Web Based COVID Detection System using Deep Learning

S Srivarshan¹, Prithvi Seshadri¹, E Kaarthikand¹, A Vijayalakshmi¹,*
¹Vellore Institute of Technology, Kelambakkam-Vandalur Road, Rajan Nagar, Chennai, Tamil Nadu, India 600127
*
vijayalakshmi.av@vit.ac.in

Abstract. Currently COVID-19 is a disease that is ravaging the entire globe. Generally, people affected with COVID-19 will come down with a mild to moderate respiratory illness. Detecting COVID-19 has become a major concern in hospitals due to the sheer number of people who claim to have been suffering from the symptoms. This work presents a solution to this problem where the patient can determine whether he/she has COVID-19 or not. This research work tries to identify whether a person is infected with COVID or not by processing the X-Ray scan of the chest area using Deep Learning with the aid of a Neural Network. X-Ray images obtained from a GitHub repository have been used to train the model. Then the model can predict using the X-Ray image obtained from the user. A web application has been developed, to make this process seamless and efficient. The trained neural network is used to process the given image. If the person is infected with COVID, the application makes use of an advanced AI searching algorithm to find the most suitable physician based on the requirements of the user.

1. Introduction

Research carried out by various organizations and institutions has shown that there are dozens of viruses in the coronavirus family. Of these, approximately seven are said to be extremely dangerous to human beings [1]. It was initially speculated that mammals such as bats are the main reasons for the propagation of these viruses to the bodies of human beings [2]. The basic structure of these coronaviruses [18] is shown in figure 1. In December 2019, several cases of the virus were reported for the first time in Wuhan, China. Coronavirus causes acute respiratory syndrome in human beings and has evolved into a global pandemic. It was the World Health Organization, WHO, that declared it to be a pandemic and termed it as the COVID-19 virus [3]. Various Research and Development centres around the world scrambled to find an effective way to diagnose the virus and to find an efficient vaccine for it.
This field of research was not only limited to the medical or bio-technical fields but is also the main concern for researchers from other avenues like Data Analysis, Machine Learning and Artificial Intelligence. They too are lobbying for effective solutions to combat the various complications that arise with a pandemic on such a large scale. The COVID-19 virus has now evolved to be the major contributor to changes in the world mortality rate. The death count of the coronavirus now has touched a peak of 41.4 Lakhs worldwide. The COVID-19 virus when it enters the respiratory tract of a human being is known to critically affect the lungs and cause symptoms similar to but more severe than Pneumonia. The lungs start filling up with a liquid, inflammation starts occurring and patches known as ‘Ground-Glass Opacity’ are known to be formed. It is a known fact that it is extremely difficult to diagnose a person with COVID-19 as it can be easily mistaken for other similar viruses. Therefore, the research was concentrated on finding a way to come up with a solution that was not only efficient but also easily available to the masses. Even though plenty of effort has gone into finding effective solutions to prevent the spread of the virus the only effective solution found was to practice social distancing and wear masks. Major metropolitan cities were also subjected to lock-downs and people were restricted from venturing outside. The reaction is that this has severely affected the country’s economy and also had an extreme psychological impact on the mental health of citizens. The amount of people suffering from the symptoms of COVID-19 is also increasing exponentially around the world. The most severely affected countries such as the United States, Brazil, and India have declared catastrophic effects on their GDP. Therefore, it has become very important to develop a system that uses artificial intelligence and machine learning to quickly diagnose COVID cases. Another important factor was that this system can be easily available to anyone who wishes to use it irrespective of platform and hardware capabilities. The development of such a system would help to detect new cases of COVID fast and effectively and would help in avoiding this global pandemic. Deep Learning techniques using neural networks to diagnose and predict have been game-changing in various research avenues. The medical field especially has various datasets such as retinal images, brain MRI’s and chest X-rays that could be used to train such models and efficiently diagnose diseases. X-rays also provide a cheap and efficient method to scan various paths of the human body in hospitals. The interpretation of these scans is generally done by experts in the radiology department. If the system could make use of these cheap scans and interpret them without the help of the experts then it would be an extremely efficient way to combat the current pandemic [4]. This would also be of help in
countries where X-rays are quite common but experts to interpret them are sparsely located. In view of the above issues, a system has been developed that uses deep learning to analyse and interpret X-ray images of the chest and predict if the person is infected with the virus. After studying various neural network models, it is found that convolutional neural networks are very effective in analysing images [5]. Thus, convolutional neural networks were decided to be used for building the model. Since the lungs have been identified as the main infection area for the COVID virus, analysis of the X-rays can help in identifying whether the person is infected by the virus. Figure 2 shows the chest X-Ray of non-COVID patients and COVID infected patients. The primary objective of the research is to develop a system that uses a convolutional neural network trained in X-ray images of healthy lungs and lungs affected by the coronavirus. The model will then be used to predict whether the person is infected with the virus. The system would also employ the usage of an advanced AI searching technique that would recommend nearby doctors to people who use the system.

![Figure 2. (a) Non-COVID Chest X-Ray (b) COVID Chest X-Ray.](image)

The model can quickly predict COVID19 cases by scanning for features such as blurring or white spots on the patient's chest X-ray image. The major parts of the proposed work can be summarized in the following lines. A CNN based on the LeNet5 architecture that can speculate the presence of Coronavirus in patients through X-ray images. An advanced AI searching technique based on a mixture of the A* algorithm and Breadth-First algorithm to recommend doctors near the patient based on a predefined set of parameters. A web application to host these services that can be accessed from all parts of the globe. The proposed model effectively detected the presence of COVID-19 with an
accuracy of 98%. The details of implementing the system, the difficulties faced and the conclusion are discussed in the following sections.

2. Literature Survey

Xiang Liu et al. [11] use various algorithm to find the most optimal path in search and rescue. Depth-first search and A-star with and without heuristic functions are used to determine the final path. Sousa et al. [12] use different training algorithms such as SVM, K-Nearest Neighbor, and Naïve Bayes to diagnose pneumonia from radiographic images through computer-aided systems in which SVM is identified to perform the best. Turker et al. [13] form a model inclusive for all genders, to detect COVID-19 with residual exemplar local binary pattern (ResExLBP). Li et al. [14] differentiate between pneumonia and non-pneumonia related lung diseases using deep learning model COVNet on CT scans. Ozturk et al. [15] use DarkNet model for binary and multiclass classification with varying accuracy for COVID-19. Wang et al. [16] introduced the concepts of transfer learning with a novel selection algorithm and fusion approach. The accuracy of multiclass classification average at 96% for 4 classes. Khan et al.[17] used CNN with pretrained Xception layer to identify COVID from x-rays. Mporas et al. [6] evaluated several popular deep learning CNN models that were pre-trained in a transfer learning environment for analysing X-ray images of the chest and predicting COVID-19 from them. This paper employed the usage of two datasets that were publicly available and used them in different setups. Some setups were based on testing each of them separately and some on mixing them both and testing them. Ismael et al. [7] elaborated on the implementation of deep learning methodologies, deep feature extraction, fine-tuning the parameters of pre-trained Convolutional Neural Networks, and the complete training of a vanilla Convolutional Neural Network model to be implemented in the classification, prediction, and labelling of chest X-rays of patients both healthy and affected by COVID-19. The various pre-trained deep learning models used were VGG16, VGG19, ResNet18, ResNet50 and ResNet101. It also describes how a Support Vector Machine classifier was used to classify the deep features. Ducho et al. [8] demonstrated the use of a grid map to plot the path of a mobile robot. It also performs several assumptions and additions to the vanilla A-star algorithm. These changes are primarily concerned with the algorithm's processing time and path optimality.

3. Proposed Work

Most of the existing works use different kinds of models such as ConvNet, CoroNet, VGG, etc., and have accuracy around 90 percentage with either an increment or decrement of 2 percent. This work has an accuracy of 97 percent due to efficient pre-processing and optimal use of necessary algorithms. Additionally, people can easily access the model through the website created and the doctor recommendation system, which uses A* algorithm, which is area-specific adds to its versatility.

3.1. Datasets Used

The data set for this work was obtained from Kaggle [10]. The X-ray images were split into two directories, test and train. Each testing and training directory is divided into 3 further directories. They are COVID-19, normal and pneumonia [9]. All these directories contained X-ray images of the chest region of the patients suffering from the respective ailments. However, for the scope of this work, the pneumonia directories in both the test and train directories were dropped. The other two directories COVID-19, normal was used for training the convolutional neural network. The X-rays that the model was trained on were 1726 in number. 460 of the images were from COVID-19 patients, while the rest were normal people's chest X-rays. The structure of the dataset directory is shown in figure 3. The number of chest X-ray images used for testing and validating the trained model was about 433. Of these 433 images, there were 116 photos of COVID-19 patients and the rest were chest X-ray scans of healthy people. The sizes of the individual images range from a few 100 kilobytes to a few megabytes.
The comparatively large size of some of these images established a strain on the hardware while training the model and increased the overall computation time.

![Figure 3. Structure of Dataset Directory.](image)

### 3.2. Data Pre-processing
Before the vanilla X-ray images can be used to train the deep learning neural network model, some operations need to be performed on it so that the model will accept these images as training data. This step is crucial in all types of machine learning algorithms and is popularly called pre-processing. There are many reasons why data pre-processing is a crucial step in machine learning. Real-world data occasionally contain unwanted data, missing values and noise. They may also sometimes contain data that is in an unusable format. Data pre-processing scans the data for these anomalies and performs the required tasks to clean the data and format it in such a way that it is suitable for a machine learning model to train or test upon. This process also exponentially increases the accuracy and efficiency of the convolutional neural network used in this article. Generally, data pre-processing consists of the following steps: obtaining the data set, importing libraries, importing datasets, locating missing data, encoding of categorical data, dividing the data set into training and test sets, and feature scaling.

### 3.3. Getting the Dataset
As mentioned in the previous section the data set was obtained from a Kaggle repository. The data set was downloaded in the form of a zip file and extracted into a local directory in the host machine. Saving the data set directly in the machine where the training will be done is an efficient method as it would greatly reduce the computational and training time.

### 3.4. Importing the Libraries
Python makes use of features called libraries to abstract large amounts of code and presents them in a user-friendly format. Machine learning also comes under the previous definition of large amounts of code that are too complex to understand or implement. Therefore, various Python machine learning libraries are available which makes the process of creating a model, training it and performing validation easy to implement. This research, makes use of a library created by Google known as Keras to create the base framework for the deep learning model and to perform predictions using it. This Keras library makes use of a Python framework known as TensorFlow. The TensorFlow website describes it as “an end-to-end open-source machine learning platform”. It provides a comprehensive and inflexible set of tools, libraries and resources to enable researchers to easily build and deploy
machine learning applications. A package named scikit-learn was utilised to offer the functions needed to examine the model's performance and produce its performance metrics. This library is built using other libraries like NumPy, SciPy and matplotlib. It's an open-source toolkit that provides easy-to-use tools for analysing predictive data. The PIL library has been imported to make use of its advanced image manipulation functions. PIL is an abbreviation for Python imaging library. A library called seaborn which provides functions to easily plot and create graphs out of data has been used for data visualization. It provides the user with an advanced Python API for drawing eye-catching, beautiful and informative statistical graphics.

3.5. Feature Scaling
This is an approach for standardising and bounding the independent features obtained from a dataset within a particular range. This is especially useful when dealing with images. The images from the data set obtained are of various sizes and dimensions. Ideally, deep learning models expect all their input to be of the same input size. So, to make this possible various scaling techniques were used to standardize and compress the input features, which in this case are X-ray images of the chest. The scaling method used here is called ImageDataGenerator. This function is obtained by importing the pre-processing module of the Keras Python library. This method generates tensor image data in batches with real-time data augmentation capabilities. While the deep learning model is still processing the input during training, it augments images in real time. This has two main advantages. The first one is that it makes the model robust. The other advantage is that it uses a minimal amount of overhead memory. This image augmentation is a process of applying different techniques to the same images which result in multiple varied copies of the same image. This is especially crucial when working with a limited data set. The different augmentations applied are transformations like shifting, rotating, flipping etc. These provide a new perspective to the model during training.

3.6. Encoding the Categorical Data
Before the data can be used for training or testing it needs to be encoded with a label. This need not be done when a CSV file is available with the name of the image and its label. However, the dataset obtained does not provide the luxury of such a CSV file. So, in this research, the labels had to be encoded via the directories in which the images were present. This is a very tedious process. However, this process can be abstracted using a function called flow\_from\_directory(). This function is part of the ImageDataGenerator class seen in the Feature Scaling section. This method is used in the script to read the images directly from the directory and encode them the label of the directory in which they are present. Here, the size of the input image has also been restricted to 150x150 pixel dimensions. The class mode is also set to categorical as there are different categories or labels, “COVID19” and “NORMAL”.

3.7. Convolutional Neural Network
The deep learning model used in this article is a convolutional neural network. It is a form of deep learning approach that assigns weights and biases to various features in an image when given an image as input. These objects are generally the pixels that constitute the base of the image. An advantage of this algorithm is that it requires low amounts of pre-processing when compared to other classification algorithms. While other algorithms create filters by hand engineering and training with data, convolutional neural networks can learn and obtain these characteristics or features. A convolutional network's architecture is similar to the human brain's neuron connection pattern. In its most basic form, it is based on the visual cortex's orientation and arrangement. Using numerous relevant filters, a convolutional network can efficiently capture the temporal and spatial characteristics in an image. Because the number of features used in the training is reduced and the weights assigned are reused, the architecture can better match the image data set. In the first section of the convolutional layer, there is an element that performs the convolution operation. The kernel or filter is the name for this element. The filter shifts during training and performs a matrix multiplication operation over a portion of the
image during each shift. Extracting high-level features from the input image, such as edges is the fundamental goal of the convolutional operation. However, the model does not have to be confined to just a single layer, and in practise, just the first layer is capable of collecting lower-level features like colour, orientation, edges etc. and more layers are stacked to the neural network, which adjusts to high-level features as well, resulting in a deep learning model that has a thorough grasp of the data set's images. Sometimes the dimensionality of the output image may be different from the dimensionality of the original image. This anomaly is corrected by using something known as Padding. After the convolutional layer, comes the pooling layer. The pooling layer is in charge of reducing the number of the spatial features. This is done to reduce the amount of computing resources needed to process the images. This is also known as dimensionality reduction. It also extracts the most important features and keeps the model's training process running efficiently. The pooling layer used here is called a Max pooling layer. The highest value from the part of the image captured by the filter is returned by this Max pooling layer. Max pooling additionally reduces the amount of noise produced during training. After all these layers comes the fully connected layer. The model can now learn nonlinear sequences of high-level features due to the addition of a fully linked layer. After all of these processes, the original image is transformed into a compatible format for the multi-level perceptron. This image is now flattened and sent into the neural network as a column vector. The SoftMax algorithm is then used for the model to identify the dominating and lower-level features present in the image over numerous iterations or epochs. There are several CNN architectures to choose from, each with its own set of advantages and disadvantages.

3.8. Proposed Architecture

The architecture of the model put forth in this work consists of four Convolutional 2D layers and MaxPooling 2D layers with ReLU as the activation function. The first convolutional layer is of size 64 and has a 3x3 filter size. Processing the image using this layer provides a feature map with size 64. This is followed by a max pooling layer with a kernel of 2x2 size which reduces the dimensions of the features. Furthermore, three convolutional layers with sizes 64, 128, 128 respectively and filter sizes 3x3 are added to retrieve the finer details from the image features. Each of these layers are followed by a max pooling layer with kernel sizes 2x2. A Flatten layer is imposed on this block to act as a convenience layer by transforming the output of the above block into a 1-dimensional vector. The Flattening converts the n-dimensional matrix obtained from the above layers into a 1-dimensional array, which can now be passed to the fully connected layer. To prevent the deep learning model from overfitting, a Dropout layer is added to it. This layer performs a 50% decay on the weights associated with the neurons that have a lesser contribution. This aids in reducing the error caused by generalization of the network. Now the features, after the weight decay, are passed to the Dense layer. This layer is a fully connected layer with 512 neurons. It is followed by the ReLU activation function. The final layer is also a Dense layer with 2 neuron and softmax activation function applied to it. This layer is the output layer. The ReLU activation function preserves the sparseness of the network. It also aids in decreasing the probability of the model not being able to properly propagate the gradient information between neurons. The final Dense layer has a softmax activation function applied to it to address the issue of the dying ReLU problem where it always outputs the same value and prevents additional learning. The Dense layers acts as a classification layer and trains on the high-level features obtained from the convolutional layers. The optimizer used is RMSProp and categorical cross-entropy is used as the loss function while training the network. RMSProp was chosen as the optimizer due to its high convergence rate when compared to other optimizers. This faster rate is due to the fact that it utilizes a gradient which degrades exponentially by its square. The history of the gradient is also removed in further iterations. The architecture of the model can be viewed at figure 4.
3.9. Modified A* Algorithm

The A-star algorithm is an advanced AI searching technique. This algorithm employs a hybrid of heuristic and greedy search techniques. As it is evaluated using a heuristic function, it is sometimes referred to as a best first algorithm. The node with the lowest heuristic value is chosen during every iteration. The method was chosen because of its low time complexity and high space complexity. The heuristic functions that are used as parameters can be integrated, modified, or added to another function. This provides a plethora of options for fine-tuning the underlying algorithm. One such change has been used in this study. Some adjustments had to be made to the vanilla algorithm in order to use it for the application. The A * algorithm was utilised to search for and select the best doctor based on the parameters provided by the user. These factors include the doctor's distance from the user and the consultation charge requested by the doctor. In practise, however, each has distinct weights, and the choice is not evenly split between the two. To account for this oddity, the heuristic function was changed. The distance between the user and the doctor is given more weight. Just 10% of the weight that had been ascribed to a distance was attributed to the physician's consultation fee. As a result, the user receives a better suited recommendation.

The heuristic function can be denoted as:

\[
H(x, y) = d(x, y) + (0.1 \times f(y))
\]

Where,
\(x\) → User
\(y\) → Doctor
\(H(x, y)\) → Heuristic Function
\(d(x, y)\) → Distance between user and doctor
\(f(y)\) → Consultation fee charged by the doctor

\[ Figure 4. Proposed Architecture. \]
4. Results and Discussions

In this study, a Convolutional Neural Network model was created that takes a chest X-Ray image as the input. Since the dataset used was of a limited size, data augmentation was performed on the images to increase the size of the training data. These images are then processed by the convolutional layers to extract the features. The dense layers are trained on these features, which in turn provide the output. The Convolutional Neural Network model is essentially a sequential model to which convolutional layers, maxpooling layers, dropout layers, and fully connected layers were added. The initial network with only one convolutional layer and a dense layer, gave an accuracy of 86%. Whereas after adding additional layers to the neural network as part of the fine-tuning process, the performance improved. By adding three additional convolutional and maxpooling layer, and by adding a dropout layer to prevent overfitting, the model was able to achieve an accuracy of 97%. The learning process of the model is shown in figure 6 and figure 7. The limited size of the dataset however, could be a factor for the high accuracy of the model. But this could not be avoided because the problem statement chosen was a relatively new area of research with few usable datasets available. Figure 6 depicts the confusion matrix of the model while testing it on the test dataset. The performance metrics of the model can be viewed in table 1. The model's prolonged training time, which appeared to be a severe flaw, was easily addressed by using a GPU instead of the host machine's CPU. Using cloud computing technologies, this disadvantage can be fully avoided. The model's deployment in a cloud computer, however, was outside the scope of this study. By applying machine learning to clean the pictures prior to training, the model's bias toward marks in the input x-ray images can be avoided. Because this approach changes the features required to determine the outcome (the presence of Covid 19), the solution was abandoned for the sake of accuracy. As a result, the model was constructed, trained, and deployed with an accuracy of 97%.
### Table 1. Performance Metrics of the model.

| METRICS                        | COVID19 | NORMAL |
|--------------------------------|---------|--------|
| Accuracy                       | 0.97    | 0.97   |
| Precision                      | 1.00    | 0.97   |
| Precision (Macro Average.)     | 0.98    | 0.98   |
| Precision (Weighted Average.)  | 0.98    | 0.98   |
| Recall                         | 0.91    | 1.00   |
| Recall (Macro Average.)        | 0.95    | 0.95   |
| Recall (Weighted Average.)     | 0.97    | 0.97   |
| F1 Score                       | 0.95    | 0.98   |
| F1 Score (Macro Average.)      | 0.97    | 0.97   |
| F1 Score (Weighted Average.)   | 0.97    | 0.97   |

### Figure 6. Learning Process of the model: Accuracy vs Epochs.
5. Conclusion and Future Scope

In this paper, it is successfully demonstrated on how deep learning in the form of a CNN can be used for the prediction of Coronavirus in patients by analysing X-ray images of the lungs. The experimental results are highly promising and give an insight into the scope of this avenue. The accuracy of the model, after pre-processing the data, feature scaling and finally performing the CNN analysis, is relatively high by predicting the right diagnosis around 97%. Although CNN works well, as the size of the data set increases, the accuracy decreases and the time taken for computation increases. Other models which use VGG or Segmentation can develop a lot more layers, which will help in increasing the size of data set and its accuracy. Additionally, the current model uses X-rays and converts them into black and white pixels. Only then is the image recognition algorithm run. This is a huge disadvantage as it can only be used to recognize ailments which use X-rays. But recently, so many other tests and equipment are present to identify a problem which will not work well with the model. For instance, CT scan or even MRI which is in black and white may face some problems with the current model. The scope of the model will greatly improve if it is able to use other forms of scans as well.

References

[1] www.niaid.nih.gov/diseases-conditions/coronaviruses
[2] Fan Y, Zhao K, Shi ZL and Zhou P 2019 Bat coronaviruses in China. Viruses11(3)p210
[3] Bilal M, Munir H, Nazir MS and Iqbal HM 2020 Persistence, transmission, and infectivity of SARS-CoV-2 in inanimate environments Case Studies in Chem. and Env. Eng.2p100047
[4] Mahmud M, Kaiser MS, McGinnity TM and Hussain A2021Deep learning in mining biological data Cognitive Computation13(1) p1-33
[5] Panwar H, Gupta PK, Siddiqui MK, Morales-Menendez R and Singh V 2020 Application of deep learning for fast detection of COVID-19 in X-Rays using nCOVnet Chaos, Solitons & Fractals138 p109944.
[6] Mporas I and Naronglerdrit P 2020 Covid-19 identification from chest x-rays2020 Int. Conf. on Biomedical Innov. and Appl. p 69-72
[7] Ismael AM and Şengür A 2021 Deep learning approaches for COVID-19 detection based on chest X-ray images Expert Syst. with Appl.164 p114054
[8] Duchoň F, Babinec A, Kajan M, Beňo P, Florek M, Fico T and Jurišica L 2014 Path planning with modified a star algorithm for a mobile robot *Procedia Eng.* **96** p59-69
[9] Hall LO, Paul R, Goldgof DB and Goldgof GM 2020 Finding covid-19 from chest x-rays using deep learning on a small dataset (*arXiv preprint arXiv:2004.02060*)
[10] www.kaggle.com/prashant268/chest-xray-covid19-pneumonia
[11] Liu X and Gong D 2011 A comparative study of A-star algorithms for search and rescue in perfect maze 2011 *Int. Conf. on Electric Information and Control Eng.* p 24-27
[12] Sousa RT, Marques O, Soares FAA, Sene II, Oliveira LL and Spoto ES 2013 Comparative performance analysis of machine learning classifiers in detection of childhood pneumonia using chest radiographs *Procedia Computer Science* **18** p2579-2582
[13] Tuncer T, Dogan S and Ozyurt F 2020 An automated residual exemplar local binary pattern and iterative ReliefF based COVID-19 detection method using chest X-ray image *Chemometrics and Intelligent Lab. Sys.* **203** p104054
[14] Li L, Qin L, Xu Z, Yin Y, Wang X, Kong B, Bai J, Lu Y, Fang Z, Song Q and Cao K 2020 Artificial intelligence distinguishes COVID-19 from community acquired pneumonia on chest CT *Radiology*
[15] Ozturk T, Talo M, Yildirim EA, Baloglu UB, Yildirim O and Acharya UR 2020 Automated detection of COVID-19 cases using deep neural networks with X-ray images *Computers in bio. and med.* **121** p103792
[16] Wang SH, NayakDR, Guttery DS, Zhang X and Zhang YD 2021 COVID-19 classification by CCSHNet with deep fusion using transfer learning and discriminant correlation analysis *Information Fusion* **68** p131-148
[17] Khan AI, Shah JL and Bhat MM 2020 CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images *Comp. Methods and Programs in Biomed.* **196** p105581
[18] www.nationalgeographic.com/science/2020/02/what-happens-to-coronavirus-covid-19-in-warmer-spring-temperatures/