COVID-19 anomaly detection and classification based on Chest X-Ray images

Nitin Bisht (21MM61R08), Nemanth Kumar (21MM61R09)

Github Repo: https://github.com/kalpiree/Predicting-diseases-using-Chest-XRay-using-Supervised-Learning

April 15, 2022

Abstract

The term COVID-19 is an abbreviation of Coronavirus 2019, a global pandemic that threatened the lives of millions of people. However, early detection provides an adequate opportunity for recovery and prevents the spread of the disease. The chest X-Ray is not only cheaper but also a quicker method compared to other imaging modalities.

This paper aims to provide different methods for the classification and the early detection of COVID-19 using X-Ray imaging. We aim to use Supervised learning methods to classify the X- ray image into either of the following four categories- Normal, Pneumonia, Lung-Opacity, and Covid-19 (Hasoon et al., 2021).

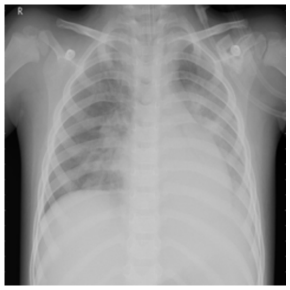
The supervised method involves image preprocessing for noise removal, ROI Detection, segmentation, feature extraction, and classification. The classification uses different machine learning models like K-Nearest Neighbours, Support Vector Machine, and Random Forest method. The deep learning method involves the use of vanilla VGG-16 architecture. The methods are trained and tested on around 5000 images distributed unequally over four classes using a train-test split, and their accuracy and the F-1 score are calculated. The detailed description of the project is available at the **Github** repository.

1 Data

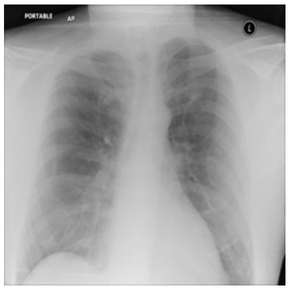
A team of researchers from Qatar University, Doha, Qatar, and the University of Dhaka, Bangladesh, along with their collaborators from Pakistan and Malaysia in collaboration with medical doctors, created a database of chest X-ray images for COVID-19 positive cases along with Normal and Viral Pneumonia images. This COVID-19, normal, and other lung infection dataset has 219 COVID-19, 1341 normal,1345 viral pneumonia chest X-ray (CXR) images and 2095 other lung infection images (Rahman et al., 2021, Tahir et al., 2022, Chowdhury et al., 2020). However, we used data augmentation techniques like to increase the size of our dataset. Figure 1 shows a sample of four classes in the data. The dataset is available on Google Drive.

(a) COVID affected lung

(b) Lung opacity



(c) Normal lungs



(d) Pneumonia affected lungs

Figure 1: Samples of different classes from the dataset.

2 Methods

2.1 Dataset Preprocessing

The original Kaggle dataset was in the form of folders containing labeled X-Ray images classified into four classes. The dataset was unsuitable to be fed into the supervised learning models. Additionally, the X-Ray image samples were noisy. Thus, we followed the following steps on the initial dataset.

2.1.1 Automating the process of finding I.D. from the link and importing libraries required

The image dataset was large enough to download multiple times, so it was downloaded to the cloud server. A function was created to automate extracting the zip folder I.D. and saving it in the cloud server. Once the data was extracted from the unzipped folder, specific python libraries were imported to perform further operations. Some important libraries used are:

- Scikit-image (a.k.a. skimage): It is a collection of algorithms for image processing and computer vision (*Skimage* n.d.)

- OpenCV is a vast open-source library for computer vision, machine learning, and image processing. It may be used to analyze photos and videos to recognize items, people, and even human handwriting. Python can process the OpenCV array structure for analysis when combined with other modules such as NumPy.
- NumPy: NumPy is a Python module that allows you to interact with arrays. It also provides functions for working with matrices, Fourier transforms, and linear algebra.
- Seaborn: Seaborn is a matplotlib-based Python data visualization package. It has a high- level interface for creating visually appealing and instructive statistics visuals.

2.1.2 Image preprocessing

The initial images were in raw form. We needed to write a preprocessing function before the feature extraction could be done, and the CSV file was fed to the supervised learning model for classification. The following steps were involved:

- **Converting to a grayscale image and applying a Gaussian filter on the image to reduce or remove noise.**

  A linear filter, a Gaussian filter is just that. It is frequently used to blur or minimize the noise in a picture. You can use them for "unsharp masking" if we combine two and remove them (edge detection). By alone, the Gaussian filter blurs edges and reduces contrast. Gaussian filters use a bell curve to weight pixels around the center pixel. This means that pixels further away have lower weights. Multiplying and adding are faster than sorting. Hence it is shorter than the median filter.

- **Automatically finding threshold value using Otsu's thresholding.**

  The threshold approach developed by Otsu is intended to transform a grayscale image into a binary image. This approach uses various image processing algorithms to accomplish histogram-based image thresholding or convert a grayscale picture to binary (Ozturk et al., 2020). The picture in the Otsu thresholding approach is assumed to be a bi-modal histogram (foreground and background and the related optimal threshold).

- **Performing morphological operations on the samples: erosion and dilation.**

Mathematical morphology (MM) tries to extract picture components that are useful in depicting region, form, and description, such as skeletons and convex hull boundaries. In addition, morphological approaches such as morphological filtering by reconstruction, thinning, and pruning transforms are explored for pre-or post-processing. Here we are performing dilation followed by two erosions:

1. **Dilation:** Morphological dilation makes objects more visible and fills in small holes in things. Lines appear thicker, and filled shapes appear larger.
2. **Erosion:** Morphological erosion removes floating pixels and thin lines so that only substantive objects remain. The remaining lines appear lighter, and shapes appear smaller.

- **Using a midpoint ellipse to create a mask.**

  This method makes use of the principle that the denser objects like bones appear white in X-Ray modality. To extract the Region Of Interest (ROI), we made use of the midpoint ellipse method. The midpoint x coordinates are chosen such that x1 is determined where the binary values turn one and x2 is the last pixel where the binary value is 1. The y axis coordinate is chosen beginning from the last rows where the entire row has binary value of one. The length of the major axis is twice the length from the midpoint to the first row from the top where the entire row has binary value of one. Figure 2 shows a sample depicting the image after extracting the region of interest. Algorithm 1 performs preprocessing operations to obtain ROI from a given lung image.

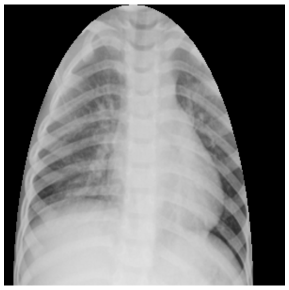
2.1.3 Feature Extraction of the Region of Interest

After extracting the Region of Interest (ROI), the next step involves feature extraction. The distribution of gray levels over the pixels in a picture region determines the texture of the part. It is frequently used to describe how an image seems fine or coarse, smooth or uneven, homogenous or inhomogeneous, and so on. The characteristics are defined to quantify attributes of an image region by utilizing spatial relations underlying a specific picture's gray-level distribution. We will find the first order and second-order features of the images of all classes.

- The **first-order statistics** provide information about the image's gray-level distribution (Aggarwal and K. Agrawal, 2012). It involves mean, standard deviation, skewness,

(a) COVID affected lung

(b) Lung opacity


(c) Normal lungs

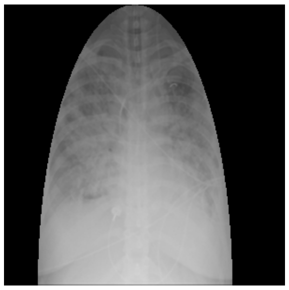

(d) Pneumonia affected lungs

Figure 2: Samples of different classes from the dataset after ROI extraction.

and kurtosis. The Variance is a measure of histogram width that indicates how far gray levels deviate from the Mean. Kurtosis is a measure of histogram sharpness, and skewness is a measure of histogram asymmetry around the Mean.

- The first-order features do not give any information about the relative positions of the various gray levels within the image. These features will not be able to measure whether all low-value gray levels are positioned together or they are interchanged with the high-value gray levels. An occurrence of some gray-level configuration can be described by a matrix of relative frequencies $P_\theta, d(I_1, I_2)$. It represents how frequently two pixels with gray levels I_1 and I_2 appear in the window, separated by a distance d in direction θ. It is called the gray level co-occurrence matrix. Using the Co-occurrence matrix, features can be defined, which quantifies coarseness, smoothness, and texture-related information that have high discriminatory power. Contrast, Correlation, Homogeneity, and Entropy are **second-level features**.

Algorithm 2 extracts first and second-order features from an image.

Algorithm 1 Pre-processing on Lung Image

```
1: function PREPROCESS($image$)
2:     $image \leftarrow rgb2gray(image)$    ▷ $rgb2gray$: converts image to grayscale.
3:     $image \leftarrow gaussfilter(image)$    ▷ $gaussfilter$: applies gaussian filter.
4:     $binary\_image \leftarrow otsu(image, otsu\_threshold)$    ▷ $otsu$: converts grayscale image to binary image.
5:     $binary\_image \leftarrow dilation(binary\_image, matrix)$    ▷ $dilation$: performs morphological dilation.
6:     $binary\_image \leftarrow erosion(binary\_image, matrix)$    ▷ $erosion$: performs morphological erosion.
7:     $binary\_image \leftarrow erosion(binary\_image, matrix)$
8:     $major\_axis, minor\_axis \leftarrow midpoint\_ellipse(binary\_image)$    ▷ $midpoint\_ellipse$: finds major and minor axis for ellipse.
9:     $mask \leftarrow ellipse(major\_axis, minor\_axis)$    ▷ $ellipse$: generates ellipse for given major and minor axis.
10:    $roi \leftarrow extract(mask, binary\_image)$    ▷ $extract$: extracts roi for given image and mask.
11:    return $roi$
```

Algorithm 2 Feature Extraction

```
1: function FEATUREEXTRACT($image$)
2:     $image \leftarrow rgb2gray(image)$    ▷ $rgb2gray$: converts image to grayscale.
3:     $first\_order \leftarrow [mean(image), std\_dev(image), skew(image), kurtosis(image)]$
4:     $second\_order \leftarrow [dissimilarity(image), contrast(image), homogeneity(image), energy(image), corr(image), asm(image)]$
5:     return $first\_order, second\_order$
```

2.1.4 Creating a CSV file with extracted features

Once the features are extracted, we will create a CSV file with all the extracted features as x- labels, and the y-label as the classification of images which will act as ground truth. The y-labels are in the form of text which we will encode into the numerical format. The CSV file is created by iteratively running the image processing and feature extraction code on all the images. Once the file is created, it is saved in the cloud for further usage. The folder containing the images after masking and generation of csv file is Google drive link.

2.2 Machine Learning Models

The CSV file obtained is given to different machine learning models with the train-test split of 80:20. We used three different kinds of supervised learning algorithms here. Algorithm 3 shows different ML models applied to the dataset. It takes input folder containing images as input and an output folder to store preprocessed images.

1. **Support Vector Machines**

   A support vector machine is a sort of machine learning approach that may be used for classification as well as regression. It primarily consists of two variations that support both linear and nonlinear issues. Linear SVM has no kernel and finds the problem's smallest margin linear solution (Taneja et al., 2015). When the solution is not linearly separable, SVM with kernels are utilized. Here for the ease of computation power, we are using linear SVM.

2. **K-Nearest Neighbours**

   K-nearest neighbors is a non-parametric classification and regression technique. It's one of the most basic machine learning techniques. It's a paradigm of lazy learning with local approximation. KNN's core concept is to look at your area, suppose the test data point is comparable to them, and deduce the result (Han et al., 2016) .We seek k neighbors and make a forecast using KNN. Here we used the elbow method to find the best value of k.

3. **Random Forests**

   Random Forest uses many decision trees ensembled utilizing the "bagging approach"; It generates the result via a majority vote in classification, but it calculates the Mean in regression (Donges, n.d.). Random Forest produces a reliable, accurate model that can handle various input data types, including binary, categorical, and continuous characteristics.

Algorithm 3 ML Models

```
1: function $feature\_extract(inp\_folder)$
2:     $df \leftarrow empty\_csv()$                ▷ $empty\_csv$: generates empty csv.
3:     for $image$ in $inp\_folder$ do
4:         $image \leftarrow preprocess(image)$
5:         $first\_order, second\_order \leftarrow featureextract(image)$
6:         $df \leftarrow update(df, img, first\_order, second\_order)$    ▷ $update$: returns updated dataframe
7:     $svm\_acc, svm\_f1 = svm(df)$                ▷ SVM model
8:     $knn\_acc, knn\_f1 = knn(df)$                ▷ KNN model
9:     $rf\_acc, rf\_f1 = rf(df)$                   ▷ Random Forest model
10:    return $svm\_acc, svm\_f1, knn\_acc, knn\_f1, rf\_acc, rf\_f1$
```

2.3 Deep Learning Model

The deep learning technique make use of VGG-16 architecture [12]. VGG-16[12] consists of 16 learnable layers, of which 13 are the convolutional layers, and the other 3 are fully connected layers. After every convolution layer in the network, we have used ReLU functions which help to prevent the exponential growth in the computation required to operate the neural network and avoid vanishing gradients.

The SoftMax activation function is used at the output as it is a multiclass classification. It is considered one of the excellent vision model architectures to date. The unique thing about VGG16 is that instead of having a large number of hyper-parameters, they focused on having convolution layers of a 3x3 filter with a stride one and always used the same padding and max pool layer of 2x2 filter of stride 2.

The input image is resized into the standard size of 224*224*3 for this VGG-16 model. The following steps are used to develop the deep learning model for unprocessed and processed images.

1. **Obtaining the input dataset**

In this step, input images of different classes are obtained. The dataset had images which were saved in a zipped file. We extracted the file and manually segregated the files into subfolders based on their labels.

2. **Data Arrangement**

   In this step, the image data we obtained is collected and stored in the variable using the Pandas library containing the image directory in the first column and the image categories in the second column. The classes used are 0: Normal, 1: Lung Opacity, 2: Viral Pneumonia, and 3: Covid.

3. **Splitting of data to train, test, and validation:**

   The variable in which the directory of images stored and the respective categories are then split to train dataset as 80%, test dataset and validation dataset each of 10%. Train and validation dataset is given to train the model and check the accuracy of both train and validation to see how the model is getting trained and working for the validation data simultaneously.

4. **Model Designing**

   The standard model VGG-16 is used for this classification model (*Keras* n.d.). It is used because the model can highly distinguish the image of different classes.

5. **Model Training**

   To train the model and observe how the model behaves when we use different optimizers, we have given two optimizers to train the model: Stochastic Gradient Descent and ADAM optimizers. They are used because ADAM is the best optimizer of all, and it has a high convergence rate and faster computation. Stochastic Gradient Descent is also a good optimizer that performs well, equivalent to the ADAM optimizer. Our reference states that it generalizes better than ADAM, so to observe the difference in the model, we have done with these two optimizers.

   (a) Plotting the Training and validation accuracy of both the models.

   (b) Plotting the Training and validation accuracy of both the models.

6. **Prediction** Using the 10% of the test data, prediction of the data is done and compared with the actual category and plotting the confusion matrix to obtain the performance metrics of the model performance.

3 Experimentation and Results

3.1 ML models

Several experiments were performed to find the robustness of our model and to check how it impacted our final F1 score and accuracy. Some key experimentations were:

1. We change the train-test split and see how their variation impacts our model efficiency.
2. In the KNN method, you are changing the distance metric from Euclidean to Manhattan and Minkowski and finding out the optimum level of nearest neighbors to be considered.
3. In SVM, we change the Gaussian kernel's soft-margin constant and width parameter.
4. In Random Forest, changes involved changing the loss function from entropy to Gini, the number of estimators, and the maximum number of features in a tree.

After performing these experimentations, the optimum hyperparameters were chosen, which led to the following conclusion:

	Random Forest	KNN	SVM
Accuracy	**0.71**	0.54	0.59
F1 Score	**0.70**	0.52	0.56

Table 1: Comparison of different ML methods in terms of accuracy and F1 score.

3.2 Deep Learning model

Here again certain experiments were carried out to test the model robustness. Some key experimentations were:

1. The optimizers were used while weighing their time complexity and computational cost. We choose Stochastic Gradient Descent and Adam Optimizer for our model. It was found that for a smaller number of epochs SGD was performing better but when we moved to higher number of epochs, Adam optimizer started to perform better.

2. We changed the train test split to find the one which suits us the best. Finally, we settled at 80:10:10 train test and validate split.

We obtained the following results after performing the above experimentations:

	Accuracy	F1 Score
SGD Optimizer	0.76	0.76
Adam optimizer	**0.84**	**0.84**

Table 2: Comparison of different DL methods in terms of accuracy and F1 score.

3.3 Performance of Dataset Preprocessing

We also decided to take a step further and did feature extraction in unprocessed image in case of supervised Image without segmenting the image to supervised learning models and gave segmented image to the deep learning model to find out how they perform to deep learning models. This is the result we got:

ML Models	Random Forest	KNN	SVM
Unprocessed Dataset	**0.73**	**0.55**	**0.53**
Processed Dataset	0.71	0.52	0.54

Table 3: Comparison of performance of different ML methods on processed and unprocessed data in terms of accuracy and F1 score.

Deep Learning Models	SGD Optimizer	Adam optimizer Score
Unprocessed Dataset	**0.84**	**0.84**
Processed Dataset	**0.76**	**0.76**

Table 4: Comparison of performance of different DL methods on processed and unprocessed data in terms of accuracy and F1 score.

The above result shows there have been little or negative impact after segmenting the lungs while there has been steep decrease in case of deep learning model. A possible reason can be there might be loss of data after segmentation. The pixels are dropped which might have resulted in incomplete images.

4 Future Work

4.1 Machine Learning Models

We used three different machine learning algorithms and achieved a f1-score of 0.69 in the best case. However, there were certain steps which we can try to improve our model further.

1. We can try ensemble learning methods based on multiple models to further improve our scores.
2. The mask we created was based on the basic principle that denser objects like bones absorb much of the radiation and appear white. We can create a better mask by using different filtering and segmentation techniques.
3. We can also use SIFT (Scale-invariant feature transform) to segregate the lungs which will hopefully improve our model further.

4.2 Deep Learning Models

1. The Deep learning model can be further developed by increasing the total epochs and by fine-tuning the parameters. It will help this model grow into real-time prediction model.
2. Several other external factors that depend on this covid-19 classification while developing into the real-time classification can be studied and identified at the top level and can implement in the model by eliminating these factors so that the accuracy model' s accuracy can be highly improved.

References

Aggarwal N. and K. Agrawal R. (2012). "First and second order statistics features for classification of Magnetic Resonance Brain Images". *Journal of Signal and Information Processing* 03.02, pp. 146–153. DOI: 10.4236/jsip.2012.32019.

Chowdhury M. E. H., Rahman T., Khandakar A., Mazhar R., Kadir M. A., Mahbub Z. B., Islam K. R., Khan M. S., Iqbal A., Emadi N. A., Reaz M. B. I., and Islam M. T. (2020). "Can AI Help in Screening Viral and COVID-19 Pneumonia?" *IEEE Access* 8, pp. 132665–132676. DOI: 10.1109/ACCESS.2020.3010287.

Donges N. (n.d.). *A complete guide to the random forest algorithm.* URL: https://builtin.com/data-science/random-forest-algorithm.

Han Y., Li J., Li J.-Z., Xing H.-W., Yang A.-M., and Pan Y.-H. (2016). "Demonstration of SVM Classification Based on Improved Gauss Kernel Function". *First International Conference on Real Time Intelligent Systems.* Springer, pp. 189–195.

Hasoon J. N., Fadel A. H., Hameed R. S., Mostafa S. A., Khalaf B. A., Mohammed M. A., and Nedoma J. (2021). "COVID-19 anomaly detection and classification method based on supervised machine learning of chest X-ray images". *Results in Physics* 31, p. 105045. DOI: https://doi.org/10.1016/j.rinp.2021.105045. URL: https://www.sciencedirect.com/science/article/pii/S2211379721010342.

Keras (n.d.). URL: https://keras.io/.

Ozturk T., Talo M., Yildirim E. A., Baloglu U. B., Yildirim O., and Rajendra Acharya U. (2020). "Automated detection of COVID-19 cases using deep neural networks with X-ray images". *Computers in Biology and Medicine* 121, p. 103792. DOI: 10.1016/j.compbiomed.2020.103792.

Rahman T., Khandakar A., Qiblawey Y., Tahir A., Kiranyaz S., Abul Kashem S. B., Islam M. T., Al Maadeed S., Zughaier S. M., Khan M. S., and Chowdhury M. E. (2021). "Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images". *Computers in Biology and Medicine* 132, p. 104319. DOI: https://doi.org/10.1016/j.compbiomed.2021.104319. URL: https://www.sciencedirect.com/science/article/pii/S001048252100113X.

Skimage (n.d.). URL: https://scikit-image.org/docs/stable/api/skimage.html.

Tahir A. M., Chowdhury M. E. H., Qiblawey Y., Khandakar A., Rahman T., Kiranyaz S., Khurshid U., Ibtehaz N., Mahmud S., and Ezeddin M. (2022). *COVID-QU-Ex Dataset.* DOI: 10.34740/KAGGLE/DSV/3122958. URL: https://www.kaggle.com/dsv/3122958.

Taneja S., Gupta C., Aggarwal S., and Jindal V. (2015). "MFZ-KNN—A modified fuzzy based K nearest neighbor algorithm". *2015 International Conference on Cognitive Computing and Information Processing (CCIP).* IEEE, pp. 1–5.

Code

1. **Machine Learning Techniques**
2. **Deep Learning Technique: Processed Dataset**
3. **Deep Learning Technique: Unprocessed Dataset**