PROJECT-I REPORT
ON
COVID-19 PREDICTION USING CHEST X-ray DATA

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF

BACHELOR OF ENGINEERING
(INFORMATION TECHNOLOGY)

Supervisor
Dr. Neelam Goel
Assistant Professor

Submitted By:
Anand kumar (UE178016)
Vikas kumar patel (UE178111)

To
Department of Information Technology
University Institute of Engineering and Technology
Panjab University, Chandigarh
2017-2021

DECLARATION

We hereby declare that the work which is being presented in this report entitled "COVID-19 PREDICTION USING CHEST X-ray DATA" towards partial fulfillment of the requirement for the award of degree of Bachelor of Engineering in Information Technology, is an authentic record of our work under the supervision of **Dr. Neelam Goel ,**Assistant Professor, Department of Information Technology, UIET, Panjab University, Chandigarh.

The work in this report has not been submitted by us for the award of any other degree or any other institute. We have taken care in all respects to honour the intellectual property right and have acknowledged the contribution of others for using them for this academic purpose.

Date: December 31, 2020

Signature of the Student

Anand Kumar (UE178016)

Vikas kumar patel (UE178111)

Acknowledgement

If words are considered as a symbol of approval and token of appreciation then let the words play the heralding role expressing my gratitude. The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. We would like to express our gratitude towards **Dr. Neelam Goel** for her guidance, constant supervision, constructive suggestions and for her support in completing the project. We would like to express our gratitude towards our parents & members of **University Institute of Engineering & Technology, Panjab University** for their kind cooperation and encouragement which helped us in completion of this project.

We have put in efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

Our thanks and appreciations go to our colleagues in developing the project and people who have willingly helped us out with their abilities.

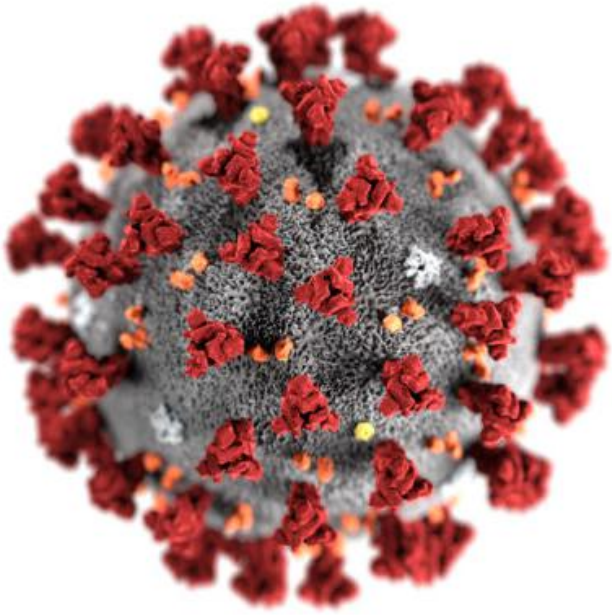
Anand Kumar (UE178016)

Vikas kumar patel (UE178111)

Table of contents

1. Introduction

Since December 2019, a novel corona-virus (SARS-CoV-2) has spread from Wuhan to the whole China, and many other countries. By April 18, more than 2 million confirmed cases, and more than 150,000 deaths were reported in the world (https://www.worldometers.info/coronavirus/). Due to unavailability of therapeutic treatment or vaccine for novel COVID-19 disease, early diagnosis is of real importance to provide the opportunity of immediate isolation of the suspected person and to decrease the chance of infection to healthy population. Reverse transcription polymerase chain reaction (RT-PCR) or gene sequencing for respiratory or blood specimens are introduced as main screening methods for COVID-19 .



However, total positive rate of RT-PCR for throat swab samples is reported to be 30 to 60%, which accordingly yields to un-diagnosed patients, which may contagiously infect a huge population of healthy people. Chest radiography imaging (e.g., X-ray or computed tomography (CT) imaging) as a routine tool for pneumonia diagnosis is easy to perform with fast diagnosis. Chest CT has a high sensitivity for diagnosis of COVID-19 and X-ray images show visual indexes correlated with COVID-19.

The reports of chest imaging demonstrated multilobar involvement and peripheral airspace opacities. During the early course of COVID-19, ground glass pattern is seen in areas that edges the pulmonary vessels and may be difficult to appreciate visually. Asymmetric patchy or diffuse airspace opacities are also reported for COVID-19 Such subtle abnormalities can only be interpreted by expert radiologists.

Considering huge rate of suspected people and limited number of trained radiologists, automatic methods for identification of such subtle abnormalities can assist the diagnosis procedure and increase the rate of early diagnosis with high accuracy. Artificial intelligence (AI)/machine learning solutions are potentially powerful tools for solving such problems.

1.1 **Problem statement**

We need to perform covid-19 prediction using chest x-ray data through deep learning.

1.2 **Objective**

- ✧ To study and understand Deep Learning and its implementation using Python.
- ✧ To study different classification techniques for covid-19 prediction.
- ✧ To collect previous chest x-ray data of the patients.
- ✧ To design the model for our covid-19 prediction problem.
- ✧ Training convolutional network model for COVID-19 detection.
- ✧ Presenting the sensitivity and confusion matrix for each model.
- ✧ To test and validate the model using testing data set.

1.3 Challenges in present scenario

Infection with the virus causing COVID-19 (SARS-CoV-2) is confirmed by the presence of viral RNA detected by molecular testing, usually RT-PCR. Detection of viral RNA does not necessarily mean that a person is infectious and able to transmit the virus to another person.

RT-PCR tests to detect severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) RNA are the operational gold standard for detecting COVID-19 disease in clinical practice.

Technical problems including contamination during sampling (eg, a swab accidentally touches a contaminated glove or surface), contamination by PCR amplicons, contamination of reagents, sample cross-contamination, and cross-reactions with other viruses or genetic material could also be responsible for false-positive results.

Also there are insufficient number of radiologists to tackle this situation. Less developing countries are more shortage of doctors or lab technicians who can perform covid-19 prediction test.

Potential consequences of false-positive COVID-19 swab test results

✧ Health-related

- For swab tests taken for screening purposes before elective procedures or surgeries: unnecessary treatment cancellation or postponement

✧ Financial

- Financial losses related to self-isolation, income losses, and cancelled travel, among other factors

✧ Psychological

- Psychological damage due to misdiagnosis or fear of infecting others, isolation, or stigmatisation

2. Technology used

(A)Hardware requirements

Hardware requirements include that hardware which is required for its working. It includes:

- INTEL i3 PROCESSOR (Minimum)
- 4 GB RAM (Minimum)
- Hard Disk - 10GB (Minimum)

(B) Software requirements-

2.1 PYTHON

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

There are two major Python versions- **Python 2 and Python 3**. Both are quite different.

Features of python-

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

2.2 Machine Learning

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that makes it more similar to humans: The ability to learn. Machine learning is actively being used today, perhaps in many more places than one would expect.

2.3 Data Visualization

Data visualization is an important skill in applied statistics and machine learning. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more.

2.4 Deep Learning

It is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks.

Deep-learning architectures such as deep neural networks, deep belief networks, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, machine vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance.

2.5 Convolution Neural Network

In deep learning, a convolutional neural network is a class of deep neural networks, most commonly applied to analyzing visual imagery. They are also known as shift

invariant or space invariant artificial neural networks, based on their shared-weights architecture and translation invariance characteristics.

3. Software/ System Analysis Design

3.1 Requirement Gathering

(i) Datasets

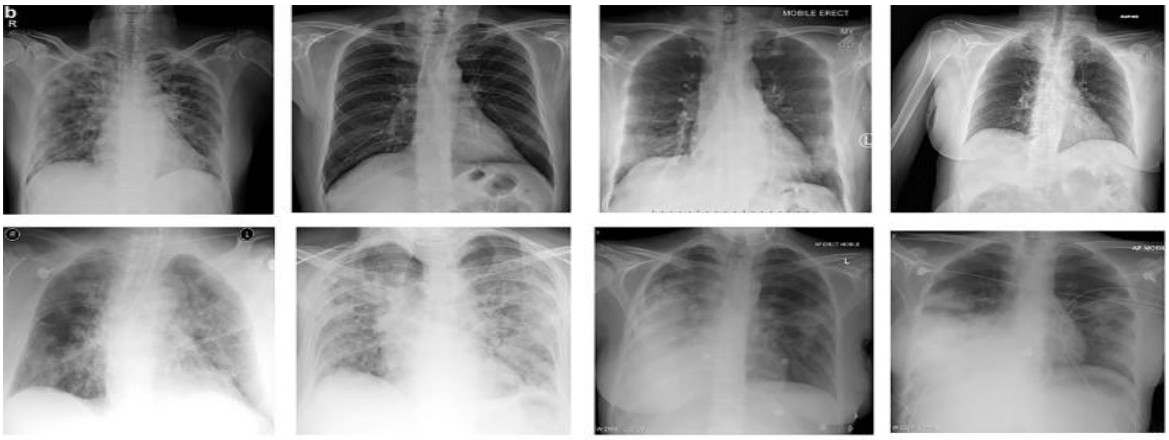
We prepared a dataset of 284 images with binary labels, for COVID-19 detection from Chest X-ray images. Chest X-ray images from two datasets formed the COVID-Xray dataset that contains 224 training and 60 test images.

One of the used datasets is the recently published **Covid-Chestxray-Dataset**, which contains a set of images from publications on COVID-19 topics, collected by https://github.com/ieee8023/covid-chestxray-dataset,. This dataset contains a mix of chest X-ray and CT images. As of May 3, 2020, it contained 250 X-ray images of COVID-19 patients, from which 203 images are anterior-posterior view. It is mentioned that this dataset is continuously updated. It also contains some meta-data about each patients, such as sex and age. Our COVID-19 images are all coming from this dataset.

Number of images per category in COVID-Xray dataset.

Split	COVID-19	Non-COVID
Training Set	112 (420 after augmentation)	112
Test Set	30	30

Below figure shows 16 sample images from COVID-Xray dataset, including 4 COVID-19 images (the first row), 4 normal images



(ii) Execution requirement:

1. A good internet connection

2. Laptop/PC with at-least 4GB of RAM.

3. Jupyter notebook

4. Gpu(google colab)

3.2 Feasibility Study

3.2.1 Technical Feasibility

The project is technically feasible as the technique used, are -

1. Deep learning,

2. python programming

3. TensorFlow + Keras

4. Machine learning libraries- Numpy, pandas, Matplotlib

3.2.2 Financial Feasibility-

This project is financially feasible as we are developing it on open source platform I.e python and data set used in this project is also available publicly. All the libraries are available easily and freely.

3.2.3 Social/Legal Feasibility

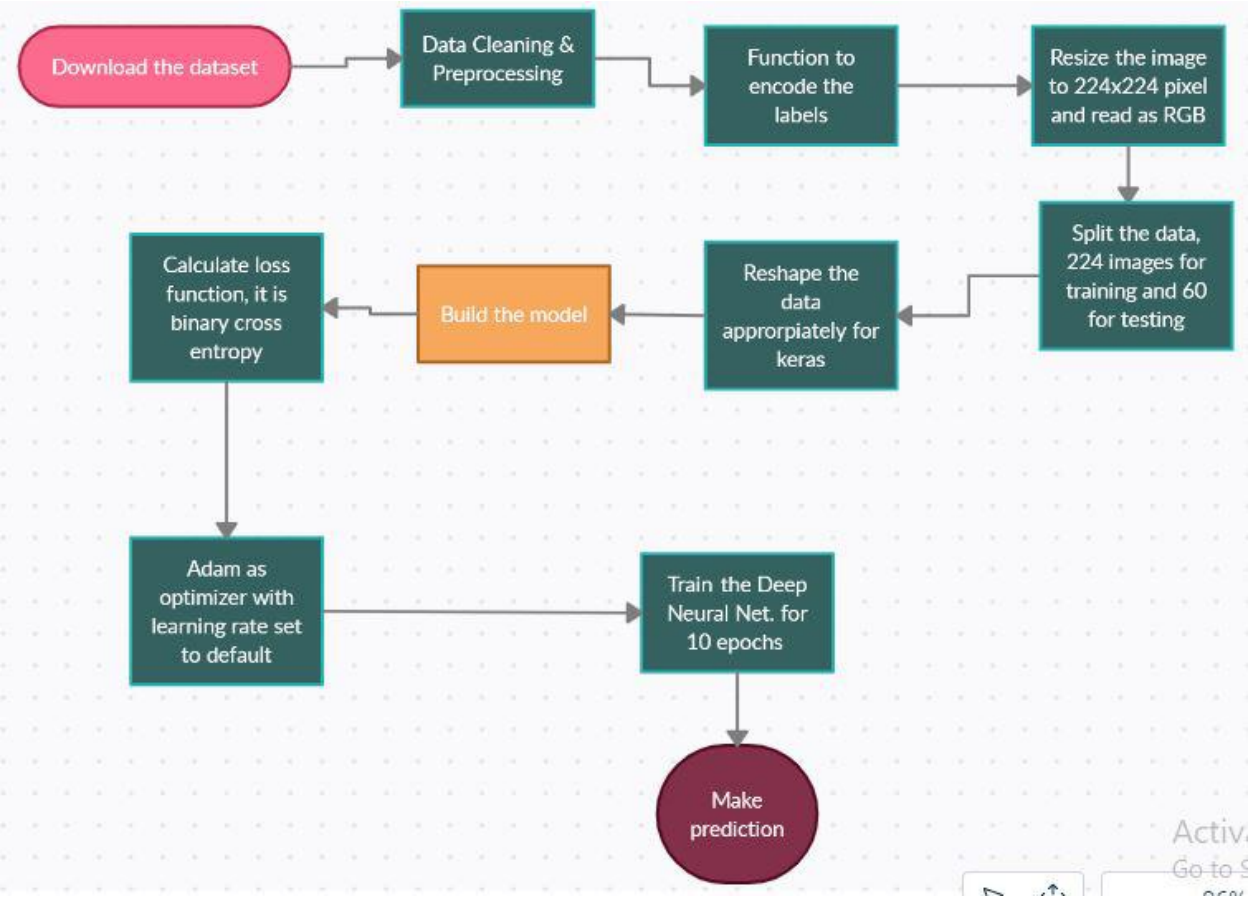
We did not require any copyrighted and trademarked material hence legally feasible.

3.2.4 Operational Feasibility

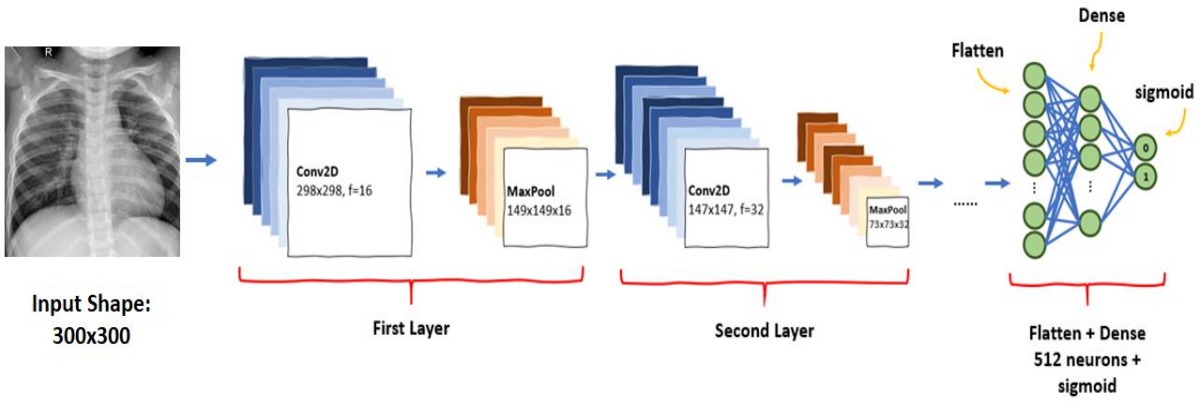
The project has been developed under strict connection with problem statement hence operationally feasible

3.3 Design/use case diagrams/ Flowcharts

The flow chart shows the working methodology for the project work , as a division of the final project into various sub tasks, to give the project a modular structure.



Flow chart for convolution neural network layers:

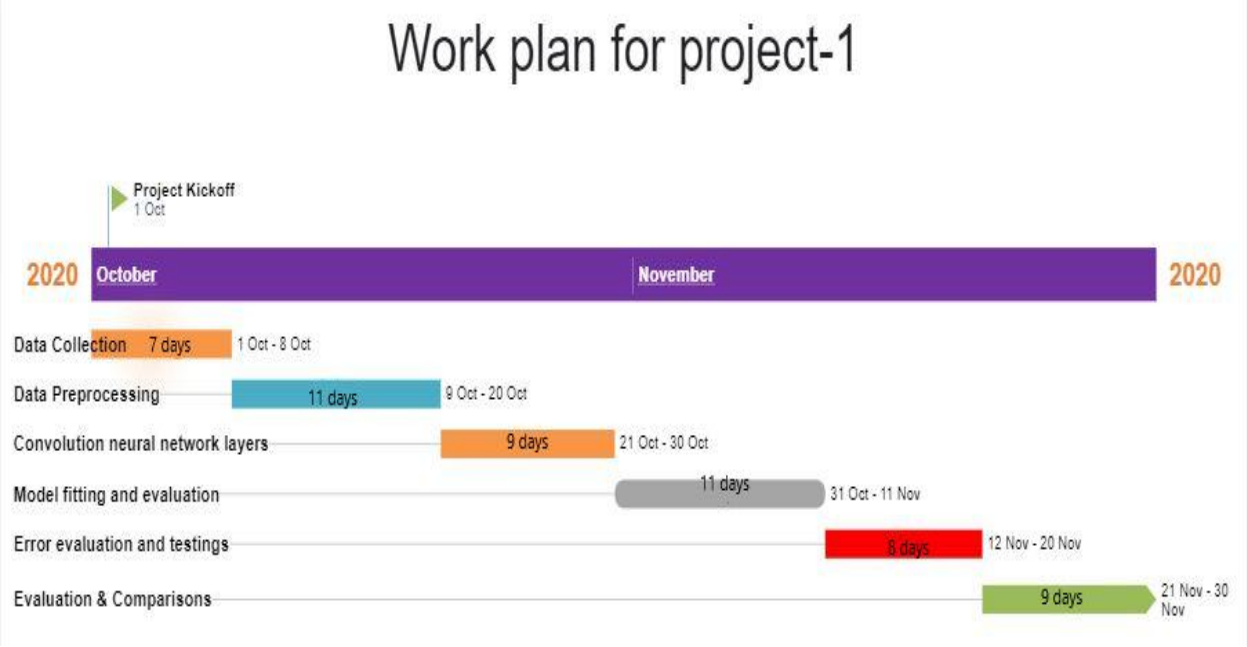


A multi layer convolution network will be built where Conv2D() and MaxPooling2D() are stack together as one layer. Then, the output of the final convolutional layer will be flattened and fit to fully connected neurons.

3.4 Implementation

3.4.1 Work plan

Showing the overall plan for the work division into modules and implementing them on the basis of given schedule time.



3.4.2 Team contribution:

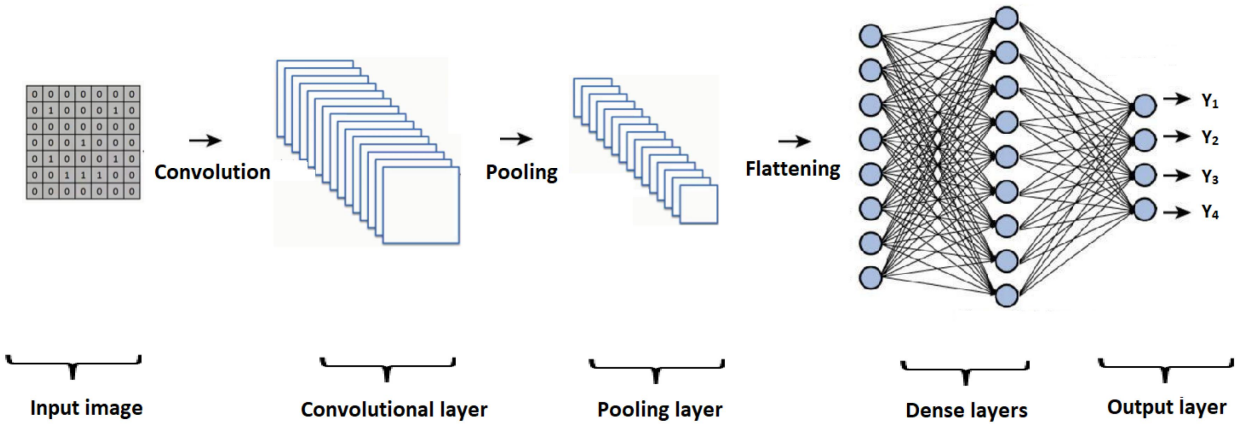
Anand kumar	✧ Project idea and Essentials ✧ Training and Implementation
Vikas kumar patel	✧ Validation and Testing ✧ Documentation

3.4.3 Datasets

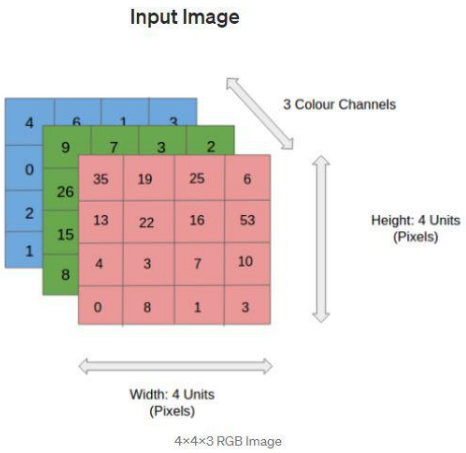
This project leveraging the data from Kaggle repository titled Chest X-Ray Images (Pneumonia).The dataset composes of two classes which are normal lung and pneumonia lung but we selected only normal lung images for our project. Chest X-Ray data for covid-19 positive patients we obtained from a open source github repository.

3.4.4 Building a convolution neural network model

CNN have following layers-



Input image-

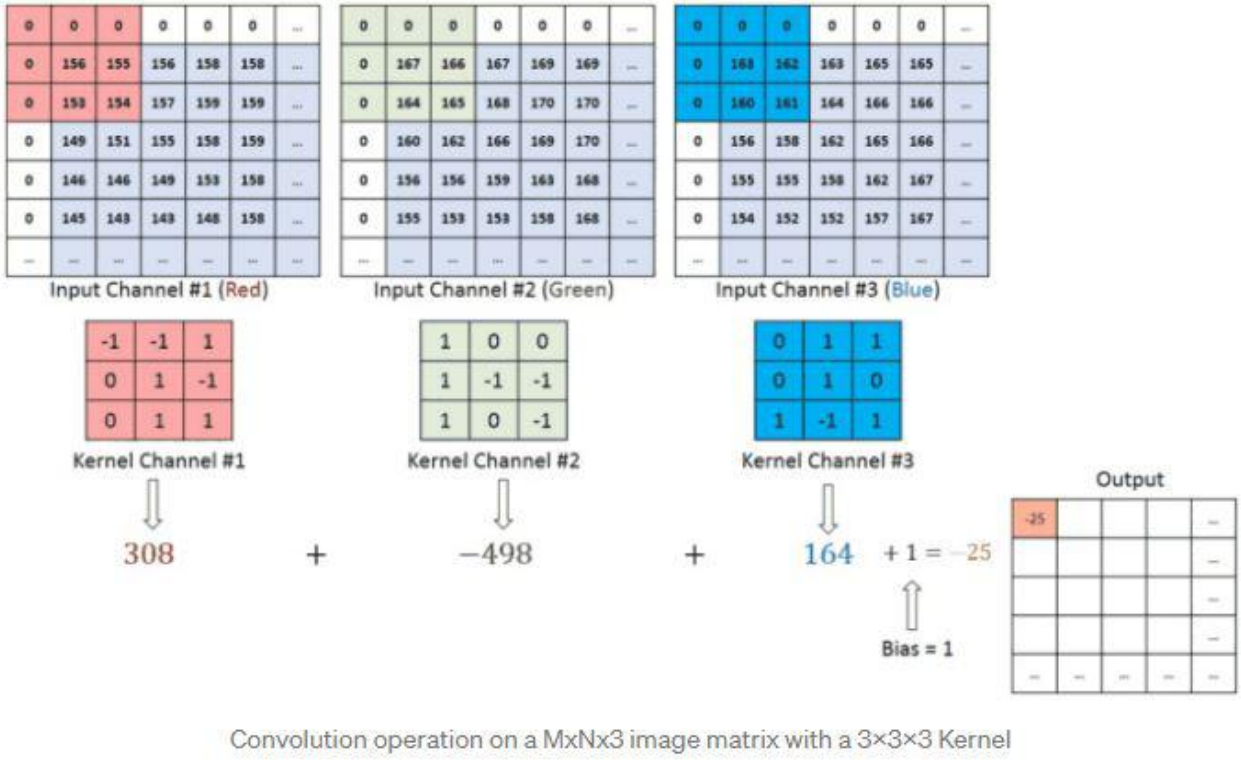


In the figure, we have an RGB image which has been separated by its three color planes — Red, Green, and Blue.

(i) Convolution

The objective of the Convolution Operation is to **extract the high-level features** such as edges, from the input image. ConvNets need not be limited to only one Convolutional Layer. Conventionally, the first ConvLayer is responsible for capturing the Low-Level features such as edges, color, gradient orientation, etc. With added layers, the architecture adapts to the High-Level

features as well, giving us a network which has the wholesome understanding of images in the dataset, similar to how we would.

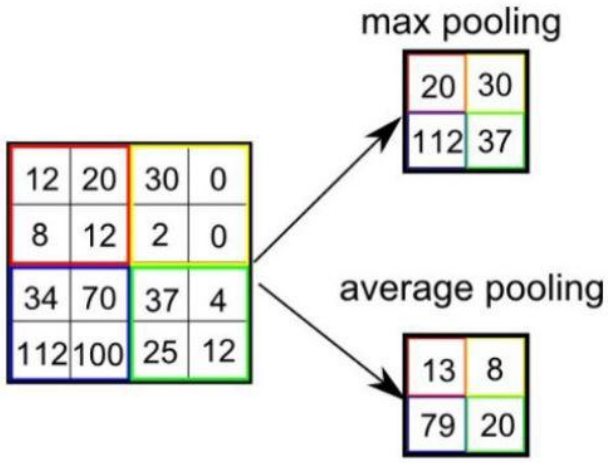


Convolution operation on a MxNx3 image matrix with a 3×3×3 Kernel

(ii)Pooling

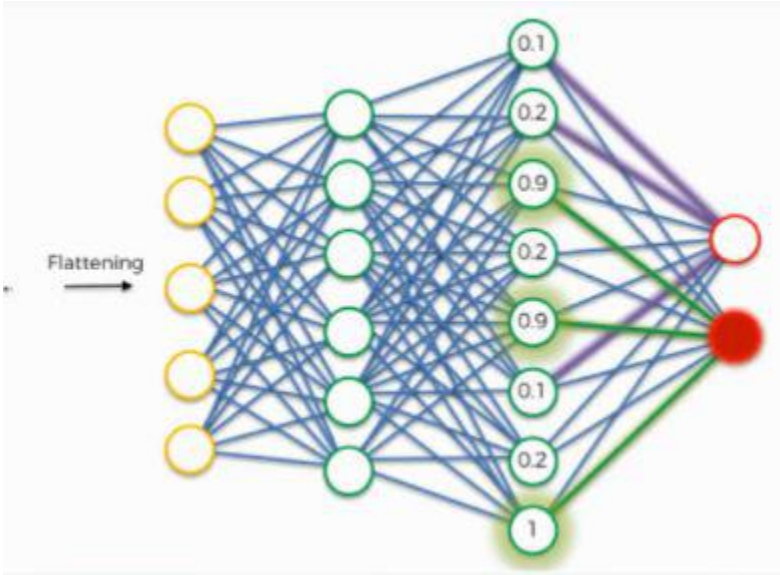
The Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model.

There are two types of Pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel.



(iii)Dense Layer/Fully connected layer

Fully connected layers connect every neuron in one **layer** to every neuron in another **layer**. It is in principle the same as the traditional multi-**layer** perceptron neural network (MLP). The flattened matrix goes through a **fully connected layer** to classify the images.

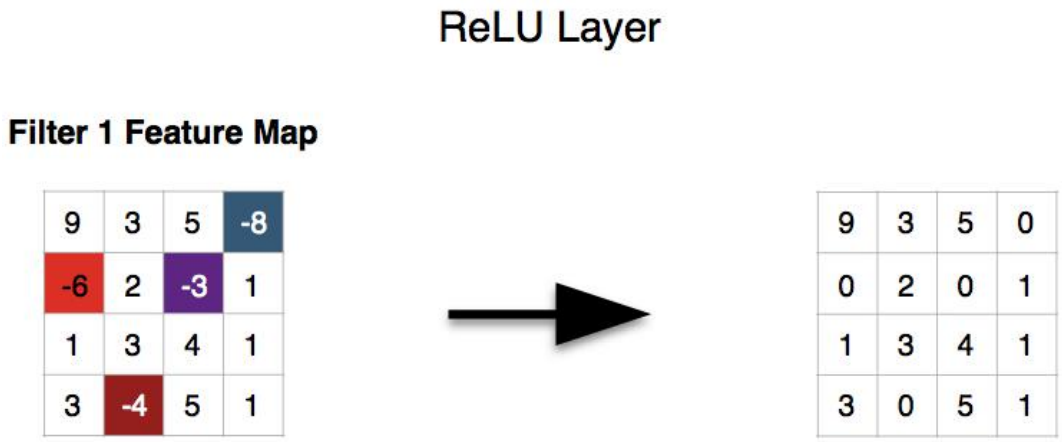


(iv)ReLu Layer

In this layer we remove every negative values from the filtered images and replaces it with zero's.

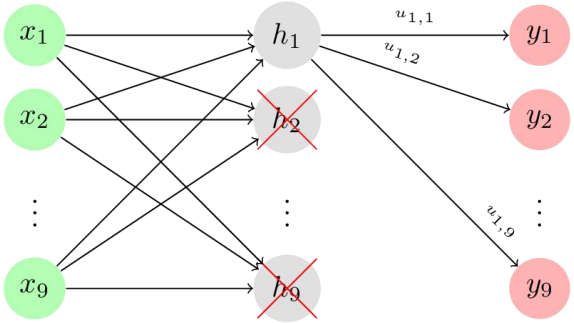
This is done to avoid the values from summing up to zero.

Rectified Linear Unit transform function only activates a node if the input is above a certain quantity, while the input is below zero, the output is zero, but when the input rises above a certain threshold, it has a linear relationship with the dependent variable.



(v)Dropout

Dropout is a technique where randomly selected neurons are ignored during training. They are "dropped-out" randomly. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not applied to the neuron on the backward pass.



3.5 Testing

✧ The error evaluation is done after splitting the complete merged datasets into 2 parts of desired ratios as Training dataset and Test dataset by selecting the random rows of the complete dataset.

✧ Then the models are trained on the training dataset and the evaluation is done on the test dataset.

✧ The optimal ratio used for test train split is generally 70-30% for test-training correspondingly, but because of highly sensitive dataset we try using various split ratios and then evaluate the final test error (this step helps us to get various new insights and scopes of the technique we are using).

✧ The test set predictions are evaluated best in terms of Binary cross entropy loss. Cross-entropy is a better measure than MSE for classification, because the decision boundary in a classification task is large (in comparison with regression).

✧ The binary cross entropy loss is hence reduced by trying various different combinations and ratios of train test splits on the dataset.

✧ We infer the results based on not only the cross entropy loss we obtain, but also through various visualizations for better insights and future scopes for these techniques.

3.6 Maintenance

The model is basically a machine learning based python code notebook for given dataset.

We can keep the model maintained by following some basic steps -

● Because the model have the heavy neural networks,
the neural network architecture has been saved as a .H5 extension file which can be easily loaded later on if the neural network is to be used or modified instead of rerunning the autoencoder training again. So, the .H5 extension model files need to be kept safe with the jupyter notebooks.

● The project uses datasets which are to be kept in the same folder where the jupyter notebook is kept because the paths for loading the datasets used in the project are explicit paths and not the relative paths, so any change in location of the files will disturb the rerunning of the project.

● Any modification if brought to the project then the modification meaning should be mentioned in the comments in the cell of the code. Because the file is large and the meanings can get lost easily for even a very small code snippet.

● The python version used in the library is “PYTHON 3.7.6”, so if any error occurs because of library version this python version must be installed in the virtual environment and then used.

4. Project Snapshots or results/outcome

In [1]:
```
TRAIN_PATH = "CovidDataset/Train"
VAL_PATH = "CovidDataset/Test"
```

In [2]:
```
import numpy as np
import matplotlib.pyplot as plt
import keras
from keras.layers import *
from keras.models import *
from keras.preprocessing import image
```

```
Using TensorFlow backend.
```

In [3]:
```
# CNN Based Model in Keras

model = Sequential()
model.add(Conv2D(32,kernel_size=(3,3),activation='relu',input_shape=(224,224,3)))
model.add(Conv2D(64,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))

model.add(Conv2D(64,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))

model.add(Conv2D(128,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(64,activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1,activation='sigmoid'))

model.compile(loss=keras.losses.binary_crossentropy,optimizer='adam',metrics=['accuracy
```

```
In [72]: model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 222, 222, 32)      896
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 220, 220, 64)      18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 110, 110, 64)      0
_________________________________________________________________
dropout_1 (Dropout)          (None, 110, 110, 64)      0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 108, 108, 64)      36928
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 54, 54, 64)        0
_________________________________________________________________
dropout_2 (Dropout)          (None, 54, 54, 64)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 52, 52, 128)       73856
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 26, 26, 128)       0
_________________________________________________________________
dropout_3 (Dropout)          (None, 26, 26, 128)       0
_________________________________________________________________
flatten_1 (Flatten)          (None, 86528)             0
_________________________________________________________________
dense_1 (Dense)              (None, 64)                5537856
_________________________________________________________________
dropout_4 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 65
=================================================================
Total params: 5,668,097
Trainable params: 5,668,097
Non-trainable params: 0
```

In [5]:
```
# Train from scratch
train_datagen = image.ImageDataGenerator(
    rescale = 1./255,
    shear_range = 0.2,
    zoom_range = 0.2,
    horizontal_flip = True,
)

test_dataset = image.ImageDataGenerator(rescale=1./255)
```

In [6]:
```
train_generator = train_datagen.flow_from_directory(
    'CovidDataset/Train',
    target_size = (224,224),
    batch_size = 32,
    class_mode = 'binary')
```

```
Found 224 images belonging to 2 classes.
```

In [7]:
```
train_generator.class_indices
```

Out[7]:
```
{'Covid': 0, 'Normal': 1}
```

In [8]:
```
validation_generator = test_dataset.flow_from_directory(
    'CovidDataset/Val',
    target_size = (224,224),
    batch_size = 32,
    class_mode = 'binary')
```

```
Found 60 images belonging to 2 classes.
```

In [9]:
```
hist = model.fit_generator(
    train_generator,
    steps_per_epoch=8,
    epochs = 10,
    validation_data = validation_generator,
    validation_steps=2
)
```

```
Epoch 1/10
8/8 [==============================] - 46s 6s/step - loss: 1.7001 - accuracy: 0.5352 -
val_loss: 0.6874 - val_accuracy: 0.5000
Epoch 2/10
8/8 [==============================] - 40s 5s/step - loss: 0.6643 - accuracy: 0.6172 -
val_loss: 0.6633 - val_accuracy: 0.6667
Epoch 3/10
8/8 [==============================] - 40s 5s/step - loss: 0.5563 - accuracy: 0.6914 -
val_loss: 0.4567 - val_accuracy: 0.9167
Epoch 4/10
8/8 [==============================] - 40s 5s/step - loss: 0.4170 - accuracy: 0.8242 -
val_loss: 0.3569 - val_accuracy: 0.9500
Epoch 5/10
8/8 [==============================] - 39s 5s/step - loss: 0.4551 - accuracy: 0.7812 -
val_loss: 0.4027 - val_accuracy: 0.9667
Epoch 6/10
8/8 [==============================] - 38s 5s/step - loss: 0.2974 - accuracy: 0.9023 -
val_loss: 0.0934 - val_accuracy: 0.9500
Epoch 7/10
8/8 [==============================] - 40s 5s/step - loss: 0.2311 - accuracy: 0.9219 -
val_loss: 0.1776 - val_accuracy: 0.9833
Epoch 8/10
8/8 [==============================] - 39s 5s/step - loss: 0.2229 - accuracy: 0.9141 -
val_loss: 0.1062 - val_accuracy: 0.9833
Epoch 9/10
8/8 [==============================] - 38s 5s/step - loss: 0.1453 - accuracy: 0.9531 -
val_loss: 0.0372 - val_accuracy: 0.9833
Epoch 10/10
8/8 [==============================] - 39s 5s/step - loss: 0.1724 - accuracy: 0.9453 -
val_loss: 0.0669 - val_accuracy: 0.9833
```

In [15]:

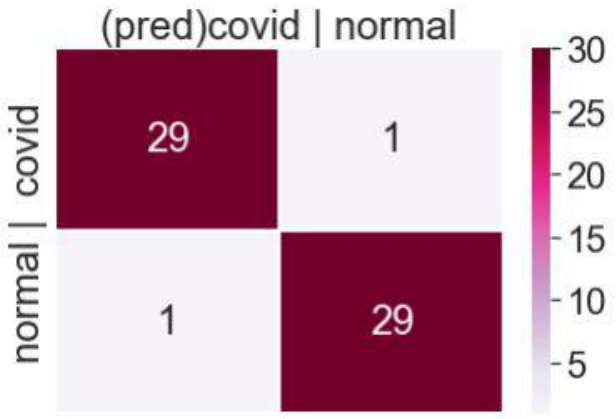
```python
# Saving of model for further implementation
from tensorflow.keras.models import load_model
model.save('covid.h5')
#loading the saved model
new_model=load_model('covid.h5')
```

In [55]:
```
# Class Activation Maps
# Grad-CAM
import os
y_actual=[]
y_test=[]
for i in os.listdir("C:/Users/anand/Desktop/project/CovidDataset/Val/Normal/"):
    img=image.load_img("C:/Users/anand/Desktop/project/CovidDataset/Val/Normal/"+i,targ
    img=image.img_to_array(img)
    img=np.expand_dims(img,axis=0)
    p=model.predict_classes(img)
    y_test.append(p[0,0])
    y_actual.append(1)
```

In [56]:
```
for i in os.listdir("C:/Users/anand/Desktop/project/CovidDataset/Val/Covid/"):
    img=image.load_img("C:/Users/anand/Desktop/project/CovidDataset/Val/Covid/"+i,targe
    img=image.img_to_array(img)
    img=np.expand_dims(img,axis=0)
    p=model.predict_classes(img)
    y_test.append(p[0,0])
    y_actual.append(0)
```

In [69]:
```
y_actual=np.array(y_actual)
y_test=np.array(y_test)
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_actual,y_test)
import seaborn as sns
sns.set(font_scale=2)
sns.heatmap(cm,cmap="PuRd",vmin=1,vmax=30,linewidth=0.9,annot=True, xticklabels=False,
plt.title("(pred)covid | normal")
plt.ylabel("normal |  covid")
```

Out[69]: Text(50.0, 0.5, 'normal | covid')



5. Future Scope

The future scope of this study involves following work:

- ✧ Due to the limited number of COVID-19 images publicly available so far, further experiments are needed on a larger set of cleanly labeled COVID-19 images for a more reliable estimation of the accuracy of these models.
- ✧ the present study does not address any potential effects of additional radiographic findings from coexistent conditions, such as pulmonary edema as seen in congestive heart failure,this is need to be addressed
- ✧ The dataset provides complicated computer vision challenging problems due to the intensity inhomogeneity in the images and irregularities in the data distribution,it should be removed.
- ✧ Better model selection techniques can be used in future.

These points may help in improving the prediction power of the models and lower the prediction error. Incorporating other convolution networks like ResNet18, ResNet50, SqueezeNet, and DenseNet-161 may lead to more sophisticated modeling.

6. Bibliography

- **https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7282464/**
- **https://www.nhlbi.nih.gov/health-topics/chest-x-ray**
- **https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia**
- **https://github.com/ieee8023/covid-chestxray-dataset/tree/master/images**
- **https://www.siemens-healthineers.com/en-in/news/mso-x-ray-imaging-for-covid-19.html**
- **https://en.wikipedia.org/wiki/Chest_radiograph**
- **https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7141645/**
- **https://www.researchgate.net/publication/340518062_Portable_chest_X-ray_in_coronavirus_disease-19_COVID-19_A_pictorial_review**