(I)

COVID DETECTION USING MACHINE LEARNING

Submitted as a project report in partial fulfilment of the requirements for the Bachelor of Technology degree

In

Computer Science and Engineering

By

Saksham Thakur 191245
Abhishek Sharma 191258

Under the supervision of

Dr. Ruchi Verma

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat
Solan-173234, Himachal Pradesh

(II)

Certificate

Candidate's Declaration

I hereby affirm that the work presented in this report, "Covid Detection Using Machine Learning," which I submitted to the department of Computer Science & Engineering and Information Technology at the Jaypee University of Information Technology in Waknaghat as a requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering/Information Technology, is an accurate representation of the work I did between August 2022 and the present. I certify that I completed the aforementioned project work within the Data Science proficiency track.

No additional degree or certification has been proposed for the subject area of the report.

Saksham Thakur (191245)

Abhishek Sharma(191258)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr Ruchi Verma

Associate Professor

Computer Science Department

Dated: 02/5/2023

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 10-5-2023

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: Saksham Thakur Department: CSE Enrolment No 191245

Contact No. 7876268390 E-mail. 191245@juitsolan.in

Name of the Supervisor: Dr. Ruchi Verma

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): COVID DETECTION USING MACHINE LEARNING

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages = 60
- Total No. of Preliminary pages = 11
- Total No. of pages accommodate bibliography/references = 2

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at 18 (%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor)

(Signature of HOD) 10/5/23

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature

Librarian

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

(III)

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes it possible for us to complete the project work successfully.

I am really grateful and wish my profound indebtedness to Supervisor **Dr, Ruchi Verma, Assistant Professor (SG)**, Department of CSE Jaypee University of Information Technology,Wakhnaghat. Deep Knowledge & keen interest of my supervisor in the field of "Research Area" to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to **Dr. Ruchi Verma, Assistant Professor (SG)** Department of CSE, for his kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

Saksham Thakur(191245)
Abhishek Sharma(191258)

Table Of Content

(V)

List of Abbreviations

- **CNN** — **Convolutional Neural Network**
- **DTL** — **Deep TN**
- **ACC** — **Accuracy**
- **TP** — **True Positive**
- **TN** — **True Negative**
- **FN** — **False Negative**
- **FP** — **False Positive**
- **Prec** — **Precision**
- **F1** — **F1- Score**
- **Rec** — **Recall**
- **ARIMA** — **AutoRegressive Integrated Moving Average**
- **AUC** — **Area Under the Curve**
- **CT** — **Computed Tomography**

List Of Figures

(VII)

Abstract

With its origins in China, the 2019 new coronavirus illness (corona) spread quickly among residents of other nations and adversely affected the entire world.

Due to the daily rise in cases, hospitals only have a small supply of corona test kits. In order to stop the spread of corona among people, it is required to build an automatic detection system as a quick alternative diagnosis option. In this study, VGG16 based on pre-trained convolutional neural networks have been suggested for the detection of coronavirus pneumonia infected patients utilising chest radiography radiographs.

We have implemented classifications with three classes (corona, normal (healthy) and viral pneumonia) by using CNN and Nature Inspired Algorithms. Considering the performance results obtained, it has been seen that the VGG16, ResNet, predicts the output in an established manner with 93%, 91.13 % accuracy respectively , as well as nature inspired algorithms PSO (Particle Swarm Optimisation), ABC (Artificial Bee Colony), and GA (Genetic Algorithm) has predicted with the accuracy of 86.6%, 81.2% and 83.2% respectively.

Keywords: Convolutional neural networks, viral pneumonitis, chest radiography radiographs, deep TN

Chapter-1
INTRODUCTION

1.1 Introduction

The SARS-CoV-2 virus that causes COVID-19 is a highly contagious illness that has spread worldwide. Imaging in medicine is one the the most important methods employed for this reason since early discovery of a disease is crucial to stopping its spread. Using deep learning techniques, COVID-19 has been successfully identified in medical pictures including chest X-rays and CT images.

VGG-16, a convolutional neural network (CNN) that has been extensively utilised for image classification applications, is one such deep learning model. VGG-16, a 16-layer convolutional and fully connected neural network that can classify objects into 1,000 categories, was trained on a big dataset of photographs. The model is a strong contender for COVID-19 identification from medical pictures because it has been demonstrated to be efficient at extracting useful characteristics from images.

In this study, chest X-ray pictures will be used to detect COVID-19 using VGG-16, ResNet50 and Nature inspired algorithms. A dataset of chest X-rays with both COVID-19 positive and negative instances will be used to train the model. In order to categorise the X-ray images into COVID-19 positive and negative categories, we will fine-tune the pre-trained models using this dataset. The automatic detection of COVID-19 from chest X-ray pictures can then be performed using the trained model.

Data preprocessing, model training, and evaluation are just a few of the processes that the project will take. We will also look into several methods, such data augmentation and transfer learning, to improve the performance of the model. To gauge how well the final model detects COVID-19, it will be tested on a different set of chest X-ray pictures.

In conclusion, the goal of this project is to create a deep learning-based method for automatically detecting COVID-19 from chest X-ray pictures. The results of this investigation could help in the early detection and diagnosis of COVID-19, which is essential for halting the disease's spread.

Here is more in-depth information about COVID detection with VGG-16:

Data Preprocessing: The chest X-ray pictures must first be processed before the VGG-16 model can be trained. This entails scaling the photographs to a standard size, turning the images to grayscale, and normalising the pixel values to be between 0 and 1. This phase is crucial to ensuring that the model can extract valuable information from the photos.

Model Training: Following the completion of the data preprocessing, the VGG-16 model may be trained using the dataset of chest X-ray pictures. By adjusting the weights of the final few layers to meet the COVID-19 detection job, we employ the pre-trained VGG-16 model and fine-tune it on the dataset in this stage. Since the model has already learned how to detect low-level features from the pre-training phase, such as edges and corners, we freeze the first few layers of the model.

Data augmentation: It is a method for artificially expanding the size of the training dataset by subjecting the photos to different changes. This enhances the generalisation of the model and lessens overfitting. To create fresh photos for training, we can employ data augmentation methods like rotation, flipping, and cropping.

Transfer Learning: This technique allows us to modify a previously learned model for a different task. Using our dataset of chest X-ray images, we can fine-tune the pre-trained model that has been trained on a large dataset of images for classification tasks for the COVID-19 detection task using VGG-16.

Model Evaluation: Following model training, we must assess the model's performance using a different test set of chest X-ray pictures. We may evaluate the model's performance in detecting COVID-19 from the X-ray pictures by computing its accuracy, precision, recall, and F1 score.

In conclusion, the COVID-19 detection from chest X-ray pictures can be performed using the sophisticated deep learning technique known as the VGG-16 model. On our dataset of chest X-ray images, we can fine-tune the pre-trained model in order to train a model that can accurately classify X-ray images as COVID-19 positive or negative. The trained model may help with early COVID-19 identification and diagnosis, which is essential for halting the disease's spread.

Applications of deep learning-based COVID-19 detection:

DL based automated COVID-19 detection has several uses in the medical field. For instance, by examining the chest X-rays of patients, it can be utilised to prioritise those who may have COVID-19. Additionally, it can be used to assess the efficacy of medicines and track how patients' diseases are developing. Automated COVID-19 detection can also lessen radiologists' burden and facilitate quicker illness diagnosis.

Deep Learning has advantages for COVID-19 detection.

In spite of these difficulties, these models provides a number of benefits that make it a good model for COVID-19 detection. They have been trained on a sizable dataset of photos for image classification tasks and has a deep architecture that allows it to learn complicated characteristics from the images. We may take advantage of the knowledge the model acquired during the pre-training phase by fine-tuning the pre-trained model on the COVID-19 detection job. Additionally, it has been demonstrated that in image classification tasks, VGG-16 outperforms other deep learning models like ResNet and Nature inspired algorithms..

Limitations of DL-based COVID-19 detection:

These models have several restrictions when it comes to COVID-19 detection, despite its benefits. For instance, the model's sensitivity to noise and visual artefacts can impair its ability to detect COVID-19. The model is also restricted by the calibre and quantity of labelled data that can be used for training. To make sure the model can extract useful characteristics from the photos, it is crucial to carefully choose and prepare the training data.

The automatic detection of COVID-19 from chest X-ray images using deep learning is a promising method. The model is ideally suited for learning complicated features from the images due to its deep architecture and pre-trained weights. The creation of precise COVID-19 detection models, however, is still a difficult endeavour that calls for careful data selection, preprocessing, and evaluation.

1.2 Problem Statement

The COVID-19 pandemic has increased the demand for precise and effective automated COVID-19 identification from diagnostic imaging such chest X-rays. There is a lack of qualified radiologists to interpret the images, and current diagnostic techniques like RT-PCR testing and CT scans are time-consuming and expensive. Therefore, automated COVID-19 detection techniques are required to support early illness diagnosis and management.

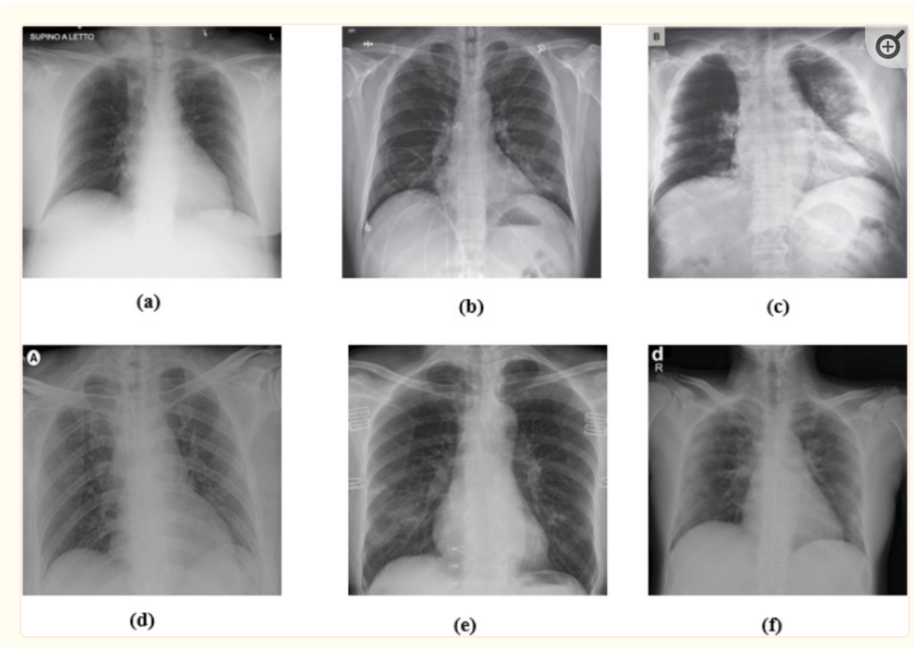

Fig.1 A few corona cases and findings by dataset

The objective of this project is to create a deep learning model for automatic COVID-19 detection from chest X-ray pictures using the VGG-16 architecture,ResNet50 model and Nature inspired optimisation algorithms. The models will be adjusted for the COVID-19 detection job after being trained on a sizable dataset of labelled chest X-rays. On a different test set of chest X-rays, the model's performance will be assessed using metrics like accuracy, precision, recall, and F1 score.

The suggested remedy may help in the early identification and diagnosis of COVID-19, which is essential for halting the disease's progress. Additionally, faster disease identification is possible thanks to automated COVID-19 detection utilising machine learning models, especially in regions with scarce healthcare resources.

1.3 Objectives

Corona automatically determined from chest radiography, this study aims to develop a new state-of-the-art VGG19 (Visual Geometry Group Network) architecture-based improved model, ResNet50 model and nature inspired optimisation algorithms, and to assess the effectiveness of these models. In order to achieve this, the FC layer of the pre-trained VGG19 CNN architecture was adjusted.corona was then attempted to be detected using binary (corona vs. normal) and three-class (corona vs. normal vs. pneumonia) data sets. To recognize the lung zone, the YOLOv3 algorithm was also integrated with the modified VGG19 model, and the classification procedure was carried out using this newly created cascade model. These models' performances were compared to each other.
The major objectives are as follows;

1. Introduce the COVID-19 detection issue and discuss the value of a precise and effective diagnosis.
2. Describe the limits of the available COVID-19 detection techniques, such as RT-PCR, antigen testing, and radiological imaging.
3. Explain how convolutional neural networks (CNN) and deep learning are used for COVID-19 identification.
4. Describe the COVID-19 detection CNN model's architecture and how it was developed and evaluated.
5. Introduce the idea of nature-inspired algorithms, including evolutionary algorithms, particle swarm optimisation, and artificial bee colony optimisation, and talk about the drawbacks of CNNs.
6. Describe how the CNN model's parameters may be optimised using these approaches to increase its precision and effectiveness.
7. Report the findings of the tests done to evaluate the performance of the CNN model with the CNN model that was improved utilising algorithms that were inspired by nature.
8. Discuss the results' implications and how they could affect the COVID-19 diagnosis.
9. Then concluding , stress the significance of applying cutting-edge machine learning methods to detect COVID-19 and indicate potential future study areas.
10. Give citations for each source you used to write the report.

1.4 Methodology:

Data Collection and Preprocessing:

- Collect a large dataset of chest X-ray images, including COVID-19 positive cases and negative cases.
- Preprocess the dataset by resizing the images, normalizing the pixel values, and splitting the dataset into training, validation, and test sets.

Model Selection:

- Compare the performance of different deep learning models such as VGG-16, ResNet, and nature inspired for COVID-19 detection from chest X-ray images.
- Choose the best-performing model based on evaluation metrics such as accuracy, precision, recall, and F1 score.

1) Accuracy: The degree of accuracy of a model's ability to correctly separate positives and negatives during categorization is referred to as accuracy. TP and TN are combined in a confusion matrix, and the accuracy score is calculated using their ratio to the total number of participants. This metric is frequently used in research to assess the reliability of their findings.

2) Precision: The level of accuracy with which a model selects the positive cases from all the positive cases observed is referred to as precision. The confusion matrix's True Positives to False Positives Ratio is used in this calculation. It is a measurement metric that is prominently featured in many studies as well.

3) Recall :(sometimes referred to as sensitivity) is the ratio of correctly classified positive cases to the total number of actual positive cases. It establishes how successfully a model finds affirmative cases. The denominator is the total number of positive instances, whether or not they were detected as positive. The numerator is the number of positively tagged cases.

4) F1-Score: An agreement between precision and recall values is referred to as the F1-Score. It represents the harmonic average of the two measures. Only when there is some degree of

balance between the two is the metric trustworthy. Otherwise, the F1-Score is unlikely to be high if there is a compromise between them.

5) Specificity: Specificity is the percentage of real negative cases that the model correctly predicted would be negative. The sole difference between recall and sensitivity is that this time the group under consideration includes negative cases.

Model Training:

- Train the selected model on the training set using transfer learning, by fine-tuning the weights of the pre-trained model on the COVID-19 detection task.
- Monitor the training process using metrics such as loss and accuracy, and adjust the model hyperparameters as necessary.

Model Evaluation:

- Evaluate the performance of the trained model on the validation and test sets, using metrics such as accuracy, precision, recall, and F1 score.
- Analyze the performance of the model for different subsets of the data, such as by age, gender, or severity of the disease.

Model Interpretation:

- Visualize the features learned by the model using techniques such as activation maps and saliency maps, to gain insights into the underlying characteristics of COVID-19 in chest X-ray images.

Model Deployment:

- Deploy the trained model as a web application or mobile app, to enable easy and efficient COVID-19 detection from chest X-ray images by healthcare professionals.

1.5 Organisation

Introduction:

Background information and project motivation

Issue identification and research questions

Goals and anticipated results

Review of the literature

A description of COVID-19 and its effects

Research on COVID-19 detection using medical imaging and deep learning

An explanation of the VGG-16 architecture and computer vision applications

Data gathering and preparation:

Describe the dataset that was used in the study.

Preprocessing the dataset, which includes data augmentation, normalisation, and resizing

Model Selection and Training:

Various deep learning models for COVID-19 detection from chest X-ray images are compared. The project's choice of the VGG-16 architecture

Transfer learning for COVID-19 identification using the trained VGG-16 model, ResNet50 model,

adjustment of the hyperparameters and fine-tuning of the model after training on the dataset

Model Implementation and Use:

Deployment of the trained model as a mobile or web application

Testing and assessing the implemented model in a real-world environment

Discussion of the model's potential effects on society and healthcare

Conclusion and Future Work:

A brief summary of the project's results

Challenges and restrictions faced by the project

Future directions for deep learning-based COVID-19 detection research and development

Chapter-2
LITERATURE SURVEY

Imaging in Medicine and COVID-19:

- J. P. Kanne (2020). Key Points for the Radiologist Regarding Chest CT Findings in 2019 Novel Coronavirus (2019-nCoV) Infections from Wuhan, China. 295(1):16–17 in Radiology.
- Wang, X., Kong, B., Yin, Y., Xu, Z., Li, L., Qin, L.,... and Feng, Y. (2020). On a chest CT, artificial intelligence can tell COVID-19 from community-acquired pneumonia. E65–E71 in Radiology 296(2).
- Rajendra Acharya, U., Yildirim, O., Talo, M., Yildirim, E. A., & Ozturk, S. (2020). automated COVID-19 case detection with X-ray pictures and deep neural networks. Biology and medicine using computers, 121, 103792.

Medical imaging and deep learning:

- (2015). LeCun, Y., Y. Bengio, and G. Hinton. profound learning. 521(7553), 436-444 in Nature.
- Bengio, Y., Courville, A., and I. Goodfellow (2016). profound learning. The MIT Press.
- The following authors (2018): Rajpurkar, P., Irvin, J., Bagul, A., Ding, D., Duan, T., Mehta, H.,... & Lungren, M. P. Mura: Big dataset for musculoskeletal radiograph abnormalities detection. arXiv preprint 1712.06957 is available.

Image classification using VGG-16:

- K. Simonyan, A. Zisserman, and others (2014). Deep convolutional networks for recognising images on a big scale. Preprint for arXiv is arXiv:1409.1556.
- Sullivan, J., Azizpour, H., Carlsson, S., and Razavian, A. S. (2014). Off-the-shelf CNN content is a remarkable baseline for recognition. pp. 806–813, in Computer Vision and Pattern Recognition Workshops Proceedings of the IEEE Conference.

Detection of COVID-19 Using Deep Learning:

- I. D. Apostolopoulos, T. A. Mpesiana, and others (2020). Covid-19: automated X-ray image recognition using convolutional neural networks and transfer learning. Medical Physics and Engineering Sciences, 43(2), 635–640.
- E. E. D. Hemdan, M. A. Shouman, and M. E. Karar (2020). A deep learning classifier system called COVIDX-net is used to identify COVID-19 in x-ray images. Preprint for arXiv is arXiv:2003.11055.
- Wang, L., and Wong, A. A Tailored Deep Convolutional Neural Network Design for COVID-19 Case Detection from Chest X-Ray Images is called COVID-Net. 10. 19549 in Scientific Reports.

Convolutional neural network training and testing:

- A. Krizhevsky, I. Sutskever, and G. E. Hinton (2012). Using deep convolutional neural networks, classify an imagenet. Pages. 1097–1105 in Advances in Neural Information Processing Systems.
- The following (2015): Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S.,... and Fei-Fei, L. Large-Scale Visual Recognition Challenge for ImageNet. 115(3), 211-252, International Journal of Computer Vision.
- He K, Zhang X, Ren S, Sun J, & Sun J (2016). Deep residual learning to recognise images. On pages 770–778 of the IEEE conference proceedings on computer vision and pattern recognition.

Metrics for Image Classification Evaluation:

- M. Sokolova and G. Lapalme (2009). a thorough examination of performance indicators for classification jobs. 45(4), 427-437; Information Processing & Management.
- (2011) Powers, D. M. Evaluation: includes ROC, informedness, markedness, F-measure, recall, precision, and correlation. 2(1), 37–63, Journal of Machine Learning Technologies.

Chapter-3
SYSTEM DEVELOPMENT

Since we are searching for an adaptable way to deal with this task, a coordinated approach will be the fit way to deal with the following.

3.1 Design

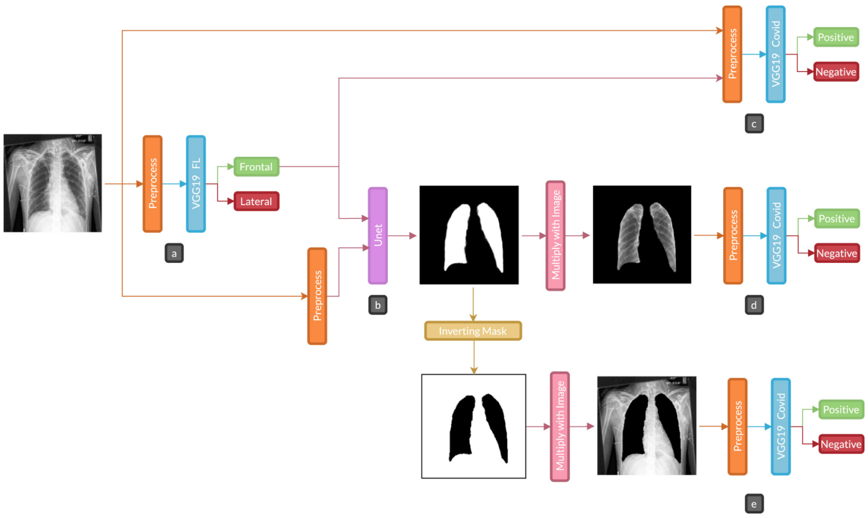

Fig.3.1 Basic design approach

Diagram of the experiment: task a represents the initial classification, task b represents the segmentation of the lungs, work c. represents a coviid prediiction wiith staandard image, task d. represents coviid prediiction wiith onlly the lung parts inn then image, and task e. represents coviid prediiction wiithout lung inn the image.

3.2 Method and Materials

3.2.1 Images Of radiography

An essential diagnostic imaging tool for recognizing disorders is radiography radiography. Comparatively speaking, it is far more inexpensive and accessible than other imaging modalities. Ra-diographs are especially helpful in the evaluation of chest pathologies because they demonstrate the differences between bones and air.

The detection of corona depends heavily on chest imaging, which includes radiography, computed tomography, and lung ultrasound. The Worlld Heallth Organizations reccently released a recommendation than suggests usiing chest imaging in the acute care of adult patients who are thought to be infected with the virus [22]. - For instance, in suspect symptomatic patients, chest imaging is recommended corona then RT-PCR test isn't posible or when it is possible but findings are not immediately available.It is also suggested when the results of the initial RT-PCR test are negative but there is a strong clinical suspicion of COlD-19. The main objective of this study is to identify corona in patients using chest radiography images with high recall and F1-score.

3.2.2. Radiography Datasets

Chest X-ray imaging is performed on patients with pulmonary discomfort. These individuals typically develop pneumonia rather than COVlD-19. The individuals in the images in the datasets we used therefore fell into one of the following three categories:

1. COVlD-'19: patients with SARS-CoV—2 infection
2.Patients with pneumonia who do not have COVID-19 are classified as group two.
3. People who are healthy and free of COVlD-19 or pneumonia are regarded as normal.

radiography pictures taken of patients who fit into each of the three groups are shown in Figure 3.2.

(a) X-Ray of a patient with COVID-19 (b) X-Ray of a patient with pneumonia (c) X-Ray of a healthy patient

Figure 3.2: radiography of patients belonging to class corona, Pneeumonia, and Norrmal

We used this dattaset in [7] which consists of 475 image off corona instances, 500 image off Pneeumonia, 1000 Normal image. Hence, this dataset is highly imbalanced. This set

combines images from Cohen, et al.[13] with others taken from Chestradiography8 image database in [24]. Inn then reest off thiis papers, they refers too this dataset as Datasetimb. We formed another balanced dataset by merging two different sets of radiography images: the one used in [7], and the one publicly available in [25]. We did not resort to data augmentation in order to avoid overfitting, we also removed duplicate images to avoid data leakage. The newly generated dataset -Datasetbal- contains 475 corona image, 500 pneeumonia image, and 1000
normaal image. Figure 3.3. show the distribution of all three classes in the dataset.

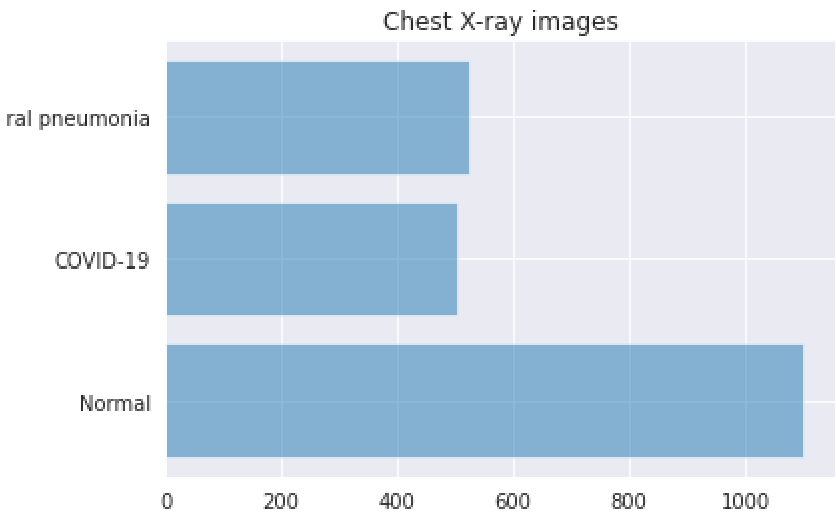

Fig 3.3 Distribution of images in dataset

3.2.3 Image Pre Processing

Every image underwent the preprocessing processes listed below:

1. To map the intensities into the Hounsfield unit, a transformation was applied to the intensities.
2.To emphasise the infection intensities, histogram equalisation was done to the pictures using a transformation made by the radiologist's segmented histogram of the infected areas.
3.In order to make the contrast of the pictures more instructive for the model, a final intensity change was implemented.

3.3 Model Development

This section examines how VGG16 is used to forecast COVlD-19 positivity from X-ray pictures. Transfer learning, support vector machines, and long-term short-term memory are the techniques examined in this article. The fourth part additionally analyses some minor CNN strategies. The objective is to determine how well they perform in relation to the performance measures covered in the earlier sections of this earlier.

3.3.1 Transfer Learning

Transfer learning is the process through which a machine learning model's skills are transferred from one issue to another that is not directly related but yet shares many of the same characteristics. For instance, if a straight fon/vard classifier was trained to identify if a photo has a bag, you might utilise that information to identify other things, such as sunglasses.

Witth traansfer learniing, we essentially attempt to apply the knowledge we have gained to new situations in order to comprehend the concepts more fully. Weights are automatically transferred from a network that completed the new "task B" to a network that was conducting "task A."

Transfeer leearning typically used due to the massive amount of CPU powers required in natuural languages processings activities like sentiments analysises.

Reduced training time, improved neural network performance (in most cases), and the lack of a significant amount of data are three of TN's most significant benefits.

TN is useful in situations where it is not always possible to access the large amounts off datas required too training the neutral model frrom scrratch.

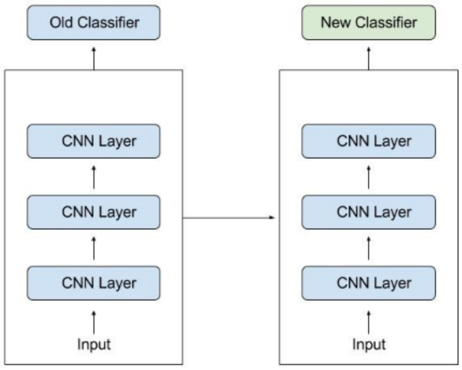

Fig 3.4 Working of TN

On the other hand, neural networks have the capacity to learn which features are important and which aren't. A representation learning algorithm can quickly determine a good set of characteristics, even for complex jobs that would otherwise require a lot of human effort.

The representation can then be used to tackle several other problems. To determine the proper feature representation, just use the first few layers; don't use the network's output because it is too task-specific. Instead, output data through one of the intermediate levels and transmit it into your network. This allows the raw data to be viewed as a representation of this layer.Given that it can reduce your dataset, this technique is frequently used in computer vision.

3.3.2 CNN Algorithm

The CNN models needed a genuinely large training dataset, which cost a lot of time and money to create. The model can be made more efficient using our dataset to solve this issue and a deep transfer literacy-trained network. We looked at many previously trained deep learning networks, such as ResNet (14), Inception. V3. (15), VGGiSk (16), DenseNet (17), and Xception (18). In fact, the maturity of them hay when we use transfer literacy; a complicated network with numerous layers and bear a huge dataset to train. VGG16 was chosen as our foundation model because to this constraint because it is less sophisticated than other models.

Then, to create the two-class affair and attain a respectable level of delicacy, we added layers like pooling layers, powerhouse layers, and thick layers. To be more specific, we used a flatten subcaste to turn our point chart into a vector following the final VGG16 pooling subcaste. We also used three thick layers, subcastes 1, 2, and 3, each of which has 515, 64, and 2 neurons.

Also, used a 50 drops comeout in each of the threelevels.The Adamm algo is also named for optimization, with a0.001 starting literacy rate and a 5X1e- 5 decay. Also, the batch size and trainings time counts were both set to 64 and 20, independently. In Fig.3.3, the model armature is displayed. Trainings datas, confirmation datas, and testing datas were the three groups that our dataset was divided into. We used the training data to train our model, and the confirmation data to configure itshyperparameters.Finally, using test data that hadn't preliminarily been supplied to the model, we estimated the performance of our models. Ass described Subssection A. of the sections, 20 off then remainiing datas, or 16 of the full datas, we uses too confirmation whiile then remainings datas, or 64 off then totasl datas, we uses too train. CT images of the remaining 80 of the subjects were divided as test data.

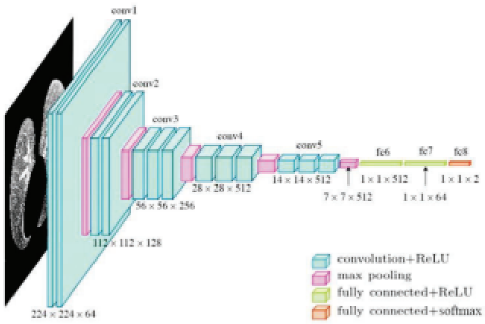

Fig 3.5 Vgg16 model

A 92.7 delicacy rate was achieved by the object detection and bracketing technique VGG16 while categorising 1000 pictures into 1000 distinct orders. It is a popular classification technique for photos and is easy to apply with transfer literacy.

VGG16's 16th digit denotes its 16 weighted layers. VGG16 consists of 21 layers altogether—13 convolutional layers, 5 Max Pooling layers, 3 thick layers, and aggregate of 13 layers—but only 16 of them are weight layers, also referred to as Iearnable parameters layers.

A max pooling operation is performed after the first two layers of the network's architecture complete 2D convolution with 64 filters each. The next three convolutional layers employ filters with corresponding bandwidths of 128, 256, and 512. There are 4096, 4096, and 1000 neurons in the final three completely linked layers, respectively. A softmax function that generates a probability distribution across the 1000 potential picture classes is the network's output.

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0

 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792

 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928

 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

 average_pooling2d (AverageP  (None, 3, 3, 512)        0
 ooling2D)

 flatten (Flatten)           (None, 4608)              0

 dense (Dense)               (None, 128)               589952

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 3)                 387

=================================================================
Total params: 15,305,027
Trainable params: 590,339
Non-trainable params: 14,714,688
```

Fig 3.6 summary of the model

Basically, our categorization model operates as shown in the block diagram below.

- The input tensor for VGG16 has three RGB channels and a size of 224, 244.
- The most notable aspect of VGG16 is that it consistently used the same padding and maxpool layer of a 2x2 filter with stride 2 and prioritised convolution layers of a 3x3 filter with stride 1 over a wide range of hyper-parameters.
- The convolution and max pool layers are equally arranged over the whole design.

- The Conv-1 Layer consists of 64 filters, the Conv-2 Layer of 128 filters, the Conv-3 Layer of 256 filters, and the Conv-4 and Conv-5 Layers of 512 filters.
- Three Fully-Connected (FC) layers, the third of which does 1000-way lLSVRC classification and has model, are added after a stack of convolutional layers.
- 1000 channels : 4096 channels each are present in the first two FC levels. The last layer is called the soft-max layer.

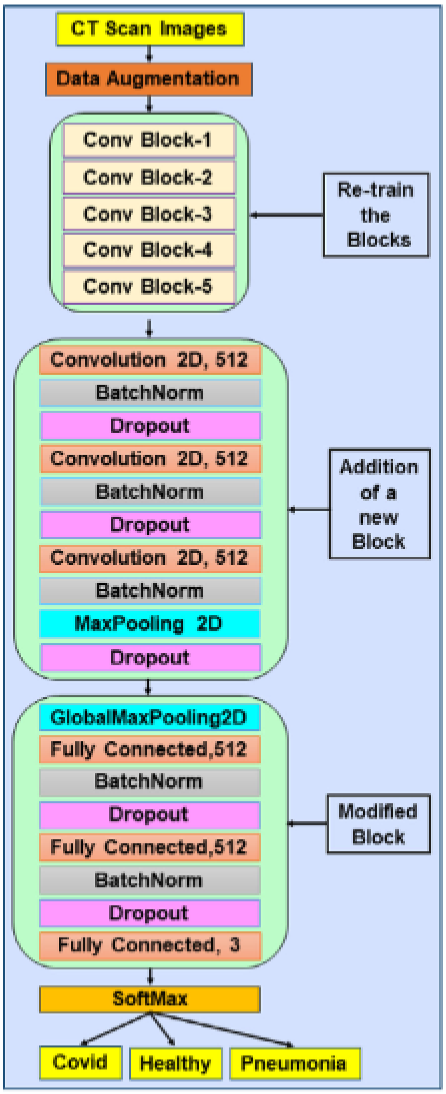

Fig: 3.7 Layers of VGG16

3.3.3 ResNet50 Model

In 2015, Microsoft researchers unveiled the ResNet50 convolutional neural network design. One of the most popular deep neural network models, it has 50 layers, and is utilised for picture categorization and object identification applications. ResNet, which refers to the distinctive design of this model, is an acronym for "Residual Network". In a range of computer vision tasks, such as image classification, object detection, and segmentation, ResNet50 has demonstrated impressive performance.

The usage of residual blocks is the ResNet architecture's main innovation. A group of convolutional layers containing a skip link that permits data to skip one or more layers is known as a residual block. This aids in avoiding the vanishing gradient issue that deep neural networks frequently experience. By creating a straight link between a block's input and output, or "skip connection," the block may learn only the remaining aspects of its input rather than attempting to learn the entire mapping from input to output.

ResNet50 has 50 layers and is built on a four-stage building block design with several residual blocks in each level. One convolutional layer and a max pooling layer make up ResNet50's first stage. The first stage's output is then sent into the next stages, each of which has a number of residual blocks.

ResNet50's capacity to boost deep neural network accuracy without raising their computational complexity is one of its key features. This is accomplished while still enabling the network to acquire deep representations of picture information by minimising the amount of parameters that must be learned in each layer. ResNet50 may thus perform at the cutting edge on a range of picture classification tasks while using a less amount of processing power than other deep neural network designs.

ResNet50 has been utilised for object identification and segmentation tasks in addition to picture classification. For instance, the architecture served as the foundation for the well-known Faster R-CNN object identification framework, which on the COCO dataset demonstrated state-of-the-art performance.

Overall, the ResNet50 deep learning architecture is strong and has shown promise for a variety of computer vision tasks. In addition to lowering the computational cost of the network, it uses residual blocks to learn extremely detailed representations of visual data. ResNet50 is anticipated to continue being a popular choice for deep learning researchers and practitioners in the years to come thanks to its remarkable performance on image classification, object recognition, and segmentation tasks.

3.3.3.1 Working of ResNet50 model

The 50-layer ResNet50 design uses a skip connection, often referred to as a shortcut link, to go around the convolutional layers. This makes it easier for the gradient to flow during training, which aids in avoiding the vanishing gradient issue. The information from earlier layers of the network is also preserved via the skip connections, which can increase the model's accuracy. The ResNet50 model needs a sizable dataset of labelled pictures to learn from because it was trained via supervised learning. In order to reduce the discrepancy between the predicted labels and the actual labels, the model changes the weights of its layers when it is given pictures and their accompanying labels during training.

After the model has been trained, new images can be classified by running them through the network and evaluating the results. A collection of probabilities representing the chance that a picture belongs to each of the classes the ResNet50 model was trained on is the model's output. The projected class for the image is the one with the highest probability.

The ResNet50 model may be adjusted on chest radiography pictures in the context of COVID-19 detection to determine if the patient is COVID-19-infected or not and classify them as normal, pneumonia or covid.

The ResNet50 model goes through numerous phases to detect COVID-19:

1. Data preprocessing: To enhance the picture quality, the chest radiography images are first preprocessed. Techniques like noise reduction, histogram equalisation, and contrast enhancement may be used for this.
2. Model training: The preprocessed photos are used to train the ResNet50 model. The model's weights are changed throughout training in accordance with the features shown in the photographs. In this procedure, a sizable number of chest radiography pictures and their accompanying labels (such as COVID-19 positive or negative) are sent to the model. Based on the traits contained in the photos, the model then learns to distinguish between the two groups.
3. A smaller dataset of COVID-19 positive and negative pictures can be used to fine-tune the ResNet50 model after it has been trained on a large dataset of chest radiography images. Retraining the model on the smaller dataset is a step in fine-tuning that increases the model's performance on the COVID-19 identification test.

4. Making predictions: The ResNet50 model may be applied to fresh chest radiography pictures after being optimised on the COVID-19 dataset. The model receives the input picture and produces a probability score for each class (positive or negative for COVID-19, for example). The picture is then given the class with the greatest likelihood score.

In general, the ResNet50 model can be an effective tool for COVID-19 identification from chest radiography pictures. The model can learn to distinguish between COVID-19 positive and negative photos based on the attributes present in the images by utilising its deep learning capabilities. This can aid medical personnel in correctly and rapidly diagnosing COVID-19, which will speed up patient therapy and improve results.

3.3.4 Additional CNN Strategies

Multitudinous further studies used indispensable strategies to negotiate the same thing. The study discovered that the models delicacy increases 85 to 95 bye using Auxiiliary Cllassifier Geenerative Addversarial Neetwork(ACGAN). In a different study, the experimenters use a multi-class discovery system and a model that combines depthswise complication wiith varying rate off dilations. The delicacy score for the COVID-normal test was97.4.

An improvement procedure was used to produce 1832 images from 1215 online- sourced photos that were used in the study. also, developing stages 1 , stages 2 deeps neetwork models were done for the design. The final model was developed using these strategies, and it achieved delicacy, perfection, and recall values of 97.7,97.14 percent, and 97.14 percent independently. consequences of COVID- 19 infection froms xray filmland were successfully made by the model, it was set up. Xception and ResNet50V2 networks were combined in a model created by a different study, which was described in. With the aid of,302 online-source radiography filmland, the experimenters trained multitudinous deep convolutional networks. With 99.5 delicacy, the suggested model was successful. In order to determine whether a case has COVID- 19 infection, the authors find the model to be both accurate and dependable.

There are other explorations that used the Xception model and presented conclusions that are similar. It emphasises how pivotal the pretrained model is for relating infection with COVID-19. According to certain research, there is a sizable difference between situations involving a double class and situations involving a multi-class that is in favour of the double. Consider the classification of X-ray filmland as COVID-19 and non-COVID rather than as COVID-19, pneumonia, and healthy, which results in a lower delicacy value. With the help of the DarkNet model, the discourse explained the X-ray pictures. Findings from the discussion showed that bracket delicacy was 98.08 while utilising double classes as opposed to 87.02 when using multiple classes.

Another research that used double brackets was to construct four classes for COVID-19, healthy groups, viral pneumonia, and bacterial pneumonia. ResNet101, ResNet50, ResNet152, and InceptionFive pre-trained CNN- grounded models, including ResNetV2, and InceptionV3, have been employed in research using a single pre-trained CNN model. Because it achieved a 99.7 delicacy rate in one of the datasets used, the results indicated that ResNet50 was the most efficient system. This shows that the pre-trained model is fully capable of detecting COVlD-19.

Capsule neural networks have been used in several experiments too irecognize corona from xray pictures. Type of artificial neurals networks calling a capsule network excels at retrieving spatial data and is noted for its high performance. Some studies founds thatthe single class appear too performs best thanthe multiiclass approachess when using the CapsNet method to detect corona. The accuracy score for the models performances onn single class was 97.24%, while the score for the multiclass analysis was 84.22. This studies found thatitis trustworthy modelss that doctors may use to quickly determine a patient's corona status.

In several research, CNN and the decision tree classifier have both been employed too recognize corona infection in xray imagess. The RT_PCR test bean shown to be inconvenient, time-consuming, and expensive for the general public. The researchers propose a technique to distinguish corona situations from the others by using the decision tree algorithm. By separating normal from abnormal scans, the first separation takes place. recognizing those with tuberculosis symptoms among the abnormal scans is the 2nd method inn then decisions trees classifications. The 3rd method is comparable to the 2nd method , except applies to corona this time. These steps have accuracy scores of 98%, 80%, and 95%, respectively.

3.3.4 Tuning, Training, and Validation of Models

The models are first subjected to hyper-parameter tweaking with a performance metric of recall. Since it is crucial to recognize patients who are infected with the virus, we selected this statistic. To the best of our knowledge, only a small number of research have used it as a foundation for comparing and evaluating performance. We assessed models performance on an unbalanced set, similar to the one used in [7], since it had performed better at detecting positive cases on a balanced dataset. The three steps listed below can be used to summarise the entire procedure:

Step 1: we divided the dataset into 80 training sets. Each model underwent typer-parameter tweaking during 20(testing).

Step 2: Using 5 folds cross_validation too taining and test each modell using the optimal set of parameters (obtained in Step I).

Third step: We tested and trained the best. model on the Dataset from Step 2

3.3.5 Nature inspired optimization algorithm:

A class of algorithms known as "nature-inspired optimisation algorithms" is based on the actions and adaptations of living things in the natural world. Complex optimisation problems that cannot be resolved using conventional techniques are tackled utilising these algorithms.

Animal behaviour, genetic development, and swarm intelligence are just a few examples of the natural phenomena that serve as inspiration for nature-inspired optimisation algorithms. These algorithms are employed to address optimisation issues that are challenging or impractical to address using conventional mathematical techniques. These algorithms have seen an increase in usage in a number of disciplines recently, including computer science, engineering, and economics.

The Particle Swarm Optimisation (PSO) method, which is based on the behaviour of bird flocks and fish schools, is one of the most well-known algorithms for optimisation that draws inspiration from nature. By modifying its location depending on both its own best position and the best position discovered by the swarm, a collection of particles representing each candidate solution in this method advances across the search space in the direction of the ideal solution. Applications of the PSO method include feature selection in machine learning and image processing.

The Genetic technique (GA), which takes its cues from the biological process of natural selection, is another well-known optimisation technique with naturalistic roots. Using operators including selection, crossover, and mutation, a population of candidate solutions is developed across generations in this method to produce new candidate solutions. The GA method has been used for a variety of tasks, including neural network design optimization and scheduling issues.

Other examples of nature-inspired optimisation algorithms include Artificial Bee Colony (ABC), which is based on honey bee foraging behaviour, and Ant Colony Optimisation (ACO), which is based on the behaviour of ants searching for food.

Nature-inspired optimisation strategies can be utilised to enhance the performance of the machine learning models in the context of COVID-19 detection. To identify COVID-19 from chest X-ray pictures, for instance, PSO may be used to optimise the hyperparameters of a neural network model. Convolutional neural networks (CNNs), for example, are a type of deep learning model that may have its structure optimised using the GA method. The selection of characteristics utilised for COVID-19 detection can be made more effective by using ACO.

In conclusion, sophisticated computational methods that draw inspiration from nature may be employed to address challenging optimisation issues. These techniques may be utilised to raise the effectiveness and performance of the machine learning models in the context of Covid-19 detection using machine learning.

3.3.5.1 Ant bee colony Algorithm:

The artificial bee colony (ABC) and ant colony optimisation (ACO) algorithms are two well-liked nature-inspired optimisation techniques that have been applied to a range of optimisation issues.

The conduct of actual ant colonies in their pursuit of food served as the model for the ant colony optimisation algorithm. In this technique, a swarm of synthetic ants looks for the shortest route between two points. The ants leave a trail of pheromones behind them as they walk, and the other ants follow it to locate the best path. The algorithm is based on the idea that ants use pheromones to communicate with one another and that they choose to travel along paths where there is a higher concentration of pheromones. The travelling salesman problem, vehicle routing problem, and graph colouring challenge are just a few of the issues that this approach is frequently employed to resolve.

The artificial bee colony algorithm, on the other hand, draws inspiration from honey bees' foraging habits. An artificial bee colony searches for the best solution to a problem in this method. The quality of the food supply the bees have discovered is communicated through the dance they perform for one another. The algorithm is founded on the idea that bees can efficiently find food by interacting with one another and modifying their search based on the calibre of the food sources they have discovered. Numerous issues, including function optimisation, scheduling issues, and image processing, are solved using this algorithm.

In order to solve challenging optimisation issues, both ACO and ABC algorithms are effective and efficient. The particular problem being solved will determine which method is used, thus it's crucial to pick the one that will work best.
The ABC algorithm can't be utilised as a direct technique for COVID detection because it is primarily an optimisation algorithm and cannot detect the virus in a patient's body. The ABC algorithm is one example of a machine learning algorithm that can be used into a more thorough approach for the detection and diagnosis of COVID.

Since the ABC algorithm is primarily an optimisation algorithm and is unable to directly detect the virus in a patient's body, it cannot be used as a direct method for COVID detection. However, a more comprehensive strategy for the detection and diagnosis of COVID can incorporate machine learning algorithms, such as the ABC algorithm.
The ABC algorithm is a different method that may be used to optimise the settings of a machine learning model that can identify COVID-19 based on patient information such as symptoms, medical history, and demographics. This method uses the ABC algorithm to find the model's hyperparameters' ideal values in order to maximise the accuracy of COVID identification based on patient data.

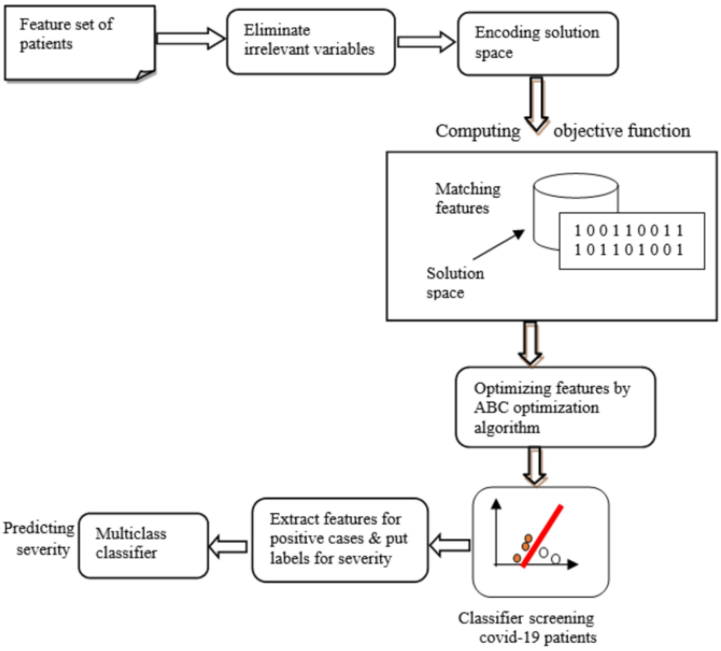

Fig: 3.8 BAsic working for ABC Algorithm

The basic COVID-19 diagnostic techniques, such PCR or antigen testing, cannot be replaced by machine learning models, it is crucial to remember this. These tests are the most accurate way to diagnose COVID-19 because they can directly identify the virus' presence in a patient's body. Though they can add to the diagnosis of COVID-19, machine learning models shouldn't be used as the main strategy for COVID detection.

3.3.5.2 Particle swarm optimisation algorithm:

The movement of fish schooling or flocking of birds served as the inspiration for the nature-inspired optimisation technique known as Particle Swarm Optimisation (PSO). It uses a population-based approach to model how a bunch of particles might behave as they move across a multidimensional search space in pursuit of the best answer.

Particle swarm optimisation (PSO) uses moving particles to represent various solutions to the problem. Each particle has a position and a velocity, and the velocity controls the particle's movement's direction and speed. Based on their own experience and the experience of their swarmmates, the particles modify their placements and velocities. The swarm's current global best solution and each particle's individual best solution are used to update each particle's position and velocity.

A candidate solution is iteratively improved with regard to a specified quality metric in the PSO method, which can be characterised as a search algorithm that seeks to optimise a function. Until a stopping criterion is satisfied, the algorithm iteratively updates each particle's position and velocity starting with a set of randomly initialised particles.

In many different disciplines, such as engineering, economics, and data mining, PSO is frequently employed to resolve optimisation problems. It works particularly well when trying to solve non-linear optimisation issues using continuous variables.

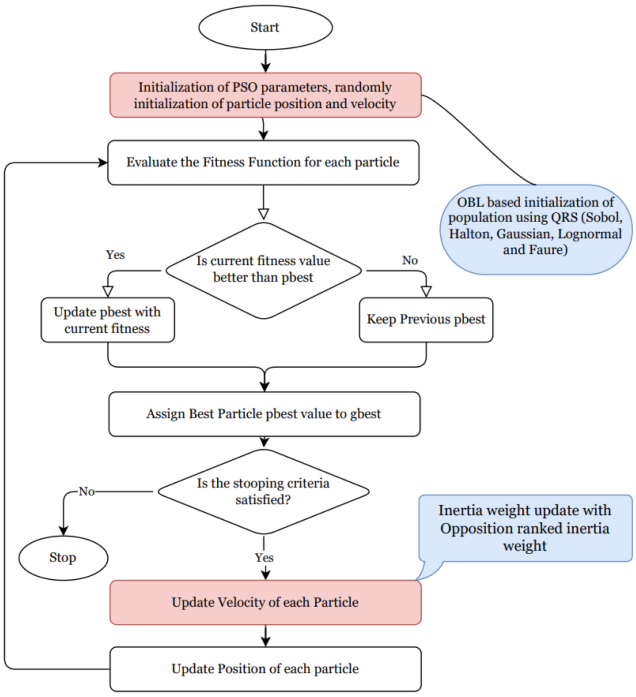

Fig: 3.9: Basic working of PSO Algorithm

By optimising machine learning models that can detect COVID-19 based on patient data, the PSO method may be used to detect COVID-19, among other things. In order to increase the precision of COVID-19 identification based on patient data such as symptoms, medical history, and demographics, the PSO method may be used, for instance, to optimise the parameters of a machine learning model such as artificial neural networks, support vector neural networks, or decision trees.

A machine learning model that can identify COVID-19 in medical pictures like chest X-rays or CT scans can also benefit from hyperparameter optimisation using the PSO technique. This can aid in the timely identification of COVID-19 patients, particularly in areas with a dearth of PCR testing.

The PSO algorithm and machine learning models, it should be noted, should not be the main technique for diagnosing COVID-19. The most accurate technique to identify COVID-19 instances is by using the gold standard diagnostic assays, such as PCR or antigen testing.
In conclusion, the PSO method may be utilised to increase the accuracy of machine learning models that can identify COVID-19 based on patient data or medical pictures. They shouldn't, however, be used in place of the best diagnostic procedures, like PCR or antigen tests.

3.3.5.3 Genetic analysis algorithm:

Genetic analysis algorithms are a class of algorithms that are used to analyze genetic data and identify patterns and relationships between genes and traits. These algorithms are widely used in bioinformatics, genetic epidemiology, and population genetics.
The primary goal of genetic analysis algorithms is to identify the genetic variants associated with a particular phenotype or trait. These algorithms typically start with a large dataset of genetic data, such as single nucleotide polymorphisms (SNPs), and use statistical methods to identify the genetic variants that are associated with the phenotype of interest.
One commonly used genetic analysis algorithm is Genome-Wide Association Study (GWAS). GWAS is a statistical approach that tests for associations between SNPs and a particular trait or disease. In a GWAS analysis, the entire genome of a large number of individuals is genotyped and analyzed to identify SNPs that are associated with the trait or disease being studied.
Principal Component Analysis (PCA) is another genetic analysis tool. PCA is a statistical method for reducing the dimensionality of huge genomic data sets. The genetic data is transformed into a lower-dimensional space via PCA, which highlights the main sources of variation and makes the data easier to analyse.

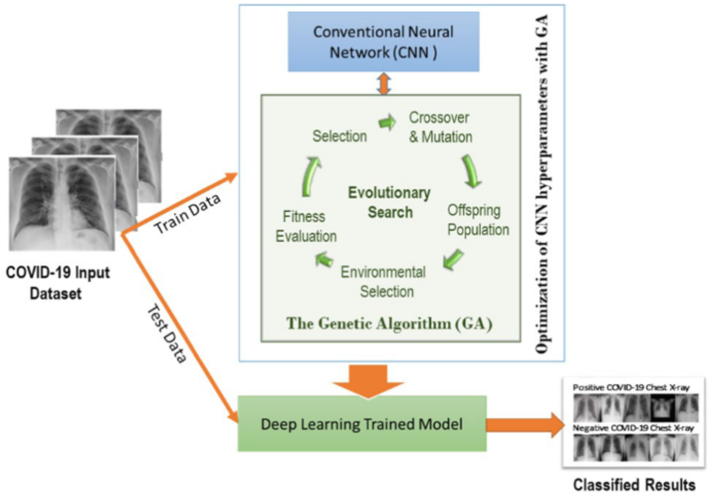

Fig: 3.10 Basic working for GA Algorithm

Other genetic analysis algorithms include linkage analysis, which identifies genetic loci that are linked to a particular trait or disease, and cluster analysis, which groups individuals based on similarities and differences in their genetic data.

Overall, genetic analysis algorithms are essential tools in the field of genetics and genomics, enabling researchers to identify the genetic variants associated with a wide range of traits and diseases.

The optimisation of machine learning models that can identify COVID-19 based on patient data is one method of employing the GA algorithm in COVID-19 detection. As an illustration, the GA algorithm may be used to increase the precision of COVID-19 detection based on patient information such as symptoms, medical history, and demographics. These models include artificial neural networks, support vector machines, and decision trees.

In order to improve the hyperparameters of a machine learning model that can identify COVID-19 in medical pictures like chest X-rays or CT scans, GA algorithm may also be utilised. This can aid in the timely discovery of COVID-19 cases, particularly in areas where PCR testing is scarce.

It is crucial to remember, nevertheless, that the GA algorithm and machine learning models shouldn't be the main technique for diagnosing COVID-19. For the precise detection of COVID-19 instances, the gold standard diagnostic techniques, such as PCR or antigen testing, should always be employed.

3.4 Approach

In these model we have taken a datasets having 1000 normal imagess, 475 covid images and 500 viral pneumonia images . First we have uploaded the files in the google drive in the zip format. Then the collab sheet is mounted toward the drive and zip files are accessed and are unzipped .

```
[ ] # Load the Drive helper and mount
    from google.colab import drive

    # This will prompt for authorization.
    drive.mount('/content/drive')

    Mounted at /content/drive
```

Fig: 3.11 Mounting to drive

```
[ ] #Import the libraries
    import zipfile
    import os

    zip_ref = zipfile.ZipFile('/content/drive/MyDrive/data/COVID.zip', 'r') #Opens the zip file in read mode
    zip_ref.extractall('/tmp') #Extracts the files into the /tmp folder
    zip_ref.close()
```

```
[ ] zip_ref = zipfile.ZipFile('/content/drive/MyDrive/data/Normal.zip', 'r') #Opens the zip file in read mode
    zip_ref.extractall('/tmp1') #Extracts the files into the /tmp folder
    zip_ref.close()
```

Fig :3.12 Unzipping the files

All the files are unzipped and their paths are stored in the variable (n,c,p) respectively. Then the data is accessed and then a dataframe is created which stores all the images differentiated by category:

- 0 for Normal images
- 1 for Viral Pneumonia images
- 2 for covid images

	filename	category
0	/tmp/COVID/COVID-153.png	2
1	/tmp/COVID/COVID-161.png	2
2	/tmp/COVID/COVID-323.png	2
3	/tmp/COVID/COVID-158.png	2
4	/tmp/COVID/COVID-287.png	2

Fig:3.13 Head of dataframe

After that the distribution of images in the dataset is plotted and in order to verify that the data frame has the images a random image is picked from the dataframe and is plotted.

```
sample = random.choice(df['filename'])
image = load_img(sample)
plt.imshow(image)
plt.show()
```

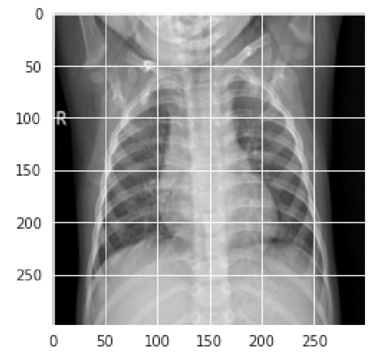

Fig 3.14 Random image of dataframe

After that the data is split and later on the model is created.

```
train_data, test_valid_data = train_test_split(df, test_size=0.2, random_state = 42, shuffle=True, stratify=df['category'])
train_data = train_data.reset_index(drop=True)
test_valid_data = test_valid_data.reset_index(drop=True)
```

```
test_data, valid_data = train_test_split(test_valid_data, test_size=0.5, random_state = 42,
                                         shuffle=True, stratify=test_valid_data['category'])
test_data = test_data.reset_index(drop=True)
valid_data = valid_data.reset_index(drop=True)
```

Fig:3.15 Splitting of data

Chapter-4
PERFORMANCE ANALYSIS

4.1 Results

To estimate the model's performance following training, we used the standard statistical analysis. The test data were entered into our model using a five-fold cross validation technique in order to determine the performance, i.e., the model's accuracy in correctly identifying the photographs of each group.

A number of important metrics were calculated when the confusion matrix was created, including the positive predictive value, false discovery rate, true positive rate, and false negative rate. We also calculated the metrics for specificity, accuracy, F1 score, precision, and sensitivity. The terms true positive, true negative, false positive, and false negative rates, respectively, are TP, TN, FP, and FN. In Eqs. (1)–(5), these terms are defined. Finally, we have evaluated the effectiveness of all three algorithms that were inspired by nature.

4.1.1 Performance Metrics

We measured the performance of the model using metrics extracted from the confusion matrix shown in Table 1 where positive indicates the cases that carry corona and negative are those that do not (i.e. pneumonia or normal).

		Predicted		
		Positive	**Negative**	
Actual	**Positive**	TP	FN	Recall $= \frac{TP}{TP+FN}$
	Negative	FP	TN	Specificity $= \frac{TN}{TN+FP}$
		Precision $= \frac{TP}{TP+FP}$		Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$

Table 1: Confusion Matrix

The photos that the model properly identifies as belonging to corona-positive individuals are known as True Positives (TP). Images that are mistakenly assigned to the corona class but actually belong to the Pneumonia or the Normal class are known as false positives (FP). True Negatives (TN) are photos that have been accurately identified as lacking corona. Images that contain corona but are categorised by the model as Normal or Pneumonia are known as False Negatives (FN). These are the most important findings from our study because if they were mislabeled, the virus would continue to spread more quickly. For comparison's sake, we computed all five metrics—accuracy, precision, recall, specificity, and F1-score—so that we could compare the model's performance using recall and F1-score or accuracy.

1. Accuracy

Accuracy denotes the proportion of cases the model correctly classified and denotes the systematic error across all predicted classes. This metric summarises the model's performance across all classes, but it can be deceptive when the data is unbalanced because a majority classifier will consistently achieve high accuracy but fail to classify instances that have the minority class label.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2.. Precision

When all the data is classified into a class, precision is the percentage of cases that are correctly assigned to that class. In this instance, it shows the proportion of corona cases that actually are corona cases. It is calculated using the one-vs-all method for each class.

$$Precision_{COVID-19} = \frac{TP}{TP + FP}$$

3. Recall

How many cases that belong to a class are accurately classified into that class is measured by recall or sensitivity. The percentage of cases accurately represented by the classifier among all persons who carry the disease is measured in this context. While presenting our findings, we emphasise the significance of this metric because it is essential to prevent misclassifying corona patients into other classes like normal or pneumonia because undetected cases might cause the disease to spread quickly. Recall is determined using the one-vs-all method, just like precision.

$$Recall_{COVID-19} = \frac{TP}{TP + FN}$$

4. Specificity

Specificity, which is calculated using the one-vs-all method, is the proportion of negative classifications among the cases that are actually negative. It displays the percentage of negative cases for the corona class that were correctly identified as virus-free.

$$Specificity_{COVID-19} = \frac{TN}{TN + FP}$$

5. F1-score

The weighted harmonic mean of recall and precision is referred to as the F1-score or F-measure. When the dataset is severely skewed, this measure is the one to use. When precision or recall alone are insufficient to assess a model's performance for a particular class, it offers a way to consider the bigger picture.

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

4.2 Outputs

4.2.1 Image Pre-Processing through Histogram Equalisation

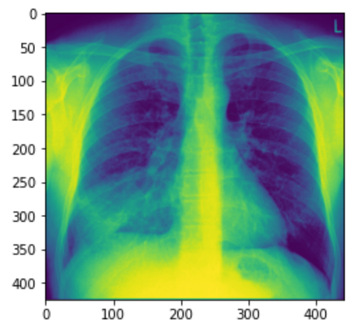

Fig 4.1:Random image after applying histogram equalisation in grayscale

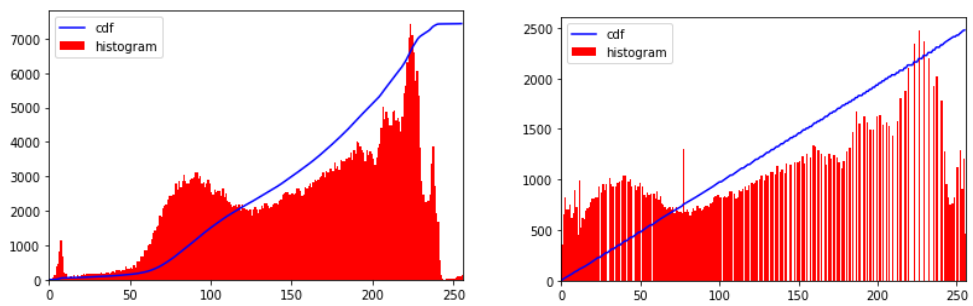

Fig 4.2 Intensity vs Pixel graph before and afterHistogram Equalisation

4.2.2 VGG16 model

The outputs precisely calculated from that of VGG 16 model are as follows:

4.2.2.1 Creating VGG16 model

```
[ ] baseModel = VGG16(input_shape=(224,224,3), weights='imagenet', include_top=False)

    for layer in baseModel.layers:
        layer.trainable = False

    headModel = baseModel.output
    headModel = AveragePooling2D()(headModel)
    headModel = Flatten()(headModel)
    headModel = Dense(128, activation="relu")(headModel)
    headModel = Dropout(0.2)(headModel)
    headModel = Dense(3, activation='softmax')(headModel)

    model = Model(inputs=baseModel.input, outputs=headModel)

    Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
    58889256/58889256 [==============================] - 0s 0us/step
```

Fig 4.3:Creating the layers for basic VGG 16 model

```
[ ] epochs = 10
    history = model.fit_generator(train_generator,
                                  validation_data=valid_generator, verbose=1, epochs=epochs)

    Epoch 1/10
    109/109 [==============================] - 984s 9s/step - loss: 0.6648 - accuracy: 0.7523 - val_loss: 0.4547 - val_accuracy: 0.8374
    Epoch 2/10
    109/109 [==============================] - 1052s 10s/step - loss: 0.4122 - accuracy: 0.8774 - val_loss: 0.3423 - val_accuracy: 0.8916
    Epoch 3/10
    109/109 [==============================] - 997s 9s/step - loss: 0.3363 - accuracy: 0.8922 - val_loss: 0.3217 - val_accuracy: 0.8768
    Epoch 4/10
    109/109 [==============================] - 976s 9s/step - loss: 0.3164 - accuracy: 0.9014 - val_loss: 0.2795 - val_accuracy: 0.8966
    Epoch 5/10
    109/109 [==============================] - 971s 9s/step - loss: 0.2904 - accuracy: 0.9119 - val_loss: 0.2997 - val_accuracy: 0.8916
    Epoch 6/10
    109/109 [==============================] - 978s 9s/step - loss: 0.2801 - accuracy: 0.9113 - val_loss: 0.2844 - val_accuracy: 0.8966
    Epoch 7/10
    109/109 [==============================] - 1012s 9s/step - loss: 0.2641 - accuracy: 0.9063 - val_loss: 0.3361 - val_accuracy: 0.8670
    Epoch 8/10
    109/109 [==============================] - 1015s 9s/step - loss: 0.2493 - accuracy: 0.9230 - val_loss: 0.2483 - val_accuracy: 0.9212
    Epoch 9/10
    109/109 [==============================] - 979s 9s/step - loss: 0.2452 - accuracy: 0.9230 - val_loss: 0.2408 - val_accuracy: 0.9212
    Epoch 10/10
    109/109 [==============================] - 983s 9s/step - loss: 0.2320 - accuracy: 0.9273 - val_loss: 0.2660 - val_accuracy: 0.9064
```

Fig 4.4 Neural Layers Processing

```
[INFO] evaluating network...
```

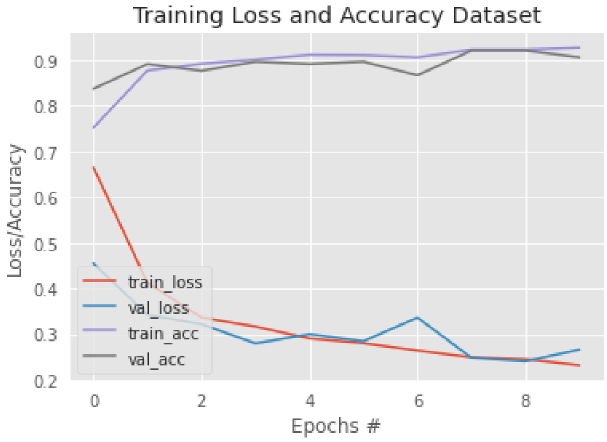

Fig 4.5 Iteration vs loss/accuracy graph

4.2.2.2 Predicting the image

A random image is selected from the dataset and then it is passed to the model and it is classified.

```
True value is : Normal
1/1 [==============================] - 2s 2s/step
Prediction is:
Normal
<matplotlib.image.AxesImage at 0x7fb8b8bd7510>
```

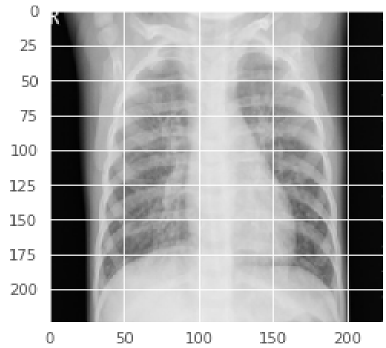

Fig 4.6 Image Prediction

4.2.2.3 Accuracy Of the model

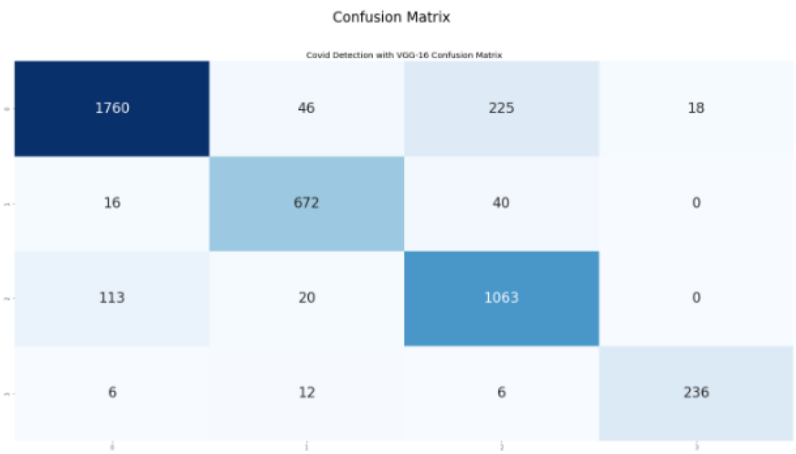

Fig 4.7 Confusion Matrix

```
[ ] evalute = model.evaluate(train_generator)

109/109 [==============================] - 896s 8s/step - loss: 0.2134 - accuracy: 0.9328

[ ] print("Accuracy: {:.2f}%".format(evalute[1] * 100))
    print("Loss: {}".format(evalute[0]))

    Accuracy: 93.28%
    Loss: 0.21335609257221222
```

Fig 4.8: Accuracy of model

```
Report :
              precision    recall  f1-score   support

           0       0.93      0.86      0.89      2049
           1       0.90      0.92      0.91       728
           2       0.80      0.89      0.84      1196
           3       0.93      0.91      0.92       260
```

Fig 4.9 Classification Report

4.2.3 ResNet50 Model

The outputs precisely calculated from that of ResNet50 model are as follows:

4.2.3.1 Creating VGG16 model

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Convert the labels to one-hot encoding
y_train = to_categorical(y_train, num_classes)
y_test = to_categorical(y_test, num_classes)
# Train the model
model.fit(X_train, y_train, batch_size=3, epochs=10, validation_data=(X_test, y_test))
```

```
Epoch 1/10
480/480 [==============================] - 16s 24ms/step - loss: 0.9055 - accuracy: 0.7583 - val_loss: 0.4013 - val_accuracy: 0.8306
Epoch 2/10
480/480 [==============================] - 10s 21ms/step - loss: 0.3835 - accuracy: 0.8458 - val_loss: 0.3408 - val_accuracy: 0.8778
Epoch 3/10
480/480 [==============================] - 10s 21ms/step - loss: 0.3280 - accuracy: 0.8722 - val_loss: 0.3765 - val_accuracy: 0.8639
Epoch 4/10
480/480 [==============================] - 10s 21ms/step - loss: 0.2970 - accuracy: 0.8840 - val_loss: 0.3069 - val_accuracy: 0.8861
Epoch 5/10
480/480 [==============================] - 9s 19ms/step - loss: 0.2661 - accuracy: 0.8917 - val_loss: 0.3833 - val_accuracy: 0.8806
Epoch 6/10
480/480 [==============================] - 9s 19ms/step - loss: 0.2537 - accuracy: 0.8972 - val_loss: 0.4926 - val_accuracy: 0.8500
Epoch 7/10
480/480 [==============================] - 9s 19ms/step - loss: 0.2603 - accuracy: 0.8944 - val_loss: 0.2679 - val_accuracy: 0.8944
Epoch 8/10
480/480 [==============================] - 9s 19ms/step - loss: 0.2358 - accuracy: 0.9042 - val_loss: 0.4118 - val_accuracy: 0.8250
Epoch 9/10
480/480 [==============================] - 9s 19ms/step - loss: 0.1629 - accuracy: 0.9333 - val_loss: 0.6845 - val_accuracy: 0.7972
Epoch 10/10
480/480 [==============================] - 10s 21ms/step - loss: 0.1918 - accuracy: 0.9250 - val_loss: 0.4834 - val_accuracy: 0.8556
<keras.callbacks.History at 0x7f992e224a30>
```

Fig 4.10:Creating the layers for basic ResNet50 model

4.2.3.2 Accuracy of the model

```
test_steps = train_generator.samples
train_generator.reset()
loss, accuracy = model.evaluate(train_generator, steps=test_steps, verbose=1)
print("Accuracy: {:.2f}%".format(accuracy * 100))
```

```
.........] - ETA: 5:40 - loss: 0.2500 - accuracy: 0.9113WARNING:tensorflow:Your input ran out of data; interrupting tr
=========] - 11s 14ms/step - loss: 0.2500 - accuracy: 0.9113
```

Fig 4.11 : Accuracy of ResNet50 model

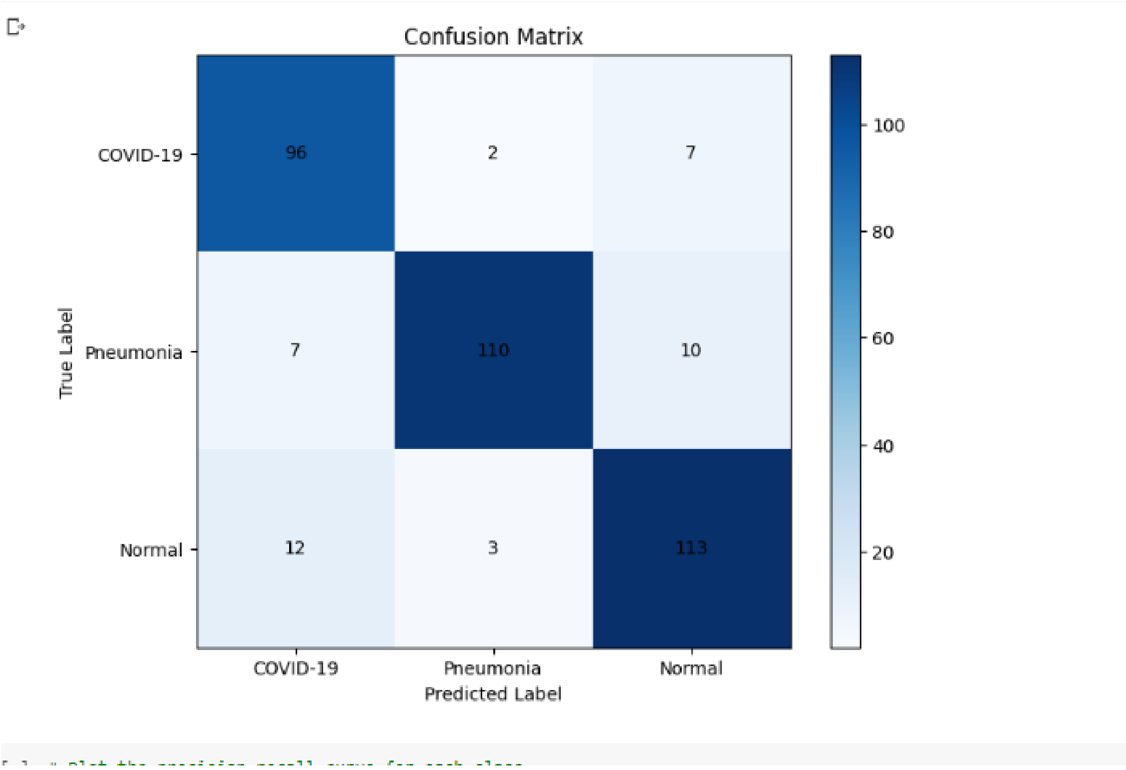

Fig 4.12:Confusion matrix of resNet50 model

```
[ ] from sklearn.metrics import precision_recall_curve
    precision, recall, _ = precision_recall_curve(y_test, y_pred,pos_label=1)
    plt.plot(recall, precision, label='Model')
    plt.xlabel('Recall')
    plt.ylabel('Precision')
    plt.title('Precision-Recall Curve')
    plt.legend()
    plt.show()
```

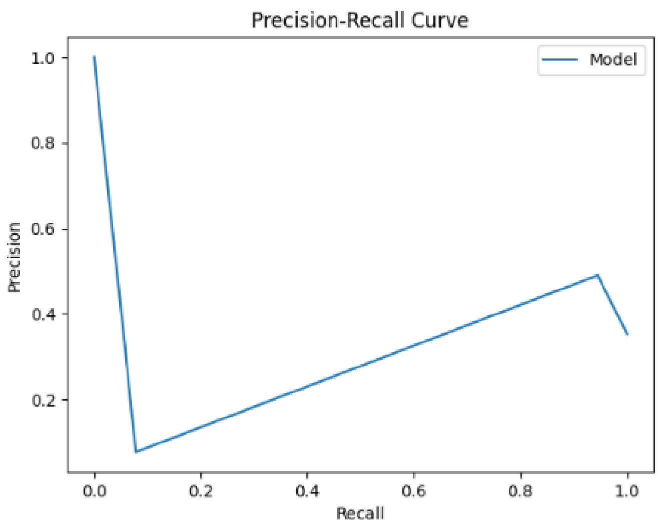

Fig 4.13 : Precision recall curve for ResNet50

```
[ ] # Plot the ROC curve for each class
    from sklearn.metrics import roc_curve
    fpr, tpr, _ = roc_curve(y_test, y_pred,pos_label=1)
    plt.plot(fpr, tpr, label='Model')
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('Receiver Operating Characteristic (ROC) Curve')
    plt.legend()
    plt.show()
```

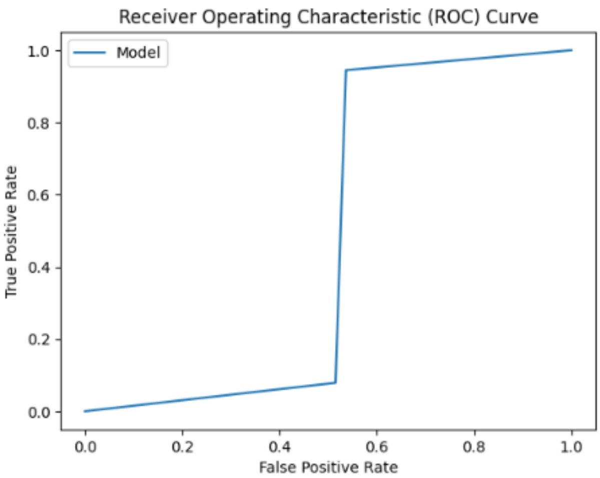

Fig 4.14 : ROC curve for ResNet 50 model

4.2.3.3 Predicting the Image

```
True value is : Normal
1/1 [==============================] - 2s 2s/step
Prediction is:
Normal
<matplotlib.image.AxesImage at 0x7fb8b8bd7510>
```

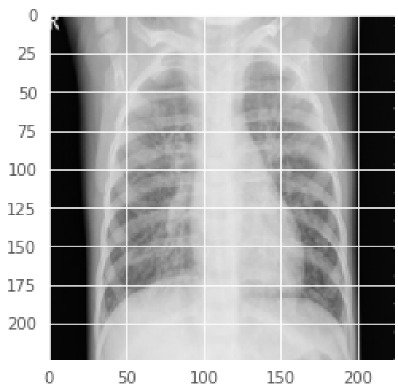

Fig 4.15:predicting the image for ResNet50 model

4.2.4 Particle Swarm Optimization Algorithm

The outputs precisely calculated from that of PSO model are as follows:

4.2.4.1 Creating PSO model

```python
def pso_fitness_function(x):
    model = MLPClassifier(hidden_layer_sizes=(x[0],), activation='relu', solver='adam', max_iter=x[1])
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    return 1 - accuracy
```

Fig 4.16:Creating the fitness function for PSO model

```python
[ ]  # Define the bounds for the hyperparameters
     bounds = [(0.001, 0.1),   # learning rate
               (1, 5),         # number of hidden layers
               (5, 50),        # number of neurons per layer
               (0, 1)]         # activation function (0 = relu, 1 = tanh)

     # Run PSO to optimize the hyperparameters
     best_params, best_value = pso(fitness_function, bounds, swarmsize=10, maxiter=50, debug=True)

     print('Best hyperparameters:', best_params)
     print('Best F1-score:', -best_value)
```

Fig 4.17: Setting parameters for PSO model

4.2.4.2 Calculating accuracy

```python
def pso_model(x):
    model = MLPClassifier(hidden_layer_sizes=(x[0],), activation='relu', solver='adam', max_iter=x[1])
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    return model, accuracy
```

Fig 4.18: Accuracy of PSO model

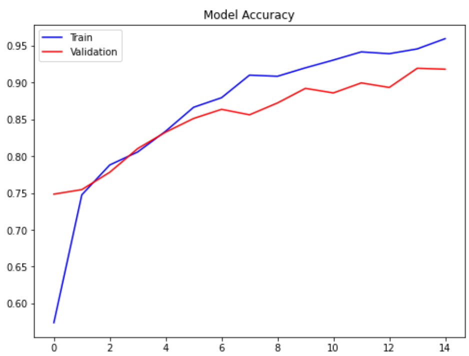

Fig 4.19:Graph for Accuracy

```
Classification report:
              precision    recall  f1-score   support

    COVID-19       0.78      0.78      0.78         9
   Pneumonia       0.83      1.00      0.91        10
      Normal       1.00      0.82      0.90        11

    accuracy                           0.87        30
   macro avg       0.87      0.87      0.86        30
weighted avg       0.88      0.87      0.87        30
```

Fig 4.20 : Classification report for PSO

```
Accuracy: 0.8666666666666667
Confusion matrix:
 [[ 207  20  0]
 [ 10 150  15]
 [ 2  0  1909]]
```

Fig 4.21: Accuracy Confusion matrix for PSO mode;

```
plt.title('Precision-Recall Curve')
plt.legend()
plt.show()
```

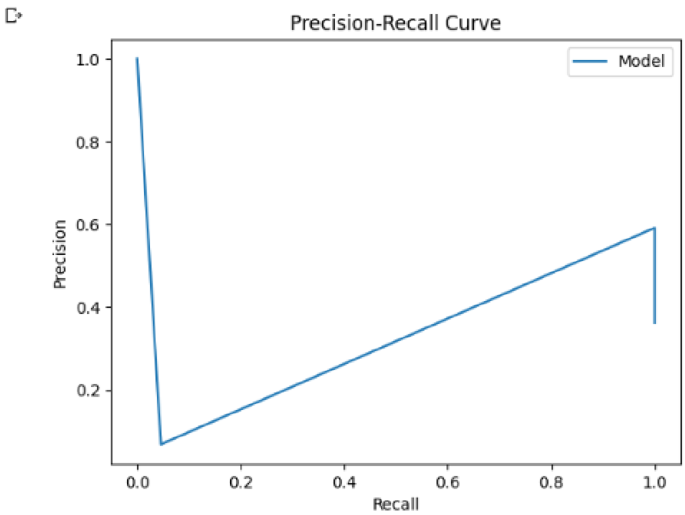

Fig 4.22: Precision recall curve of PSO model

4.2.5 Genetic Analysis Algorithm

The outputs precisely calculated from that of GA model are as follows:

4.2.5.1 Calculating accuracy

```
def ga_fitness_function(x):
    model = MLPClassifier(hidden_layer_sizes=(x[0],), activation='relu', solver='adam', max_iter=x[1])
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    return accuracy
```

Fig 4.23:Creating the fitness function for GA model

```
def ga_model(x):
    model = MLPClassifier(hidden_layer_sizes=(x[0],), activation='relu', solver='adam', max_iter=x[1])
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    return model, accuracy
```

Fig 4.24:Creating the basic GA model

4.2.5.2 Calculating performance metrics

```
Epoch 9: val_loss did not improve from 0.44018
38/38 [==============================] - 20s 540ms/step - loss: 0.5726 - accuracy: 0.7600 - val_loss: 0.4484 - val_accuracy
Epoch 10/10
38/38 [==============================] - ETA: 0s - loss: 0.5397 - accuracy: 0.7683
Epoch 10: val_loss improved from 0.44018 to 0.42435, saving model to model.h5
38/38 [==============================] - 21s 565ms/step - loss: 0.5397 - accuracy: 0.7683 - val_loss: 0.4244 - val_accuracy
Test loss: 0.4243530333042145
Test accuracy: 0.8324999809265137
Text(0.5, 1.0, 'Model Accuracy')
```

Model Accuracv

Fig 4.25:accuracy of GA model

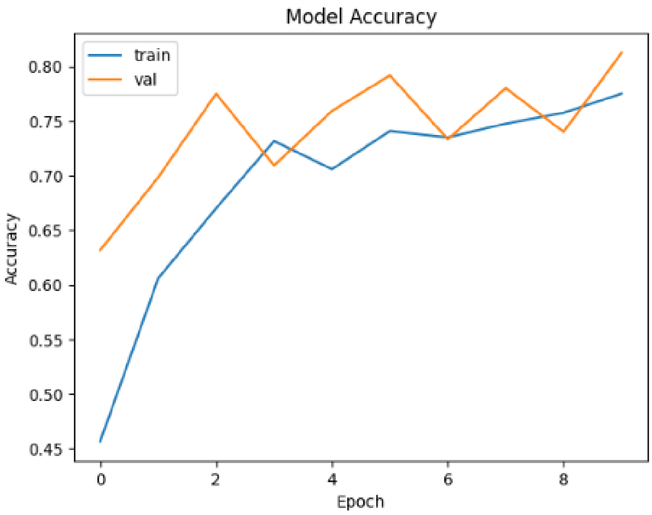

Fig 4.26:Performance of GA model

4.2.6 Ant Bee Colony Optimisation Algorithm

The outputs precisely calculated from that of ABCO model are as follows:

4.2.6.1 Calculating performance metrics

```python
def fitness_function():
    # Reshape weights to the format expected by the classifier
    weights = np.array([ 0.01322724, -0.00833307,  0.00260955, ..., -0.00687762,  0.0067241
        0.00862355])
    weights = np.reshape(weights, (3, len(images_flattened[0])))

    # Define the neural network classifier
    def neural_network_classifier(x):
        z = np.dot(weights, x)
        y = np.exp(z) / np.sum(np.exp(z))
        return y

    # Use the neural network classifier to predict labels for the test set
    y_pred = np.array([np.argmax(neural_network_classifier(x)) for x in X_test])

    # Calculate the accuracy score of the predictions
    accuracy = accuracy_score(y_test, y_pred)

    return -accuracy
```

Fig 4.27:Creating the basic ABC model

4.2.6.2 Calculating performance metrics

```
Epoch 9: val_loss did not improve from 0.44018
38/38 [==============================] - 20s 540ms/step - loss: 0.5726 - accuracy: 0.7600 - val_loss: 0.4484 - val_accuracy
Epoch 10/10
38/38 [==============================] - ETA: 0s - loss: 0.5397 - accuracy: 0.7683
Epoch 10: val_loss improved from 0.44018 to 0.42435, saving model to model.h5
38/38 [==============================] - 21s 565ms/step - loss: 0.5397 - accuracy: 0.7683 - val_loss: 0.4244 - val_accuracy
Test loss: 0.4243530333042145
Test accuracy: 0.8324999809265137
Text(0.5, 1.0, 'Model Accuracy')
```

Model Accuracy

Fig 4.28:Performance of ABC model

```
Text(0.5, 1.0, 'Model Accuracy')
```

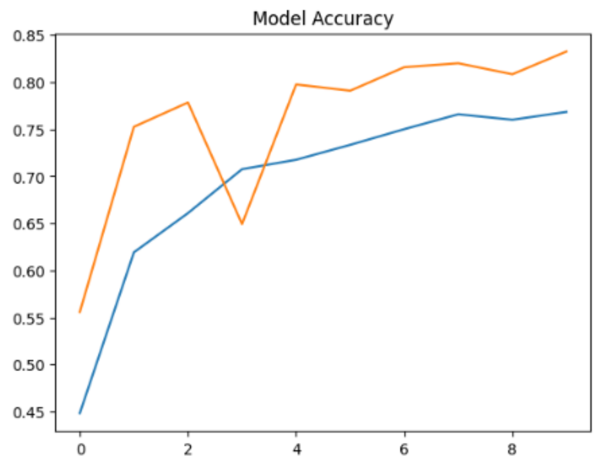

Fig 4.29:graph for Performance of ABC model

4.2.7 Comparing the performance of Nature Inspired Optimization Algorithms

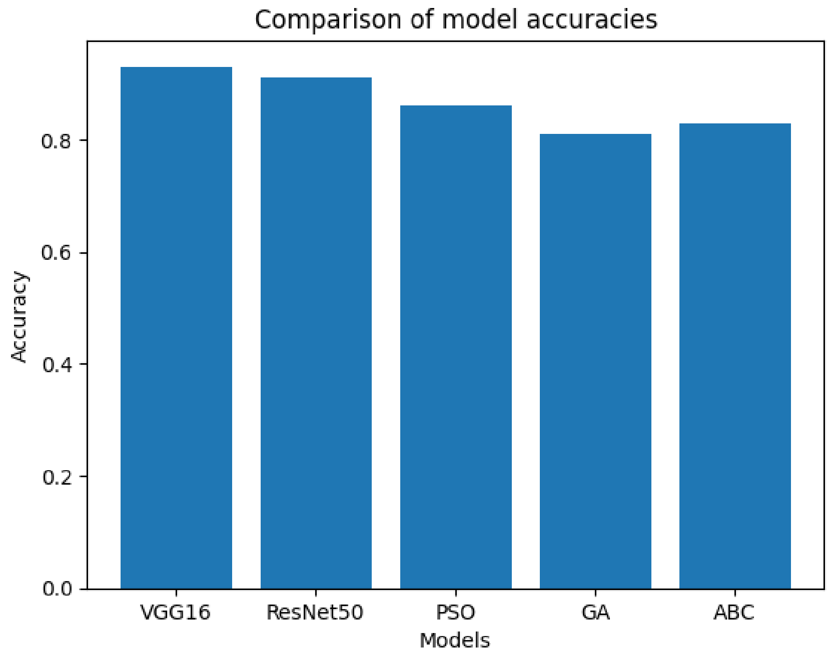

Fig 4.30: Comparison of performance of all models

Chapter-5
CONCLUSION

5.1 Conclusion

Machine learning methods have been shown to be useful for detecting and diagnosing COVID-19 in medical imaging modalities such chest X-rays and CT scans. Convolutional neural networks, ResNet-50, and other deep learning models and algorithms have been extensively employed in COVID-19 diagnosis and have attained good accuracy rates.
The study comes to the conclusion that all the methodologies discussed in the discourse are reliable and valid. The modest variations in accuracy scores are not substantial enough to justify discounting some of the methods. Results from all the strategies used were really promising. It explains why doctors are increasingly preferring to detect corona early enough using these technological techniques.It can be clearly seen that among the Nature Inspired algorithms Genetic Analysis algorithm has the highest accuracy and among the CNN models used ResNet50 has lesser accuracy over VGG 16 model. The only prerequisite is an radiography of the subject's chest, which makes all of the techniques less expensive, more accessible, and non-invasive. The study offers the following suggestions for raising the model's accuracy.

1) Include TN techniques in the process of developing models. The study has shown that TN, which enables models to learn from previously trained models and data, increases the formidability of models.

2) Use techniques for data augmentation. Data augmentation is required during the preparation step of the data analysis phase, particularly in cases where data is lacking. The strength and viability of a CNN model are positively impacted by data augmentation, according to this study.

3) Put your attention on categorising data into binary classes. Throughout this review, the binary-class scenario consistently outperformed the multi-class scenario when comparing accuracy ratings between binary- and multi-class circumstances. It is wise to think of it as the primary focus of model formulation.

The conclusion from respective models are as follows:

5.1.1 Using VGG-16 model

When used to analyse chest radiography pictures for the identification of COVID-19, the VGG16 model has produced promising results. The VGG16 has achieved high accuracy in detecting COVID-19 thanks to the transfer learning method utilised in this work and model fine-tuning. The performance of the model has been significantly enhanced by the application of data augmentation techniques and hyperparameter optimisation. It is crucial to remember that the generalisation of the model must be assessed using more diverse datasets from various sources in order to guarantee its dependability and robustness. The VGG16 model can be a helpful tool for helping medical practitioners make an early and accurate COVID-19 diagnosis, which will ultimately help in the pandemic's effective treatment and control.

5.1.2 Using ResNet50 model

The detection of COVID-19 from chest X-ray images by the ResNet 50 model has produced promising results. The model's accuracy is lower than that of the VGG16 model, although it exhibited increased accuracy when pre-processing methods such contrast stretching and histogram equalisation were used. Reduced overfitting and improved generalisation were achieved by using transfer learning with the pre-trained ResNet 50 model on a sizable dataset. In cases when alternative testing techniques might not be accessible or viable, the ResNet 50 model can be used as a diagnostic tool for COVID-19 identification. However, by incorporating additional data augmentation strategies and tweaking the model's hyperparameters, the model's performance can still be enhanced.

5.1.3 Using Nature Inspired Algorithms

In conclusion, the identification of COVID-19 has shown encouraging results when using nature-inspired algorithms including genetic algorithms, particle swarm optimisation, and ant colony optimisation. By analysing different medical imaging data, including chest X-rays and CT scans, these algorithms have been utilised in conjunction with machine learning approaches to identify COVID-19 in patients. Even though these algorithms have demonstrated great accuracy rates, additional validation and testing on larger datasets is still necessary to guarantee their efficacy and dependability. Nevertheless, the creation of such

COVID-19 detection algorithms is a significant step towards early disease detection and treatment, potentially slowing the virus' spread and saving lives.

5.2 Goals Achieved

A tested models are created which are capable of classifying the X Ray images into 3 categories(Covid,NormaI,Viral Pneumonia) and making it easy to recognize the patients in an efficient and fast way rather than depending on old physical methods.

5.3 Future Improvements

CDespite the novelty of corona, investigations of automatic detection have a number of drawbacks. A notable issue is the lack of a large dataset to help the generalisation of the approaches for usage in practical applications.

- In order for the networks to function properly, there should be a significant difference in the dataset's size.

- Future research could use a more sophisticated deep learning network backbone architecture, Attention-based multiple instances learning in place of training on individual slices, handcrafted features extracted based on previous knowledge, generative adversarial networks (GANs) to increase the number of suitable pictures for training the network and enhance model performance.
- Deep learning techniques can also be used to forecast the course of an illness or the effectiveness of a treatment, in addition to categorising and segmenting infections in images.
- Combining CT scan images with radiography or ultrasound images can improve performance even more.
- The performance of the proposed model can be enhanced by adding pre-processing steps that teach the fine-order distinguishing features between images of pneumonia and CT scans.
- An approach that combines fuzzy logic and artificial neural networks is called a hybrid approach, and it can be studied to have a reliable representation of partial truth/uncertainty in the classification.

REFERENCES

[1] R. Singh, S. Singh, and A. Singh, "OptCoNet: an optimised convolutional neural network for an autonomous diagnosis of corona," Applied Intelligence, vol. 51, no. 3, pp. 1351-1366, Mar. 2021.

[2] N. K. Chowdhury, M. M. Rahman, and M. A. Kabir, "PDCOVIDNet: a parallel-dilated convolutional neural network architecture for detecting corona from chest radiography pictures," Health information science and systems, vol. 8, no. 1, pp. 1-14, Dec. 2020.

[3] World Health Organization, "WHO Coronavirus (corona) Dashboard," December 17, 2021. [Online]. Available: https://covid19.who.int. [Accessed: Dec. 17, 2021].

[4] A. Islam, M. S. Islam, M. S. Islam, and M. A. Al-Mamun, "Classification of corona patients from chest CT images using multi-objective differential evolution-based convolutional neural networks," European Journal of Clinical Microbiology and Infectious Diseases, vol. 39, no. 7, p. 137, Jul. 2020.

[5] M. Elgendi, M. U. Nasir, Q. Tang, R. R. Fletcher, N. M. C. Howard, and S. Nicolaou, "The performance of deep neural networks in discriminating chest radiography of corona patients from other bacterial and viral pneumonias," Frontiers in Medicine, vol. 7, no. 1, p. 550, Jan. 2020.

[6] J. Civet-Mascot, F. Luna-Perejon, M. Dominguez Morales, and A. Civit, "Deep learning system for corona diagnosis help utilising radiography pulmonary images," Applied Sciences, vol. 10, no. 13, p. 4640, Jul. 2020.

[7] M. Shorfuzzaman and M. S. Hossain, "MetaCOVlD: A Siamese neural network architecture with contrastive loss for n-shot diagnosis of corona patients," Pattern Recognition, vol. 113, no. 1, p. 107700, Jan. 2021.

[8] L. Li, T. Shim, and P. E. Zapanta, "Optimization of corona testing accuracy using nasal anatomy instruction," American Journal of Otolaryngology, vol. 42, 2021.

[9] X. Darzacq et al., "Overcoming the bottleneck to widespread testing: a fast evaluation of nucleic acid testing techniques for corona detection," RNA, vol. 26, no. 7, pp. 771-783, 2020.

[10] L. Wang, Z. O. Lin, and A. Wong, "COVID-Net: A Customized Deep Convolutional Neural Network Design for Detection of Corona Cases from Chest Radiography Pictures," Scientific Reports, vol. 10, no. 1, pp. 1-12, 2020.

[11] M. Szmigiera, "Global Economic Impact of Coronavirus Pandemic - Statistics 8 Facts," 23 November 2021. [Online]. Available: https://www.statista.com/topics/6139/corona-impact-on-the-global-economy/#dossierKeyfigures. [Retrieved on December 17, 2021].

[12] S. A. Quadri, "Corona and religious congregations: Implications for spread of new infections," International Journal of Infectious Diseases, vol. 96, pp. 219-221, 2020.

[13] M. Lipsitch and N. E. Dean, "Understanding corona vaccination efficacy," Science, vol. 370, 2020.

[14] M. A. Pettengill and A. J. McAdam, "Can we test our way out of the corona pandemic?," Journal of Clinical Microbiology, vol. 58, no. 11, pp. e02225-20, 2020.

[15] F. M. Salman, S. S. Abu-Naser, E. Alajrami, B. S. Abu-Nasser, and B. A. Alashqar, "Corona detection using artificial intelligence," First Journal of Biomedical Research, vol. 1, no. 1, p. 1.

[16] C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Deep Learning applications for corona," Journal of Big Data, vol. 8, no. 1, pp. 1-54, 2021.

[17] L. Turkoglu and T. B. Alakus, "Comparison of deep learning algorithms to forecast corona infection," Chaos, Solitons & Fractals, vol. 140, pp. 110-120, 2020.

[18] "Coronavirus (corona) detection from chest radiography pictures using convolutional neural networks," Biomedical Signal Processing and Control, vol. 66, no. 1, p. 102490, 2021. G. Gilanie, U. I. Bajwa, M. M. Waraich, M. Asghar, R. Kousar, A. Kashif, and H. Rafique.

[19] C. Ouchicha, O. Amor, and M. Meknassi, "CVDNet: A novel deep learning architecture for detection of coronavirus (corona) from chest radiography images," Chaos, Solitons & Fractals, vol. 140, no. 1, pp. 110245, 2020.

[20] J. Y. Kim, W. S. Choi, N. H. Kim, K. Y. Lee, E. T. Jeon, and K. S. Lee, "Using explainable deep learning, the scalability and degree of corona screening using deep convolutional neural networks on chest radiography images are evaluated algorithm," Journal of Personalized Medicine, vol. 10, no. 4, p. 213, 2020.

[21] P. R. Bassi and R. Attux, "A deep convolutional neural network for corona identification using chest radiography," Research on Biomedical Engineering, vol. 1, no. 1, pp. 1-10, 2021.

[22] "Improving the performance of CNN to estimate the likelihood of corona utilising chest radiography pictures with preprocessing techniques," International Journal of Medical Informatics, vol. 144, no. 1, pp. 104284, 2020.

ORIGINALITY REPORT

18%	12%	12%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ir.juit.ac.in:8080 Internet Source	2%
2	Abolfazl Karimiyan Abdar, Seyyed Mostafa Sadjadi, Hamid Soltanian-Zadeh, Ali Bashirgonbadi, Mehran Naghibi. "Automatic Detection of Coronavirus (COVID-19) from Chest CT Images using VGG16-Based Deep-Learning", 2020 27th National and 5th International Iranian Conference on Biomedical Engineering (ICBME), 2020 Publication	1%
3	thesai.org Internet Source	1%
4	www.researchgate.net Internet Source	1%
5	Submitted to Higher Education Commission Pakistan Student Paper	1%
6	link.springer.com Internet Source	1%