# COVID-19 Detection using Chest X-Ray Images


## Abstract

Early detection of COVID-19 may also aid in developing the best treatment plan and disease management strategy. Decisions on containment In this study, we show how switch learning from deep learning models can be used to detect COVID-19 using images from three of the most commonly used scientific imaging modes: X-Ray, Ultrasound, and CT scan. Covid-19 is a fiercely disputed study issue since it is a potentially lethal pandemic. The goal of our research is to create an intelligent method for identifying covid-19 in chest X-ray pictures using Machine Learning techniques. We suggested a hybrid model-based Principal Component Analysis (PCA), Support Vector Machine (SVM), and Xgboost technique for identifying Covid-19 in chest X-ray images in this work. To execute the recognition process, our technique combines the best qualities of PCA and SVM. We employed the PCA approach to extract features from X-ray pictures, SVM as a binary classifier, and Xgboost to increase the performance of our model and minimize overfitting. Our model produces satisfactory results with a less complex model architecture.



**Keywords**: Support Vector Machine (SVM), Principal Component Analysis (PCA), X-ray chest pictures, Covid-19, and the Xgboost.


## Introduction

The infectious disease caused by coronavirus, dubbed COVID-19 after its discovery in 2019, began in China but has since spread globally. The protein spikes on the surface of the coronavirus resemble a crown, hence the Latin name "corona," which means "crown." SARS stands for Severe Acute Respiratory Syndrome. The main cause of this newly discovered disease is the Coronavirus 2 (SARS-CoV-2) virus, which is genetically similar to the SARS Coronavirus (SARS-CoV) virus, which was first identified in 2003. Coronaviruses such as SARS-CoV-2, SARS-CoV, and Middle East respiratory syndrome Coronavi-rus (MERS-CoV) were initially spread among a variety of people, animals and spread to humans as a result of increased animal-human contact or virus mutations [1]

[1] github link

The disease spreads quickly among humans via respiratory droplets expelled by an infected person while coughing, talking, or sneezing in close proximity to other people. The severity of an infected person's illness can range from mild to moderate symptoms like fever, cough, sore throat, and tiredness to life-threatening cases like pneumonia, shortness of breath, organ failure, and death. People with pre-existing medical conditions, such as chronic respiratory disease, diabetes, high blood pressure, heart disease, or cancer, as well as the elderly, are more likely to develop complications from this infectious disease.

RT-PCR (Reverse Transcriptase Polymerase Chain Reaction), a test that aids in the detection of the virus's genetic material, is currently used to diagnose Coronavirus disease. Some antibody or serology tests are available to check for virus antibodies. Diagnostic tests, such as RT-PCR, take time and can produce false-negative results. To combat this disease, antiviral medications and vaccines are being developed. Rapid and accurate detection of this lethal infectious disease is critical for containing the current pandemic. COV-ID-19 can be diagnosed by radiologists using a chest computed tomography (CT) scan and can be used as an initial screening technique to identify infected patients, according to recent research.

Millions of people have died as a result of the Covid-19 outbreak all across the world. Since then, the public health care system has had to contend with an unprecedented challenge. Due of the COVID 19 outbreak, it is imperative to be diagnosed correctly and quickly. Machine learning (ML) and artificial intelligence (AI) can be extremely useful tools in the fight against the Covid-19 epidemic. The automatic detection of patients who are Covid-19 infected using chest X-ray pictures can be accomplished utilizing a variety of machine learning (ML) algorithms, which are presented in detail in this study. Chest X-ray images of both Covid and non-Covid patients that were gathered from various sources make up the dataset that has been used.

Based on a few parameters, including accuracy, recall, precision, F1-score, and the AUC-ROC curve, a performance comparison analysis of the various ML algorithms has been carried out. With an accuracy of up to 83.9%, recall, precision, and F1-score of 81.3%, respectively, and an AUC of 83.3%, XGBoost has outperformed all other classifiers.[2]

[2] github link

## Motivation

COVID-19, Corona Virus Disease-2019, caused by a new Corona Virus 2 Severe Acute Respiratory Syndrome (SARS-CoV-2). An effective screening of this virus can enable quick and accurate COVID-19 diagnosis, reducing the burden on the healthcare system. A detailed analysis of the provided dataset can be used to construct various types of machine learning algorithms, the performance of which can be computed and further evaluated. Random Forest outperformed all other Machine Learning models in the following case, including SVM, XGBoost, and Bagging Classifier.

In comparison to other strategies for great disorder prediction within algorithms, SVM produces excellent results. This is a Machine Learning (ML) model for two-group classification problems that used a classification algorithm. It basically creates a learning model that assigns new examples to one of two groups.Statistics mining is useful for obtaining important information.

## Objective

Create a PCA with ML model to detect COVID-19 positive patients using X-rays. That is, to train a model using a dataset of normal and COVID-19 images. positive patient's X-Rays, and to improve its ability to successfully diagnose the patient using bullet points from their X-Ray List objectives

- To achieve maximum efficiency.
- Consider large data sets.
- To reduce time complexity and improve accuracy.

## Related Work

The World Health Organization has declared a worldwide pandemic of the novel coronavirus Covid-19. Without public awareness and precautionary measures, it is extremely difficult to prevent community transmission even during lockdown. This Android app updates the locations of the areas in a Google map that have been identified as containment zones. The application also alerts users if they enter a containment zone. To identify COVID-infected people, healthcare and travel data are processed using machine learning algorithms rather than traditional healthcare systems. This study compared multiple algorithms for processing patient

[3]data and determined that the Boosted Random Forest was the best method for data processing. To improve performance, we use the grid search to adjust the BOOSTED RANDOM FOREST method's hyper parameters.

Our work eliminates the need to re-compare existing COVID-19 patient data processing algorithms. The ability to handle patient demographics, travel information, and subjective health data along with image data (scans) will allow researchers to continue working on a system that will improve the prediction of COVID-19 health-related outcomes for patients.

The data obtained from the application is not made public and is only used by the government to trace, track, and manage COVID-19. To use the Aarogya Setu application, users must first log into the app using a one-time password and fill out basic demographic information about themselves in order to determine whether they are in a safe area. The current covid19 prediction is based on the patients' X-ray images. However, looking at x-ray images with bare eyes makes it difficult to determine whether the person is truly infected with coronavirus or if the symptoms are caused by something else. As a result, the accuracy is subpar, and prediction is not successful always.

Tulin et al. [1] developed an automatic model for COVID-19 detection using Chest X-ray images. This model classified using two methods: binary classification (images of COVID and NoFindings) and multi-class classification (images of COVID, Pneumonia, and NoFindings). They used a DarkNet model as a classifier in their study for "You Only Look Once" (YOLO), a real-time object identification system. They used a total of 17 convolutional layers. They achieved a binary classification accuracy of 98.08% and a multi-class classification accuracy of 87.02%.

In their publication, Khan et al. [2] proposed the "CoroNet" model, a CNN model for COVID-19 diagnosis utilising chest radiography images. The "Xception Architecture," a pretrained model developed using the ImageNet dataset, serves as the foundation for the suggested solution. It is trained using data that was compiled for research from several publicly available datasets. Models have an 89.6% success rate on average. For the four classes, the recall and precision rates for COVID-19 cases are 93% and 98.2%, respectively. The accuracy

[3] github link

of classification for the three classes COVID, pneumonia, and normal is 95%.

## Proposed Framework

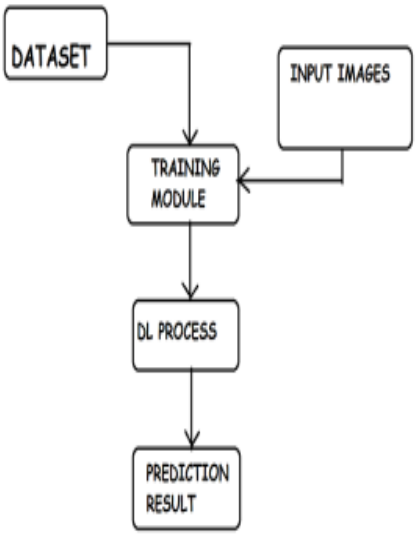
1) The dataset of chest x-rays with covid positive and non positive labels is used in the suggested system.

2) The proposed system can accept user input.

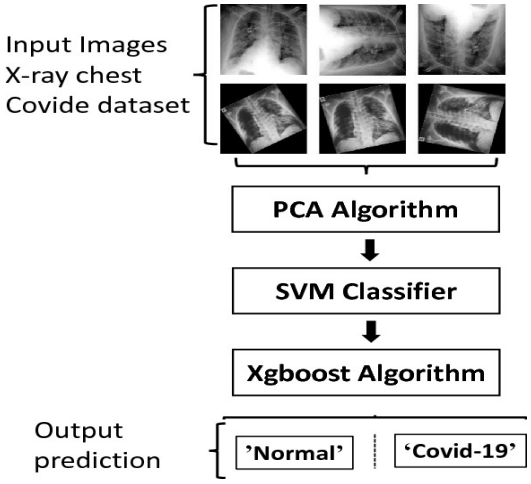
3) The system under consideration can perform PCA operations by dividing the data into training and testing sets.

4) The proposed system can compare the extracted feature from the user's input x-ray image with a dataset that has already been trained and contains a variety of characteristics.

**Working Model:**




**Architecture**




## Pseudo code

Step 1: First, a dataset of chest x-ray pictures that is readily accessible is used to construct the system.[4]

[4] github link

Step 2: To prepare the dataset for utilisation, steps including feature extraction and exploration are carried out.

Step 3: After the dataset is complete, it will be divided into a train dataset and a test dataset.

Step 4: Accuracy will be assessed after training and testing a pre-trained ML model.

Step 5: After all of the preparations have been made, the user can submit actual test results for COVID19 Prediction.

Step 6: The Streamlit Python Module is used to give the user interface.

## Analysis

The following breakdown quickly depicts various learning variables discovered in our studied applications: Regulated Learning improves a blunder work in terms of predicted and real names. These ground truth markings necessitate manual explanation. Solo Learning does not use names. This includes bunching calculations that look for natural design in information. Self-Supervised Learning streamlines a misfortune effort as for the anticipated and true names. Unlike Supervised Learning, these marks are derived from a distinct recording metric rather than human remark. Semi-Supervised Learning use a combination of human-labeled and unlabeled data for portrayal learning. Move Learning displays beginning preparation with the portrayal obtained from a previous assignment. This previous task is typically ImageNet-based directed learning in "Normal Language Processing" or Internet-scale language showing.

Carry out various tasks Learning concurrently improves various misery work, normally by interleaving refreshes or adding regularization punishments to try. not to fight against urges from every misfortune Pitifully Supervised Learning refers to guided learning using heuristically marked information rather than carefully identified information.

## Implementation

### PCA-SVM[5]

[5] github link

Input: Given $N$ observations along with the class labels $(x_i, t_i)$, $x_i \in R^m, t_i \in R^C$.
Output: PCA-SVM model for classification.
(1) Procedure PCA-SVM
(2) Identify the relationship among features through a covariance matrix.
(3) Through the linear transformation or eigendecomposition of the covariance matrix, we get eigenvectors and eigenvalues.
(4) Transform our data using eigenvectors into principal components.
(5) Quantify the importance of these relationships using eigenvalues and keep the important principal components.
(6) Data extracted from PCA and will be given as input.
(7) Initialize weights $W$ and bias $b$ with any arbitrary number.
(8) Feature optimization.
(9) TrainDataSet, TestingDataSets = Data.Split(Ratio).
(10) Define $y$.
(11) Define $f(y)$.
(12) Define $J(W)$.
(13) Calculate $\nabla_w J(w)$.
(14) Repeat for minimum $J(W)$:
Update $W$.
Call steps 10-13.
End.
(15) Calculate accuracy.
(16) Return accuracy.
(17) End procedure.

## SVM Classifier

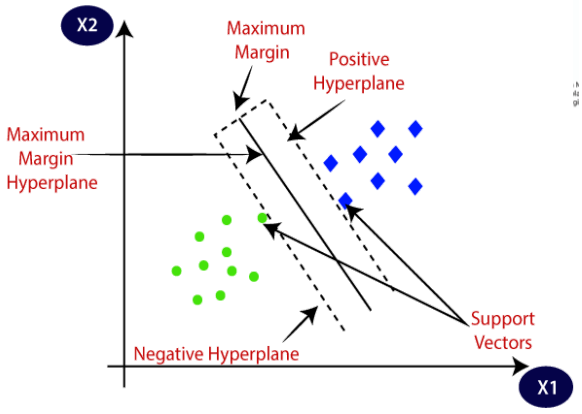
The data in SVM Classification might be either linear or non-linear. An SVM Classifier can be configured using a variety of kernels. We can set the kernel to 'linear' for a linear dataset.

A non-linear dataset, on the other hand, has two kernels: 'rbf' and 'polynomial.' The data is mapped to a higher dimension in this case, making it easier to construct the hyperplane. It is then brought down to the lower dimension.

Support Vector Machine, or SVM, is a prominent Supervised Learning technique that is used for both classification and regression issues. However, it is mostly utilised in Machine Learning for Classification difficulties.

The SVM algorithm's purpose is to find the optimum line or decision boundary for categorising n-dimensional space so that we may easily place fresh data points in the correct category in the future. A hyperplane is the optimal choice boundary.

SVM selects the extreme points/vectors that aid in the creation of the hyperplane. These extreme examples are referred to as support vectors, and the method is known as the Support Vector Machine. Consider the picture below, which shows two distinct categories that are classified using a decision boundary or hyperplane:




### XGBoost

XgBoost is an acronym for Extreme Gradient Boosting, which was proposed by University of Washington academics. It is a

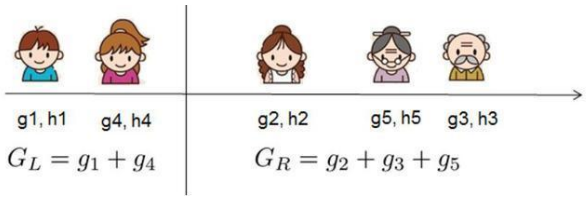
C++ package that optimises the training for Gradient Boosting.[6]

The science behind XgBoost Before delving into the mathematics of Gradient Boosting, consider this simple example of a CART that predicts if someone will enjoy a fictional computer game X. The following is an example of a tree:

The prediction scores of each individual decision tree are then added together to yield When you look at the example, you'll notice that the two trees aim to compliment each other. We may write our model mathematically in the way

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \epsilon \mathcal{F}$$

Example




Sample

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, eval_metric='mlogloss',
              gamma=0, gpu_id=-1, importance_type='gain',
              interaction_constraints='', learning_rate=0.300000012,
              max_delta_step=0, max_depth=6, min_child_weight=1, missing=nan,
              monotone_constraints='()', n_estimators=100, n_jobs=16,
              num_parallel_tree=1, objective='multi:softprob', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=None, subsample=1,
              tree_method='exact', use_label_encoder=False,
              validate_parameters=1, verbosity=None)
```

**Bagging classifier**

A bagging classifier is an ensemble meta-estimator that applies base classifiers one at a time to arbitrary subsets of the original dataset, then combines the individual predictions (either by voting or by averaging) to produce a final prediction. A meta-estimator of this kind can frequently be used to reduce the variance of the estimator by adding randomization to the process of developing a black-box estimator (such as a decision tree).

This approach uses a variety of literary works. When random subsets of the dataset are chosen as random subsets of the samples, this process is known as pasting. "Bagging" samples are those taken using replacement. When random subsets of the dataset are created as random subsets of the features, the method is known as "Random Subspaces". Last but not least, when base estimators are built on subsets of both

[6] github link

samples and features, the technique is known as Random Patches.


## Data Description

Link

https://www.kaggle.com/datasets/pranavraikokte/covid19-image-dataset/versions/2

**About this directory**

It contains around 137 cleaned images of COVID-19 and 317 in total containing Viral Pneumonia and Normal Chest X-Rays structured into the test and train directories.

Patients who have been infected with Covid-19 have had their chest X-ray images labelled as 1, whereas those who have not been infected with Covid-19 have had their images l[7]abelled as 0.

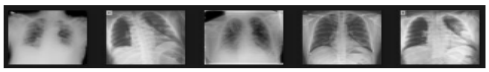


Fig. 1. COVID chest X-Ray images

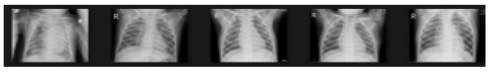


Fig. 2. PNEUMONIA chest X-Ray images

Fig. 3. NORMAL chest X-Ray images

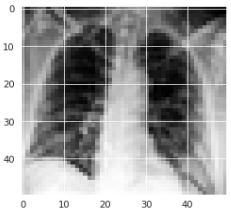
## Results

Data Visualization

+ Code + Markdown

```
dim = (50,50)
plt.imshow(X_train[0],cmap='gray')
```

```
[16]: <matplotlib.image.AxesImage at 0x7fc6dd054b50>
```




### SVM Accuracy

```
[23]: from sklearn.metrics import accuracy_score
      print("accuracy Score -->",accuracy_score(y_test,y_pred)*100)
```

```
accuracy Score --> 86.36363636363636
```

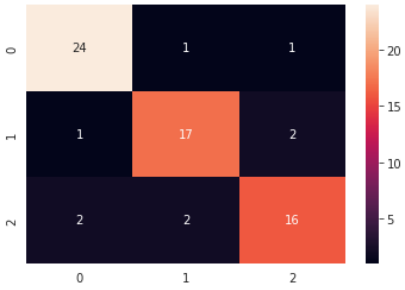
### Classification Report SVM

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.89 | 0.92 | 0.91 | 26 |
| 1 | 0.85 | 0.85 | 0.85 | 20 |
| 2 | 0.84 | 0.80 | 0.82 | 20 |
| accuracy | | | 0.86 | 66 |
| macro avg | 0.86 | 0.86 | 0.86 | 66 |
| weighted avg | 0.86 | 0.86 | 0.86 | 66 |

[7] github link

Confusion Matrix

```
[26]: <AxesSubplot:>
```




## XGboost Accuracy

```
[29]: #Predicting the test set result
y_pred= classifier.predict(X_test_pca)
from sklearn.metrics import accuracy_score
print("accuracy Score -->",accuracy_score(y_test,y_pred)*100)
```

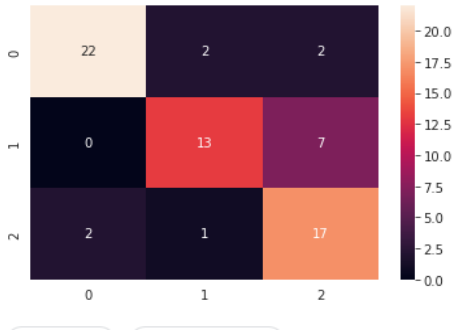
```
accuracy Score --> 78.78787878787878
```

## Classification Report XGboost

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.92 | 0.85 | 0.88 | 26 |
| 1 | 0.81 | 0.65 | 0.72 | 20 |
| 2 | 0.65 | 0.85 | 0.74 | 20 |
| accuracy | | | 0.79 | 66 |
| macro avg | 0.79 | 0.78 | 0.78 | 66 |
| weighted avg | 0.81 | 0.79 | 0.79 | 66 |

## Confusion Matrix

```
[31]: <AxesSubplot:>
```

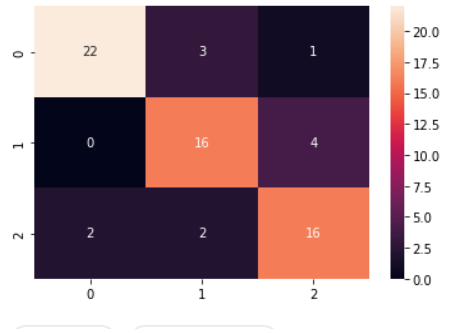



## Bagging Classification report

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.92 | 0.85 | 0.88 | 26 |
| 1 | 0.76 | 0.80 | 0.78 | 20 |
| 2 | 0.76 | 0.80 | 0.78 | 20 |
| accuracy | | | 0.82 | 66 |
| macro avg | 0.81 | 0.82 | 0.81 | 66 |
| weighted avg | 0.82 | 0.82 | 0.82 | 66 |

## Bagging Confusion Matrix

```
[83]: <AxesSubplot:>
```




## References

1. Tulin Ozturk, Muhammed Talo, Eylul Azra Yildirim, Ulas Baran Baloglu, Ozal Yildirim, and U. Rajendra Acharya, "Automated detection of COVID-19 cases using deep neural networks with

[8]X-ray images," Computer Biology and Medicine, 2020 Jun; 121: 103792. Online since April 28, 2020 4.

2. "CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images," Asif Iqbal Khan, Junaid Latief Shah, and Mohammad Mudasir Bhat, Computer Methods and Programs in Biomedicine, Volume 196, 2020,105581, ISSN 0169-2607
3. Sonbhadra, S.K., Agarwal, S., Nagabhushan, P.: Target specific mining of COVID-19 scholarly articles using one-class approach. Elsevier (2020)
4. Xu, X., Jiang, X., Ma, C., et al.: Deep learning system to screen coronavirus disease 2019 pneumonia, pp. 1e29 (2020)
5. Sun, J., et al.: A prospective observational study to investigate performance of a chest X-ray artificial intelligence diagnostic support tool across 12 US hospitals, arXiv preprint arXiv:2106.02118 (2021)
6. Baloch, S., Baloch, M.A., Zheng, T., Pei, X.: The coronavirus disease 2019 (COVID-19) pandemic. In: The Tohoku Journal of Experimental Medicine, pp. 271–278. Tohoku University Medical Press (2020)
7. https://towardsdatascience.com/machine-learning-basics-support-vector-machine-svm-classification-205ecd28a09d#:~:text=Overview%20of%20SVM%20Classification&text=In%20SVM%20Classification%2C%20the%20data,rbf'%20and%20'polynomial'.
8. https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm
9. https://www.geeksforgeeks.org/xgboost/

[8] github link