Project Report On

COVID-19 and Pneumonia Detection from Chest X-Rays with Hyperparameter Tuning using Machine Learning

Submitted by

Bikram Mondal (10200117062)
Adarsha Karmakar (10200117077)
Abhishek Singh (10200117078)
Anirban Jash (10200117072)

Under the guidance of

Prof. Malay Kumar Pakhira

Department of Computer Science and Engineering
Kalyani Government Engineering College
(Kalyani, Nadia, West Bengal)

কল্যাণী - ৭৪১ ২৩৫
নদীয়া, পশ্চিমবঙ্গ

Kalyani 741 235
Nadia, West Bengal,India

পত্রাঙ্ক /Ref. No.:
তারিখ / Date :

কল্যাণী গভঃ ইঞ্জিনিয়ারিং কলেজ
Kalyani Government Engineering College
(Govt. of West Bengal)

Certificate of Approval

This is to certify that this report of project, entitled **"COVID-19 and Pneumonia Detection from Chest X-Rays using Machine Learning"** is a record of bona-fide work, carried out by **Bikram Mondal, Adarsha Karmakar, Abhishek Singh** and **Anirban Jash** under my supervision and guidance.

In my opinion, the report in its present form is in partial fulfillment of all the requirements, as specified by the ***Kalyani Government Engineering College*** and as per regulations of the ***Maulana Abul Kalam Azad University of Technology***. In fact, it has attained the standard, necessary for submission. To the best of my knowledge, the results embodied in this report, are original in natureand worthy of incorporation in the present version of the report for the B. Tech programme in Computer Science and Engineering in the year 2021.

Guide / Supervisor

Malay Kumar Pakhira

Prof. Malay Kumar Pakhira
Department of Computer Science and Engineering
Kalyani Government Engineering College

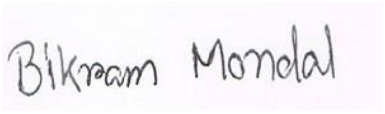
Examiner(s)

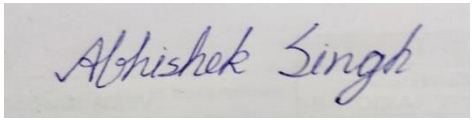
Head of the Department
Computer Science and Engineering
Kalyani Government Engineering College

DECLARATION

I declare that this written submission represents my ideas in my own words and where other's ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principlesof academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly citedor from whom proper permission has not been taken when needed.

Signature





CONTENT

ABSTRACT

The COVID-19 pandemic is causing a major outbreak in more than 150 countries around the world, having a severe impact on the health and life of many people globally.

One of the crucial steps in fighting COVID-19 is the ability to detect the infected patients early enough, and put them under special care. Detecting this disease from radiography and radiology images is perhaps one of the fastest ways to diagnose the patients. Some of the early studies showed specific abnormalities in the chest radiograms of patients infected with COVID-19.

Inspired by earlier works, we study the application of deep learning models to detect COVID19 patients from their chest radiography images. Along with training our model, we will also try to understand how different hyper parameters affect the performance of our model.

.

INTRODUCTION

The novel coronavirus severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), which causes the respiratory disease coronavirus disease-19 (COVID-19), was first identifiedas a cluster of cases of pneumonia in Wuhan, Hubei Province of China on December 31, 2019. Within a month, the disease had spread significantly, leading the World Health Organization (WHO) to designate COVID-19 a public health emergency of international concern. On March 11, 2020, the WHO declared COVID-19 a global pandemic. As of August 18, 2020, the virus has infected > 21 million people, with > 750,000 deaths worldwide. The spread of COVID-19 has had a dramatic impact on social, economic, and health care issues throughout the world, which has been discussed elsewhere.

Prior to this century, members of the coronavirus family had minimal impact on humanhealth.

However, in the past 20 years, outbreaks have highlighted an emerging importance of coronaviruses in morbidity and mortality on a global scale. Although less prevalent than COVID-19, severe acute respiratory syndrome (SARS) in 2002 to 2003 and Middle East respiratory syndrome (MERS) in 2012 likely had higher mortality rates than the current pandemic. Based on this recent history, it is reasonable to assume that we will continue to see novel diseases with similar significant health and societal implications. The challenges presented to health care providers (HCPs) by such novel viral pathogens are numerous, including methods for rapid diagnosis, prevention, and treatment. In the current study, we focus on diagnosis issues, which were evident with COVID-19 with the time required to develop rapid and effective diagnostic modalities.

We have previously reported the utility of using artificial intelligence (AI) in the histopathologic diagnosis of cancer. AI was first described in 1956 and involves the field of computer science in which machines are trained to learn from experience. Machine learning (ML) is a subset of AI and is achieved by using mathematic models to compute sample datasets. Current ML employs deep learning with neural network algorithms, which can recognize patterns and achieve complex computational tasks often far quicker and with increased precision than can humans. In addition to applications in pathology, ML algorithmshave both prognostic and diagnostic applications in multiple medical specialties, such as radiology, dermatology, ophthalmology, and cardiology. It is predicted that AI will impact almost every aspect of health care in the future.

In this article[1], we examine the potential for AI to diagnose patients with COVID-19 pneumonia using chest radiographs (CXR) alone. Employing AI to both screen and diagnose emerging health emergencies such as COVID-19 has the potential to dramatically change how we approach medical care in the future.

LITERATURE SURVEY

Utility of using artificial intelligence (AI) in the histopathologic diagnosis of cancer has been previously reported[2].

AI was first described in 1956 and involves the field of computer science in which machines are trained to learn from experience. Machine learning (ML) is a subset of AI and is achieved by using mathematical models to compute sample datasets.

Current ML[3][4] employs deep learning with neural network algorithms, which can recognize patterns and achieve complex computational tasks often far quicker and with increased precision than can humans.

In addition to applications in pathology, ML algorithms have both prognostic and diagnostic applications in multiple medical specialties, such as radiology, dermatology, ophthalmology, and cardiology.

It is predicted that AI will impact almost every aspect of health care in the future including the corona virus[5][6].

OBJECTIVES

The objectives of this report are:

- To discover what assessments of the different picture pre-processing strategies can be applied to this investigation;
- To recognize whether can these strategies significantly affect the mistakepace of chosen characterization models;
- To inspect the capability of the model
- To suggest which calculations can improve the exactness
- To build a webpage that takes chest X-ray image as input and provides which type of disease it is i.e Normal, Simple Pneumonia or Covid.

REQUIREMENTS AND DESIGN

➢ HARDWARE REQUIREMENTS:

- 2.6 GHz Processor
- 1 GB RAM
- 128 GB HDD

➢ SYSTEM TOOLS REQUIRED:

- Python 3.5+
- Scikit-Learn (latest version)
- Numpy
- Matplotlib
- Keras
- Pandas
- HTML
- Operating System: Windows, Linux, Mac
- Programming Language: Python
- Editor: Google Colab
- Software: Chrome, Mozilla

DATA COLLECTION

COVID-19 RADIOGRAPHY DATABASE (Winner of the COVID-19 Dataset Award by Kaggle Community)

A team of researchers from Qatar University, Doha, Qatar, and the University of Dhaka, Bangladesh along with their collaborators from Pakistan and Malaysia in collaboration with medical doctors have created a database[7] of chest X-ray images for COVID-19 positive cases along with Normal and Viral Pneumonia images. In our current release, there are 1200 COVID-19 positive images, 1341 normal images, and 1345 viral pneumonia images. We will continue to update this database as soon as we have new x-ray images for COVID-19 pneumonia patients.

We have created a database of chest X-ray images for COVID-19 positive cases along with Normal and Viral Pneumonia images. In our current release, there are 1200 COVID-19 positive images, 1341 normal images, and 1345 viral pneumoniaimages. We will continue to update this database as soon as we have new x-ray images for COVID-19 pneumonia patients. Main objective is to researchers can use this database to produce useful and impactful scholarly work on COVID-19, which can help in tackling this pandemic.

We have developed the database of COVID-19 x-ray images from the Italian Society of Medical and Interventional Radiology (SIRM) COVID-19 DATABASE[1], Novel Corona Virus 2019 Dataset developed by Joseph Paul Cohen and Paul Morrison and Lan Dao and images extracted from 43 different publications. ImageFormats: All the images are in Portable Network Graphics (PNG) file format and resolution is 1024-by-1024 pixels, and 256-by-256 pixels, which can be easily converted to 224-by-224 or 227-by-227 pixels typically required by the popular Convolutional Neural Networks (CNNs).

PRE-PROCESSING OF DATA

- Resizing of the Images: All the images are in Portable Network Graphics (PNG) file format and resolution is 1024- by-1024 pixels, and 256-by-256 pixels, which can be easily converted to 224-by-224 or 227- by-227 pixels typically required by the popular Convolutional Neural Networks (CNNs).
- Converting images into grayscale
- Converting the pixels to numpy arrays
- Normalization

ALGORITHM AFTER PRE-PROCESSING

- Split the dataset in training set and testing set.
- Build and train the model using CNN.
- Pass the testing set through the model for validation purpose.
- Adjust various parameters and hyperparameters such as epochs, batch size and find out which dataset combination turns out to be the most effective and accurate one.
- Build a webpage that will take scanned images of chest X-rays and after predicting it will say whether it is Normal, Simple pneumonia or Covid affected.

CNN (Convolutional Neural Network)

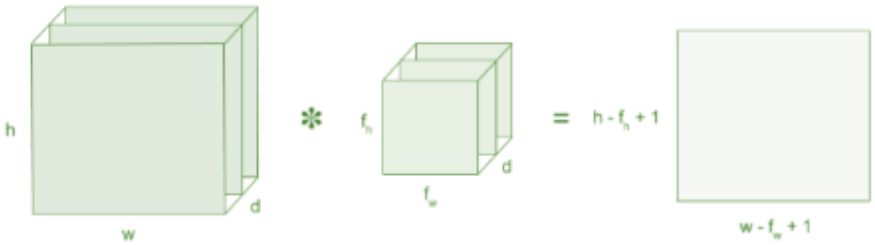
A Convolutional Neural System (CNN)[8] is a multi-layer neural feed-forward system with deep supervised learning design, which can be viewed as a two- section combination: programmed feature extractor and trainable classifier. The classifier and weights of the back-propagation algorithm in the feature extractor areapplied. In addition, CNN can likewise separate topology attributes from pictures. It abstracts features from the essential picture in the main layer and classifies the pattern with the last layer.

In neural systems, Convolutional neural system (ConvNets or CNNs) is one of the fundamental classes to do pictures acknowledgment, pictures arrangements. Image recognition, acknowledgment faces and so forth., are a portion of the zones where CNNs are generally utilized.

Actually, profound learning CNN models to train and test, each information picture will go it through a progression of convolution layers with filters (Kernels), Pooling, fully connected layers (FC) and apply Softmax capacity to order an item with probabilistic qualities somewhere in the range of 0 and 1.

Convolution is the primary layer to extricate highlights from an info picture. Convolution protects the connection between pixels by learning image features utilizing little squares of information. It is a numerical activity that takes two inputs, for example, image matrix and a filter or kernel.

- An image matrix (volume) of dimension **(h x w x d)**
- A filter **(f_h x f_w x d)**
- Outputs a volume dimension **(h - f_h + 1) x (w - f_w + 1) x 1**



Consider an 6x6 greyscale image in the below figure 8.1

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

Now we will convolve this 6x6 image matrix with 3x3 filter given in figure 8.2

We take the first 3 X 3 matrix from the 6 X 6 image and multiply it with the filter.Now, the first element of the 4 X 4 output matrix will be the sum of the element- wise product of these values, i.e., 3*1 + 0 + 1*-1 + 1*1 + 5*0 + 8*-1 + 2*1 + 7*0
+ 2*-1 = -5. To calculate the second element of the 4 X 4 output, we will shift our filter one step towards the right and again get the sum of the element-wise product.In figure 8.3 we get the output matrix.

-5	-4	0	8
-10	-2	2	3
0	-2	-4	-7
-3	-2	-3	-16

Padding

When we have NxN image matrix and FxF filter matrix the output matrix will beof (N- F+1) x (N-F+1).

There are basically two hindrances here:

1. Each time we apply a convolutional activity, the size of the picture contracts.

2. Pixels present toward the side of the picture are utilized just a couple of number of times during convolution when contrasted with the focal pixels. Subsequently, we don't concentrate a lot on the corners since that can prompt datamisfortune.

To overcome this issues padding is needed i.e., adding of pixels all around thematrix.

After padding the input matrix:

- Input : N x N
- Padding : p
- Filter : F x F
- Output matrix: (N+2p-F+1) x (N+2p-F+1)

Strides

Stride is the quantity of pixels shifts over the information lattice (input matrix). At the point when the stride is 1 then we move the filters to 1 pixel at once. At the point when the step is 2 then we move the filters to 2 pixels one after another, etc. Stride assists with decreasing the size of the picture, an especially valuable component.

After Striding the input matrix:

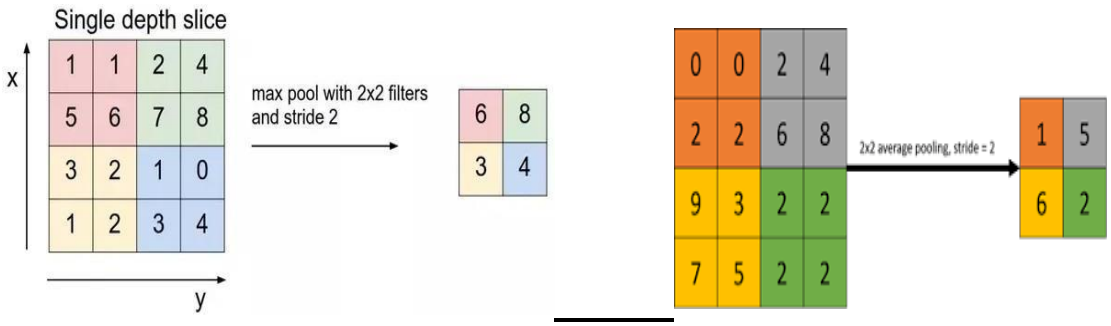
- Input : N x N
- Padding : p
- Strides : s
- Filter : F x F
- Output matrix : [(N+2p-F)/s+1] x [(N+2p-F)/s+1]

Pooling Layers

Pooling layers segment would lessen the quantity of parameters when the pictures are excessively huge. Spatial pooling likewise called subsampling or down sampling which lessens the dimensionality of each guide however holds significantdata. Spatial pooling can be of various sorts:

1. Max Pooling: Takes the largest element from the rectified feature map.
2. Avg Pooling: Takes the avg of all elements.
3. Sum Pooling: Takes the sum of all elements.

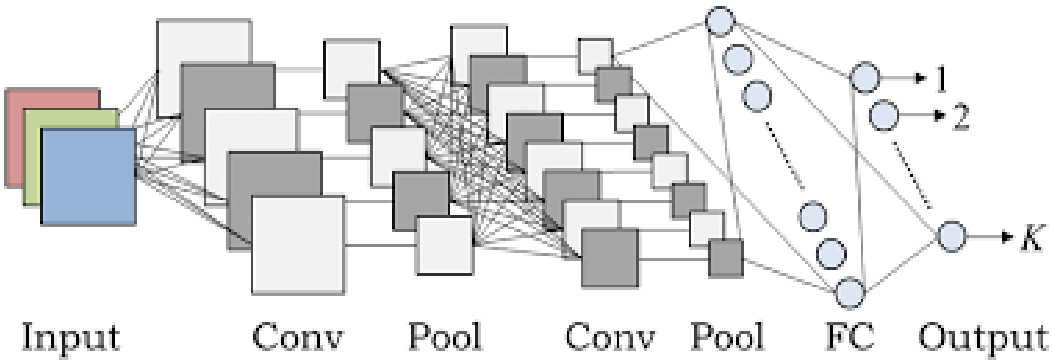
Figure 8.5 shows maximum pooling and average pooling.



Fully Connected Layer (FC)

In FC layer, we straightened our lattice into vector and feed it into a completely associated layer like a neural system. Actually, the feature map matrix gets converted as vectors. With all these connected Layers a model is created. Finallyusing activation functions like Softmax or sigmoid
the output is classified.

The following figure shows the working of CNN.



Advantages of CNN:

1. They are very efficient and accurate.
2. Works very well in image recognition problems.

Disadvantages of CNN:

1. If the processor of the system is not good it takes a lot of time to train thedata.
2. High computational cost.

PARAMETERS OF THE MODEL

After adding the layers, we need to compile our model[9]. Compiling the Model takes three parameters:

1. **Loss:** We have used here categorical cross entropy loss function as there are two or more label classes. In these loss function labels are to be provided in a one_hot representation.

2. **Optimizer:** It is required for compiling the Keras model. We have used two different optimizers (1. adam 2.adadelta) and checked out the accuracy for both of them. We will see them in result section.

3. **Metrics:** A metric is a function that is used to judge the performance of the model.
Here we have used accuracy metrics that calculates how often our predictions matches our labels.

After compiling our model we fit our training data to the model. We passed our training images and training labels to fit function. Batch size is number of samples per gradient update. If unspecified, batch_size will default to 32.
One Epoch is when the Full dataset is passed forward and backword through the neural network only once. One Epoch is too big to be feeded to an algorithm in once so we have divided it into 128 batches. As we increase our Epoch our accuracy of the model increases till a certain extent then the accuracy becomes constant. We will observe these things in the Result section.

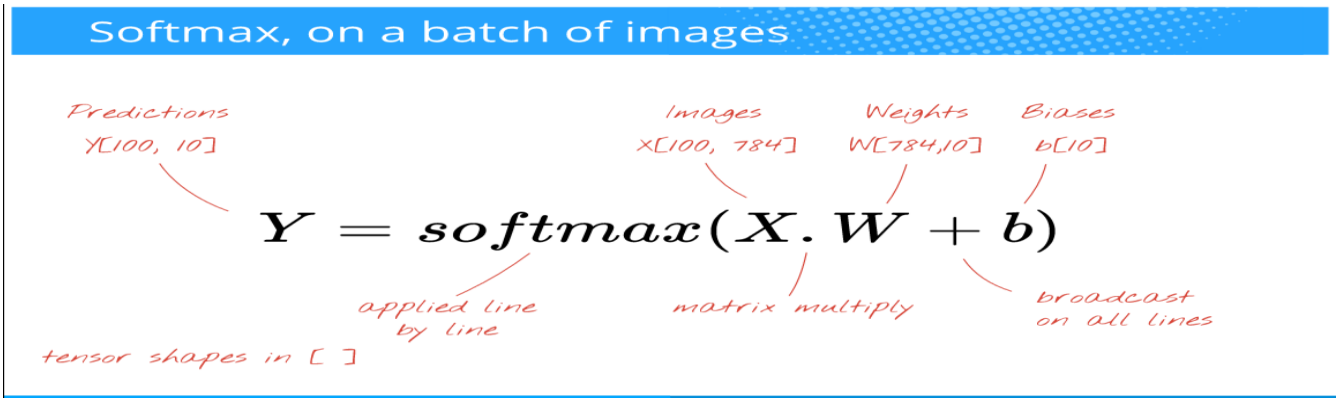
Verbose just help us in looking in the training progress for each Epoch. Verbose can have values as 0,1,2. For 0 we will get to see nothing (silent), for 1 we will see an animated progress bar and for 2 will just mention the no. of Epochs.

At last, we move to the last step of our model to evaluate the testing dataset. After evaluation we can get the accuracy of our model and the errors caused by our model.

DIFFERENT HYPERPARAMETERS

Batch Size:

The batch size is a hyperparameter of gradient descent that controls the number of training samples to work through before the model's internal parameters are updated.



Drop Out:

A single model can be used to simulate having a large number of different network architectures by randomly dropping out nodes during training. This is called dropout

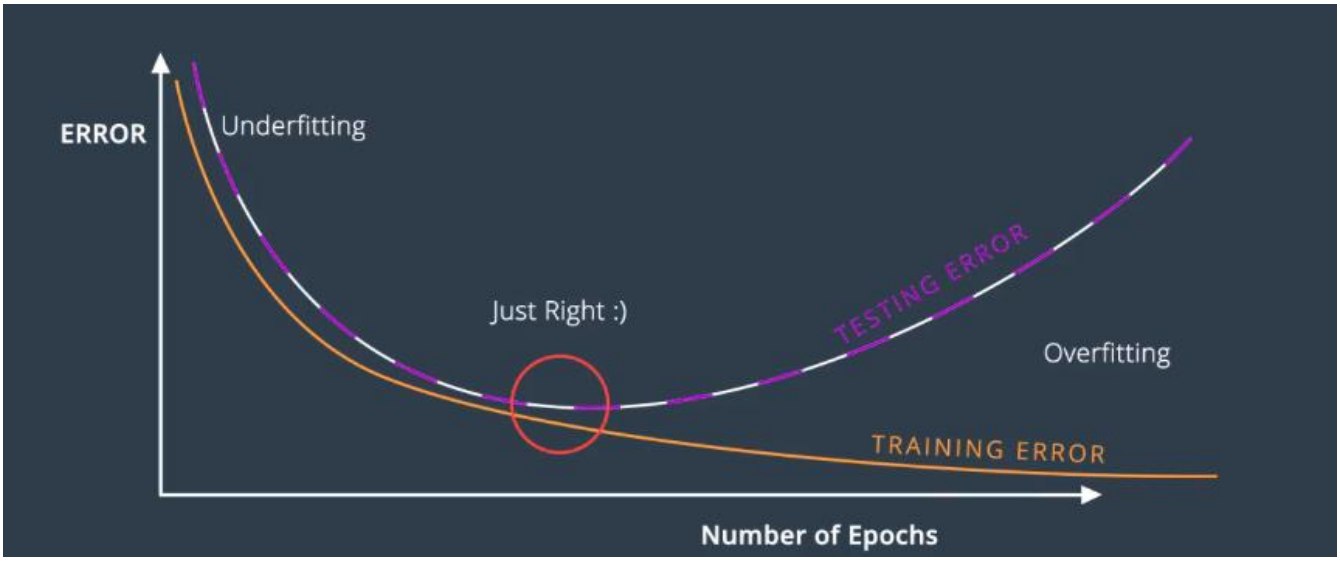
A very computationally cheap and remarkably effective regularization method to reduce overfitting and improve generalization

The default interpretation of the dropout hyperparameter is the probability of training a given node in a layer, where 1.0 means no dropout, and 0.0 means no outputs from the layer.

Epoch:

One Epoch is when an entire dataset is passed forward and backward through the neural network only once.

Since one epoch is too big to feed to the computer at once we divide it in several smaller batches.



CODE

project.py - C:/Users/BIKRAM MONDAL/Desktop/project.py (3.7.3)

File Edit Format Run Options Window Help

```
import os
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials

auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)

download = drive.CreateFile({'id': '14_4PbKRNhIyFh5__-iFR8pLiDUJSu9xI'})
download.GetContentFile('Covid_dataset.zip')
!unzip Covid_dataset.zip

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import keras
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D
from keras.layers import Dense, Dropout
from keras.layers import Flatten, BatchNormalization
import cv2

def load_images_from_folder(folder):
    images = []
    for filename in os.listdir(folder):
        img = cv2.imread(os.path.join(folder,filename),0)
        if img is not None:
            images.append(img)
    return images
```

```
'''
We are intentionally combining the train and test data sets.
So that we can vary the ratio as and when required directly from the code itself.
'''

folder1=r"/content/Covid-19 Image Dataset/test/Covid"
folder2=r"/content/Covid-19 Image Dataset/test/Normal"
folder3=r"/content/Covid-19 Image Dataset/test/Viral Pneumonia"
folder4=r"/content/Covid-19 Image Dataset/train/Covid"
folder5=r"/content/Covid-19 Image Dataset/train/Normal"
folder6=r"/content/Covid-19 Image Dataset/train/Viral Pneumonia"

li1=load_images_from_folder(folder1)
li2=load_images_from_folder(folder2)
li3=load_images_from_folder(folder3)
li4=load_images_from_folder(folder4)
li5=load_images_from_folder(folder5)
li6=load_images_from_folder(folder6)
li=li1+li2+li3+li4+li5+li6
n=len(li)

labl=[]
#Since there are only 3 categories, we have directly used the one-hot encoded values for 1,2 and 3
for i in range(len(li1)):
    labl.append([1,0,0])
for i in range(len(li2)):
    labl.append([0,1,0])
for i in range(len(li3)):
    labl.append([0,0,1])
for i in range(len(li4)):
    labl.append([1,0,0])
for i in range(len(li5)):
    labl.append([0,1,0])
for i in range(len(li6)):
    labl.append([0,0,1])

for i in range(n):
 li[i]=cv2.resize(li[i],(480,640))
 li[i]=li[i]/255.0

li = np.array(li)
li = li.reshape(li.shape + (1,))
labl = np.asarray(labl).astype('int32')
```

```
def baseline_model():
    model = Sequential()
    model.add(BatchNormalization(input_shape=(640,480,1)))
    model.add(Conv2D(32, (3, 3), padding='same', activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2,2)))
    model.add(Dropout(0.25))

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

    model.add(Flatten())
    model.add(Dense(128, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(3, activation='softmax'))
    return model

model = baseline_model()
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

X_train, X_test, y_train, y_test = train_test_split(li, labl, test_size=0.20)
history = model.fit(X_train, y_train,
          batch_size=32,
          epochs=3,
          verbose=1,
          validation_data=(X_test, y_test))

loss_train = history.history['loss']
loss_val = history.history['val_loss']
epochs = range(1,4)
plt.plot(epochs, loss_train, 'g', label='Training loss')
plt.plot(epochs, loss_val, 'b', label='validation loss')
plt.title('Training and Validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

```
loss_train = history.history['accuracy']
loss_val = history.history['val_accuracy']
epochs = range(1,4)
plt.plot(epochs, loss_train, 'g', label='Training accuracy')
plt.plot(epochs, loss_val, 'b', label='validation accuracy')
plt.title('Training and Validation accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()

y_pred=model.predict(X_test)
y_pred=np.array(y_pred)
y_test=np.array(y_test)
y_pred_=[]
y_test_=[]
for i in range(len(y_pred)):
  m=max(y_pred[i])
  for j in range(3):
    if(y_pred[i][j]==m):
      y_pred_.append(j)
      break
for i in range(len(y_test)):
  for j in range(3):
    if(y_test[i][j]==1):
      y_test_.append(j)
      break
#print(y_test_)
#print(y_pred_)
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test_, y_pred_)

import seaborn as sns
sns.heatmap(cm,cmap="plasma",annot=True)
```

```
!pip install flask_ngrok

from flask_ngrok import run_with_ngrok
from flask import Flask, request, render_template
app=Flask(__name__, template_folder='/content/templates', static_folder='/content/templates')
run_with_ngrok(app)
@app.route("/")
def home():
  return render_template("index.html")
@app.route("/predict",methods=['POST'])
def predict():
  f=request.files['file']
  f.save(f.filename)
  def pred_one_img(img):
    img=cv2.resize(img,(480,640))
    img=img/255.0
    img = img.reshape(img.shape + (1,))
    return img
  img=pred_one_img(cv2.imread("/content/"+str(f.filename),0))
  prediction=model.predict(np.expand_dims(img,axis=0))
  class_level=np.where(prediction[0]==max(prediction[0]))[0][0]
  if(class_level==0):
    cls="COVID"
  elif(class_level==1):
    cls="Normal"
  else:
    cls="Viral Pneumonia"
  return render_template("index.html", Prediction=cls)

app.run()
```

HTML CODE[10][11]

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6" crossorigin="anonymous">

    <title>COVID-19 Detection</title>
  </head>
  <body class="bg-dark">
    <div class="p-3 mb-2 bg-warning text-white">
       <div class="container text-center">
          <h1 class="text-dark">Hi, there</h1>
       </div>

    </div>
    <div class="container text-center color:$gray-900 margin-top:5em">
       <h1 class="text-dark">8</h1>
       <h1 class="text-warning">COVID'19 & PNEUMONIA DETECTION</h1>
       <img class="thumbnail img-responsive" src="{{url_for('static', filename='ui_image.png')}}" alt="Chania" width="1200"
height="543">
       <h3 class="text-warning">Upload the chest X-ray</h3>
    </div>
    <div class="container text-center">
       <form action="{{url_for('predict')}}" method="post" enctype=multipart/form-data>
                    <div class="form-group>
                                 <label for="file">Select image:</label>
                                 <input type="file" id="file" class="form-control" name="file" accept="image/*">
                                 <input type="submit" value="Submit">
                    </div>
       </form>
       <h1 class="text-dark">8</h1>
       <h5 class="text-white">{{Prediction}}</h5>
    </div>

   <!-- Optional JavaScript; choose one of the two! -->

    <!-- Option 1: Bootstrap Bundle with Popper -->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
JEW9xMcG8R+pH31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9Ekf" crossorigin="anonymous"></script>

    <!-- Option 2: Separate Popper and Bootstrap JS -->
    <!--
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.1/dist/umd/popper.min.js" integrity="sha384-
SR1sx49pcuLnqZUnnPwx6FCym0wLsk5JZuNx2bPPENzswTNFaQU1RDvt3wT4gWFG" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.min.js" integrity="sha384-
j0CNLUeiqtyaRmlzUHCPZ+Gy5fQu0dQ6eZ/xAww941Ai1SxSY+0EQqNXNE6DZiVc" crossorigin="anonymous"></script>
    -->
  </body>
</html>
```

RESULT

A) Using Optimizer as "adam"

1. No of Epoch = 3

```
Epoch 1/3
8/8 [==============================] - 31s 1s/step - loss: 256.2269 - accuracy: 0.3863 - val_loss: 8.5725 - val_accuracy: 0.3438
Epoch 2/3
8/8 [==============================] - 6s 724ms/step - loss: 97.7721 - accuracy: 0.7093 - val_loss: 0.7424 - val_accuracy: 0.8594
Epoch 3/3
8/8 [==============================] - 6s 723ms/step - loss: 19.9058 - accuracy: 0.8394 - val_loss: 0.4864 - val_accuracy: 0.8750
```

2. No of Epoch = 5

```
Epoch 1/5
8/8 [==============================] - 9s 884ms/step - loss: 457.9508 - accuracy: 0.3718 - val_loss: 2.3380 - val_accuracy: 0.3281
Epoch 2/5
8/8 [==============================] - 6s 805ms/step - loss: 110.4951 - accuracy: 0.6559 - val_loss: 0.7530 - val_accuracy: 0.6875
Epoch 3/5
8/8 [==============================] - 6s 812ms/step - loss: 13.0188 - accuracy: 0.8734 - val_loss: 0.5903 - val_accuracy: 0.8594
Epoch 4/5
8/8 [==============================] - 7s 845ms/step - loss: 4.8364 - accuracy: 0.8218 - val_loss: 0.7935 - val_accuracy: 0.7344
Epoch 5/5
8/8 [==============================] - 6s 805ms/step - loss: 1.5137 - accuracy: 0.8828 - val_loss: 0.9296 - val_accuracy: 0.6875
```

3. No of Epoch = 10

```
Epoch 1/10
8/8 [==============================] - 8s 891ms/step - loss: 364.7466 - accuracy: 0.4003 - val_loss: 5.1026 - val_accuracy: 0.3281
Epoch 2/10
8/8 [==============================] - 6s 818ms/step - loss: 231.8271 - accuracy: 0.5906 - val_loss: 0.5688 - val_accuracy: 0.7812
Epoch 3/10
8/8 [==============================] - 6s 814ms/step - loss: 46.9758 - accuracy: 0.8102 - val_loss: 0.5010 - val_accuracy: 0.9062
Epoch 4/10
8/8 [==============================] - 6s 811ms/step - loss: 17.5379 - accuracy: 0.8602 - val_loss: 0.6690 - val_accuracy: 0.7031
Epoch 5/10
8/8 [==============================] - 7s 843ms/step - loss: 7.6740 - accuracy: 0.8189 - val_loss: 0.8274 - val_accuracy: 0.6562
Epoch 6/10
8/8 [==============================] - 6s 813ms/step - loss: 1.0602 - accuracy: 0.9097 - val_loss: 0.8251 - val_accuracy: 0.7656
Epoch 7/10
8/8 [==============================] - 7s 849ms/step - loss: 0.8272 - accuracy: 0.9010 - val_loss: 0.8195 - val_accuracy: 0.7812
Epoch 8/10
8/8 [==============================] - 6s 811ms/step - loss: 0.3975 - accuracy: 0.8707 - val_loss: 0.8107 - val_accuracy: 0.8281
Epoch 9/10
8/8 [==============================] - 6s 808ms/step - loss: 0.2454 - accuracy: 0.8812 - val_loss: 0.8013 - val_accuracy: 0.8281
Epoch 10/10
8/8 [==============================] - 6s 806ms/step - loss: 0.3611 - accuracy: 0.8759 - val_loss: 0.7745 - val_accuracy: 0.8594
```

B) Using Optimizer as “adadelta”

1. No of Epoch = 3

```
Epoch 1/3
8/8 [==============================] - 35s 1s/step - loss: 1.5413 - accuracy: 0.3994 - val_loss: 1.0902 - val_accuracy: 0.5156
Epoch 2/3
8/8 [==============================] - 6s 745ms/step - loss: 0.7243 - accuracy: 0.7208 - val_loss: 1.0727 - val_accuracy: 0.6719
Epoch 3/3
8/8 [==============================] - 6s 744ms/step - loss: 0.4840 - accuracy: 0.7986 - val_loss: 1.0544 - val_accuracy: 0.8125
```

2. No of Epoch = 5

```
Epoch 1/5
8/8 [==============================] - 9s 905ms/step - loss: 1.3917 - accuracy: 0.4328 - val_loss: 1.0895 - val_accuracy: 0.4219
Epoch 2/5
8/8 [==============================] - 7s 837ms/step - loss: 0.7046 - accuracy: 0.6873 - val_loss: 1.0757 - val_accuracy: 0.4062
Epoch 3/5
8/8 [==============================] - 7s 831ms/step - loss: 0.6176 - accuracy: 0.7532 - val_loss: 1.0661 - val_accuracy: 0.7812
Epoch 4/5
8/8 [==============================] - 7s 834ms/step - loss: 0.4073 - accuracy: 0.8142 - val_loss: 1.0555 - val_accuracy: 0.8281
Epoch 5/5
8/8 [==============================] - 7s 869ms/step - loss: 0.3307 - accuracy: 0.8937 - val_loss: 1.0320 - val_accuracy: 0.7656
```

3. No of Epoch = 10

```
Epoch 1/10
8/8 [==============================] - 31s 1s/step - loss: 1.6699 - accuracy: 0.4922 - val_loss: 1.0758 - val_accuracy: 0.5469
Epoch 2/10
8/8 [==============================] - 6s 742ms/step - loss: 0.6004 - accuracy: 0.7341 - val_loss: 1.0596 - val_accuracy: 0.6719
Epoch 3/10
8/8 [==============================] - 6s 747ms/step - loss: 0.4620 - accuracy: 0.8440 - val_loss: 1.0407 - val_accuracy: 0.6719
Epoch 4/10
8/8 [==============================] - 6s 743ms/step - loss: 0.3855 - accuracy: 0.8628 - val_loss: 1.0358 - val_accuracy: 0.7969
Epoch 5/10
8/8 [==============================] - 6s 784ms/step - loss: 0.2691 - accuracy: 0.9076 - val_loss: 1.0245 - val_accuracy: 0.7969
Epoch 6/10
8/8 [==============================] - 6s 742ms/step - loss: 0.3093 - accuracy: 0.8829 - val_loss: 1.0073 - val_accuracy: 0.7812
Epoch 7/10
8/8 [==============================] - 6s 742ms/step - loss: 0.3225 - accuracy: 0.8670 - val_loss: 0.9862 - val_accuracy: 0.8281
Epoch 8/10
8/8 [==============================] - 6s 742ms/step - loss: 0.3166 - accuracy: 0.8807 - val_loss: 0.9726 - val_accuracy: 0.8750
Epoch 9/10
8/8 [==============================] - 6s 743ms/step - loss: 0.2571 - accuracy: 0.9161 - val_loss: 0.9565 - val_accuracy: 0.8750
Epoch 10/10
8/8 [==============================] - 6s 738ms/step - loss: 0.3054 - accuracy: 0.8948 - val_loss: 0.9390 - val_accuracy: 0.9062
```

So, after training and testing we get the following results based on **No. of Epoch** and **Optimizer used.**

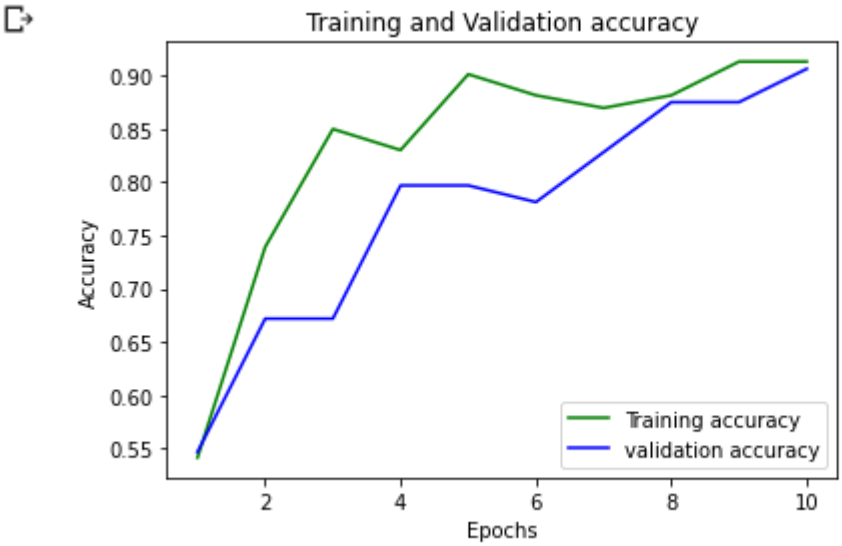
Accuracy table based on No. Of Epoch and Optimizer used:

	Epoch = 3 (Accuracy %)	**Epoch = 5 (Accuracy %)**	**Epoch = 10 (Accuracy %)**
Optimizer: adam	**87.50 %**	**68.75 %**	**85.94 %**
Optimizer: adadelta	**81.25 %**	**76.56 %**	**90.62 %**

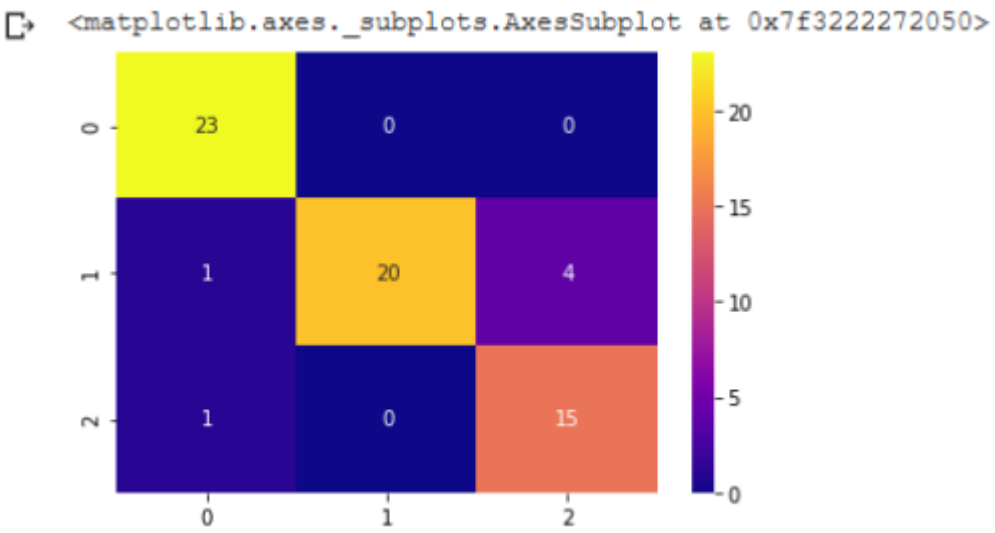
Accuracy of the model: 90.62 %

Plot of No. of Epochs vs. Loss:

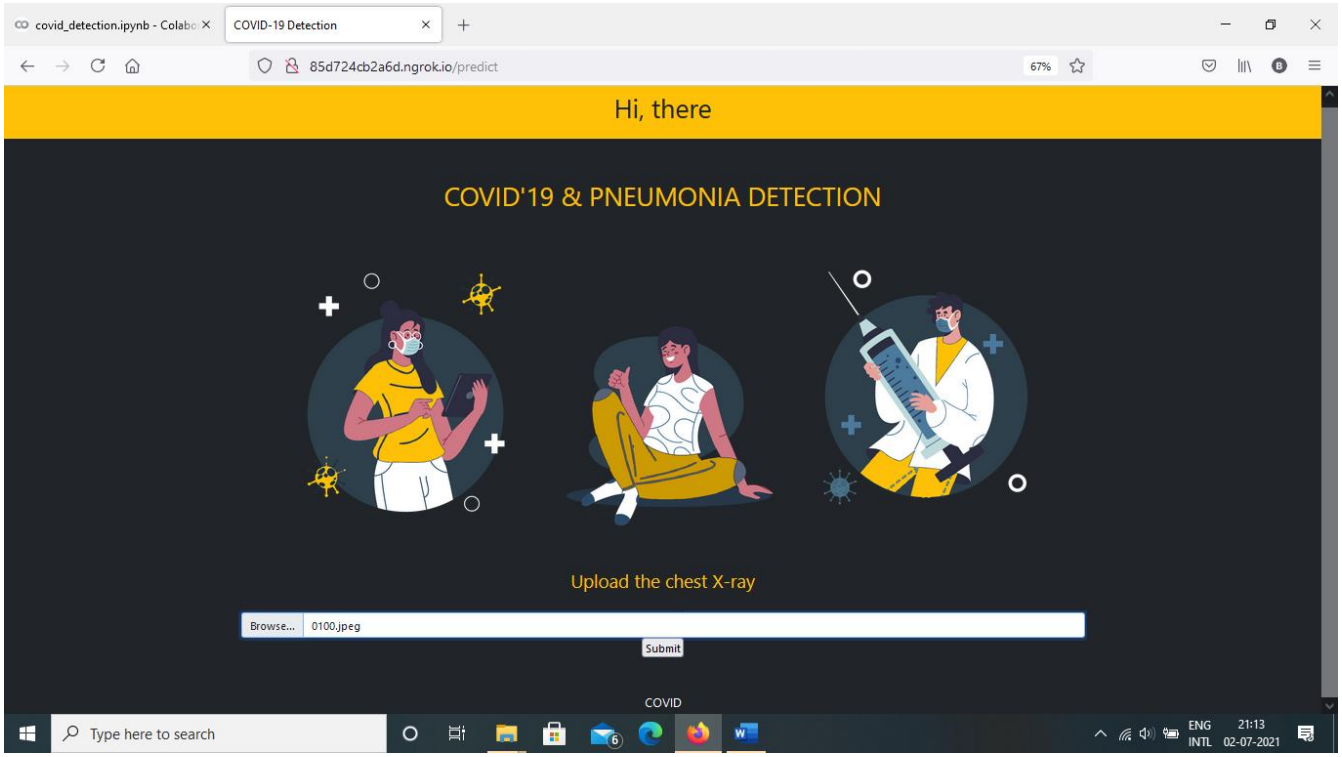
Plot of No. of Epochs vs. Accuracy:



Confusion Matrix[9]:



Webpage:


covid_detection.ipynb - Colabo
COVID-19 Detection
85d724cb2a6d.ngrok.io/predict
67%
Hi, there
COVID'19 & PNEUMONIA DETECTION
Upload the chest X-ray
Browse...
0100.jpeg
Submit
COVID
Type here to search
ENG
INTL
21:13
02-07-2021

ADVANTAGES AND DISADVANTAGES

- **ADVANTAGES:**

  - Since ReLu is used, because it's simple, fast, and it empirically seems to work well. Empirically, early papers observed that training a deep network with ReLu tended to converge much more quickly and reliably than training a deep network with sigmoid activation.
  - We have the flexibility of choosing the best possible options for hyper-parameters instead of hard-coding just one set of values.

- **DISADVANTAGES:**

  - Does not include regularization factor yet. (Because we have not trained extensively on an extremely large dataset)
  - This works fine for now but may turn out to be problematic if the model starts overfitting.

CONCLUSION

As using machine learning algorithms are used like KNN, SVM, Neural networks along with different parameters and feature scaling vectors, we also saw the different comparison among the classifiers in terms of the most important feature of accuracy and timing. Accuracy can alter as it depends on the splitting of training and testing data, and this can further be improved if the number of training and testing data is provided. There is always a chance to improve accuracy if the size of data increases. Every classifier has its own accuracy and time consumption. We can also include the fact that if the power of CPU changes to GPU, the classifier can perform with better accuracy and less time and better results can be observed.

FUTURE SCOPE & CONFINEMENTS

- Working with most recent updated dataset to extract features more accurately.
- Building a user-friendly android application so that one can upload the digital copy of his chest X-ray and can be aware of whether he is affected with Covid or not.

Limitations

In our preliminary study, we have demonstrated the potential impact AI can have in multiple aspects of patient care for emerging pathogens such as COVID-19 using a test as readily available as a CXR. However, several limitations to this investigation should be mentioned. The study is retrospective in nature with limited sample size and with X-rays from patients with various stages of COVID-19 pneumonia. Also, cases of non-COVID-19 pneumonia are not stratified into different types or etiologies. We intend to demonstrate the potential of AI in differentiating COVID-19 pneumonia from non-COVID-19 pneumonia of any etiology, though future studies should address comparison of COVID-19 cases tomore specific types of pneumonias, such as of bacterial or viral origin.

Furthermore, 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, pleural effusions (which can be seen with COVID-19 pneumonia, though rarely), interstitial lung disease, etc. Future studies are required to address these issues. Ultimately, prospective studies to assess AI- assisted radiographic interpretation in conditions such as COVID-19 are requiredto demonstrate the impact on diagnosis, treatment, outcome, and patient safety asthese technologies are implemented.

REFERENCES

[1]. Chih-Cheng Lai, Tzu-Ping Shih, Wen-Chien Ko, Hung-Jen Tang, and Po-Ren Hsuehe, Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) and coronavirus disease-2019 (COVID-19): The epidemic and the challenges, PubMed Central (PMC - NIH/NLM), doi: 10.1016/j.ijantimicag.2020.105924, 2020 Feb 17

Atul Sharma, Swapnil Tiwari, Manas Kanti Deb, and Jean Louis Martyc, Severe acute respiratory syndrome coronavirus-2 (SARS-CoV-2): a global pandemic and treatment strategies, PubMed Central (PMC - NIH/NLM), doi: 10.1016/j.ijantimicag.2020.106054, 2020 Jun 10

[2]. Andrew A. Borkowski, Catherine P. Wilson, Steven A. Borkowski, L. Brannon Thomas, Lauren A. Deland, Stefanie J. Grewe, and Stephen M. Mastorides, Comparing Artificial Intelligence Platforms for Histopathologic Cancer Diagnosis, PubMed Central (PMC - NIH/NLM), Fed Pract.; 36(10): 456–463, 2019

[3]. https://towardsdatascience.com/detecting-covid-19-using-deep-learning-262956b6f981

[4]. https://www.researchgate.net/publication/341980921_COVID-19_Prediction_and_Detection_Using_Deep_Learning

[5]. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7127800/

[6]. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7286265/

[7]. https://www.kaggle.com/pranavraikokte/covid19-image-dataset

[8]. https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/

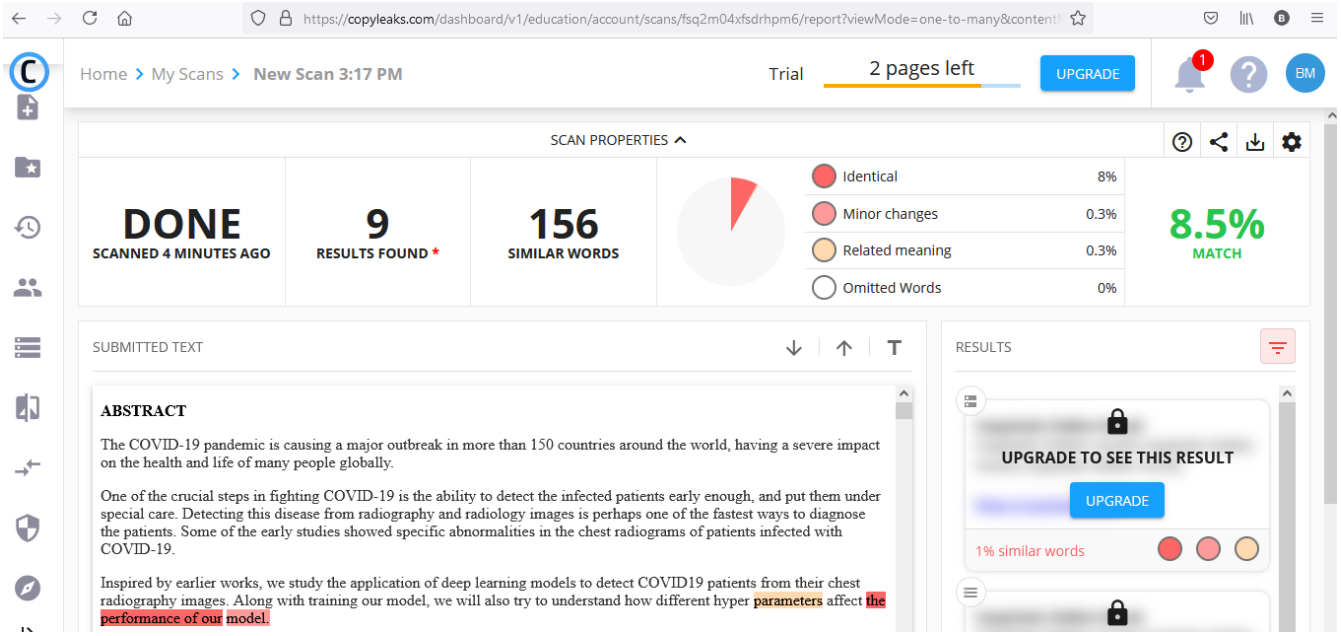
[9]. https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html

[10]. https://pypi.org/project/flask-ngrok/

[11]. https://www.kdnuggets.com/2020/05/build-deploy-machine-learning-web-app.html

PLAGIARISM REPORT

We have used Copyleaks to generate the below attached report which shows 91.5% unique content and 8.5% common texts or definitions that has been taken from other research papers and websites which has been mentioned in the reference.



Signed by: