MANAV RACHNA
vidyavatarikshaa

MANAV RACHNA UNIVERSITY

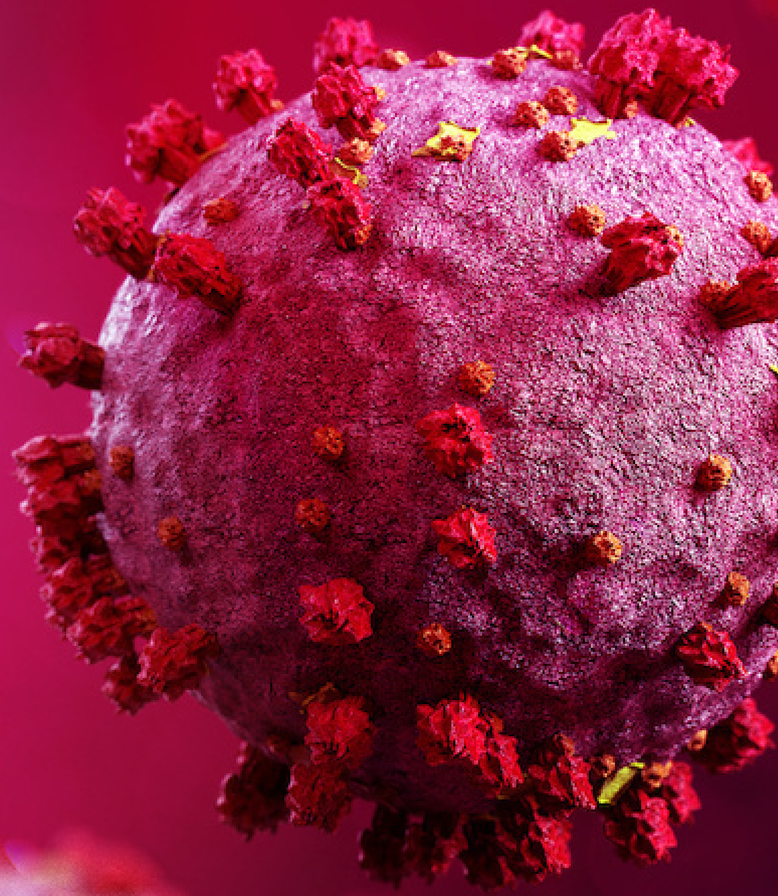
(FORMERLY MANAV RACHNA COLLEGE OF ENGINEERING
NAAC ACCREDITED 'A' GRADE INSTITUTION)

Declared as State Private University under section 2f of the UGC act, 1956

# COVID CLASSIFICATION

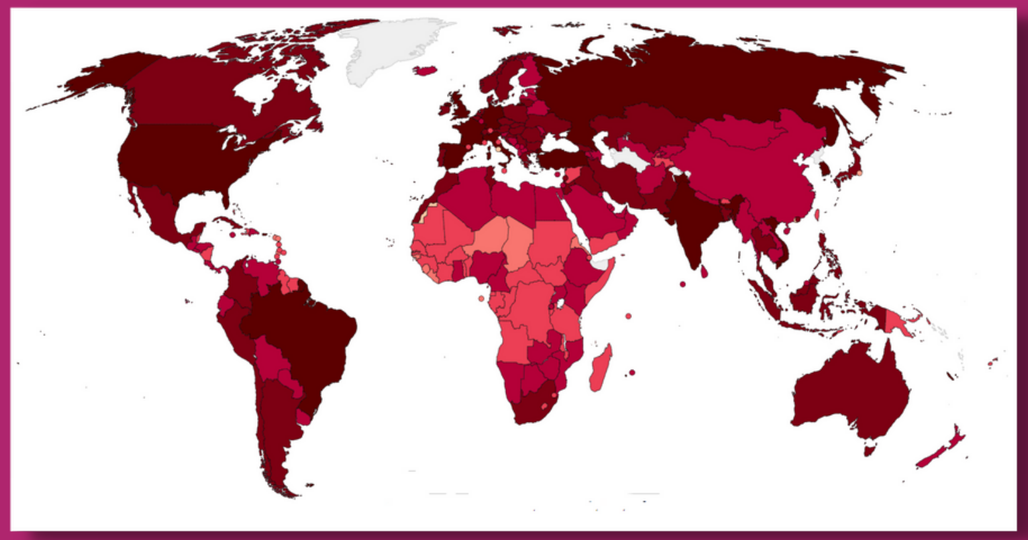
USING X RAY IMAGING

DOMAIN- HEALTHCARE / MACHINE LEARNING



# CORONA-VIRUS

- The novel coronavirus diseases 2019 has created a critical and urgent threat to global health.
- This pandemic continues to challenge medical systems worldwide, and have become the biggest threat to humanity in the recent times.



## Problem

Manual detection of COVID-19 diseases by the clinicians is  time consuming and complex

## Problem

The most validated diagnosis test (RT-PCR) has long been in shortage in developing countries.

....

# HEALTHCARE-MACHINE LEARNING-IMAGE PROCESSING

- AI methods (ML), coupled with biomedical analysis can play a crucial role during pandemics.
- The model is designed to meticulously classify the covid-19 patients using **IMAGE PROCESSING**.
- The model is able to detect COVID-19 diseases by analyzing chest X-RAYS images.
- The image processing technique is a rapid and finer approach.
- The model provides effective inspection just by analyzing an image.
- The model imparts the specific area in the images for better treatment

# PROJECT PIPELINE

- **PRE-PROCESSING**

Data preprocessing is the concept of changing the raw data into a clean data set.

- **FEATURE EXTRACTION**

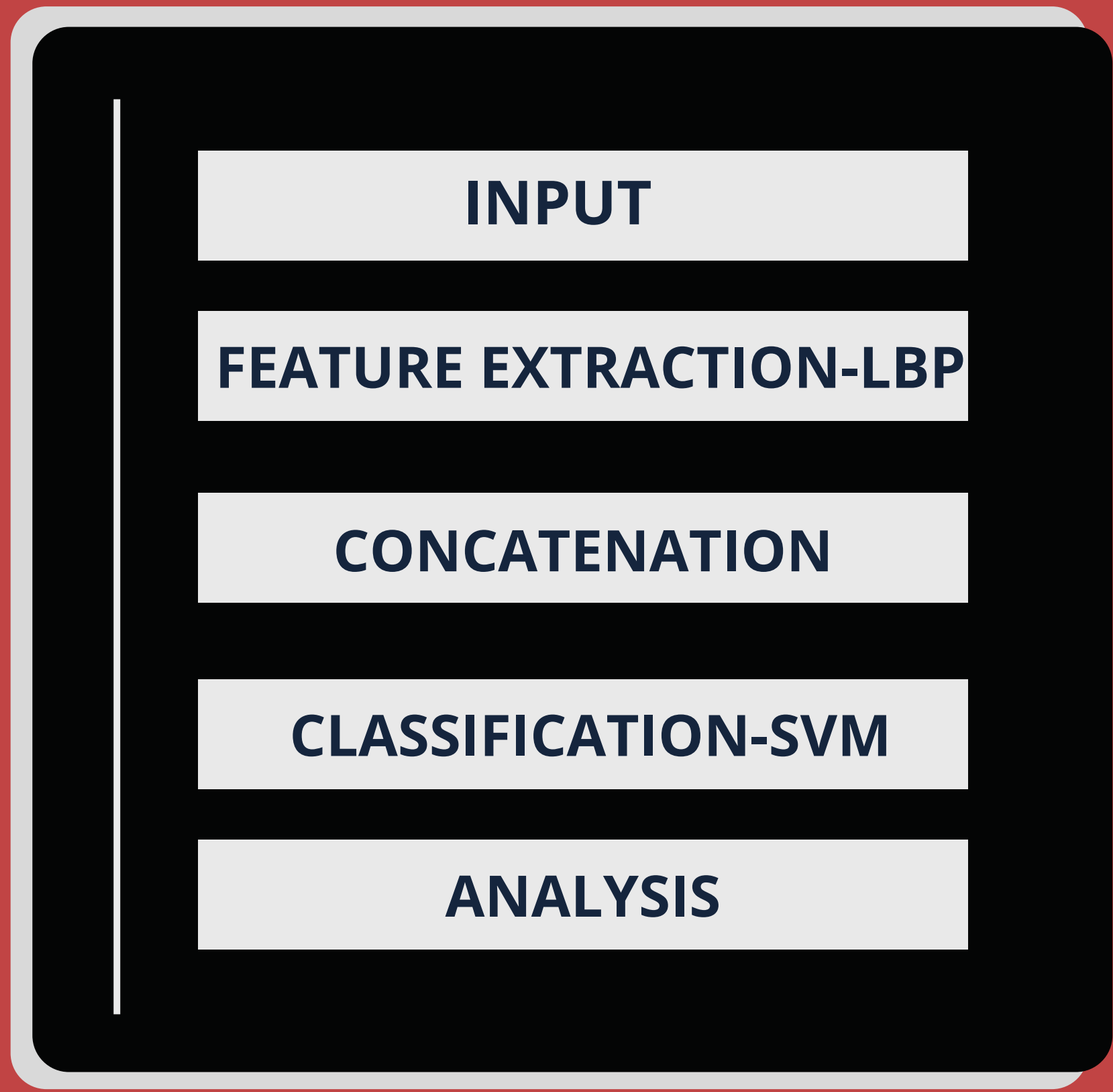
Local Binary Pattern (LBP) has been used for feature extraction from the pre-processed images
LBP is an efficient texture operator which thresholds the neighboring pixels based on the value of the current pixel.

- **FEATURE SELECTION**

Dominant features are then selected for further classification by getting rid of noise in data.

- **CLASSIFICATION**

Support Vector Machine(SVM) using the linear kernal has been used for the identification of new observations on the basis of trained data.



```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
%cd drive/
```

```
/content/drive
```

```
%cd MyDrive/
```

```
/content/drive/MyDrive
```

```
%cd COVID/Normal/
```

```
/content/drive/MyDrive/COVID/Normal
```

```
%cd ..
```

```
/content/drive/MyDrive/COVID
```

```
import glob
import os
from matplotlib import pyplot as plt
import numpy as np
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
```

```
%ls-l
```

```
total 8
drwx------ 2 root root 4096 Apr  9 12:49 Affected/
drwx------ 2 root root 4096 Apr  9 07:51 Normal/
```

```
normal_files = os.listdir('Normal/')
print(normal_files)
affected_files = os.listdir('Affected/')
print(affected_files)
```

```
%cd Normal/
```

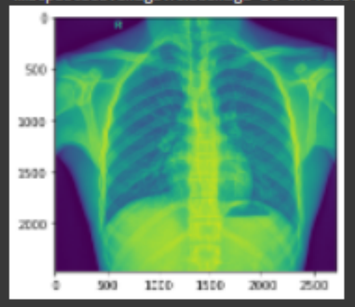
```
/content/drive/MyDrive/COVID/Normal
```

```
def extract_feat(img):
  out_img= np.zeros((img.shape[0], img.shape[1]))
  for i in range(1, img.shape[0]-1):
    for j in range(1, img.shape[1]-1):
      n=[0]*8
      cen= img[i, j]
      n[0]=img[i, j-1]>cen
      n[1]=img[i-1, j-1]>cen
      n[2]=img[i-1, j]>cen
      n[3]=img[i-1, j+1]>cen
      n[4]=img[i, j+1]>cen
      n[5]=img[i+1, j+1]>cen
      n[6]=img[i+1, j]>cen
      n[7]=img[i+1, j-1]>cen
      temp=0
      for k in range(len(n)):
        temp+=n[k]*(2**k)
      out_img[i, j]=temp
  return out_img
```

```
def final_feat(img):
  lbp_img=extract_feat(img)
  hist1=np.histogram(lbp_img, bins=256)
  return(hist1[0])
```

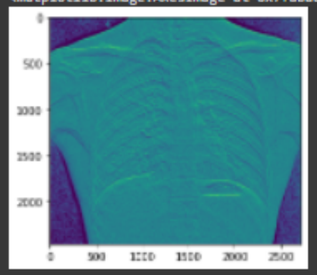
```
img_arr=plt.imread(normal_files[0])
print(img_arr)
plt.imshow(img_arr)
```

```
[[1 1 1 ... 8 7 2]
 [1 1 1 ... 8 7 2]
 [1 1 1 ... 8 6 2]
 ...
 [0 0 0 ... 5 4 2]
 [0 0 0 ... 5 3 0]
 [0 0 0 ... 5 3 0]]
<matplotlib.image.AxesImage at 0x7fdbaf313890>
```




```
plt.imshow(lbp_img)
```

```
<matplotlib.image.AxesImage at 0x7fdbaddd0d90>
```




```
hist1=np.histogram(lbp_img, bins=256)
print(hist1[0])
```

```
[975589 128618  43874  89249  96247   7584  78154 107698  44612  10892
  2335   4035  78543  10589 100482  94297 128769  11203  10816   6888
  7593    459  10552   3885  90780   6931   3992   5925 107821   3754
 93039  41113  45981  10974   6961  13217   8736    582  12785  14907
  2933   1219    286    684   4326   1114   8955  10854  90962   7044
 12915   8418   8532    435  16946   7418 119675   5191   6854   7908
 88818   2403 105760  43363  98698   7690   8408   8536   9526    471
  6085   3293   8587    579   1083    842   6093    451   5042   2541
  7629    501    596    393    518     38    434    131   8442    428
   806    894   3336    161   2516   1086  82872  11030  12977  16944
  6187    454   9460   8471   4376    968    661    784   6901   1166
 10236   9259 112634   4012  15264   7496   3342    146   8451   1784
 84629   2442   8626   7055  45876   1350  53089  15182  45114  90001
  2896 114523   8662   8579   4363  88202   7010  13188    253   6981
 12996  17280   8591 105968  11265   6872   1247   5212    558    427
  1051   2471  13307   8570    648   7963  15043   7224  10690  43396
  2595   4393    289   6997   1026    840    657   8758    275    682
    44    792    651    796    885   9628   4423   6479    665   8102
   902    934    763   7115   7035   8091    830  20836   8694   7034
  9303  74515  81782 110310   4288  82587   6229   3395   6918  45701
 13138  15126    670   8738   9295   8444   9943  52663  11109   3916
  1027   2491    505    149   1092   1381  17188   7370    789   6902
  8301   1765   9029  15065 108261 102072   9197 106817   5160   2654
 10427  53597   8928  10972    948   9284  10278   9281  23091  99931
105120  44604  11291  44656   2553   1221   9308  15392 109231  44721
  9506  75070  53483  15389 100464 216535]
```

```
hist1[0].shape
```

```
(256,)
```

```
%pwd
normal_features=np.zeros((1, 256))
for file in normal_files:
  img1=plt.imread(file)
  feat1=final_feat(img1)
  feat1=np.reshape(feat1, (1, feat1.shape[0]))
  normal_features=np.vstack((normal_features, feat1))
```

```
normal_features= normal_features[1:,:]
normal_features.shape
```

```
(20, 256)
```

```
affected_features= affected_features[1:,:]
affected_features.shape
```

```
(20, 256)
```

```
X=np.vstack((normal_features, affected_features))
X.shape
```

```
(40, 256)
```

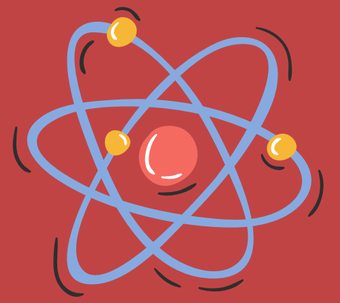
```
y=[0]*20 + [1]*20
y
```

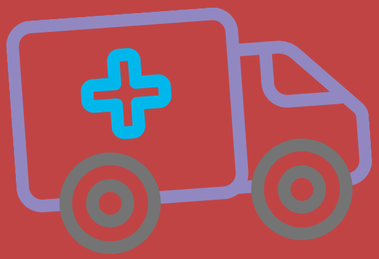
```
[0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1]
```

```
clf=SVC(C=0.04, kernel = "linear")
clf.fit(X,y)
y_pred = clf.predict(X)
acc = accuracy_score(y, y_pred)
tn, fp, fn, tp = confusion_matrix(y, y_pred).reshape(-1)
recall = tp/(tp+fn)
precision = tp/(tp+fp)
f1_score = (2*recall*precision)/(recall + precision)
```

```
acc
```

# PROJECT SPECIALITY





MODEL IS NOT RESTRICTED TO PARTICULAR VARIANT

MODEL CAN BE USED WORLDWIDE FOR MEDICAL TREATMENT AND RESEARCH

MODEL WILL ASSIST THE CLINISTS/DOCTORS AND MITIGATE THE BURDEN ON HEALTH CARE SYSTEM

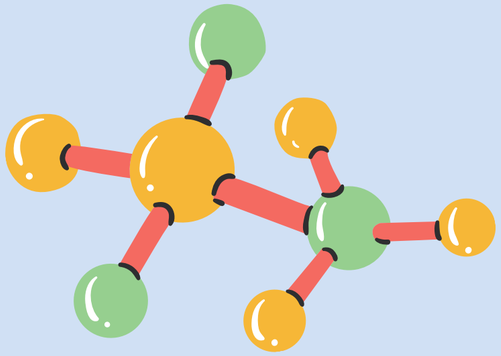
MODEL ABILITY INCREASES WITH QUANTITY OF DATA AND MAY LEAD TO BETTER PERFOMANCE

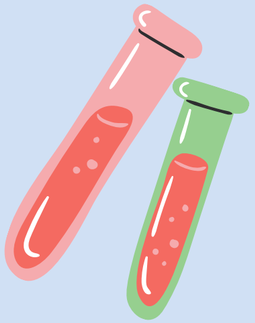


# FUTURE SCOPES

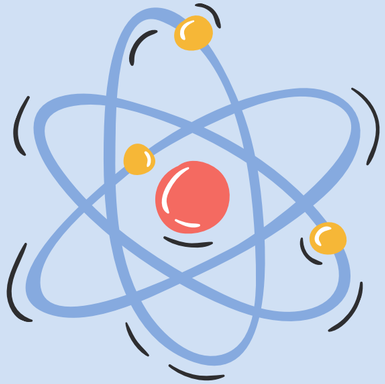
- BEST COMBINATIONS OF FEATURE EXTRACTOR AND CLASSIFIER  FOR GENERIC RESULTS
- CONFIGURE WITH DEEP LEARNING
- FEATURE OF VIRUS DETECTION
- PROCESSING WITH LARGE DATA
- WEB APPLICATION
- ANDROID / IOS APPLICATION

# WE STAND WITH EVERY FRONTLINE WORKER





# STAY SAFE




**MANAV RACHNA UNIVERSITY**

***TANNU SANGWAN***
***B.TECH CSE ,4th semester***
***ABHEER MEHROTRA***
***B.TECH CSE ,4th semester***