Computer-aided detection of Covid-19 from X-ray images.

A Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of
Bachelor of Technology
in
Information Technology

by
**Shivanshu Tripathi(20170050), Sourav Sarkar(20178017)
and Basant Kumar Pandit(20178083)**
Group: **IT-08**

to the
COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY
ALLAHABAD, PRAYAGRAJ
December,2020

UNDERTAKING

We declare that the work presented in this report titled "*Computer-aided detection of Covid-19 from X-ray images.*", submitted to the Computer Science and Engineering Department, Motilal Nehru National Institute of Technology Allahabad, Prayagraj, for the award of the ***Bachelor of Technology*** degree in ***Information Technology***, is our original work. We have not plagiarized or submitted the same work for the award of any other degree. In case this undertaking is found incorrect, We accept that our degree may be unconditionally withdrawn.

December,2020
Allahabad

(Shivanshu Tripathi)

(Sourav Sarkar)

(Basant Kumar Pandit)

CERTIFICATE

Certified that the work contained in the report titled "*Computer-aided detection of Covid-19 from X-ray images.*", by *Shivanshu Tripathi, Sourav Sarkar, Basant Kumar Pandit*, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Dr. Vibhav Prakash Singh)
Computer Science and Engineering Dept.
M.N.N.I.T. Allahabad

December,2020

Preface

The starting of the year 2020, introduced the world to a new novel Corona Virus named Covid-19 which soon became a pandemic. According to the official site of World Health Organization, the Covid-19 pandemic has already caused a major outbreak in over 150 countries, having drastic impact on the life and health of people worldwide. With more than 67.7 million confirmed cases and over 1.5 million fatalities at the time of writing this report, the Covid-19 has became a challenge for the Mankind.

Being a contagious disease, the most important step to keep a check on the spread of the virus is to detect it in a person as soon as possible. Current medical methods which detect the virus with a good accuracy but are time taking. Since the Covid-19 virus affects the respiratory system, we had the idea to detect the virus using the X-ray images of the chest. To do this we used a dataset made publicly available by the medical professional and then we are using Computer Vision, Machine Learning, and Deep Learning techniques to detect Covid-19 from the chest X-ray of a patient.

To have clarity in our discussion about detection and classification of Covid-19, we have taken the help of various published research works from reputed journals.

Acknowledgments

We would like to thank everyone who contributed for the completion of our project. We are especially grateful to our mentor Dr. Vibhav Prakash Singh for assisting us in our project. Without his expertise and suggestions we would never have been able to complete our work.

We would also like to extend our gratitude to Dr. Shashwati Banerjea, Assistant Professor of Computer Science and Engineering Department, for the formation of groups for the project which enabled us to work on this project.

Abstract

One of the critical step in reducing the spread of Covid-19 is the ability to detect the patient infected by it as soon as possible and isolate them. Maybe one of the best ways to diagnose patients is to identify this disease through radiographic and radiological photos. Any of the early studies had shown clear anomalies in patients infected with COVID-19 in the chest radiographs. We research the application of deep learning models to detect COVID-19 patients from their chest radiography images, inspired by earlier works.

In this proposed project, we used the dataset made publicly available by the medical professionals which included the chest X-ray images of various people belonging to categories like normal X-ray, X-ray of people affected by Covid-19, X-ray of people affected by pneumonia from bacteria and virus. In all the dataset consisted of 5910 images of chest X-ray.Firstly we tried to classify and detect Covid-19 using Machine Learning Techniques like SVM and Random Forest by using texture based feature extraction techniques like LBP and GLCM as input and achieved an accuracy of 93%. Then to improve the accuracy we applied Image Enhancement techniques like Contrast Limited Adaptive Histogram Equalization (CLAHE) to improve the contrast in X-ray images and then we used Deep Learning Techniques namely Convolutional Neural Networks (CNNs) and achieved and accuracy of 96% in detecting Covid-19.

Contents

Chapter 1

Introduction

The World Health Organization (WHO) has announced in recent months that a new virus called COVID-19 has spread rapidly in many countries worldwide[1]. Rapid COVID-19 identification can help to monitor the spread of the disease.Usually, both pneumonia symptoms and chest X-ray scans are associated with the diagnosis of COVID-19. The chest X-ray is the first imaging technique that plays an important role in the diagnosis of COVID-19 disease.

Early diagnosis is of real significance because of the unavailability of clinical medication or vaccination for novel COVID-19 disease, to provide an opportunity for rapid isolation of the suspected individual and to minimise the risk of infection for a healthy population. The key screening methods for COVID-1919 are reverse transcription polymerase chain reaction (RT-PCR) or gene sequencing for respiratory or blood specimens.[32].However it is estimated that the average positive RT-PCR rate for throat swab samples is 30 to 60 percent, resulting in undiagnosed patients who can infect a large population of healthy people[33]. With rapid diagnosis, chest radiography imaging (e.g., X-ray or computed tomography (CT) imaging) is easy to conduct as a standard method for diagnosing pneumonia. Chest CT has a high resistance to COVID-19 diagnosis[9].Ground-glass (57 percent) and mixed attenuation (29 percent) are the most widely recorded opacities. The pattern of ground glass is seen in areas that surround the lung vessels during the early course of COVID-19 and can be difficult to visually appreciate[17].

For COVID-19, asymmetric patchy or diffuse airspace opacities are also reported

(Rodrigues, 2020). Only specialist radiologists may interpret such subtle anomalies. Automatic methods for detecting such subtle anomalies will help the diagnostic process and increase the rate of early detection with high precision, given the enormous rate of suspicious individuals and the small number of qualified radiologists. Fig. 1 shows image of the normal x-ray of person chest, a COVID-19 positive person's X-ray , and one of acute respiratory syndrome (SARS).

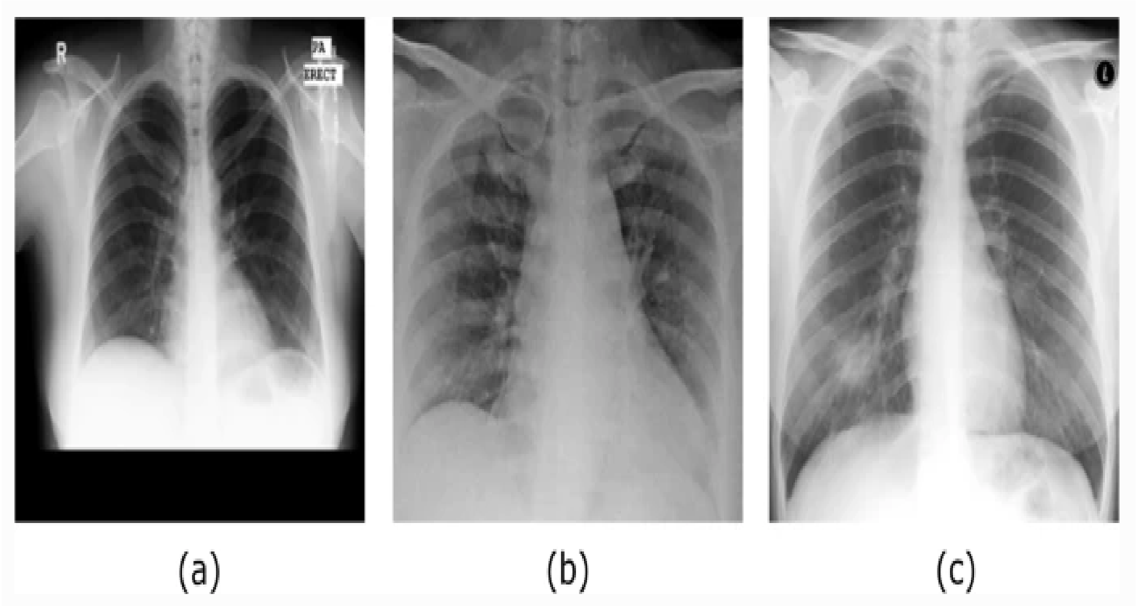


Figure 1: Examples of a) normal, b) COVID-19, and c) SARS chest x-ray images.

.

Firstly we processed the dataset of images and made labels according to the disease and then we applied the image processing technique Contrast Limited Adaptive Histogram Equalization (CLAHE) for increasing the contrast of images and reducing noise.

Then from the dataset of given images, features were extracted using the texture-based feature extraction techniques like Local binary Patterns (LBP), Grey-Level Co-Occurrence Matrix (GLCM) features, and first-order statistics. Then all seven possible combinations of the feature vector were tested by splitting the dataset between the training set and testing set with 80% data in the training set. Several classification models were trained with the testing set, and then for validation purposes, the testing set was given to the trained model and the confidence estimation of that classifier was calculated using metrics like accuracy from Confusion Matrix.

After we shifted to applying Deep Learning Techniques for classification using Convolutional Neural Networks (CNNs). We build our own custom CNN model and trained the model using 80% of dataset.

1.1 Motivation

The idea of this project originated from the interest of computer-based medical diagnosis. In this project, the idea was to develop a program that could detect and classify Covid-19 disease using the X-ray images of patients chest with high accuracy. Medical diagnosis is an interesting field with a broadened scope due to advancements in computer vision, machine learning and deep learning technology.

This project will assist in detecting Covid-19 in minimal time from the chest X-ray and therefore could control its spread. It is a small step towards helping the medical fraternity by using computer vision and computer science applications in the medical field.Because all the techniques for the Covid-19 test take time and they are not 100% accurate so having a second opinion by taking the chest X-ray and processing it through our tool which takes very little time will make sure whether a patient has Covid-19 or not.

1.2 Objective

The main objective and the motive of this project is to explore the utilization of computer vision, texture-based feature extraction techniques and then using state of the art machine learning algorithms and deep learning techniques to detect Covid-19 from images of chest X-rays.

1.3 What is Coronavirus Disease (Covid-19)

Coronavirus disease 2019 (COVID-19) is a contagious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2).In December 2019, in Wuhan, China, the first case was found. Since then it has spread globally, resulting in an ongoing pandemic.

Symptoms of COVID-19 are variable, but often include fever, cough, fatigue, breathing difficulties, and loss of smell and taste. Symptoms begin one to fourteen days after exposure to the virus,. There are no signs for around one in five infected individuals. While most people have mild symptoms, some people have acute syndrome of respiratory distress (ARDS). ARDS can be precipitated by cytokine storms, multi-organ failure, septic shock and blood clots. . Longer-term damage to the organs (lungs and heart in particular) has been noted.The COVID-19 virus spreads primarily when an infected person is in close contact with another person. The virus produces tiny droplets and aerosols that can spread from the nose and mouth of an infected person when they breathe, cough, sneeze, sing, or talk.Other individuals are infected if the virus gets into their mouth, nose or eyes. The virus which also spread by fomites (contaminated surfaces), although the primary transmission mechanism is not known to be this..

Chest CT scans in people with a high clinical suspicion of infection can be helpful in diagnosing COVID-19, but routine screening is not recommended.. In early infection with peripheral, asymmetric, and posterior distribution, bilateral multilobar ground-glass opacities are common. Mad paving (lobular septal thickening with variable alveolar filling) subpleural dominance. Asymmetric ground-glass peripheral opacities without pleural effusions provide characteristic image features of symptomatic individuals on chest radiographs and computed tomography (CT). Below image show the CT scan of patient affected by Covid-19.

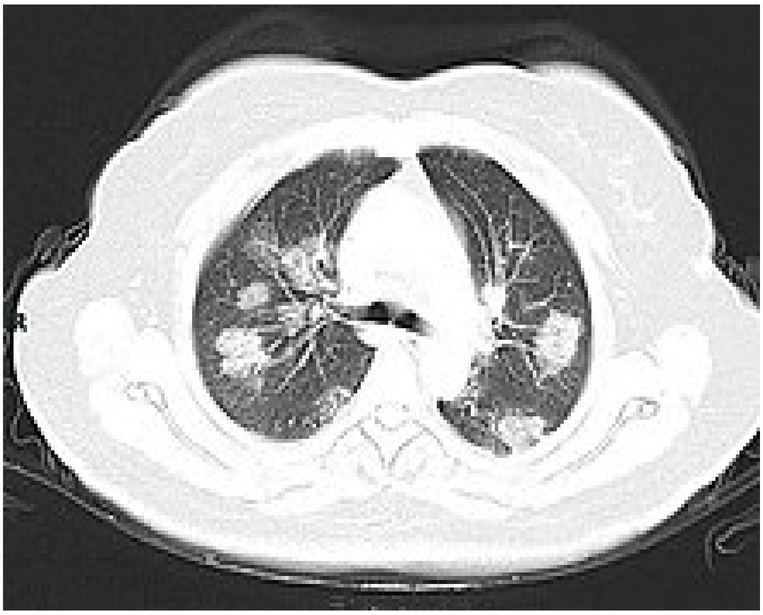


Figure 2: A CT scan reveals lesions (bright regions) in the lungs of a person with COVID-19..

.

Chapter 2

Related Work

For automated classification of digitised chest files, many classical machine learning methods have previously been used. [11][21]. For example, in [25]using a Support Vector Machine SVM classifier, three statistical features were calculated from the lung texture to differentiate between malignant and benign lung nodules.With Backpropagation Network, a grey-level co-occurrence matrix approach was used to identify images as regular or cancerous[29].Deep learning approaches have demonstrated their dominance over the classic approaches to machine learning with the availability of ample annotated images[29][8][12].

One of the most common deep learning approaches with superior achievements in the medical imaging domain is CNN architecture[31]. CNN's primary success is attributed to its ability to automatically learn features from domain-specific images, unlike the classical methods of machine learning. Fine-tuning mechanism transfer learing demonstrated outstanding results in chest X-ray image classification.[16][30]. Detection of pneumonia from chest X-ray using Deep Learning[10].Chest X-ray Analysis using Machine Intelligence Research for HIV/TB Screening[15].Some preliminary work has also been done in detecting Covid-19 from X-ray images using Deep Learning[6] and Transfer Learming[26].

Chapter 3

Proposed Work

3.1 Overview of Our Work

Our work is related to use of computer vision, machine learing and deep learning techniques in detection of Covid-19 from X-ray images.We have used python libraries namely NumPy[3], Scikit-Learn[5] for Machine Learning Models, Scikit-Image[4] for Image processing,and Keras[2] for Deep Learning .We used Kaggle Kernels[1] for simulating our code.

The input dataset of images is first analysed.Firstly from the dataset of given images, To improve the contrast of images, Contrast Limited Adaptive Histogram Equalization(CLAHE) and noise reduction was done using Adaptive Median Filters, then features were extracted using the feature extraction techniques Local Binary Patterns (LBP), the Grey-Level Co-OccurrenceMatrix (GLCM) and the first order statistics.Then all seven possible combinations of the feature vector were tested by splitting the dataset between the training set and validation set with 70% data in the training set. Before classification, the feature vectors are preprocessed and normalized. In particular, the feature vectors are preprocessed scaling to unit variance and eliminating the mean(centering).

For classification of tissue we applied SVM on on trainig set of all seven combinations of feature vectors extracted, thus using hybrid features.SVM is chosen because it helps us to defeat the scourge of dimensionality that occurs when examining our high-dimensional feature vector. For verification and comparison, the performance of different classifiers, for example, Naive Bayes (NB), k-closest neighbors (kNN), and Random Forest (RF), are additionally examined. For validation purposes, the testing set was given to the trained model and the confidence estimation of that classifier was calculated using the metrics like precision, recall,and accuracy from Confusion Matrix. By this we achieved an accuracy of 93% when we used feature vector combination of LBP and GLCM and Random Forest as classification algorithm.

With no scope of improvement in Machine Learning we switched to Deep Learning and build our own custom Convolutional Neural Networks model (CNN) as they are used extensively in medical images classification and trained the model on our dataset and achieved an accuracy of 96% and compared the result obtained from Deep Learning Techniques and Machine Learning Techniques.

3.2 Data Analysis And Pre-processing

In this project, we have used the dataset of chest X-ray images which was made available to us by the medical fraternity.The dataset consists of total 5910 images of chest X-rays belonging to several category like an X-ray of healthy person,person having Covid-19 , person suffering from pneumonia and person suffering from SARS.Below image gives example of chest -Xray of person having Covid-19 and the visualization of our dataset.

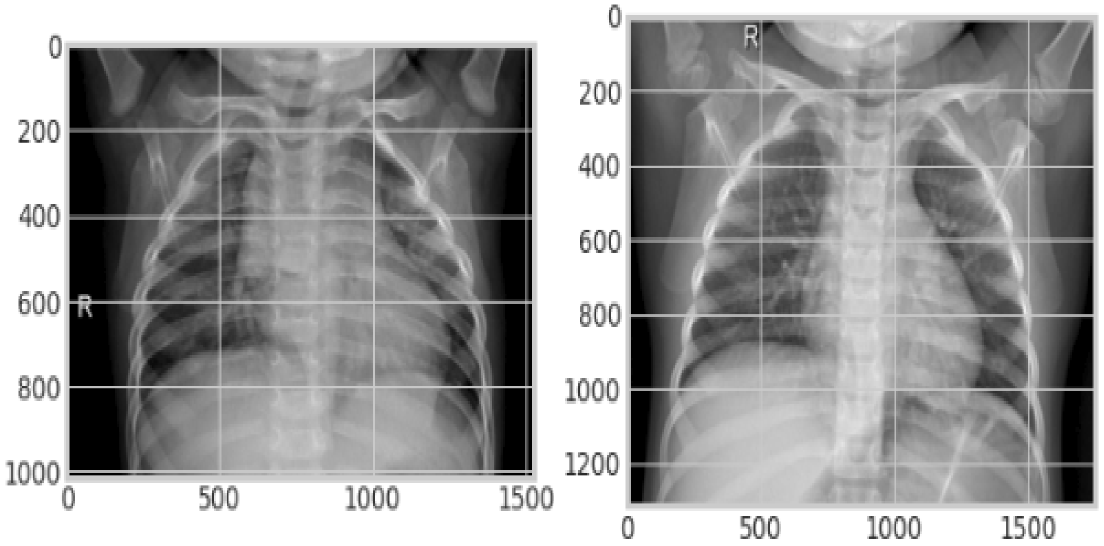


Figure 3: Chest X-ray of Covid-19 patient

.

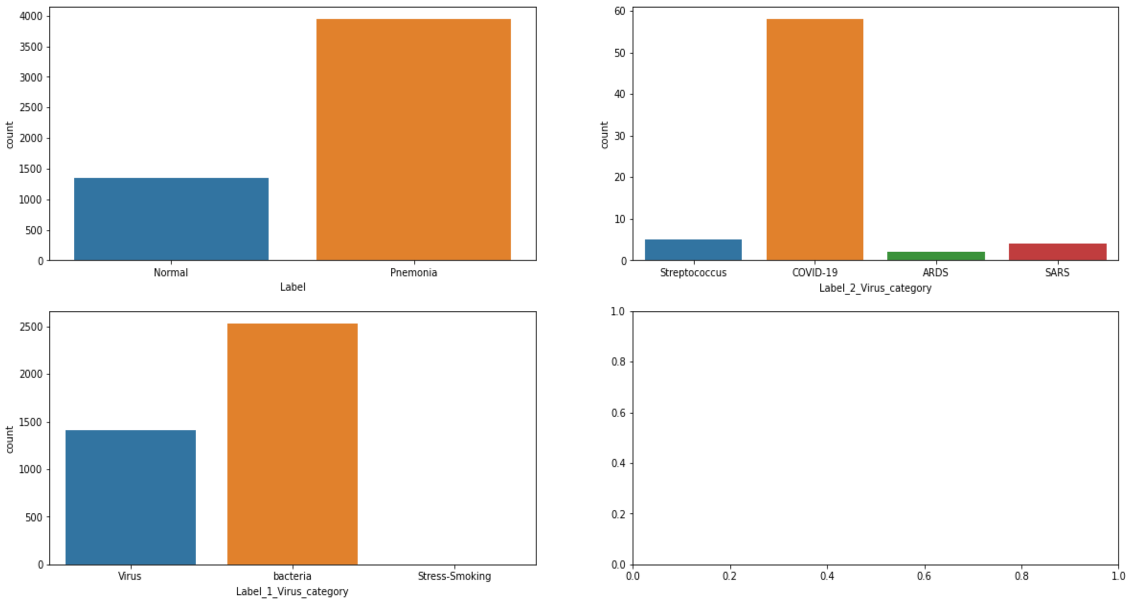


Figure 4: No. of images belonging to different class

.

After that we analysed the histogram of different classes of images to see the chances of enhancement in the images and to choose best image processing algorithm to enhance our images before we proceed further, below images show the histogram of some samples of images belonging to healthy class and Covid-19 class.

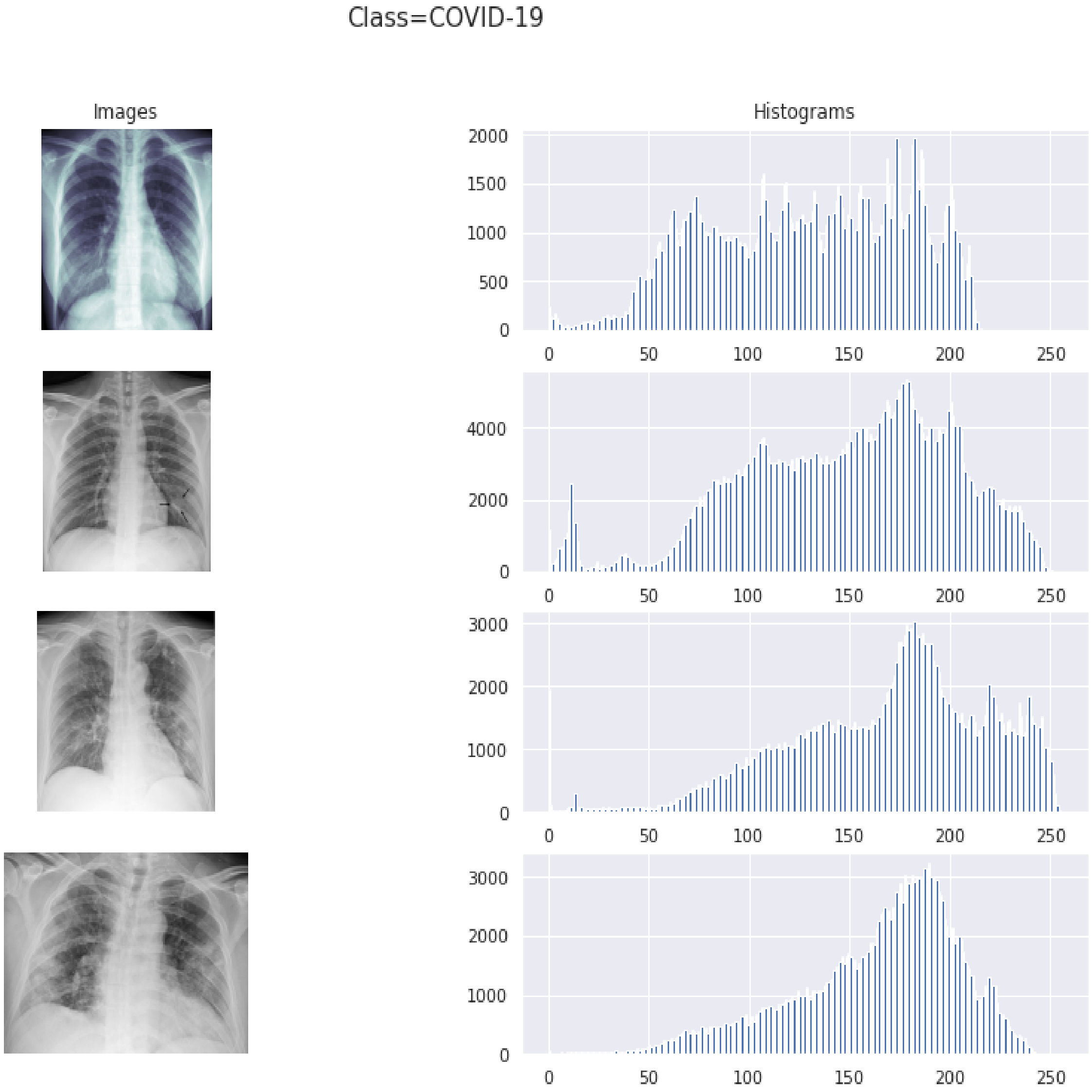


Figure 5: Histogram plots of Chest X-ray of Person havng Covid-19

.

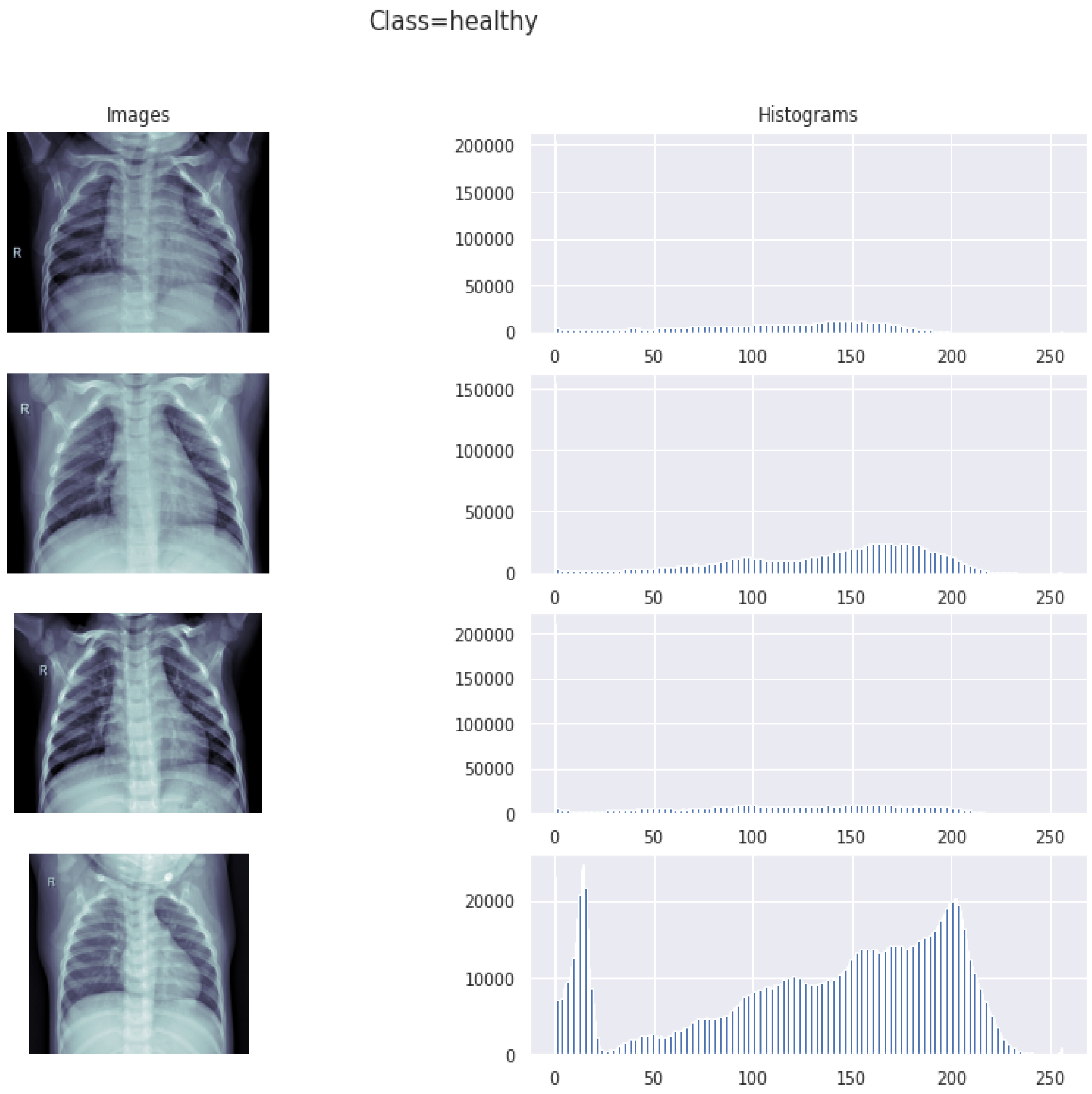


Figure 6: Histogram plots of Chest X-ray of Healthy Person

.

To improve the contrast of our images, we then applied Contrast Limited Adaptive Histogram Equalization before further processing and applied Adaptive Median Filters to reduce noise in the images.

3.3 Feature Extraction

Since we deal with Chest X-ray gray-scale images and the asymmetric ground-glass peripheral opacities without pleural effusions provide characteristic image features on chest radiographs and computed tomography (CT) of individuals who are symptomatic and have Covid-19, we have extracted texture-based features for our project.. In this project we have used three feature extraction techniques Local Binary Pattern(LBP), the Grey-Level Co-Occurrence Matrix (GLCM) and the first-order statistics:

3.3.1 Local Binary Patterns (LBP)

Out of many texture-based global descriptors, LBP[27] is generally considered as revolutionary for the examination of textures of images related to the medical field. LBP gives low-complexity and is invariant of grey-scale, well coordinating the pre-requisites of this project. The first articulation of LBP (LBPR, P) presented in the writing requires characterizing, for a pixel $c = (c_x, c_y)$, a spatial roundabout neighborhood of span R with P similarly divided neighbor points $(p_n, n\epsilon(0, p-1)))$:

$$LBP_{R,P}\,(c) = \sum_{n=0}^{P-1} s(g_{pn} - g_c)2^n$$

where g_{pn} and g_c signify the pixel c's gray value and of its n'th neighbor p_n's gray value, respectively, and s is described as follows,

$$s(g_{p_n} - g_c) = \begin{cases} 1 & , g_{p_n} >= g_c \\ 0 & , g_{p_n} < g_c \end{cases}$$

The most frequently used LBP variation is the uniform rotation-invariant $(LBP_{R,P}^{riu2})$[13] Because the endoscope posture is continuously changing during the examination of the larynx, the uniform rotation-invariant of LBP is appropriate for his project. From $LBP_{R,P}^{riu2}$ the computed histogram $(H_{LBP_{riu2}})$ is normalized using the L2 norm, and the resultant is used as the feature set extracted using LBP.

For the features extracted using LBP, Total nine combinations of the $LBP_{R,P}^{riu2}$ were computed using all possible combinations of (R;P), with $R\epsilon\,\{1, 2, 3\}$ and $P\epsilon\,\{8, 16, 24\}$

and the corresponding $H_{LBP_{riu2}}$ sets from these nine combinations were concatenated to form the final feature vector of $H_{LBP_{riu2}}$. This was done to give a progressively precise elucidation of the texture of images.

Local Binary Pattern Code Snippet

1. Extracting features using LBP

```
lbp=feature.local_binary_pattern(gray,numPoints,radius,method="uniform")
(hist,_)=np.histogram(lbp.ravel(),bins=np.arange(0,numPoints+3),range=(0,numPoints+2))
```

2. Normalization

```
 feature_lbp=preprocessing.normalize(feature_lbp,norm="l2")
```

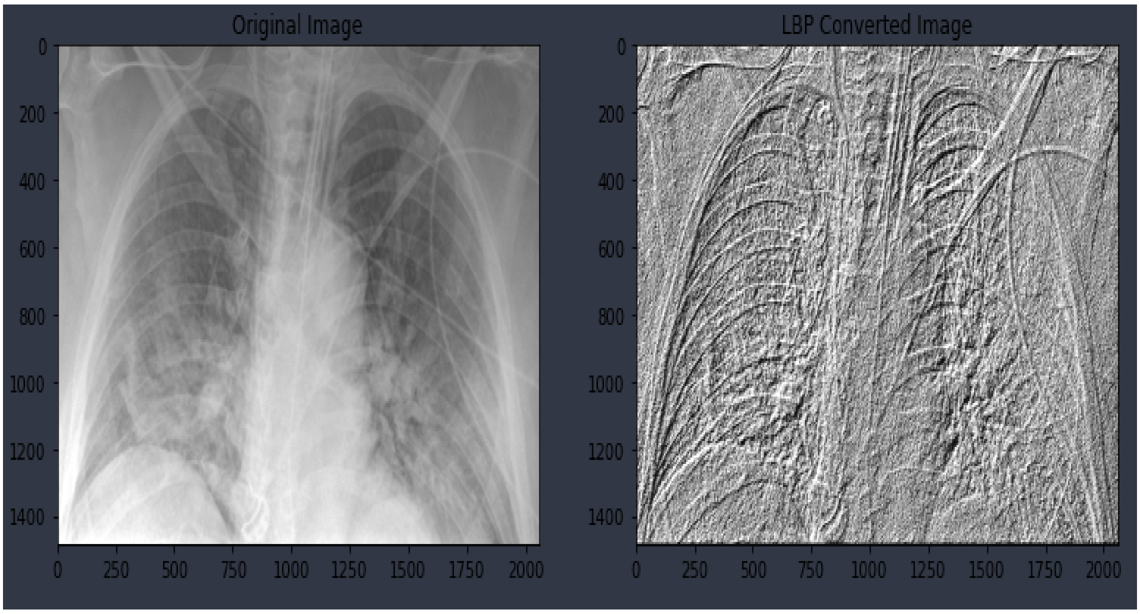


Figure 7: X-ray Image Before And After Applying LBP

.

3.3.2 Grey-Level Co-Occurrence Matrix (GLCM)

For comparison, the GLCM[18], a second broadly utilized descriptor, is tried. GLCM computes how many times a pair of pixels (c,q) with definite values and in a predefined spatial relationship happen in an image. The spatial relationship is determined by and d, which are the distance and the angle between c and q. The GLCM height (H) equal to the GLCM width (W), correlates with the intensity of grey-levels of the number of quantized images. For width(w) is equal to height(h) intensity grey-levels, the GLCM calculated with values of and d is described as follows,

$$GLCM_{\Theta,d}(h,w) = \begin{cases} 1 & , I(c) = h \wedge I[c_x + d \cdot \cos(\Theta), c_y + d \cdot \sin(\Theta)] = w \\ 1 & , I(c) = h \wedge I[c_x - d \cdot \cos(\Theta), c_y - d \cdot \sin(\Theta)] = w \\ 0 & , otherwise \end{cases}$$

From the normalized $GLCM_{\Theta,d}$, as suggested in Ref.[18], a feature set (F_{GLCM}) is extracted, which consists of GLCM contrast, correlation, energy, and homogeneity. The normalized $GLCM_{\Theta,d}$,It is obtained by dividing each $GLCM_{\Theta,d}$ entry by the number of all entries, which expresses the likelihood of gray-level events.

Total twelve combinations of $GLCM_{\Theta,d}$ was calculated using every conceivable mix of (,d), with $d\epsilon\,\{1,2,3\}$ and $\Theta\epsilon\,(0,\pi/4,\pi 2,3\pi/4)$, From these twelve variations, the corresponding ($FGLCM$) sets were concatenated to form the final feature vector of ($FGLCM$).

Grey-Level Co-Occurrence Matrix Code Snippet

1. Setting range of parameters and properties

```
ANGLES=[0.,np.pi/4.,np.pi/2.,3.*np.pi/4.]
DISTANCES=[1,2,3]
properties=["correlation","contrast","homogeneity","energy"]
```

2. Creating GLCM Matrix

```
 glcm=greycomatrix(gray,distances=[rad],angles=[thetha],levels=None,symmetric=True,normed=True)
```

3. Extracting Properties From GLCM Matrix

```
lf.append(greycoprops(glcm,k)[0,0])
```

4. Normalization

```
feature_glcm=preprocessing.normalize(feature_glcm,norm="l1")
```

3.3.3 First-Order Statistics

Variance, Intensity mean, and Entropy in each image patch is calculated and concatenated to give a feature set constructed from the intensity-based feature. The entropy is described as follows,

$$entropy = -\sum_{i=0}^{i=255} h_i \log_2(h_i)$$

where h_i alludes to the counts of histogram of the i(=0255) bin of the image.All these were concatenated to form feature vector extracted from first-order statistics (Stat1).

First-Order Statistics Code Snippet

1. Extracting Features like mean, variance and entropy

```
(hist,_)=np.histogram(gray.ravel(),bins=np.arange(0,256),range=(0,256))
lf.append(hist)
lf=np.array(preprocessing.normalize(lf,norm="l1"))
feature.append(np.mean(lf))#Intensity Mean
feature.append(np.var(lf))#Variance
feature.append(measure.shannon_entropy(gray))#Entropy
```

3.3.4 Hybrid Features

Besides these descriptors, we tested all possible combinations of feature vectors constructed from combining one or more of the feature vectors obtained from $H_{LBP_{riu2}}$,F_{GLCM} , Stat1. Thus, giving us the hybrid feature vectors namely (($H_{LBP_{riu2}}$) +Stat1), ((F_{GLCM}) +Stat1), (($H_{LBP_{riu2}}$) + (F_{GLCM})), and (($H_{LBP_{riu2}}$) + (F_{GLCM}) + stat1).

3.4 Machine Learning Techniques for Classification

3.4.1 Support Vector Machine (SVM)

Recently, A modern theory of statistical learning, viz.. For classification, the Support Vector Machine (SVM) has received growing attention. SVM was developed in 1995 at the AT&T Bell Laboratories by Vapnik and his colleagues. SVMs is initially designed to solve problems with pattern classification, such as optimal recognition of characters, face identification and text classification, etc. But soon, in other domains, such as function approximation, regression estimation, they find wide applications.

SVMs[23] Centered on the structural risk minimization principle, they are non-linear and used for classification, regression, and time-series prediction. Using a kernel function, SVMs map input data into a high-dimensional space through non-linear mapping, and then perform a linear regression of this space. The key benefit of SVM is that it is always special and globally optimal for the solution obtained.

It is a type of supervised learning method that analyzes the data and arranges it into one of the given categories. The algorithm creates a hyperplane or a line separating the data into different classes. A hyper plane in an n-D feature space can be represented by the following equation:

$$f(\mathbf{x}) = \mathbf{x}^T\mathbf{w} + b = \sum_{i=1}^{n} x_i w_i + b = 0$$

Dividing by $||\mathbf{w}||$, we get

$$\frac{\mathbf{x}^T\mathbf{w}}{||\mathbf{w}||} = P_{\mathbf{w}}(\mathbf{x}) = -\frac{b}{||\mathbf{w}||}$$

indicating that the projection of any point $\mathbf{x}$ on the plane onto the vector $\mathbf{w}$ is always $-b/||\mathbf{w}||$, i.e., $\mathbf{w}$ is the normal direction of the plane, and $|b|/||\mathbf{w}||$ is the distance from the origin to the plane.

As per the SVM[19] algorithm, it finds the points nearest to the line from all the classes. These points are known as support vectors. Then, it computes the distance between the support vectors and the line. This distance is known as the margin.

The objective of the algorithm is the maximization of the margin. The optimal hyperplane or the optimal line is the one whose margin is maximum. The algorithm tries to create a decision boundary in a way so that the partition between classes is as wide as possible.

In this work we have used the SVM[24] classification method to classify laryngeal tissues.SVM is picked since they permit defeating the curse of dimensionality that emerges while investigating our high-dimensional feature vector.

Kernel functions:

The kernel function converts the nonlinear input space into a high-dimensional space for features. The issue can be solved as a linear issue in this space. The following are some of the most famous kernel functions:

RBF kernel: TThe Gaussian kernel or radial basis function (RBF) is defined as:

$$K(x, x') = e^{-\gamma||xx'||^2}$$

where $||xx'||^2$ The Euclidean distance between the function vectors x and x' R is a, and $\gamma\epsilon$ Parameter-defined by the consumer.

The grid-search space for C and Y was set to $[10^{-3}, 10^{3}]$ and $[10^{-7}, 10^{-1}]$, respectively, with six values divided uniformly on a log10 scale in both cases. The kernel used is “Gaussian” kernel.

Support Vector Machine Model Code Snippet

1. Applying SVM for C=1000, gamma = 1, random_state=0

```
from sklearn.svm import SVC
classifier = SVC(C=10000,kernel='rbf',gamma=1,random_state = 0)
```

2. Fitting model on training dataset

```
 classifier.fit(x_train,y_train)
```

3. Predicting value for test dataset

```
 y_pred=classifier.predict(x_test)
```

4. Generating classification report and confusion matrix

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
cm=confusion_matrix(y_test,y_pred)
```

3.4.2 K-Nearest Neighbours (kNN)

KNN[20] is a non-parametric, slow learning calculation. The motivation behind this algorithm is to utilize a record wherein the data points are dispersed into a few classes to anticipate the classification of another sample point. The KNN calculation presumes that similar things exist in close vicinity. As it is often seen, similar things are-near to each other.

It captures the principle of resemblance (sometimes referred to as distance, proximity, or closeness) with Any measurement of the distance on a graph between points. However, the straight-line distance (also known as the Euclidean distance) is a familiar and the most frequently used option. After the distance is calculated, it selects K nearest points and classifies the new data point according to the majority class in the set of K nearest points.

The number of neighbors for kNN with a grid-search space set to [2,10] with nine values divided uniformly. The metric used is "Minkowski" and the value of "p" is set to 2 in order to use the "Euclidean" distance to measure the distance between the neighboring points in the dataset.

K-Nearest Neighbours Model Code Snippet

1. Applying kNN for n_neighbors=[2,10], metric = 'minkowski', p=2

```
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=k,metric = '
    minkowski',p=2)
```

2. Fitting model on training dataset

```
 classifier.fit(x_train,y_train)
```

3. Predicting value for test dataset

```
 y_pred=classifier.predict(x_test)
```

3.4.3 Random Forest

Random forest[22] or random decision forests is an ensemble learning technique for classification that works by building multiple decision trees at the time of training the model and yielding the class that is the mode of the classes.

Random forest, as the name suggests, comprises a large number of individual decision trees that function as an ensemble. Each individual decision tree in the random forest brings about a class prediction and the class which receives the maximum votes becomes the model's prediction. The reason that the random forest algorithm performs so well is that a large number of relatively uncorrelated models (trees) working as a team outperforms any of the independent constituent decision tree models.

The number of trees in the forest for the Random Forest algorithm with a grid-search space set to [40,100] with six values divided uniformly. The criterion used is "Entropy".

Random Forest Model Code Snippet

1. Applying Random Forest for criterion = ’entropy’ and random_state=0

```
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators=(k*10),
    criterion = ’entropy’,random_state=0)
```

2. Fitting model on training dataset

```
 classifier.fit(c_train,d_train)
```

3. Predicting value for test dataset

```
 d_pred=classifier.predict(c_test)
```

3.4.4 Naïve Bayes

A naive Bayes[14] classifier is a machine learning algorithm that uses Bayes' theorem to categorize objects. Naïve Bayes classifiers presume naive, or strong, self-sufficiency between properties of data points. Medical diagnosis, spam filters, and text analysis are well-known uses of the Naive Bayes classification algorithm. It is a probabilistic machine learning model that is used for the task of classification. The essence of the classifier is built on the Bayes theorem.

Bayes Theorem :

Using Bayes theorem, the Naïve Bayes algorithm can find the likelihood of A occurring, considering that B has already happened. Here, A is the hypothesis, and B is the evidence. The assumption made here is that the characteristics do not depend on each other, i.e. existence, The other is not affected by one particular feature. That is why this algorithm is also called naïve. The Bayes Theorem helps us to assess the likelihood of an incident when It is possible to split the universe into two or more disjointed sections.

$$\Pr[E] = \sum_{i=1}^{n} \Pr[E|A_i] \Pr[A_i]$$

Frequently this formula is also known as the *law of total probability.* The law of total probability states that the likelihood of an event E is a weighted average of the conditional probability of E provided that event A_i has happened over all the total possibilities of A_i. TWhen it is hard to calculate $\Pr[E]$ directly, this formula can be useful but it can be computed with additional information about A_i. If $A_1, A_2, \ldots A_n$ form a partition of the sample space and E is an event of the sample space then *Bayes Theorem* says

$$\Pr[A_i|E] = \frac{\Pr[E|A_i] \Pr[A_i]}{\sum_{i=1}^{n} \Pr[E|A_i] \Pr[A_i]}$$

Naïve Bayes Model Code Snippet

1. Applying Naïve Bayes Classifier

```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
```

2. Fitting model on training dataset

```
 classifier.fit(np.array(x_train).astype(np.float),np.array(
   y_train).astype(np.float))
```

3. Predicting value for test dataset

```
  y_pred=classifier.predict(np.array(x_test).astype(np.float))
```

4. Generating classification report and confusion matrix

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
print(classification_report(np.array(y_test).astype(np.float),
    y_pred))
cm=confusion_matrix(np.array(y_test).astype(np.float),y_pred)
```

3.5 Deep Learning Techniques

We moved towards Deep Learning from Machine Learning techniques because we weren't seeing any scope of improvement in our Machine Learning approaches and after researching and reading several papers based on application of CNN on medical images we tried building our CNN model for our problem.

3.5.1 Convolution Neural Network (CNN)

A Convolutionary Neural Network (C-NN) [28]is a type of Deep Learning algorithm that takes the image as an input, assigns different aspects of the image to weights and biases, and gains the ability to distinguish one from another. It is a network architecture that learns from data directly and removes the need for manual extraction of functions.CNN models transfer input data through a sequence of philtre convolution layers (Kernels), pooling, completely linked layers and applying Softmax to classify an object with probabilistic values between 0 and 1. The following figure is a full CNN flow for processing an input image and classifying the objects based on values.

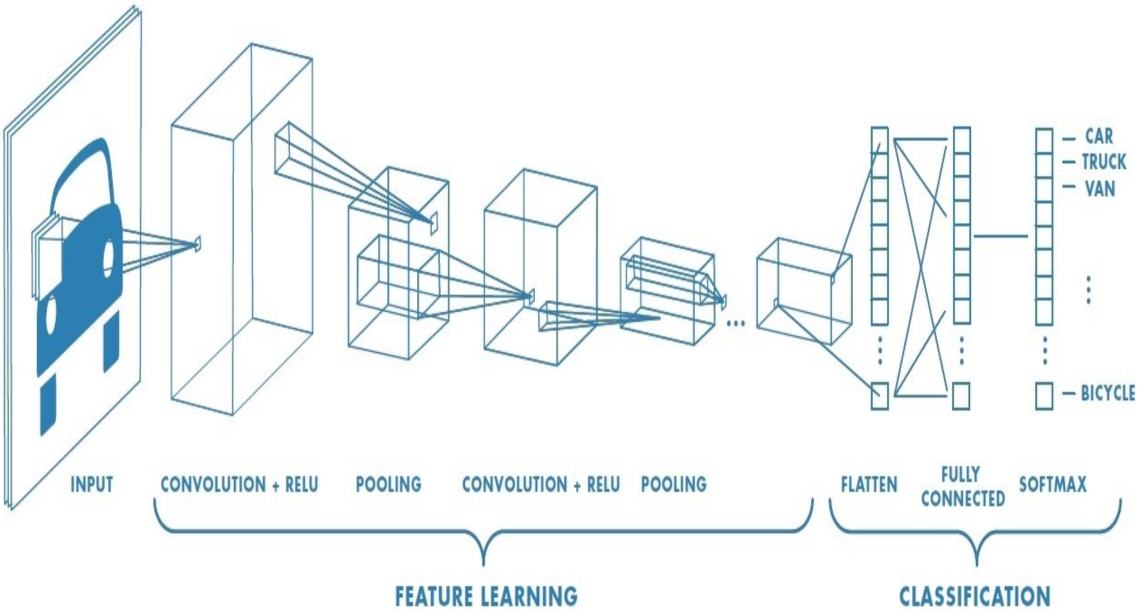


Figure 8: Neural network with several layers of convolution

.

Why we used C-NN?

- CNNs remove the need for manual extraction of features-the CNN immediately discovers the characteristics.
- CNNs generate results of recognition that are highly accurate.
- For new recognition activities, CNNs can be retrained, which allows you to expand on preexisting networks.

How CNNs work:

There may be tens or hundreds of layers in a convolutionary neural network that each learns to detect various features of an image. Filters are applied at various resolutions to each training image and the output of each convoluted image is used as the input to the next layer. The philtres can be very basic features such as brightness and edges, and features that describe the object uniquely increase in complexity.A CNN is composed of an input layer, an output layer, and several hidden layers in between, much like other neural networks.

Convolution, activation, or ReLU, and pooling are the three most common layers.

- **Convolution**: A series of convolutionary filtres are placed through the input images, each of which activates certain features of the images.
- **Rectified linear unit (ReLU)**: By mapping negative values to zero and retaining positive values, this facilitates quicker and more efficient preparation. This is often referred to as activation, since the next layer is carried forward by just the activated features.
- **Pooling**: By performing nonlinear downsampling, it simplifies the performance. The number of parameters that the network needs to learn is decreased.

Over tens or hundreds of layers, these operations are replicated, with each layer learning to recognise various features.

Classification Layers

The CNN architecture moves to classification after studying features in several layers. The next-to-last layer is a completely connected layer that generates a K-dimensional vector, where K is the number of classes that can be predicted by the network. For each class of any image being graded, this vector contains the probabilities. To provide the classification output, the final layer of the CNN architecture utilises a classification layer such as softmax.

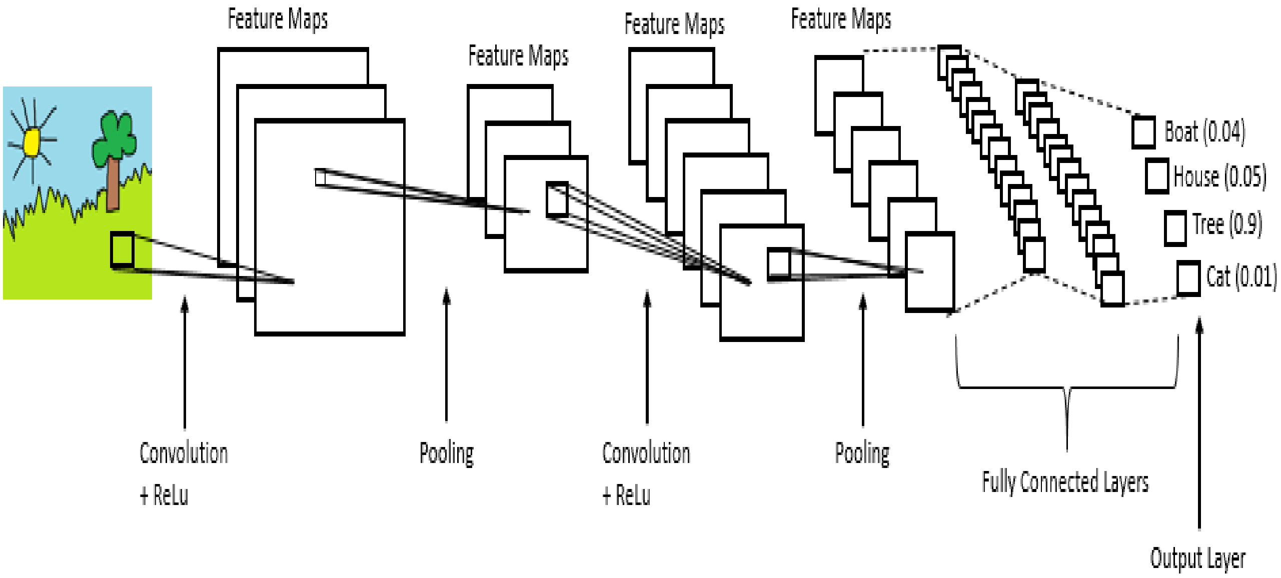


Figure 9: Complete CNN Architecture

.

3.5.2 Our Model Architecture

In our custom made cnn architecture we have used multiple convolution layers with several max pooling layers.

We have used 5 convolution layers each with different no. of filters(kernels) and rectified linear unit(ReLU) activation. For max pooling after every convolution step, we have used pool size of 2x2 with strides of 2.

The first convolution layer has 32 filters, second layer has 64 filters, third layer has 128 filters, fourth layer has 256 filters and the fifth layer has 1024 filters.

In the full connection step, we have used 2 hidden layers, first of which consists of 1024 neurons and second layer contains 128 neurons with rectified linear unit activation in each layer.

Since our problem is a multi-class classification problem, so we have used 4 output neurons in the last layer with softmax activation to classify normal,COVID-19,bacteria and pneumonia images.

Code Snippet:

1. First Layer Code

```
model.add(Conv2D(filters=32,kernel_size=5,activation="relu",input_shape=(IMG_SHAPE,IMG_SHAPE,1)))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(BatchNormalization())
```

2. Second Layer Code

```
model.add(Conv2D(filters=64,kernel_size=5,activation="relu"))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(BatchNormalization())
```

Chapter 4

Results And Analysis

Calculated feature vectors and the associated number of features. $LBP_{R,P}^{riu2}$, a normalized histogram of rotation-invariant uniform LBPs; F_{GLCM} , GLCM-based features; stat1, entropy, variance, and intensity mean. All the computed features vectors and their associated number of features are given in table below:

Table 1: Number of features in each combination of feature vectors used.

Feature Vector	Number of Features
LBP	162
GLCM	48
Stat1	3
LBP+Stat1	165
GLCM+Stat1	51
LBP+GLCM	210
LBP+GLCM+Stat1	213

All seven possible combinations of the feature extracted using three different techniques namely Local Binary Patterns, Grey-Level Co-Occurrence Matrix, and First Order Statistics were given to the four classifiers namely SVM, kNN, Random Forest and Naive Bayes and the accuracy was evaluated.

We calculated the accuracy of the classifier for a given combination of feature vector using the confusion matrix of the validation set where accuracy is given by the number of images rightly categorized upon the total number of images in the testing set.

Table 2: The accuracy scores of each classier with each combination of the feature vector.

Feature Vector	SVM	Naive Bayes	kNN	Random Forest
LBP	0.90	0.71	0.83	0.91
GLCM	0.69	0.60	0.77	0.89
Stat1	0.58	0.61	0.71	0.85
LBP+Stat1	0.89	0.63	0.80	0.89
GLCM+Stat1	0.72	0.68	0.77	0.90
LBP+GLCM	0.90	0.72	0.84	0.93
LBP+GLCM+Stat1	0.90	0.72	0.83	0.93

After this, for validation purposes from the confusion matrix, metrics like precision, recall, and accuracy were calculated. We calculated the class-specific recall ($rec_{class} = \left\{rec_{class_j}\right\} j\epsilon\,[1,4]$) for the assessment of the performance of the classifiers.

$$rec_{class_j} = \frac{TP_j}{TP_j + FN_j}$$

where TP_j is the number of images of the j'th class rightly categorized (true positive of the j'th class) and FN_j is the number of images of the j'th class incorrectly classified to one of the three remaining classes (false negative of the j'th class).

Table 3: The weighted average of recall with respect to the number of data in each class label for all the combination of feature vector and classifiers

Feature Vector	SVM	Naive Bayes	kNN	Random Forest
LBP	0.98	0.87	0.99	0.99
GLCM	0.96	0.88	0.98	0.99
Stat1	0.97	0.96	0.95	0.96
LBP+Stat1	0.97	0.80	0.97	0.99
GLCM+Stat1	0.97	0.97	0.97	0.99
LBP+GLCM	0.98	0.90	0.99	0.99
LBP+GLCM+Stat1	0.99	0.90	0.99	0.98

We further calculated the class-specific precision ($prec_{class} = \left\{prec_{class_j}\right\} j\epsilon\,[1,4]$) for the assessment of the performance of the classifiers,

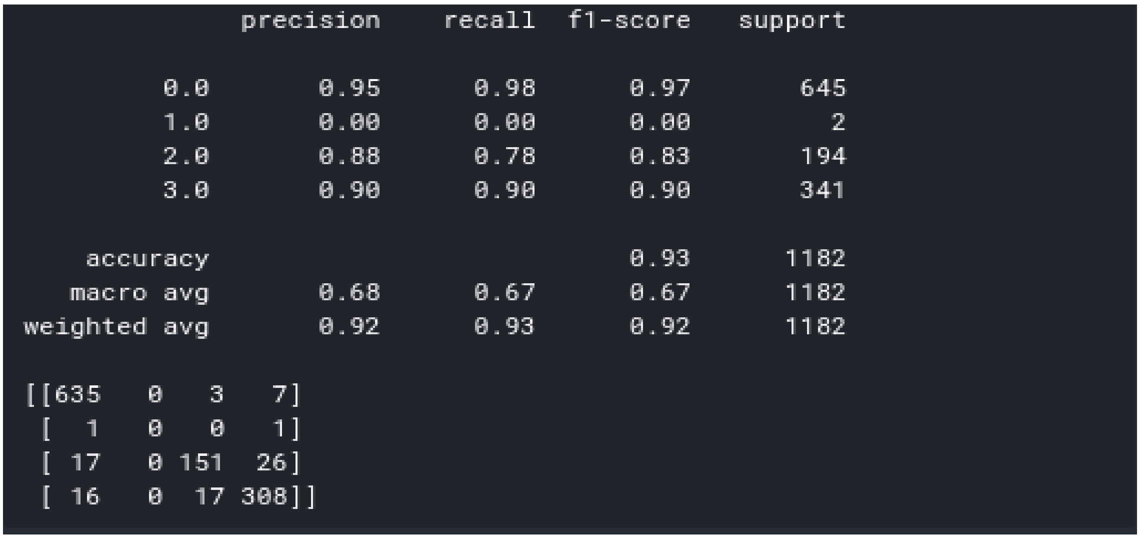
$$prec_{class_j} = \frac{TP_j}{TP_j + FP_j}$$

where FP_j is the number of false-positives of the j'th class.

Table 4: The weighted average of precision with respect to the number of data in each class label for all the combination of feature vector and classifiers

Feature Vector	SVM	Naive Bayes	kNN	Random Forest
LBP	0.96	0.77	0.87	0.93
GLCM	0.74	0.74	0.82	0.91
Stat1	0.59	0.73	0.75	0.90
LBP+Stat1	0.95	0.81	0.85	0.92
GLCM+Stat1	0.98	0.61	0.87	0.94
LBP+GLCM	0.96	0.79	0.89	0.95
LBP+GLCM+Stat1	0.96	0.78	0.87	0.95

From the above evaluation, we can see that Random Forest performed best with number of trees equal to 70, and using hybrid features of LBP, GLCM and First-Order Statistics



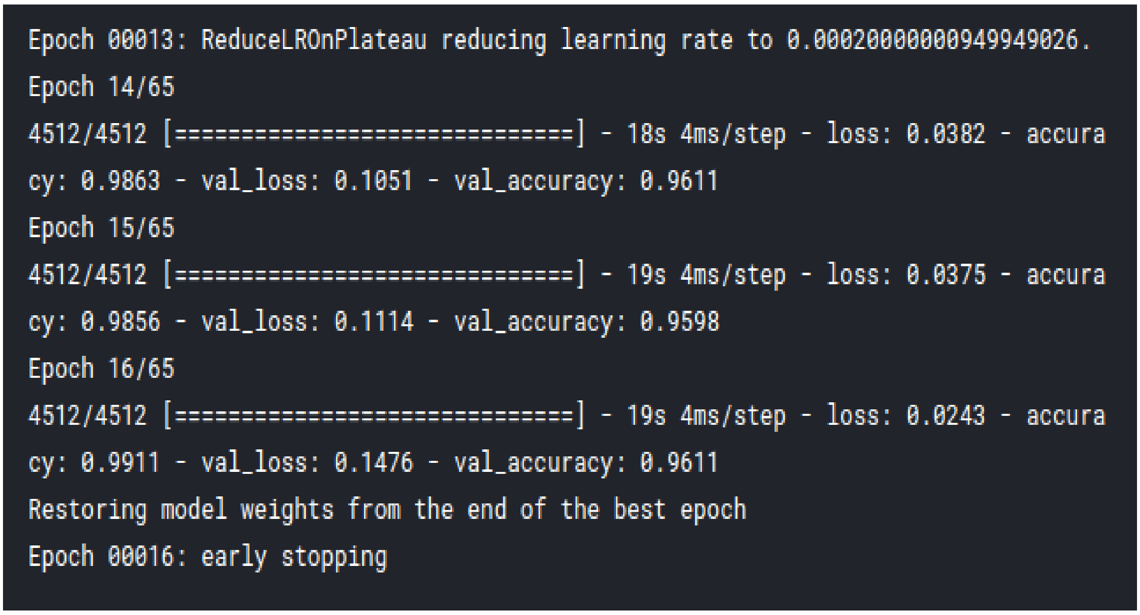
```
              precision    recall  f1-score   support

         0.0       0.95      0.98      0.97       645
         1.0       0.00      0.00      0.00         2
         2.0       0.88      0.78      0.83       194
         3.0       0.90      0.90      0.90       341

    accuracy                           0.93      1182
   macro avg       0.68      0.67      0.67      1182
weighted avg       0.92      0.93      0.92      1182

[[635   0   3   7]
 [  1   0   0   1]
 [ 17   0 151  26]
 [ 16   0  17 308]]
```

Figure 10: Classification Report Of Random Forest with hybrid feature vector as input on testing set.

We analysed that when comparing individual features ($H_{LBP_{riu2}}$, F_{GLCM} , stat1), $H_{LBP_{riu2}}$ gave the best classification performance. $H_{LBP_{riu2}}$ performed better than F_{GLCM}. The best results were given by the classifiers when the hybrid features using combinations of features extracted using LBP, GLCM, and First Order Statistics were combinedly used for classification.

Although the machine learning approaches gave good results, but two improve are result we switched to Deep Learning, namely CNN and build our own custom model consisting of 5 convolutional layers.

We trained our model using 4512 images and validated our model on 797 images and while training we set number of epochs to 65. And we achieved an accuracy of 96% from it. Below is the screenshot of accuracy achieved during training our model.



```
Epoch 00013: ReduceLROnPlateau reducing learning rate to 0.00020000000949949026.
Epoch 14/65
4512/4512 [==============================] - 18s 4ms/step - loss: 0.0382 - accura
cy: 0.9863 - val_loss: 0.1051 - val_accuracy: 0.9611
Epoch 15/65
4512/4512 [==============================] - 19s 4ms/step - loss: 0.0375 - accura
cy: 0.9856 - val_loss: 0.1114 - val_accuracy: 0.9598
Epoch 16/65
4512/4512 [==============================] - 19s 4ms/step - loss: 0.0243 - accura
cy: 0.9911 - val_loss: 0.1476 - val_accuracy: 0.9611
Restoring model weights from the end of the best epoch
Epoch 00016: early stopping
```

Figure 11: Screenshot of accuracy achieved

.

We also plotted the accuracy vs no. of epochs curve and loss vs no. of epochs curve

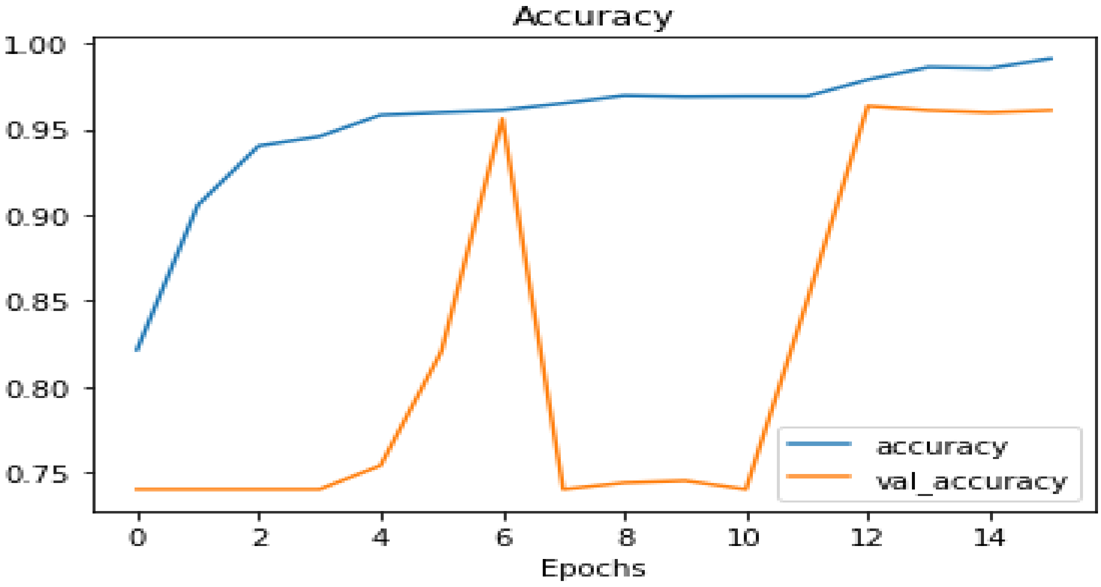


Figure 12: Accuracy vs no. of Epochs curve

.

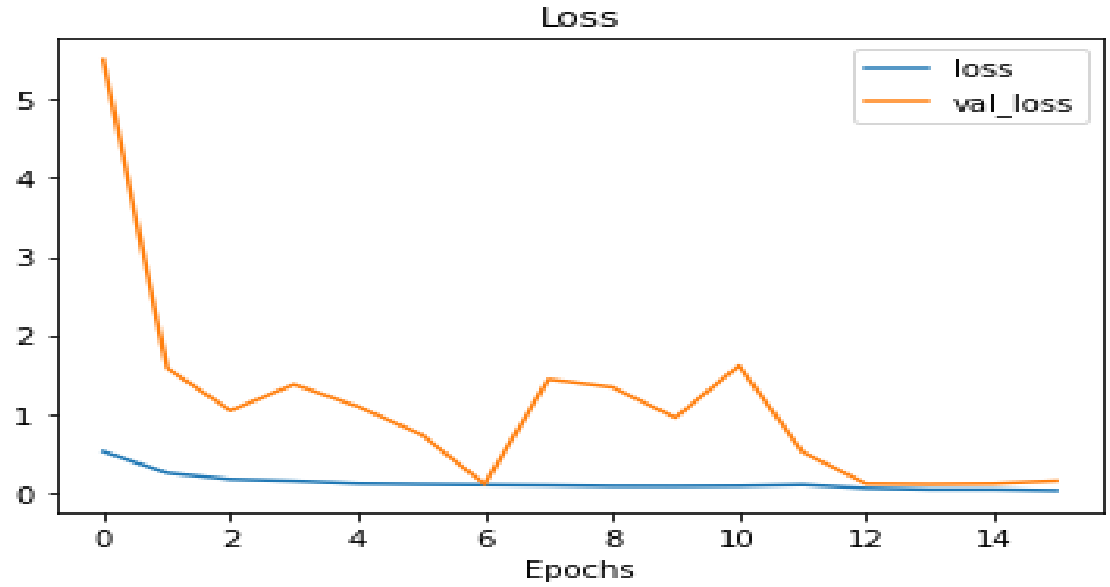


Figure 13: Loss vs no. of Epochs curve

.

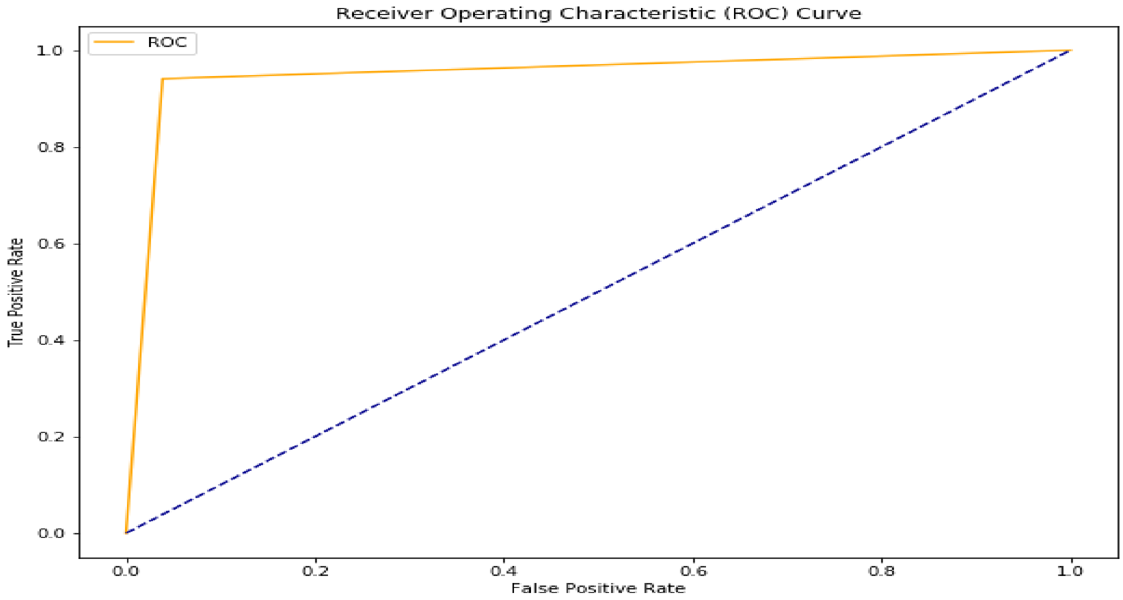


Figure 14: ROC Curve

.

From the above results we can see that the CNN performed much better giving us an accuracy of 96% than the Machine Learning Techniques which used handcrafted feature extraction and gave accuracy of 93%. Also we achieved an accuracy higher than that mentioned in the paper *"Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network."*[6] and comparable to the paper *Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network*[7]

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This project assessed an inventive way to the computer-aided classification of Covid-19 using Chest X-ray images. Distinct textural features were tried to explore the best feature vector to detect covid-19: texture-based global feature extraction techniques ($H_{LBP_{riu2}}$ and F_{GLCM}) and first-order statistics (stat1). When comparing individual features ($H_{LBP_{riu2}}$, F_{GLCM} , stat1), $H_{LBP_{riu2}}$ gave the best classification performance. In general, $H_{LBP_{riu2}}$ performed better than F_{GLCM}. The best results were given by the classifiers when the hybrid features using combinations of features extracted using LBP, GLCM, and First Order Statistics were combinedly used for classification.

Random Forest has demonstrated similar execution performence as for SVM and kNN, while huge contrasts were found regarding NB. This is likely because of NB not having the option to deal with a high-dimensional feature vector, for example, our own.

In conclusion, the most significant result was shown by the Random Forest classifier when hybrid features combining all three feature vectors obtained from LBP, GLCM, and First-order statistics were used giving an accuracy of 93%.

Furthemore, custome made CNN model of 5 layers gave better result than that of any Machine Learning Algorithms probably because CNN have been observed to perform better on medical images and radilogical images.CNN model gave an accuracy of 96%

5.2 Future Work

For future work, it would be interesting to exploit the techniques of Feature Reduction for our Machine Learning Models as our feature vectors very of very high dimensions.

We also plan to automate this process of Covid-19 detection by building a user interface through which directly the result whether a patient is Covid-19 positive or negative can be found by just give the Chest X-ray as input.

References

[1] Kaggle kernel. https://www.kaggle.com/kernels.

[2] Keras: the python deep learning ap. https://keras.io/.

[3] Numpy: The fundamental package for scientific computing with python. https://numpy.org/.

[4] Scikit-image. https://scikit-image.org/.

[5] Scikit learn. https://scikit-learn.org/stable/.

[6] Abbas, A., A. M. . G. M. Classification of covid-19 in chest x-ray images using detrac deep convolutional neural network. *Appl Intell* (2020).

[7] Abbas, Asmaa Abdelsamea, M. . G. M. Classification of covid-19 in chest x-ray images using detrac deep convolutional neural network. *10.1101/2020.03.30.20047456.* (2020).

[8] Abbas A, A. M. Learning transformations for automated classification of manifestation of tuberculosis using convolutional neural network. *In: 2018 13Th international conference on computer engineering and systems (ICCES), IEEE, pp 122–126* (2018).

[9] Ai T., Yang Z., H. H. Z. C. C. C. L. W. T. Q. S. Z. X. L. Correlation of chest ct and rt-pcr testing in coronavirus disease 2019 (covid-19) in china: a report of 1014 cases. *Radiology* (2020).

[10] Amit KumarJaiswal, PrayagTiwari, S. D. G. A. K. J. J. Identifying pneumonia in chest x-rays: A deep learning approach. *ScienceDirect* (2019).

[11] ANDIL E, ÇAKIROĞLU M, E. Z. M. K. C. A. Artificial neural network-based classification system for lung nodules on computed tomography scans. *In: 2014 6Th international conference of soft computing and pattern recognition (soCPar),* (2014).

[12] ANTHIMOPOULOS M, CHRISTODOULIDIS S, E. L. C. A. M. S. Lung pattern classification for interstitial lung diseases using a deep convolutional neural network. *In: 2018 13Th international conference on computer engineering and systems (ICCES), IEEE, pp 122–126* (2016).

[13] BARBALATA, C., AND MATTOS., L. S. Laryngeal tumor detection and classification in the endoscopic video. *IEEE Journal of biomedical and health informatics 20* (2014).

[14] D., L. D. Naive (bayes) at forty: the independence assumption in information retrieval. *European Conf. on Machine Learning, pp. 4–15, Springer;* (1998).

[15] F. PASA, V. GOLKOV, F. P. D. C. . D. P. Efficient deep network architectures for fast chest x-ray tuberculosis screening and visualization. *Scientific Reports volume 9, Article number: 6268* (2019).

[16] GAO M, BAGCI U, L. L. W. A. B. M. S. H. R. H. P. G. D. A. S. R. E. A. Holistic classification of ct attenuation patterns for interstitial lung diseases via deep convolutional neural networks. *. Comput Meth Biomechan Biomed Eng Imaging Visual 6(1):1–6* (2018).

[17] HANSELL D.M., BANKIER A.A., M. H. M. T. M. N. R. J. F. S. glossary of terms for thoracic imaging. *Radiology* (2008).

[18] HARALICK R. M., E. A. Textural features for image classification,. *IEEE Trans. Syst. Man Cybern. SMC-3(6), 610–621 (1973).10.1109/TSMC.1973.4309314* (1973).

[19] J., B. C. A tutorial on support vector machines for pattern recognition,. *Data Min. Knowl. Discovery 2(2), 121–167* (1998).

[20] Keller J. M., Gray M. R., G. J. A. A fuzzy k-nearest neighbor algorithm,. *IEEE Trans. Syst. Man Cybern. SMC-15(4), 580–585* , (1985).

[21] Krizhevsky A, Sutskever I, H. G. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* (2012).

[22] L., B. Random forests,. *Mach. Learn. 45(1), 5–32* (2001).

[23] Lin, Yuanqing, F. L. S. Z. M. Y. T. C. K. Y. L. C., and Huang., T. Large-scale image classification: fast feature extraction and svm training. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2011).

[24] Lin Y., e. a. Large-scale image classification: fast feature extraction and svm training. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR),* (2011).

[25] Manikandan T, B. N. Lung cancer detection using fuzzy auto-seed cluster means morphological segmentation and svm classifier. *J Med Syst 40(7):181* (2016).

[26] Mohd Zulfaezal Che Azemin, Radhiana Hassan, M. I. M. T. M. A. M. A. Covid-19 deep learning prediction model using publicly available radiologist-adjudicated chest x-ray images as training data: Preliminary findings. *International Journal of Biomedical Imaging, vol. 2020, Article ID 8828855, 7 pages* (2020).

[27] Nanni L., Lumini A., B. S. Local binary patterns variants as texture descriptors for medical image analysis,. *Artificial intelligence in medicine 49,* (2010).

[28] S. Albawi, T. A. M., and Al-Zawi, S. Understanding of a convolutional neural network. *SInternational Conference on Engineering and Technology (ICET), Antalya, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.* (2017).

[29] Sangamithraa P, G. S. Lung tumour detection and classification using ek-mean clustering. *In: 2016 International conference on wireless communications, signal processing and networking (wiSPNET), IEEE, pp 2201–22061* (2016).

[30] Shin HC, Roth HR, G. M. L. L. X. Z. N. I. Y. J. M. D. S. R. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE Trans Med Imaging 35(5):1285–1298* (2016).

[31] Sun W, Zheng B, Q. W. Computer aided lung cancer diagnosis with deep learning algorithms. *Medical imaging 2016: computer-aided diagnosis, vol. 9785, p. 97850z. International society for optics and photonics* (2016).

[32] Wang W., Xu Y., G. R. L. R. H. K. W. G. T. W. Detection of sars-cov-2 in different types of clinical specimens. *JAMA* (2020).

[33] Yang, Y., Y. M. S. C. W. F. Y. J. L. J. Z. M. e. a. Laboratory diagnosis and monitoring the viral shedding of 2019-ncov infections. *MedRxiv.* (2020).