 72-layer architecture with 18 deep layers. The architecture of this network aimed at enabling large amounts of convolutional layers to function efficiently.
  - However, the addition of multiple deep layers to a network often results in a degradation of the output. This is known as the problem of *vanishing gradient* where neural networks, while getting trained through back propagation, rely on the gradient descent, descending the loss function to find the minimizing weights.
  - The primary idea of ResNet is the use of jumping connections that are mostly referred to as **shortcut connections** or identity connections.
  - The aim of introducing these shortcut connections was to resolve the predominant issue of *vanishing gradient* faced by deep networks. These shortcut connections remove the vanishing gradient issue by again using the activations of the previous layer.



- Mobilenet v3
  - MobileNetV3 is a CNN that uses a combination of hardware-aware network architecture search (NAS) complemented by the NetAdapt algorithm, and then subsequently improved through novel architecture advances. Advances include (1) complementary search techniques, (2) new efficient versions of nonlinearities practical for the mobile setting, (3) new efficient network design.
  - Another novel idea of MobileNetV3 is the incorporation of an squeeze-and-excitation block into the core architecture. The core idea of the squeeze-and-excitation blocks is to improve the quality of representations produced by a network by explicitly modelling the inter-dependencies between the channels of its convolutional features.

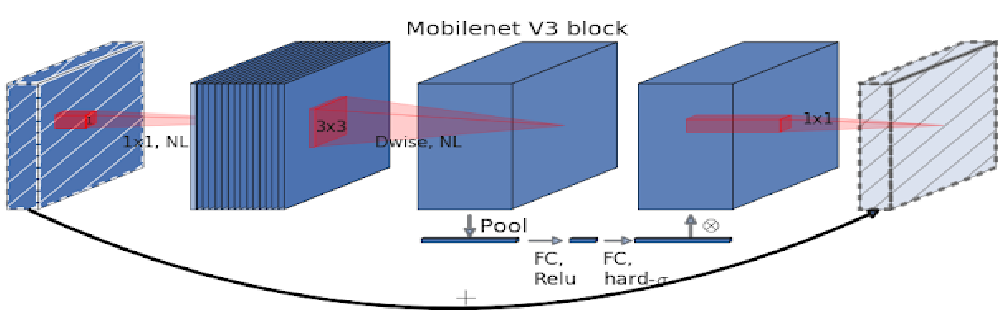


Figure 1: MobileNet-v3 Architecture

- ResNet 50
    - The ResNet-50 model consists of 5 stages each with a convolution and Identity block. Each convolution block has 3 convolution layers and each identity block also has 3 convolution layers. The ResNet-50 has over 23 million trainable parameters.
- VGG-16
    - The 16 in VGG16 refers to 16 layers that have weights. In VGG16 there are thirteen convolutional layers, five Max Pooling layers, and three Dense layers which sum up to 21 layers but it has only sixteen weight layers i.e., learnable parameters layer.
    - Most unique thing about VGG16 is that instead of having a large number of hyper-parameters they focused on having convolution layers of 3x3 filter with stride 1 and always used the same padding and maxpool layer of 2x2 filter of stride 2.
    - This makes the decision functions more discriminative, it would impart the ability to the network to converge faster and it also reduces the number of weight parameters in the model significantly.

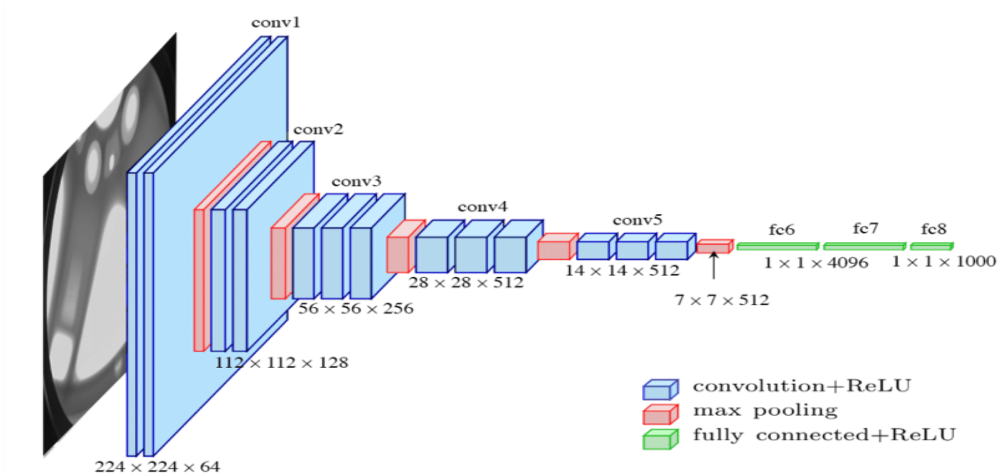


Figure 2: VGG-16 Architecture

- VGG-19
    - VGG19 is a variant of VGG model which in short consists of 19 layers (16 convolution layers, 3 Fully connected layer, 5 MaxPool layers and 1 SoftMax layer).

- Scratch VGG
  - We made a model from scratch inspired from the VGG architecture in which we have used 5 convolution layers along with 3 linear classification layers, so it's like a VGG-8 model
  - We optimized SGD for this model by use of Logits and CrossEntropy loss function. Logits structure helps in attaining a stable minima for the model.
  - Along with that we also implemented logging of minibatch loss at every epoch
  - Parameters of MaxPool and Convolution layers are same as in VGG-16.

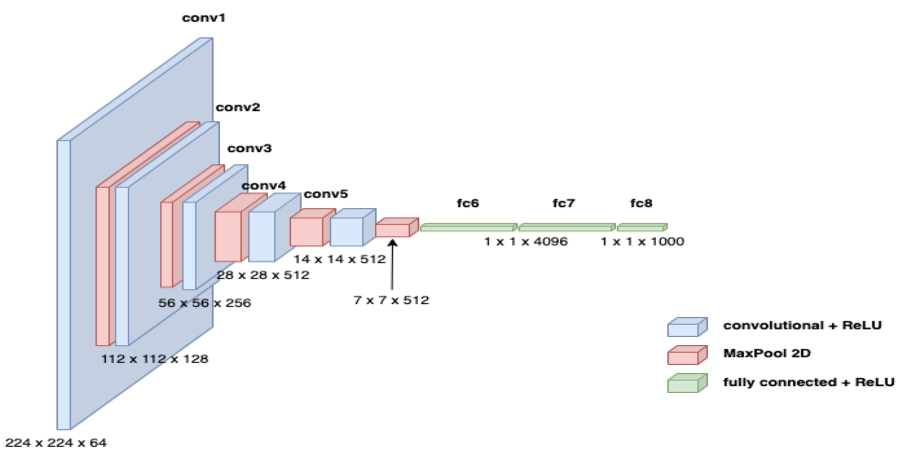


Figure 3: Self Made Architecture of our scratch VGG model

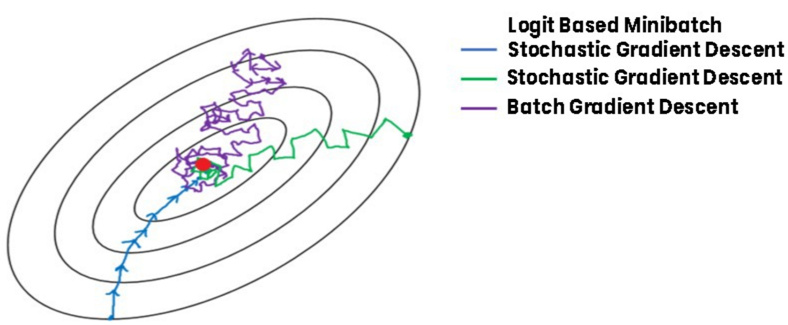
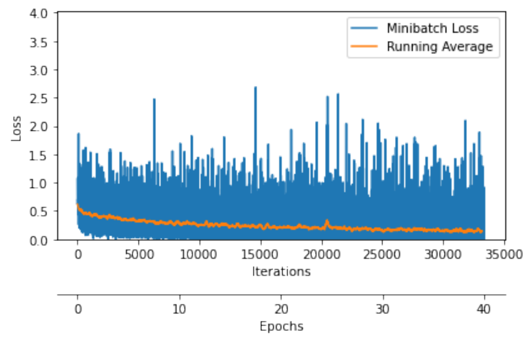
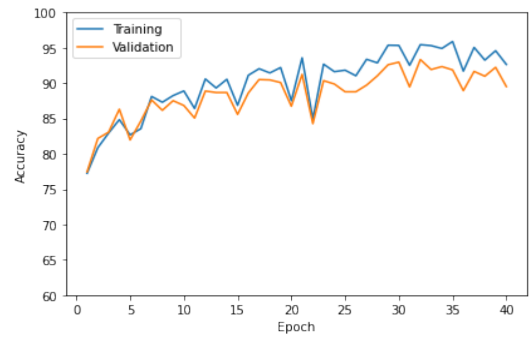
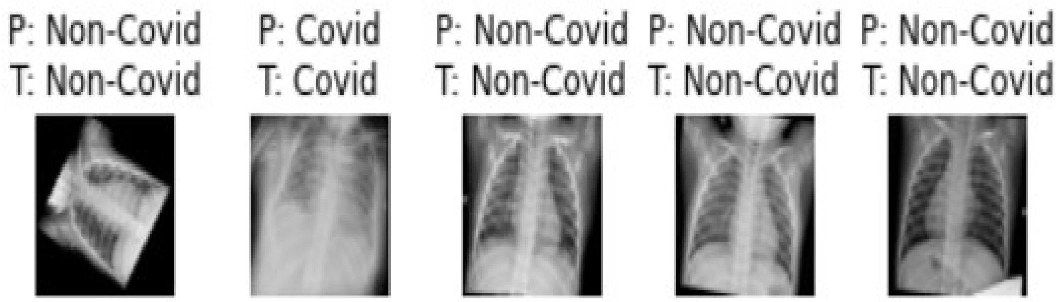


Figure 4: Imporved SGD(Logit based Mini-batch SGD







6.3 Comparison of Models

We used Various Pre-trained models, which we, fine tuned on the last linear classification layers and along with that made 1 model from scratch based on VGG Architecture, for comparing their performances on various evaluation metrics, the statistics for the same are given below in the table

Models report				
Model	**F1 score**	**Sensitivity**	**Specificity**	**Accuracy**
ResNet-18	0.7294	0.6769	0.8569	0.8133
ResNet-50	0.7491	0.8140	0.7553	0.7864
MobileNet-V3	0.8380	0.8370	0.8787	0.8607
VGG-16	0.7032	0.0.5580	0.9778	0.7958
VGG-19	0.8363	0.8861	0.8357	0.8565
Scratch VGG	0.8841	0.8239	0.9603	0.9334

Our scratch model has achieved the best accuracy of **93.34%** compared to others

6.4 CXR images Segmentation

Image segmentation is a method in which a digital image is broken down into various subgroups called Image segments which helps in reducing the complexity of the image to make further processing or analysis of the image simpler. Segmentation in easy words is assigning labels to pixels.
This concept is used when we want to particularly focus on checking the presence or absence of a specific area of interest (AOI) .

6.4.1 U-Net Model

Its architecture can be broadly thought of as an encoder network followed by a decoder network.Unlike classification where the end result of the the deep network is the only important thing, semantic segmentation not only requires discrimination at pixel level but also a mechanism to project the discriminative features learnt at different stages of the encoder onto the pixel space.

- The encoder is the first half in the architecture diagram . It usually is a pre-trained classification network like VGG/ResNet where you apply convolution blocks followed by a maxpool downsampling to encode the input image into feature representations at multiple different levels.
- The decoder is the second half of the architecture. The goal is to semantically project the discriminative features (lower resolution) learnt by the encoder onto the pixel space (higher resolution) to get a dense classification. The decoder consists of upsampling and concatenation followed by

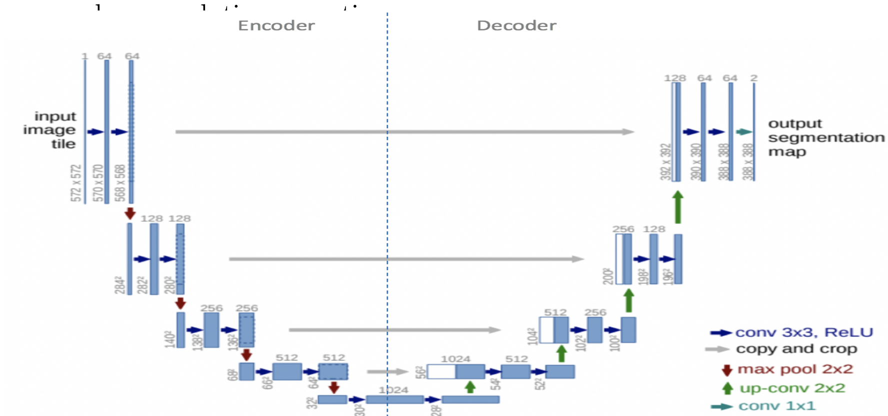
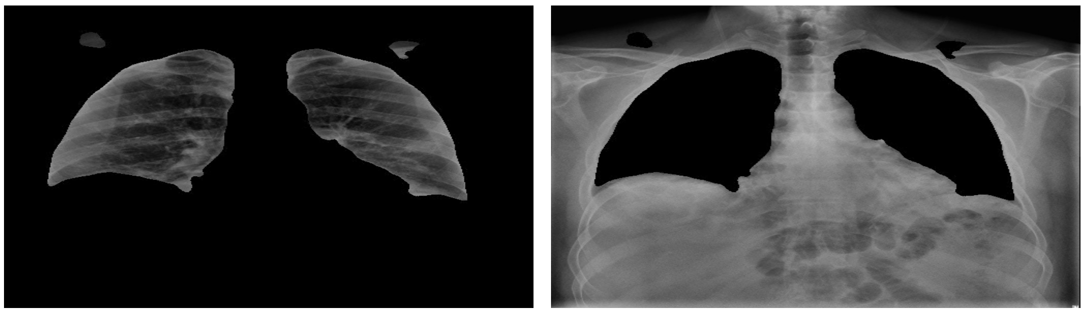


Figure 5: U-Net architecture, Blue boxes represent multi-channel feature maps, while while boxes represent copied feature maps. The arrows of different colors represent different operations

The model returns the following 4 things :- *1) Mask, 2) inverse (complimentary) Mask, 3) Masked image (Mask multiplied with original CXR image), 4) Inverse Masked image (Inverse Mask multiplied with original CXR image)*



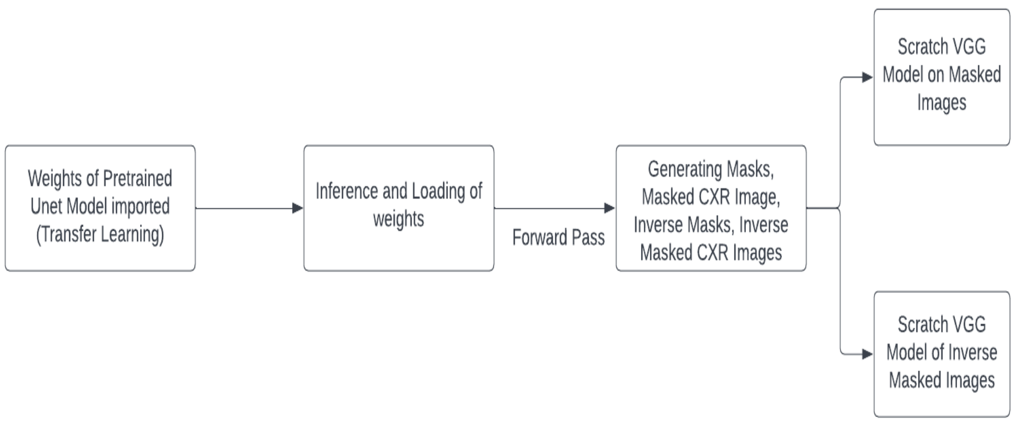


Figure 6: Pipeline for Applying Scratch VGG on Segmented images

Models report				
Model	**F1 score**	**Sensitivity**	**Specificity**	**Accuracy**
Scratch VGG	0.8897	0.8803	0.9195	0.9129
UNet+VGG (Masked Image)	0.8114	0.8177	0.8652	0.8728
UNet+VGG (Inverse Masked Image)	0.8897	0.8803	0.9195	0.9129

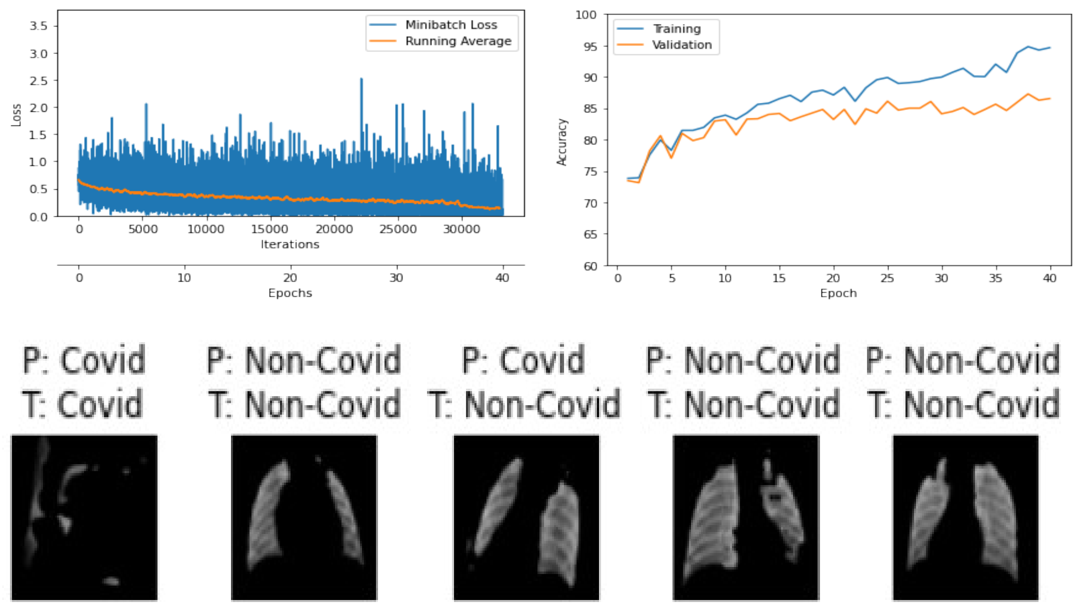


Figure 7: UNET+VGG on masked images

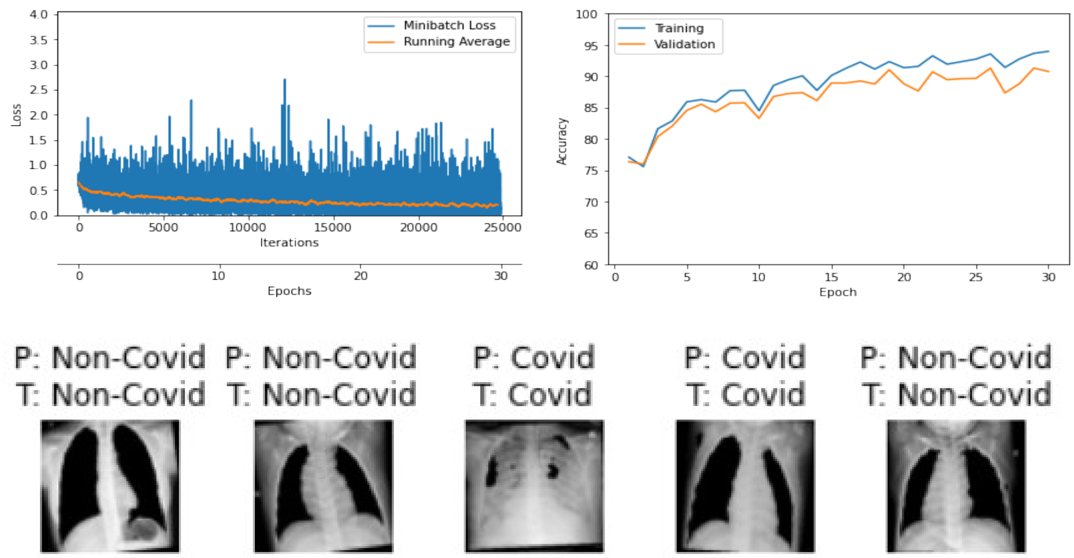


Figure 8: UNET+VGG on Inverse masked images

7 Combined Multi-Modular Approach

- We used our 3 models :- 1) Scratch VGG (trained on CXR images), 2) Scratch VGG (trained on segmented masked images) and 3) Scratch VGG (trained on Inverse masked images) as feature extractors and concatenated the features of all and classified through a series of 3 Dense classification layers as shown in the pipeline.
- we can train this model fully or just train the Dense classification layers by freezing the 3 feature extracting models, this would give the best results due to a good balance of focus on chest features, Non-chest features, Overall Image features.

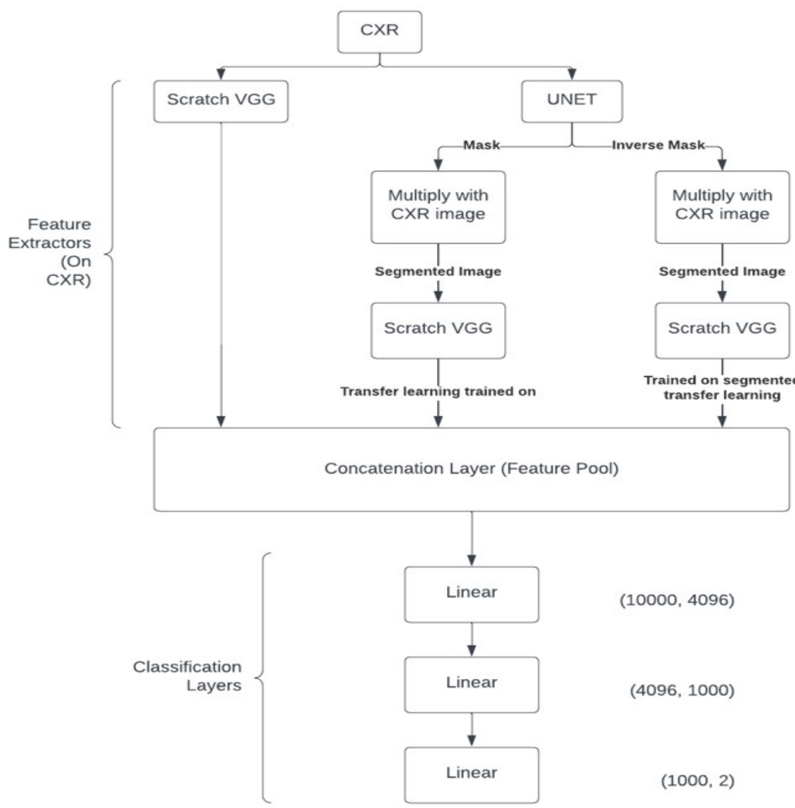


Figure 9: Combined Multi-Modular Approach

8 Deployment

We formed a website using streamlit and python, in which we loaded our best model's architecture and weight, for making the prediction on CXR images received from user as input. We finally returned the prediction along with the true labels and images back to the user.
Following is the interface of website :-

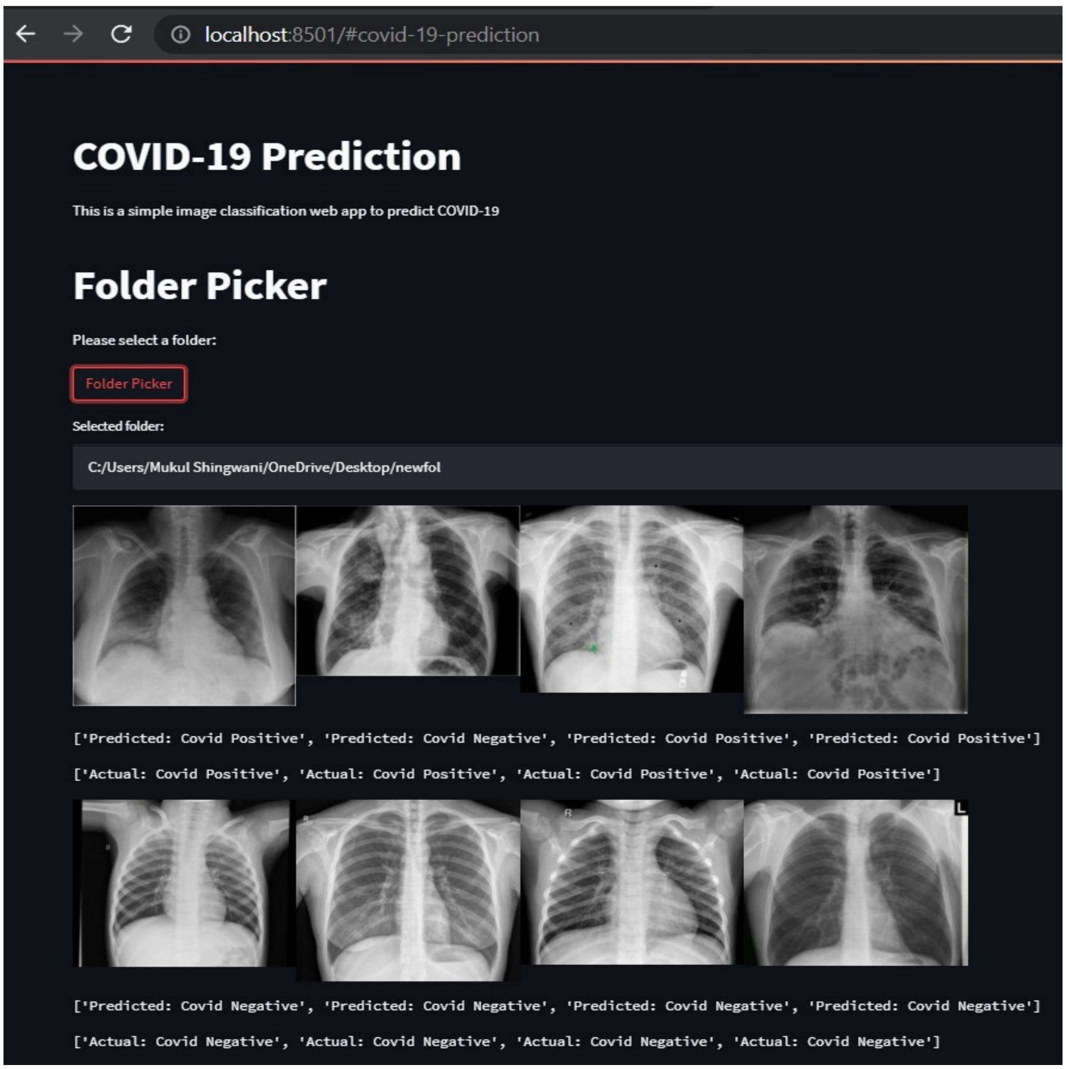


Figure 10: Interface of Website

9 Contributions

We have done most of the work collectively, since there were multiple complex things which required brainstorming and team efforts.

- **Mukul Shingwani** :- Implementation and pre-processing of supervised technique model (Random Forest), ResNet-18, ResNet-50, VGG-19, Latex Report and Website
- **Saurabh Modi** :- Implementation and pre-processing of unsupervised techniques, Scratch VGG, VGG-16 and ensemble on U-Net models.
- **Mitarth Arora** :- Implementation of supervised technique model (Decision tree and GNB), MobileNet-v3 and Latex Report

Github Link : View Project

10 References

- [1] Dataset Link for chest X-ray images
- [2] COVID-19 detection science direct link
- [3] U-net dataset link for segmentation
- [4] paper on collaborative learning of Deep Neural Networks
- [5] playlist on Deep Learning and RNN's
- [6] Improved Logits Minibatch Stochastic Gradient Descent