APPENDIX I

CODES FOR CNN MODEL

#loading dataset

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
import os
import cv2
```

In [2]:

```
Datadir = "C:/train/traindata"
CATEGORIES = ["covid", "non"]

for category in CATEGORIES:
    path = os.path.join(Datadir, category)   #path to image directory
    for image in os.listdir(path):
        img_array = cv2.imread(os.path.join(path, image), cv2.IMREAD_GRAYSCALE)
        #plt.imshow(img_array, cmap="gray")
        #plt.show()
        break
    break
```

In [3]:

```
#print(img_array)
#print(img_array.shape)
```

In [3]:

```
image_size = 28

new_array = cv2.resize(img_array, (image_size, image_size ))
#plt.imshow(new_array, cmap="gray")
#plt.show()
```

In [4]:

```
#collecting training data

train_data = []

Datadir = "C:/train/traindata"
CATEGORIES = ["covid", "non covid"]

def training_data():
    for category in CATEGORIES:
        path = os.path.join(Datadir, category)   #path to image directory
        classification_num = CATEGORIES.index(category)
        for image in os.listdir(path):
            try:
                img_array = cv2.imread(os.path.join(path, image),
cv2.IMREAD_GRAYSCALE)
                new_array = cv2.resize(img_array, (image_size, image_size ))
                train_data.append([new_array, classification_num])
            except Exception as e:
                pass

training_data()
```

In [5]:

```
print(len(train_data))
```

```
9544
```

In [6]:

```
import random       #to shuffle data
random.shuffle(train_data)
```

In [7]:

```
for img in train_data[:10]:
    print(img[1])
```

```
1
1
1
1
1
0
1
1
0
1
```

In [8]:

```
#packing data into variables to use

X = []
Y = []

for features, label in train_data:
    X.append(features)
    Y.append(label)

X = np.array(X).reshape(-1, image_size, image_size, 1)
Y = np.array(Y)
```

In [9]:

```
# saving the data

import pickle

pickle_out = open("X.pickle", "wb")
pickle.dump(X, pickle_out)
pickle_out.close()

pickle_out = open("Y.pickle", "wb")
pickle.dump(Y, pickle_out)
pickle_out.close()
```

In [10]:

```
#to load back data


pickle_in = open("X.pickle", "rb")
X = pickle.load(pickle_in)
```

In [11]:

```
#CNN

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Activation, Dropout, Conv2D, Flatten,
MaxPooling2D
```

In [12]:

```
#lets normalize the data

x = X/255.0
```

In [17]:

```
model = Sequential()
model.add(Conv2D(64, (4,4), input_shape = x.shape[1:]))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size = (3,3)))

model.add(Conv2D(64, (4,4)))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size = (3,3)))

model.add(Flatten())
model.add(Dense(64))

model.add(Dense(1))
model.add(Activation('sigmoid'))

model.compile(loss="binary_crossentropy", optimizer="adam", metrics=['accuracy'])

model.fit(x, Y, batch_size=64, validation_split=0.1, epochs=10, shuffle=True)
```

In [19]:

```
model.save('bcnnmodel.h5')
```

In [13]: