Exploring Deep Learning and Detecting COVID-19 using Radiographs on VGG16 architecture

Shrey Bhonsle
School of Computer Science and Technology,
Dr. Vishwanth Karad MIT-WPU

Krishna Wankhede
School of Computer Science and Technology,
Dr. Vishwanth Karad MIT-WPU

Sanmay Ramteke
School of Computer Science and Technology,
Dr. Vishwanth Karad MIT-WPU

Aman Shaik
School of Computer Science and Technology,
Dr. Vishwanth Karad MIT-WPU

Under the Mentorship of
Prof. Shakti Kinger
School of Computer Science and Technology,
Dr. Vishwanth Karad MIT-WPU

Disha Gidwani
School of Computer Science and Technology,
Dr. Vishwanth Karad MIT-WPU

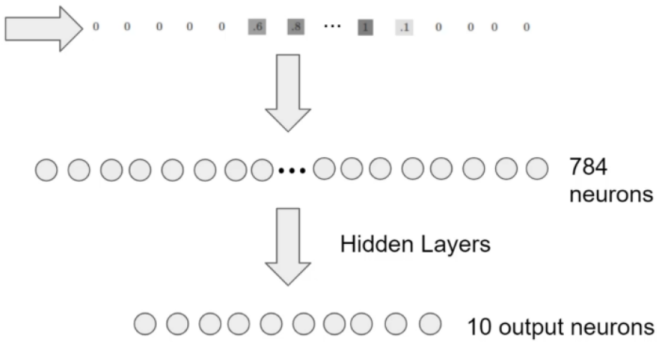
Abstract: COVID-19 pandemic has caused a devastating effect on both daily lives, public health, and the global economy. As per World Health Organization, currently there are more than 400000000 estimated confirmed cases and more than 5000000 confirmed deaths all over. In this study of CNN, we start from basics for Deep learning,understanding ANN by MNIST dataset, and it's drawbacks. After which we discuss the previous research paper's model and their architecture used. In the end, we discussed what are the possible future works and how it can be achived . The model that we trained, aims for reducing efforts of radiologists to detect COVID-19. The model is developed using a custom Convolutional Neural Network (CNN) that differentiates between COVID-19 and healthy x-ray images so that the patient can be diagnosed and quarantined on time to prevent the spread of the pandemic. Here, we are using VGG16 architecture for training our dataset. While training the model we focussed on increasing our precesion as high as possible.

Keywords: COVIDD-19, CNN, VGG16, DEEP LEARNING, CHEST RADIOGRAPHS

1. INTRODUCTION

So first we start withArtificial Neural Networks (ANN). So ANN has multiple neurons at each layersm its the most basic and simple neural networks. Here, we will understant it through a MNIST dataset, which is the classic dataset for deep learning. It consists of handwritten single digits from 0-9.
In MNIST dataset, a single image of a digit can be reperesented as array. The size of each image in MNIST dataset is 28 * 28. The first task is to flatten up the array into single array and feed it to our first layer. Also, a point to be noted here is that ANN is also reffered as Feed-Forward Neural network because inputs are processed only in the forward direction.

After flattening up, we will feed the array to first layer that consist of 784 neurons. After the first layer, we will have next layers with say 120 neurons, followed by 84 and finally the output of 10 neurons.



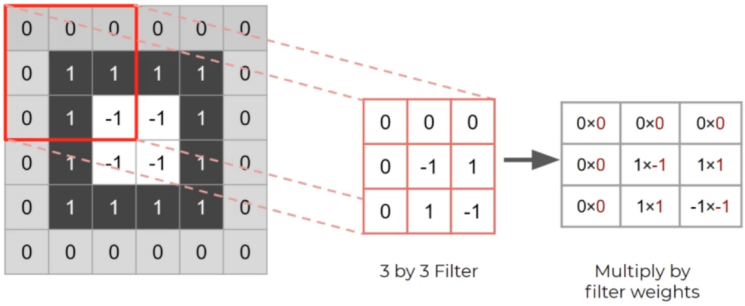
So, how should we activate the neuron for next layer? For that we use activation functions like sigmoid, relu, etc. So what this function does is, they activate neuron on the basis of their learnings and weights. After the activation, the neurons are feed to next layer to make predictions.
Now, when we discuss of total number of parameters, parameteres are used for training of a model, they are essentially number of connections from one layer to the other and so on. So, in here parameters would be:

724*120 = 94080
120*80 = 10080
84*10 = 840
Total params : more than 1L

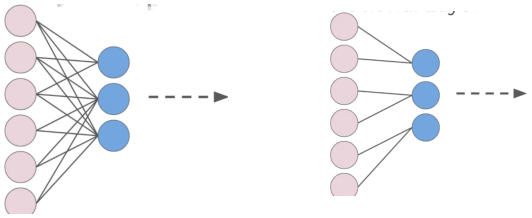
Next, from pytorch library we can use neural network functionality to form a layers, describing it's input, output neurons and its corresponding activation function. Furthur we can split data to test and train and perform model's evaluation after training is completed. Here, people have got around 95 or even greater accuracy using ANN.

If ANN is simple, why ain't we using that in our model:
The numbers of trainable parameters that we got was over 1 Lakh for just 28 * 28 image
We lost all 2D information by flattening our image.
Mostly will work only if images are centered or placed at similiar position.

To understand the concept of CNN we start with:
Filters: So filters are basically image kernals which is a small matrix that is applied to whole image.



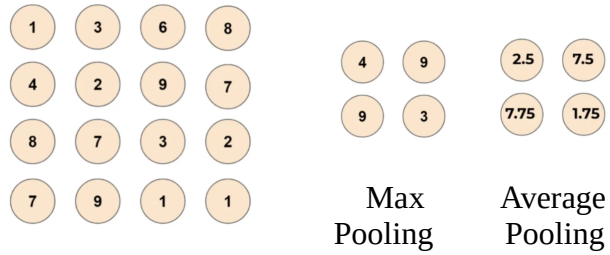
So, here we have taken grayscale image, and we apply 3*3 filter and we get reslutant image on the basis of filter weights.
In the context of CNN, these filters are known as convolution filters and passing them over a image is called **Convolution.**
In CNN, convolution layer is created when we apply multiple image filters to the input image. Then we train image to the best filter value. Unless ANN, in CNN we use local connectivity, i.e., set of neurons in one layers are connected to only subsets of neurons in next layer. Let's compare 1-D CNN and ANN.



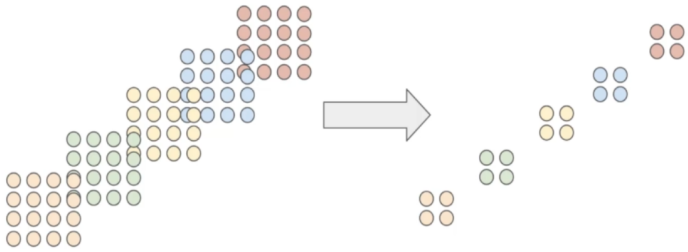
1-D ANN 1-D CNN

One convolution layer that we have, has one filter now, but then, it can have multiple filters, then its the job of network to figure out what are the best weights to this filter. Hence, in this way we conclude with comparision of Artificial Neural Network and now move on to 2-D Convolution Neural Network.
In simple terms a convolution layer is stack of filters used for convolution Network to figure out the best weights. Here we will be dealing with 2-D image which are grayscale, hence we will be using 1-D tensor or convolution. But as we go upto coloured image, we have 3D tensors/convolutions each dimension for one colour channel. Now, coming back to convolution layers, they are often fed into another convolution layer. This forms patterns within patterns for later convolution layers

Pooling Layers: Pooling layers are basically used to reduce parameters like, even after using CNN, we are reducing parameters, but convolution layers would help us futhur reduce parameters as they still remain in huge numbers. Pooling layers use various downsampling techniques. Taking the example of max pooling and and average pooling, we have a filter and we apply max pool to that filter with stride =2.



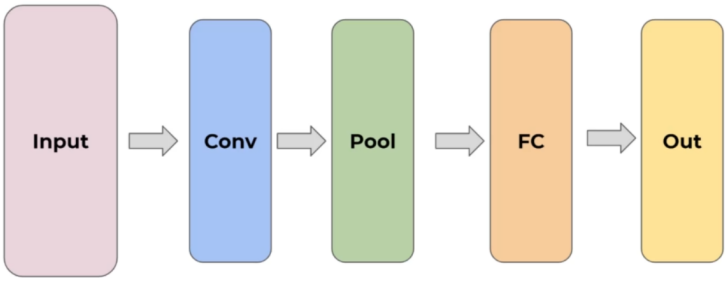
In general,



Hence, we found that pooling layer reduces number of parameters greatly.

Dropout Layers: It is another sort of layer that will help in decreasing training time. It turns OFF the units randomly along with their furthur connections.

Based on these concepts, CNN can have multiple type of architectures, but to make it in nutshell in CNN we will have Input layer, followed by convolution layers, dropout layes, poolings, etc. And in the end we will have a funtion that connects to results and has output layer that has same number of neuron as that of out classes.

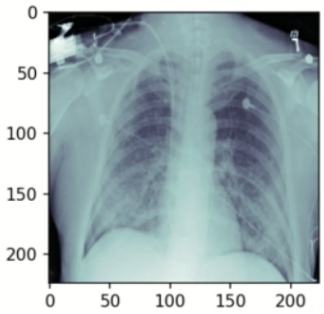


Related works:

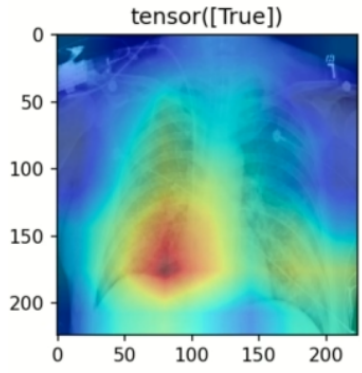
So, the previous implementations include using various architectures and datasets, but most of the dataset were from kaggle dataset, and few of them were from GitHub repos. The accuracy amonst them remained near 92-97 % but they had less precesion for COVID. The links for papers are mentioned in reference of this paper. Amongs the paper we also refered to build a model using NIFTI datasets, we tried to implement the UNET architecture but did not succed. But as mentioned in Experiments area of these report, we have proposed an idea to deal with it.

In one of the papers we saw them using GradCam to detect where is the model activated and on what basis their model was saying that person has covid or not. We really liked their approach.

Like the image down here is COVID +



But there was an assertion made on the statement using GradCam :

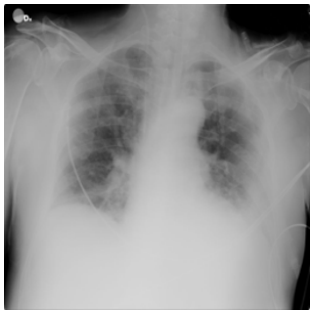


As approved by by one of the radiologist, this is almost hundred percent prediction. Though their model precesion was not that accurate as others had but they had this one special feature.

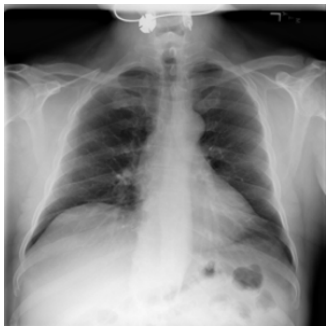
DATASETS AND FEATURE

So currently the dataset that we have is from kaggle.com, wherein the data consisted of around 3.6k COVID infected Chest xray images, and there were around 11k healthy Chest xrays, apart from that the dataset consists of other categories like lung opacity and viral pneumonia. We would have liked including that classes too, but due to hardware restrictions we were not able to do so. For furthur research if anyone wants to go for more classes, just make few changes in our code and model like output classes , i.e., from 2 to n number of classes that we want.

Since, the data was imbalanced, we **Undersampled class,** that is, we reduced the data from rest of the classes and made it relative to COVID-19 class. For future application of these method we used private data from *UNIVERSITY OF RUSSIA* which are in **NIFTI** format, which is used for neuroimagening. The current data we have is in normal PNG format. Few of which are shown below:

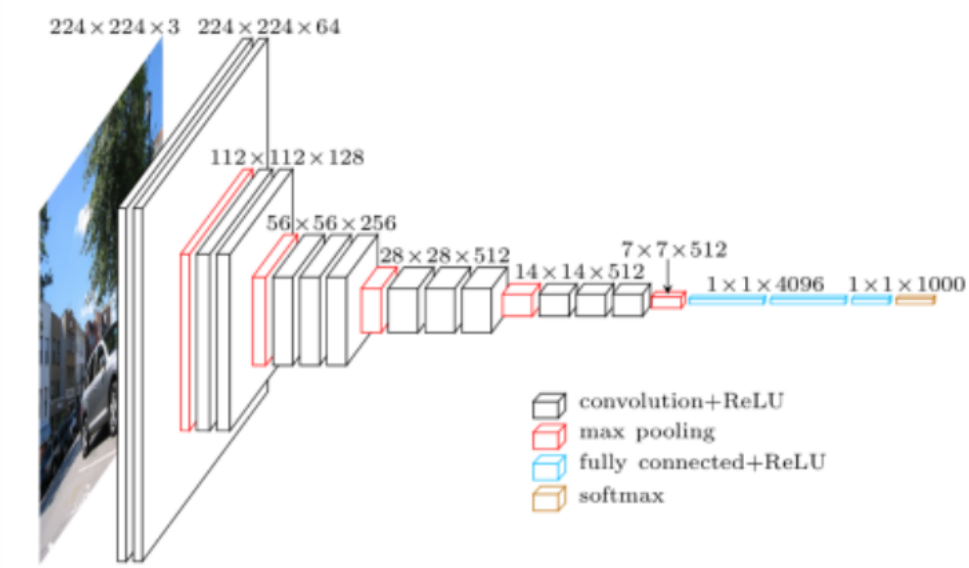


COVID +VE



NORMAL – HEALTHY

VIRAL PNEUMONIA



VGG16 ARCHITECTURE

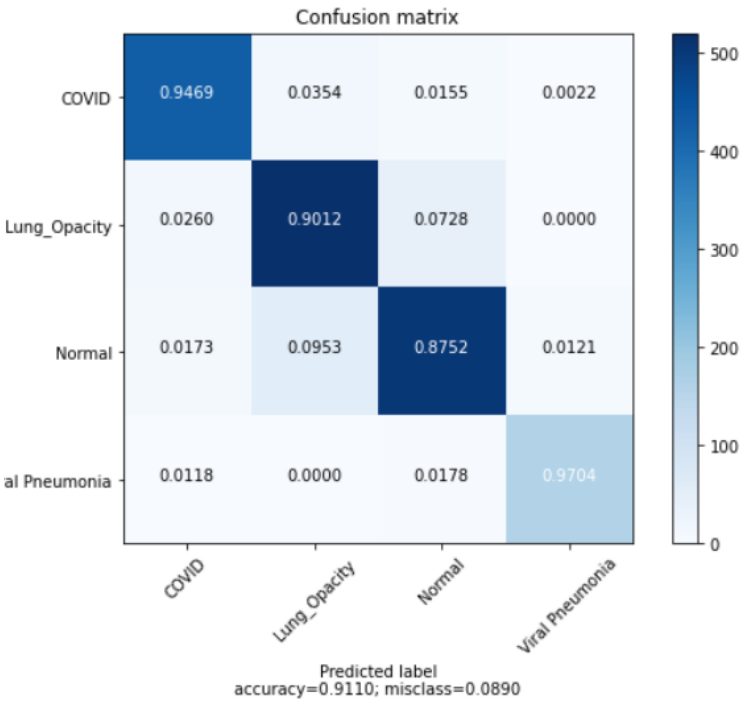
Architecture, Methods/Learning Algorithms

As Discussed earlier about CNN, and layers, several architectures were build by researchers like: LeNet-5, AlexNet (2012),
ZFNet(2013), GoogLeNet/Inception, VGGNet, ResNet(2015) and many more. This architecture were well developed and most succedded architectures.
In our project, we are going to use VGG16 architecture.
As in Diagram, its displayed that how are the layers arranged and all, the coincidence is we also have the image of same resolutions so, this diagram represents the perfect model of our project. Now, lets get into detail of our model building, as discussed earlier, we had unsampled classes as they were unbalanced. Furthur we split the data into test and train, for model training. Then as mentioned earlier, we need a connection Function for our model that corresponds to the number classes that we want our model to predict. Hence we append our model with, serveral dense and dropout layers at the end and final layer that corresponds to 4 neurons that are as per our numbers of class to be classified.
The number of epochs in training are kept to be 50, batch size, you can keep as per your hardware capablity. Currently, the learning rate is kept as:

LEARNING_RATE = 1e-5

Again we can vary it and make changes accordingly. Also we used Tunning of our VGG16 model, for which credit goes to Mr. Matheus Henrique and his work on VGG16 that made us implement his idea for tunning after training the model.
The technique that he used was to fine tune the model, till we get unfreezed VGG16 layers.

The accuracy we got is less as compared to many other works done by researchers, but the thing that we gave importance here was precesion, here we got about 95% of precesion of covid detection, though it is considered to be less, as we should always try to achive the max efficiency as we could.



Here, to depict our accuracy of model, we have made a confusion matrix, as in matrix we can see, Normal classes are predicted less efficiently.
We have taken a personal review of these project by one of the Radiologists, where he confirms that the model predicts the Healthy X-ray to Covid because of a bit of white puss and more white pixels in Xrays.

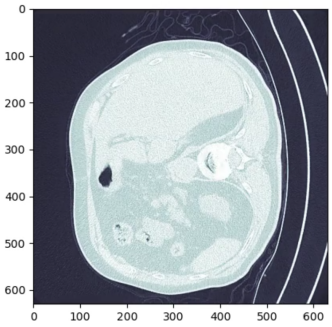
EXPERIMENTS:

So, we tried using unet to segment the image and using it to train our model, but we were facing error in the same. We wanted to build a model such that before training, using UNET we wanted to segment the image and store the vectored image along with its masks. Then we wanted that mask along with the image to the train in VGG16 architecture and aimed our model to learn only from that part of images. The Idea would give out a more precise model as it wont learn from the background. The problem is called Image segmentation with classification, till now, we didn't find any research papers with such implemented model, hence this could be done in future as it would increase it's efficiency. As shown in figure below, we can train the network such that it predicts mask and crops image accordingly. The Credit here is of AbdulWahab Kabani and Mahmoud R. El-Sakka who made the research on this, and made it possible, their aim was: to train the network is trained on images along with masks which indicate where the object is in the image. They treated the problem as a multi-class classification. Therefore, the last layer layer has a softmax activation. Furthermore, during training, the mutli-class log loss is minimized just like any classification task. In the former problem, we used this network to localize the head of the whale while in the later we used it to localize the heart left ventricle from MRI images. Most localization networks regress a bounding box around the region of interest. Unlike these architecture, we treat the problem as a classification problem where each pixel in the image is a separate class.

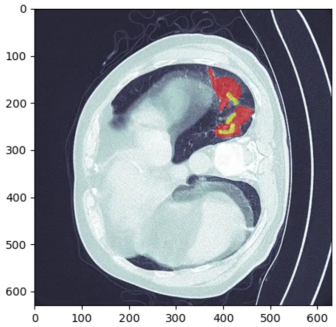
Conclusion

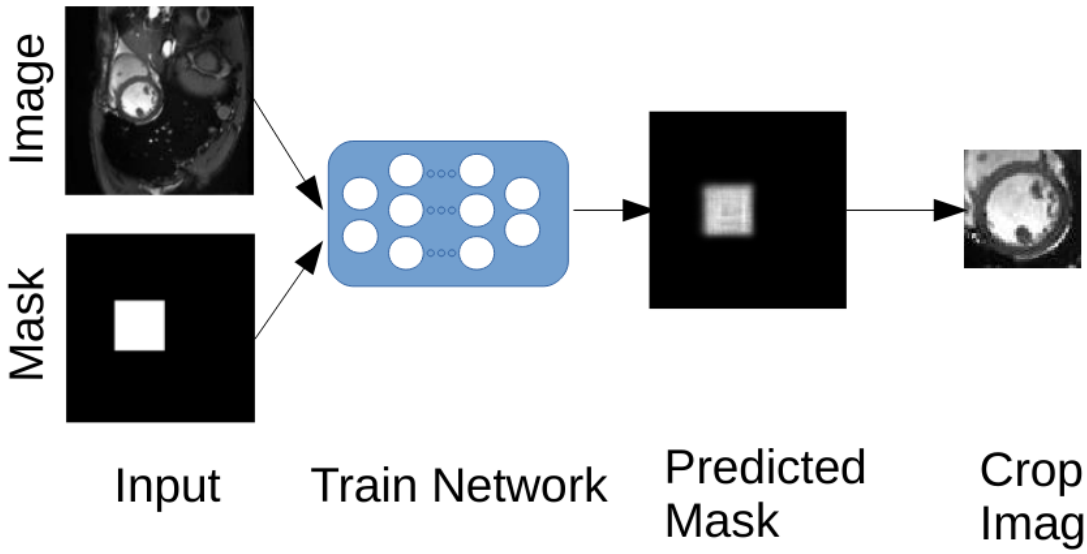
Hence, with bit of modification in VGG16, we were able to train a model with prety good accuracy but not enough for deployment in medical fields. Further studying a bit about medical images dataset, we came in conatact with nifti images and its corresponding masks for COVID-19. Images, I tried using Azure cloud with 56 GB ram and all but the model did not trained, I did some modification on my previous 3-D liver segmentation and tumor segmentation. Here are the few pictures that are relatble:

Frame0:

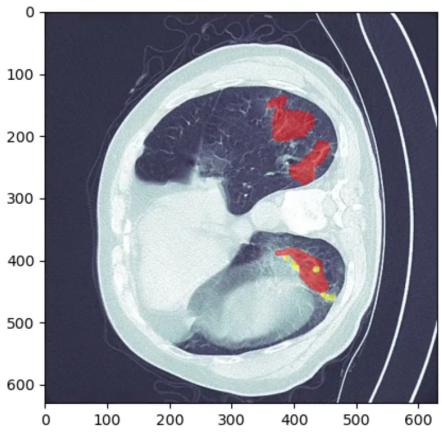


Frame7:

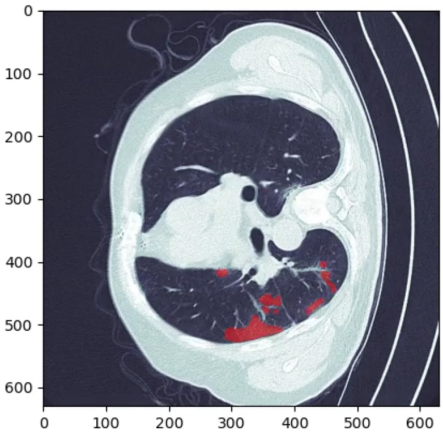




Frame 9:



Frame 14:



References

[1] COVID-19 Detection using Convolutional Neural Network Architectures based upon Chest X-rays Images
[2] COVID-19 Detection Using Deep Learning Algorithm on Chest X-ray Images
[3] Covid-19 Detection using CNN Transfer Learning from X-ray Images
[4] Detection of COVID-19 from Chest X-Ray Images Using Convolutional Neural Networks
[5] Detection Of Pandemic Virus Covid-19 Using CNN