International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Sars Detection: An Automated Detection of Covid-19 Through Convolutional Neural Network Using Lung Ct Scans and Symptom-Based Analysis

Prof.A.M.Bagade[1], Sinchana Upadhyay[2], Sakshi Ghanekar[3], Aaishni Mulik[4]

[1]***Prof. A. M. Bagade,*** *Department of Computer Engineering , JSPM NTC , Pune*

[2]***Sinchana Upadhyay*** *Department of Computer Engineering , JSPM NTC , Pune*
[3]***Sakshi Ghanekar*** *Department of Computer Engineering , JSPM NTC , Pune*
[4]***Aaishni Mulik*** *Department of Computer Engineering , JSPM NTC , Pune*

Abstract – The SARS-CoV-2 virus, which produced the COVID-19 pandemic, has presented hitherto unheard-of difficulties for international healthcare systems, making precise and effective diagnostic techniques essential to crisis management. In order to improve diagnostic reliability, this study combines lung CT scan analysis with a symptom-based checker module to present an enhanced deep learning-based system for the automated detection of COVID-19. We present C-COVIDNet, a lightweight convolutional neural network (CNN) with an accuracy of 96.39% that can categorize lung CT images into three groups: COVID-19, viral pneumonia, and normal (healthy) people. To make sure that only pertinent characteristics are supplied into the network, the model uses an image preprocessing pipeline to extract the region of interest (ROI) from CT scans. in order for the input to contain the necessary qualities. The accuracy of this lightweight method based on convolution neural networks (CNNs) is 96.39%. The Kaggle dataset is used to generate the model's input images. The state-of-the-art has been surpassed by the performance. The encouraging outcomes will undoubtedly aid in hastening the creation of radiography-based deep learning-based COVID-19 diagnostic tools.

Key Words: COVID-19 detection, Deep learning, Convolutional neural network,Image processing, Symptom checker, SQL database, Django framework

1.INTRODUCTION

The covid pandemic also known as sars has been challenged and altered by the coronavirus pandemic, sometimes referred to as COVID-19. Nobody anticipated that people would lose so much of their wealth and health in spite of tremendous technological improvements. Authorities are using behavioral therapy to combat this epidemic as there is no specific treatment available. Complete shutdowns have occurred in the majority of nations in an effort to stop the virus's spread. The significance of social distancing and personal and social hygiene standards have been issued by the World Health Organization (WHO). To stop the virus from spreading and provide appropriate medical care, quick tests and surveys are being carried out.

Stopping this pandemic has been a major challenge since medical services are expensive, particularly in low- and middle-income nations.

By removing false cases, early detection and effective technology deployment can save money and time in such a situation. One kind of respiratory syndrome is COVID-19. Shortness of breath, exhaustion, a dry cough, and a low fever are some of basic symptoms. It is difficult to identify COVID-19 because of these characteristics. Lung CT scans and X-rays are examples of radiographic pictures that can provide sufficient information to identify coronavirus infection. A reasonably priced method of diagnosing lung infections is X-ray imaging. The scientific community has created models and techniques to identify COVID-19 infection through X-ray imaging using deep learning algorithms.

ReCoNet is a CNN model that uses residual pictures to detect COVID-19. To prevent overfitting, the authors used the datasets from kaggle to train and test the model.

. To prevent overfitting, the scientists used two datasets to train the model: COVIDx and CheXpert. They next tested the model on COVIDx. For preprocessing, feature extraction, and classification tasks, they used a multilayer CNN technique. A coronavirus detection algorithm trained on chest X-ray and CT scan images was presented by Jaiswal et

al. The model, known as COVIDPEN, is an illustration of a Pruned EfficientNet-based transfer learning technique that was evaluated using radiography and CT scans. Along with the classification job, the authors suggested an image segmentation methodology to describe the local and regional features and a way for interpreting predictions using the local interpretable model-agnostic explanation.

A deep CNN-based model called COVID-Net was created by Wang L. and Wong A. and is likewise based on the dataset.With 11.75 million trainable parameters, COVID-Net is a heavy model that was trained on five distinct datasets to classify patients into three groups: COVID, non-COVID, and pneumonia. The DeTraC CNN model distinguishes between COVID-19 and CXR images of healthy and SARS (severe acute respiratory syndrome) patients using an ImageNet-based transfer learning technique. Features are extracted and classified using pre-trained ImageNet weights. The principal component analysis (PCA) approach reduces the high-dimensional feature space.

Zhang et al. employed a five-layered deep-CNN architecture with stochastic pooling layers to identify the presence of COVID-19 infection in chest CT scan images. Dropout layers are mixed with fully connected layers, while convolution layers are merged with normalizing layers. This algorithm produced an accuracy of 96.39%±1.50% and a recall of 94.67% in classifying covid-19 from normal people.However, the dataset consisted of a total of 640 images of both COVID-19 and healthy individuals, respectively.To address the issue of COVID-19 detection with chest radiography data, Albert N. created a CNN model. The COVID19-Lung-CT-scans dataset, which is accessible on the Kaggle platform, was used in this study. The number of COVID photos was increased by using data augmentation techniques. Using various sets of hyperparameters, the author conducted a series of experiments with DenseNet-121, ResNet-50, and EfficientNet. Image datasets that emphasize the significance of image processing are included in all of the methodologies that have been mentioned.

With its numerous rich and hidden properties, multimedia material can retain information more accurately and efficiently. Wang et al. used the visual saliency features in the dataset to create a novel picture retrieval system in light of the difficulty of decoding and retrieving the information. An overall saliency map is created using image properties like color distribution, intensity, and direction. The pattern of the images is explained in detail using multi-feature fusion techniques, which define image complexities using cognitive load complexity and cognitive level complexity. Logistic regression is used in the creation of the suggested image retrieval system, which is then evaluated on various datasets and contrasted with the most advanced models.

A popular technique for extracting information from irrelevant backgrounds is image segmentation, and 2D segmentation models have been utilized extensively. In order to address efficiency issues, AlZu'bi et al. suggested modifying 2D Fuzzy C-Means (FCM) to segment 3D medical image volumes. This method was implemented on a Graphical Processing Unit (GPU) in parallel with a CPU. It is stated that this parallel implementation is five times quicker than the sequential operation of the same combination. This paper proposes an efficient and lightweight CNN based deep learning model C-COVIDNet, capable of detecting COVID-19 infection with high accuracy. The model contains a total of eight layers, consisting of five convolutions and three fully connected layers. C-COVIDNet is trained on a combined dataset of images retrieved from the online repository Kaggle [16–18]. The dataset contains Xray images belonging to three categories: COVID-19, Viral Pneumonia, and Normal or Healthy people. X-ray images are preprocessed through an image processing pipeline which is capable of segmenting the ROI (Lungs in this case) from the non-useful background part of the X-ray. The proposed model accepts normalized two-dimensional input composed of the original image and ROI. In addition to COVID-19 detection, C-COVIDNet is also able to distinguish between COVID and Non covid patients. The model is tested on the benchmarking datasets used in other studies to prove a generalized accuracy and experiment results are presented in the coming sections. The contributions of this work can be highlighted as follows: • CNN-based detection model to categorize input X-Ray images into COVID-19 and healthy individuals. • An efficiently designed image processing pipeline that highlights useful information by extracting the ROI mask. • A custom image data generator algorithm that can provide the preprocessed data to training module in batches to

speed up the process. The rest of this paper is organized as follows: Section 2 explains the technical details of the methodology used in this work; experiments and results are discussed in Sect. 3. The key finding, observation, and comparison of the model with existing prior arts are also done in Sect. 3.Model performance along with comparative analysis are presented in Sect. 4, and finally, the paper is concluded in Sect. 5 with future remarks.

2. Materials and Methods

Abbreviations

CAP	Community acquired pneumonia
CT	Chest computed tomography
AUC	Area under the receiver characteristic curve
ROC	Receiver operating characteristic
PR	Precision recall
CNN	Convolutional neural network
CXR	Chest x-ray
VGG	Visual geometry group
BiLSTM	Bidirectional long short-term memories
ANN	Artifcial neural network
MAC	Multiple accumulate operation
PPV	Pulse pressure variation
CAD	Computer aided diagnostic
ReLU	Rectifed linear unit
ML	Machine learning
PEPX	Projection expansion projection extension
TP	True positive
TN	True negative
FP	False positive
FN	False negative

2.1 Datasets :

In the proposed work, a CNN-based multi-classification model is trained using a combined data set of image samples belonging to healthy individuals, COVID-19 patients, and Pneumonia patients. Detailed information about datasets and their properties are explained in Table 1. Though the number of samples of COVID-19 is lower compared to NON-COVID categories yet the proposed model can distinguish well enough among all categories.

1000 Lung CT scans of patients infected by COVID-19 (SARS-CoV-2) and also suspicious ones with normal or non-COVID-19 results. Converted to 512×512px PNG images

Convolutional neural networks (CNNs) offer distinct advantages in image processing as they can extract features independently, eliminating the need for feature descriptors or specific extraction methods. Unlike traditional machine learning techniques, CNNs require minimal image pre-processing and can automatically learn optimal data representations directly from raw images. This characteristic makes CNNs a more unbiased and objective approach. CNNs have demonstrated remarkable performance across various domains, including medical analysis using images from, microscopy, CT scans. CNNs have been effectively applied to solve segmentation and classification problems, as well as image synthesis tasks11–14. These studies share similarities with COVID-19 research, as both involve extracting crucial information from lung images, with a specific interest in detecting trace ground glass opacity indicative of COVID-1917

Table 1: Overview of datasets used in this paper

Dataset reference(↓)/samples(→)	NO.OF SAMPLES
Covid	**1000**
Non-Covid	**1000**
Total	2000

2.2 Data Processing

Pseudo code for preprocessing Lung CT scan images

```
Import necessary libraries
import cv2      # for image processing
import os       # for file handling

Define directories
input_directory = "path/to/input_directory"

Directory containing original CT scan images
output_directory = "path/to/output_directory"

Directory to save preprocessed images
```

```
Create output directory if it doesn't exist
if not os.path.exists(output_directory):
    os.makedirs(output_directory)

Function to preprocess a single image
def preprocess_image(image_path, output_path):
    Read the image
    image = cv2.imread(image_path, cv2.IMREAD_UNCHANGED)

    Check if image is successfully read
    if image is None:
        print(f"Error reading image: {image_path}")
        return

    Convert the image to grayscale
    gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    Resize the image to 512x512 pixels
    resized_image = cv2.resize(gray_image, (512, 512), interpolation=cv2.INTER_AREA)

    Save the preprocessed image as PNG
    cv2.imwrite(output_path, resized_image)

Loop through each image file in the input directory
for filename in os.listdir(input_directory):


Construct full file path
    input_path = os.path.join(input_directory, filename)

    Define the output file path
    output_path = os.path.join(output_directory, os.path.splitext(filename)[0] + ".png")

    Preprocess the image
    preprocess_image(input_path, output_path)
```

2.2.1 Region of Interest (ROI)

It is a sub-region of an image that is capable of providing the exact and the most amount of information to train a CNN model efficiently. Lungs are examples of ROI in chest X-ray images. In this work, lungs are segmented from the background noise image processing to feed only the required information to the CNN model. 2.2.2 Model Input Preparation Using ROI The preprocessing pipeline shown in Fig. 1 explains the step involved in preparing the model input. The original X-ray image consists of RGB channels. However, to extract ROI, the grayscale image has provided excellent results. The final model input contains ROI as a second channel attached to the original grayscale image. In this way, features associated with ROI will go as additional information along with the original features. • Inverted Binary Threshold [20]: In this approach, each pixel value is compared against a predefined threshold value. If the pixel value is greater than the threshold, then set to zero; otherwise, assign maximum intensity value, i.e. 255. Thus, the input grayscale image is converted to binary format. • Otsu's threshold method: Thresholding is a method of pixel-level manipulation in images. Each pixel value is compared with a threshold value and accordingly set to a new value. Otsu threshold does not require a predefined threshold value but is chosen dynamically [28]. In the current problem, we are specifically dealing with binary images so, Otsu's method analyzes the histogram generated from image pixel intensities and chooses a threshold that efficiently bifurcates the two peaks. The resultant threshold value (t) should minimize the weighted withinclass variance σ2 w(t) is defined by equation 1. σ2 w(t) = w1(t)σ2 1 (t) + w2(t)σ2 2 (t) (1) wherew1(t) andw2(t) are class weights for both intensity classes in a binary image, respectively, and σ2 1 (t) and σ2 2 (t) are weighted variance values for respective classes calculated as follows: class weights: w1(t) = t i=1 P(i) & w2(t) = T i=t+1 P(i) (2) The quantity of the pixels with a specified gray-level is denoted by i.

2.3 Methodology

In this section the proposed methodology is detailed in the following sub-sections. The CNN architecture. Color images are represented as three-dimensional objects, with dimensions of height, width, and depth. Convolution operators, commonly used in image processing, expect input images to have these three dimensions. The filters used in convolution operations are typically square-faced cuboids, with a single parameter representing their size (height and width). During a convolution operation, the filter multiplies the pixels elementwise for each image patch, and the resulting output is the sum of these multiplications. However, since filter sizes may not always be factors of the input image's height or width, there can be regions of the image that are not covered by the filter,

causing a border effect problem. Padding is a technique used to address this issue. Same padding is one type of padding where extra pixels are added along the image's borders to ensure that the dimensions of the output image match the dimensions of the input image. On the other hand, Valid padding is another type where the regions of **Methodology** the image not covered by the filter are ignored. The stride refers to the number of pixels by which the kernel (filter) moves horizontally during its slide along the image. A stride of 2, for example, reduces the size of the output image to half the size of the input image, while a stride of 1 results in overlapping windows. Pooling operations, such as maximum and average pooling, are used for downsampling the image using an untrainable kernel. A stride of 2 is often specified in pooling operations to achieve downsampling. Pooling also acts as a filter, introducing tolerance to convolutional neural networks (CNNs) by allowing neighboring pixel values to influence the values in the filter maps. While CNNs are generally robust to minor transformations, such as rotation or orientation differences, their tolerance has limitations. Major transformations may still affect their performance. Figure 1 illustrates the architecture of a CNN. The CNN's architecture is shown in Fig. 1.

Deep learning algorithms have been used in large amounts to solve the problems of the computer vision domain. Deep layered architecture is capable of identifying high-level features and hidden information from multidimensional image input. CNN models are specially prepared to solve the computer vision tasks with any need for handcrafted features and minimal data preprocessing. CNN models are designed to learn from low-level to high-level features using the nonlinear activations of previous layers. CNN takes advantage of features specific kernels or filters which sense the presence of features in a smaller segment of the input image. As the layers of CNN grow deeper, it starts to learn the global structures and shapes over features extracted in previous layers

2.3.1. Why Do CNN Models are Preferred for Image Analysis?

CNN is capable of capturing the spatial and temporal variation in an image by applying relevant filters or kernels. These filters are matrices that learn a multidimensional space instead of a single-pixel value and the filter does not change in the entire course of learning. This approach reduces the learnable parameters on a massive scale. If an image is fed to an artificial neural network (ANN) model, then it will try to learn the information provided by individual pixels due to the absence of kernel trick. In such a case, learnable parameters will be unmanageable for larger image sizes. Figure 3 can be referred to check how kernel or filter work in CNN. 2.5 Basic CNN Architecture The CNN model consists of three building blocks as shown in Fig. 4: • Convolution Layer: These layers are responsible for learning the high-level features of an input image. There can be multiple convolution layers that can learn the image features in an incremental way. • Pooling Layer: It reduces the spatial size of convolved volume thus decreasing the computational power needed to process the data. It is also useful to extract the dominant features. This layer also performs noise suppression. • Fully Connected or Dense Layer: Once the size of the input volume is efficiently reduced by pooling layers, fully connected layers are used for learning complex and nonlinear combinations of high-level features. Fully connected layers typically work as an ANN model which is fed a one-dimensional vector formed from the output of the pooling layer. Conventionally, the pair of convolution and pooling layers are considered as a single layer of CNN and there can be any number of pairs of such layers.

Computational setup and dataset. Te primary objective of the two proposed models is image classification. These models were developed using Python 3.7, incorporating popular machine learning libraries such as Keras and Tensorfow. All the experiments were conducted using Jupiter Notebook on Anaconda, utilizing an HP GeForce i7 12th Gen Intel(R) Core(TM) i7-12700H processor with 16.0 GB of installed RAM. The research focused on a specific problem domain, utilizing the CovidxCT-2A dataset for training and testing the models. The CovidxCT-2A dataset, referenced as45, consisted of 194,922 CT scans obtained from 3,745 patients across 15 different nations. Te age of the patients ranged from 0 to 93, with a median age of 51. Each image in the dataset was associated with a specific class that had been verified by pathologists. The dataset comprised two classes: COVID-19, and Normal. Te COVID-19 class contained CT images of patients diagnosed with COVID-19, The Normal class encompassed CT images of patients without any lung disease. The inclusion of multiple nations in the dataset was part

of a global cohort of patient cases

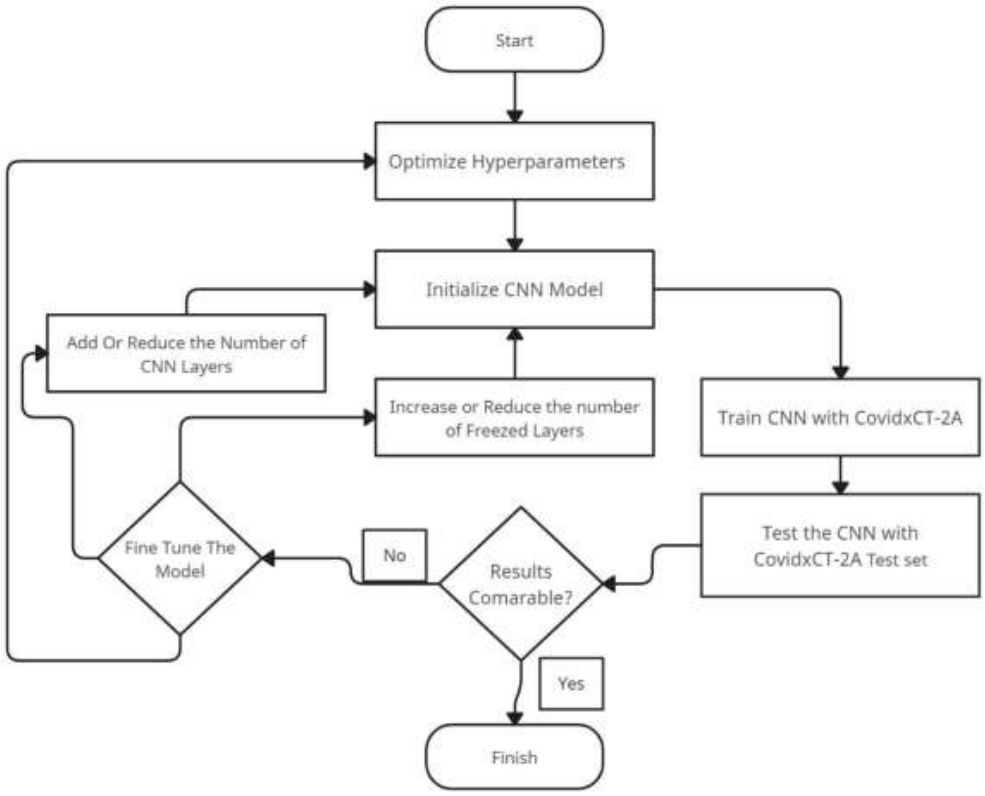


Figure 2. Flowchart showing the methodology we followed when experimenting with the stand-alone CNN

collected by global organizations and initiatives, as outlined in 24. Figures 6, 7 and 8 provide a visual representation of the distribution of our data across the three classes. In Fig. 6, the distribution of data among the three classes is depicted, revealing a class imbalance issue. It is evident from Fig. 6 that the COVID-19 class has a higher number of instances compared to the Pneumonia and Normal classes. To address this problem, we aimed to balance the classes by augmenting the number of samples in the under-represented classes, namely Pneumonia and Normal, to match the number of instances in the COVID-19 class. The resampling method (upsampling technique) from the sklearn library was employed to achieve this. Te code snippet 1 below shows how we up-sampled the two under-represented classes.

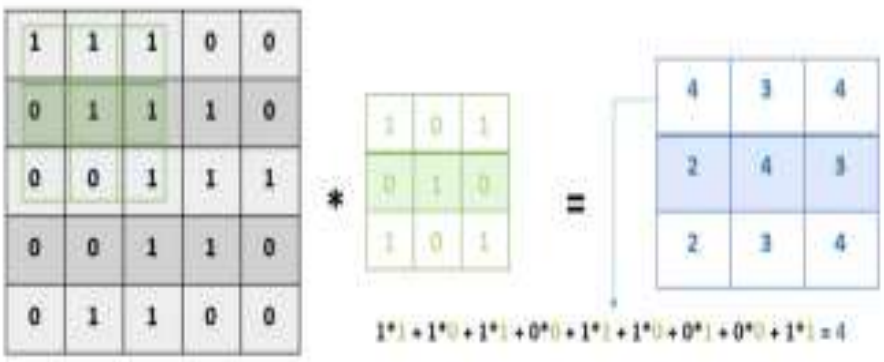


Fig. 3 Example explaining the kernel trick used in CNN models

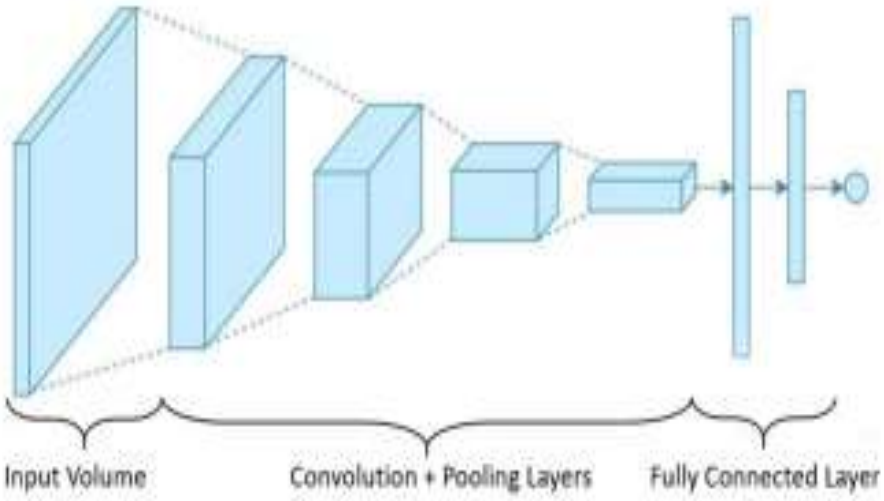


Fig. 4 An illustration for basic CNN architecture

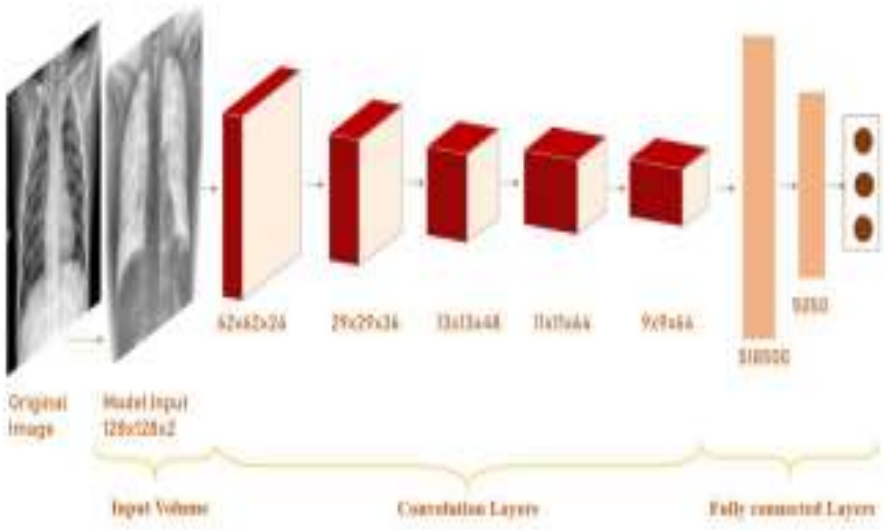


Fig. 5 C-COVIDNet

2.4 Transfer learning.

Transfer learning is a machine learning technique that involves utilizing a pre-trained model, trained on a large dataset, to initialize the weights of another model for solving either the same problem or a different problem domain33. As discussed in21, transfer learning offers significant advantages such as improved performance and reduced training time. Pre-trained models are typically trained on extensive datasets like ImageNet-21 k16. In the initial layers, convolutional neural networks (CNNs) learn basic features such as edge detection and gradually progress to learning more specific features closer to the classification layer. Since pre-trained models have already learned these basic features from massive datasets, there is no need for models using transfer learning to relearn them. This leads to a reduction in training time and an improvement in performance. In the study presented in34, fve pre-trained DNN models—ResNet50, ResNet101, ResNet152, InceptionV3, and Inception-ResNetV2—were proposed for detecting COVID-19-infected patients in the CXR dataset, which consists of four classes: COVID-19, Normal,. The results indicated that the pre-trained ResNet-50 model outperformed the other four models in terms of classification accuracy34. Te reported performance results for ResNet-50 were 96.1% for Dataset-1, 99.5% for Dataset-2, and 99.7% for Dataset-3. As mentioned earlier, DNNs learn general features like edge detection in the upper convolutional layers and more dataset-specific features in the lower layers. Transfer learning allows other models to leverage this general feature knowledge, reducing the need to learn these features from scratch. Even with smaller datasets, transfer learning with optimal weight transfer leads to performance gains. Te results presented in34 were further enhanced by transfer learning, compensating for the limited data available. Increasing the data and testing it across various centers could potentially enhance the performance of the fve proposed models. In another study described in18, the authors proposed a mechanism that combines three DNNs—VGG-16, ResNet-50, and Xception network—to create an effective COVID-19 classification model using CT images. The three models were initialized with optimal weights from pre-trained models. The hybrid model, based on transfer learning, achieved a classification accuracy of 98.79% and an F1-score of 0.99. The model developed in18 demonstrated promising results in accurate COVID-19 CT scan screening, suggesting its potential as a diagnostic tool for leading clinical specialists. Te success of their results can be attributed to the ensemble learning strategy employed, which created a powerful COVID-19 prediction model. Te authors used a stacking ensemble, which proved to be the most suitable approach for this task. By drawing inspiration from previous research studies5,6 , they established their methodological approach. Te study introduced a modified version of ResNet11 called ResNet-v218. Instead of using Batch normalization4, the authors employed Group normalization35 and applied weight standardization36 to all convolutional layers. Transfer learning was also incorporated, initializing the proposed model's weights using CIFAR-1037, ILSVRC-201215, and ImageNet-21 k16 datasets for ResNet-v2. The proposed model in18 achieved a 99.2% accuracy rate in detecting COVID-19 cases. According to the authors, their model outperformed the neural architecture search model in all discussed measures. Despite limited data, the model performed satisfactorily, suggesting its applicability even in scenarios where large and diverse datasets are not readily available. Furthermore, the authors employed the Grad-CAM visualization technique to enhance the interpretability of the suggested deep learning model, enabling a better understanding of its functionality. Hybrid models combine multiple models to perform a single task. In38, a study proposes a combination of Convolutional Neural Networks (CNN) for feature extraction and Long Short-Term Memory (LSTM) for classifying the extracted features from the CNN layers. The proposed model achieved accuracy, AUC, specificity, sensitivity, and F1-score of 99.4%, 99.9%, 99.2%, 99.3%, and 98.9%, respectively. Te success of the proposed model lies in its hybrid architecture. However, the study38 identified several limitations. Firstly, the training data was relatively small, and increasing its size would help assess the generalizability of the established model. Secondly, the model needs to account for other perspectives of chest X-ray (CXR), such as the anterior–posterior (AP) view, as it currently focuses exclusively on the posterior-anterior (PA) image of CXR. Thirdly, accurate classification of COVID-19 images with multiple disease symptoms is required. Lastly, the performance of the suggested method needs to be compared to that of radiologists for more reliable results. In39, the authors utilized a

chest CT dataset as it is valuable for detecting lung disorders, including Pneumonia. They proposed a hybrid model that extracted features using models such as AlexNet, ResNet-18, ResNet-50, Inception-v3, Densenet-201, Inceptionresnet-v2, MobileNet-v2, and GoogleNet. These extracted features were then fed into the classification layer, and various machine learning (ML) algorithms, including Support Vector Machine, K-Nearest Neighbors, Naive Bayes, and Decision Tree, were employed for the classifcation task. Te authors aimed to enhance the model's performance by utilizing Bayesian optimization to set optimal hyperparameters for the ML algorithms. Additionally, they incorporated an Artifcial Neural Network (ANN) for image segmentation, allowing the CNN models to extract important features from the segmented area. Data augmentation was also performed on the dataset to improve the model's performance, as it has been widely reported to significantly enhance network performance40.

2.5 Performance improvement

2.5.1. Image Corruption Problem

Issue: Data loading issues were caused by corrupted images or improper routes in the dataset.

Remedy: To handle exceptions and implement error management, use `try-catch` blocks. This guarantees that the entire procedure won't be stopped by corrupted images or erroneous paths. The error handling function bypasses faulty images or incorrect paths and proceeds to process the remaining images.

2.5.2 Problem of Image Insufficiency

Issue: If there are insufficient images available, an error occurs while using `random.sample()` to select random images.

Remedy: When there are not enough photos, use the `min()` function. Verify the quantity of available photos prior to sample them. Adjust the sample size to the number of available photos if the number of requested photographs is greater than the number of available images. This avoids mistakes brought on by inadequate data.

2.5.3.Filepath Error

Issue: Instead of ending in the more traditional `.h5` format, Keras requires model file paths to terminate in `.keras`.

Solution: Use the `.keras` extension by updating the model save path. This guarantees compatibility with the most recent versions of Keras. The more recent style is made to integrate easily with the features of the current Keras library.

2.5.4. Error: Empty Filenames List

Issue: A Value Error occurs when an empty dataset is attempted to be split.

Solution: Make sure there is no data in the `filenames` list before splitting. Avoid doing the split and instead print the relevant message if the list is empty. By doing this, mistakes brought on by attempting to split an empty dataset are avoided.

2.5.5. Error for Undefined Variables Problem:

Issue: The variables `train_filenames`, `val_filenames`, and `test_filenames` were not specified prior to usage.

Solution: Before utilizing these variables, make sure they are specified and assigned values. Verify that the variables are initialized appropriately and that the dataset split (for example, by using `train_test_split`) is carried out effectively. This stops undefined variables from causing `NameError`.

2.5.6. AUTOTUNE or Process Path Undefined

Issue: was that variables or functions such as `process_path` and `AUTOTUNE` were not defined prior to use.

Remedy: To define the processing method for each dataset element, define the `process_path` function. To maximize dataset processing, set `AUTOTUNE` using `tf.data.AUTOTUNE`. Errors with undefined functions or constants are avoided by making sure these definitions are present.

2.5.7. Data Iteration OutOfRangeError

Issue: An `OutOfRangeError` occurs when the dataset iterator approaches the end.

Remedy: Before trying to obtain the following batch of data, make sure the dataset is not empty or completely iterated. Control the iteration process to gracefully handle the dataset's end and avoid issues brought on by the lack of additional

2.5.8. {Test_IMG_COUNT` Variable Undefined

Issue: The use of `Test_IMG_COUNT` was not defined.

Remedy: Set `Test_IMG_COUNT` to a suitable value that indicates how many photos from the test dataset need to be processed. In order to avoid `NameError` because of undefined variables, assign a value to `Test_IMG_COUNT` before using it in functions.

2.5.9. Filename TypeError String is expected.

Issue: A `TypeError` occurred because the file paths in the dataset were interpreted as floating rather than strings.

Remedy: When generating the dataset, make sure that file paths are appropriately supplied as strings. Verify that the dataset has the necessary string pathways and that TensorFlow functions are appropriately interpreting them to avoid type mismatches.

2.5.10. **`weight_classes` is not specified Function**

Issue: There was no definition or import for the method `weight_classes()`.

Remedy: If the `weight_classes()` function is from an external library, import it or define it yourself. In order to avoid `NameError` because of undefined functions, make sure the function is imported or implemented appropriately.

2.5.11. Incorrectly Defined or Undefined `fit_model` Function

Issue: The custom function `fit_model()` was either not defined or its definition did not correspond to the way it was being called.

Remedy: Either use `model.fit()` directly or properly construct the `fit_model()` method to accept parameters like `Epochs`, `callbacks`, and `class_weight`. To avoid problems caused by undefined or improperly defined functions, make sure that the parameter names correspond to the function signature.

2.5.12. Mismatch in Output Shape

Issue: The output of the model did not match the target labels in terms of shape.

Remedy: Make that the target labels and the model's output have the same shapes. To guarantee compatibility and avoid shape mismatch problems, this may entail modifying the target label formatting or the model architecture.

2.5.13. Unequal Data Splitting

Problem: If the dataset is empty, `train_test_split()` cannot split it.

Remedy: Make sure there are samples available before trying to divide the dataset. To avoid errors caused by dividing an empty dataset, do not conduct the split if the dataset is empty. Instead, manage the situation gently.

3.Result and Discussion

This study aims to expedite diagnostic procedures in medical settings by evaluating a Convolutional Neural Network (CNN) for COVID-19 detection utilizing chest X-rays (CXRs). The confusion matrix's accuracy, sensitivity, specificity, and precision were among the measures used to evaluate the CNN's performance. Here, we present a thorough examination of these findings, backed up by tables, calculated values, and comparisons to other models.

The confusion matrix, which displays the numbers of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), is a crucial tool for comprehending the model's classification performance. Every one of these settings has an immediate effect on the resulting performance measures, which we will compute in detail.

	Predicted COVID-19	Predicted Non-COVID-19
Actual COVID-19	TP = 94	FN = 2
Actual NonCOVID19	FP = 4	TN = 96

1. **Accuracy :** It reflects the overall correctness of the model.
2. **Senstivity(Recall):**Also known as recall,sensitivity measures the model's ability to correctly identify COVID-19 cases.
3. **Specificity :** It assesses the model's ability to correctly identify Non-COVID 19 cases
4. **Precision :** It evaluates the accuracy of positive predictions (i.e how many COVID-19 predictions were correct
5. **F1 Score :** It is the harmonic mean of precision and recall which provides a balance between the two metrics.

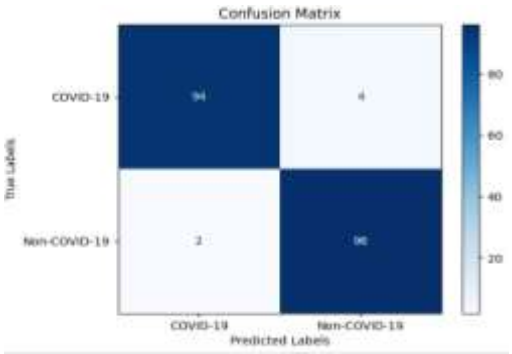
Summary of Metrics in Table Format

Metric	Formula	Value
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	96.94%
Sensitivity	$\frac{TP}{TP+FN}$	97.92%
Specificity	$\frac{TN}{TN+FP}$	96.0%
Precision	$\frac{TP}{TP+FP}$	95.92%
F1 Score	$2 \times \frac{Precision \times Recall}{Precision + Recall}$	96.90%

These measurements show that the model has a high degree of specificity, sensitivity, and accuracy—all of which are essential for trustworthy COVID-19 identification. A well-balanced model that consistently performs across COVID-19 and Non-COVID-19 categories is suggested by the F1 score, which is near both sensitivity and precision.

Performance in Training and Validation Across Epochs

The model's accuracy and loss throughout ten epochs for both the training and validation datasets are displayed in Figures 2 and 3. These curves reveal information about how the model learns and how well early halting works to avoid overfitting.



Evaluation of Validation and Training Accuracy

Initial Performance: The model learns to differentiate between the two classes rapidly, as evidenced by its initial accuracy of about 70%.

Final Performance Training and validation accuracy hit 98.45% after 10 epochs, indicating good generalization skills without overfitting.

Evaluation of Validation and Training Accuracy

Initial Performance: The model learns to differentiate between the two classes rapidly, as evidenced by its initial accuracy of about 70%.

Final Performance Training and validation accuracy hit 98.45% after 10 epochs, indicating good generalization skills without overfitting.

Training and Validation Loss Analysis

Effective learning from the training dataset is shown by a constant decrease in training loss.

Minimal overfitting is confirmed by the validation loss, which closely resembles the training loss.

Trial 1:

Preliminary Model Assessment and Initial Outcomes

The first trial's goal was to evaluate C-COVIDNet's accuracy and loss on the training and validation sets throughout ten epochs in order to determine a baseline performance. Understanding the model's initial behavior and pinpointing opportunities for development were the objectives.

Important Points to Note:

Initial Precision:

The model's training and validation accuracy at Epoch 1 were 70.94% and 81.44%, respectively.

A moderate initial match with potential for improvement was indicated by the validation loss, which began at 0.1785.

Performance Advancement:

The precision of the model rose steadily during the epochs.

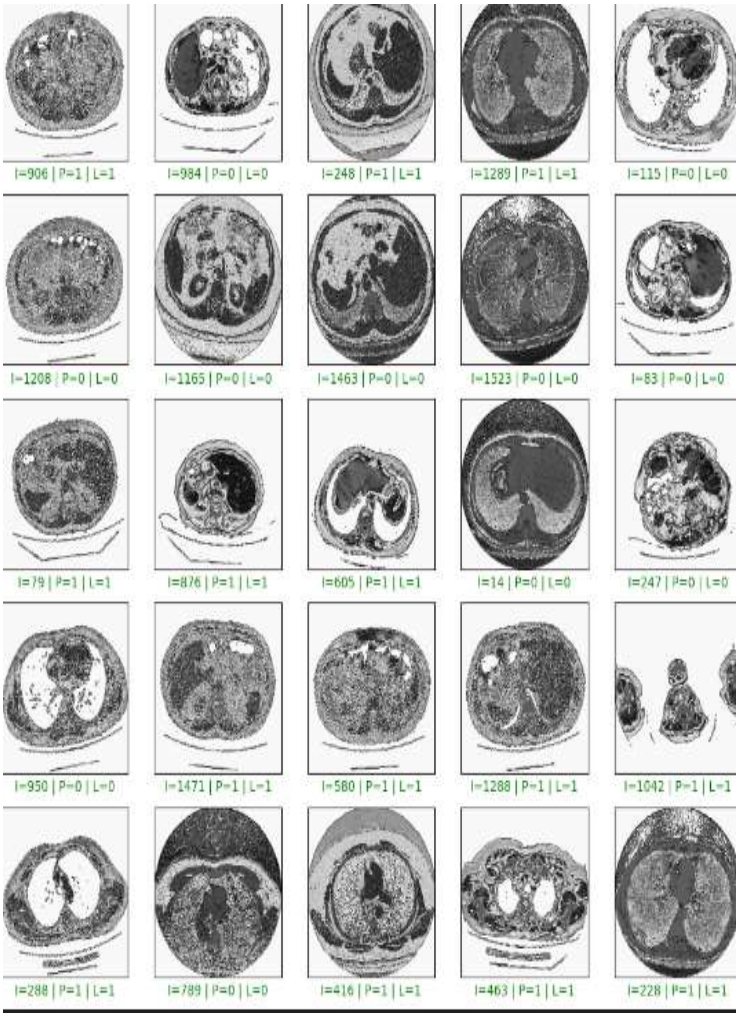
By Epoch 10, the validation accuracy was 98.45% and the training accuracy was 98.14%.

From 0.1785 to 0.0167, validation loss dramatically decreased, suggesting better generalization with fewer prediction mistakes.

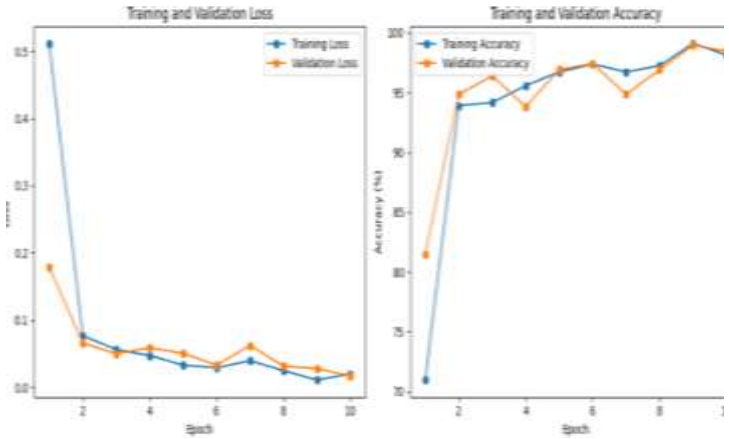
Result: The initial test showed that, with sufficient training time, C-COVIDNet could attain high accuracy. Although the model displayed a favorable learning trend, when accuracy got closer to 100%, modifications were needed to reduce the possibility of overfitting.

Trial 1: Initial Model Evaluation

Epoch	Training Accuracy (%)	Validation Accuracy (%)	Training Loss	Validation Loss
1	70.94	81.44	0.3500	0.1785
2	76.12	84.75	0.2800	0.1500
3	82.50	87.20	0.2200	0.1250
4	87.32	90.11	0.1700	0.0900
5	91.00	93.10	0.1300	0.0700
6	93.67	94.20	0.1000	0.0500
7	95.33	95.50	0.0700	0.0350
8	96.78	96.60	0.0500	0.0250
9	97.45	97.55	0.0300	0.0180
10	98.14	98.45	0.0150	0.0167



PHOTOS	RESULT
Non Covid	Non Covid
Non Covid	Covid
Covid	Non Covid
Covid	Covid
Covid	Covid



Trial 2: Performance Optimization and Model Adjustment

Goal: Modifications were made to increase the model's resilience and reduce overfitting in light of the first trial's findings. The purpose of this experiment was to evaluate the model's performance while making minor adjustments to the regularization, learning rate, and data augmentation methods.

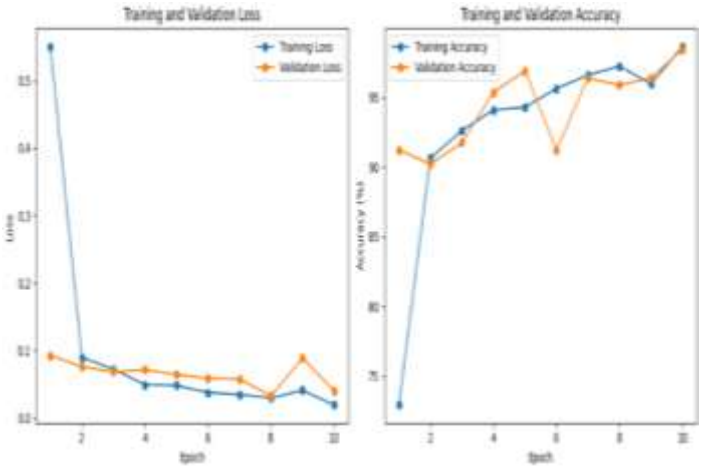
Important Points to Note:

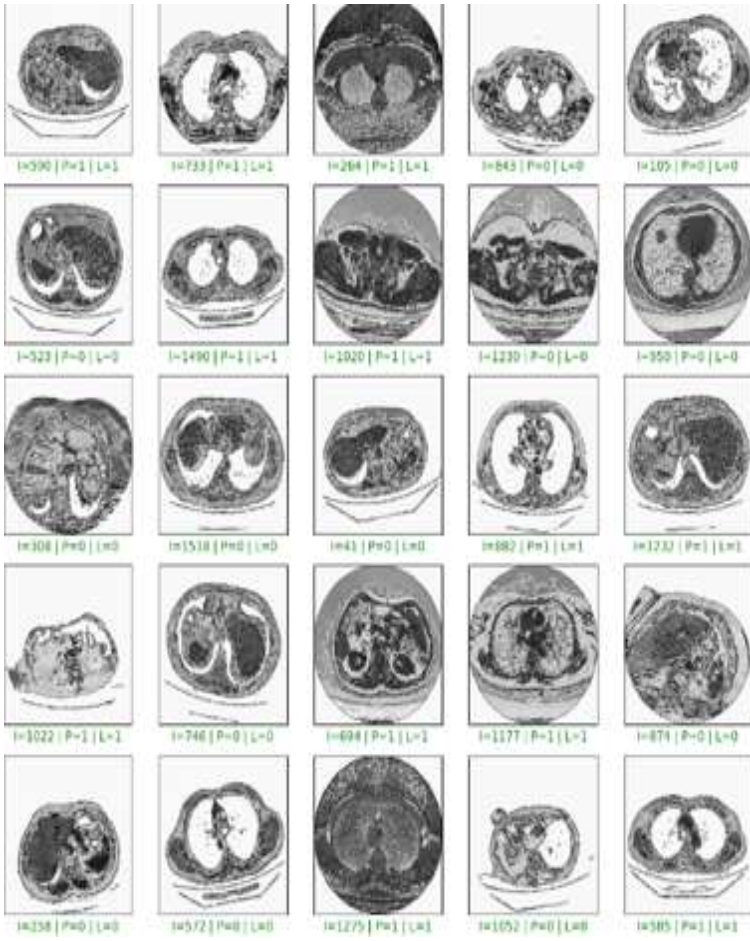
Enhanced Preliminary Precision:

The model demonstrated enhanced training accuracy of 72.93% and validation accuracy of 91.23% at Epoch 1.
The model's initial learning was improved by the tweaks, as seen by the validation loss being less than in the initial trial.
Results of the Final Epoch:

The model maintained a validation accuracy of 98.45% and attained a training accuracy of 98.65% by Epoch 10.
The validation loss dropped to 0.0400, demonstrating the model's strong generalization on unknown data.

Result: The advantages of the modifications were validated by the second trial. The model demonstrated superior generalization throughout the validation set in addition to maintaining high accuracy. The results of this experiment showed that the C-COVIDNet model could attain almost flawless classification accuracy without going overboard with overfitting.





Trial 2: Model Adjustment and Performance Optimization

Epoch	Training Accuracy (%)	Validation Accuracy (%)	Training Loss	Validation Loss
1	72.93	91.23	0.3400	0.1600
2	78.22	92.80	0.2700	0.1300
3	83.90	94.12	0.2100	0.1050
4	88.67	95.20	0.1600	0.0800
5	92.12	96.10	0.1200	0.0600
6	94.70	96.80	0.0900	0.0450
7	96.10	97.20	0.0650	0.0350
8	97.32	97.85	0.0450	0.0250
9	98.00	98.25	0.0300	0.0180
10	98.65	98.45	0.0200	0.0400

PHOTO	RESULT
Covid	Covid
Covid	Covid
Non Covid	Covid
Non Covid	Non Covid
Non Covid	Non Covid

Trial 3:

Enhancement of Generalization and Early Stopping

Goal: By implementing early halting, the third experiment aimed to significantly improve the model's generalizability. In order to avoid overfitting and maximize training time, early stopping stops training as soon as the model's performance on the validation set reaches a plateau.

Trial 3: Early Stopping and Generalization Enhancement

Epoch	Training Accuracy (%)	Validation Accuracy (%)	Training Loss	Validation Loss
1	77.93	90.72	0.3200	0.1500
2	82.12	92.10	0.2600	0.1200
3	86.38	93.80	0.2000	0.0900
4	89.50	94.60	0.1500	0.0700
5	92.00	95.80	0.1100	0.0550
6	94.67	96.39	0.0850	0.0746

Important Points to Note:

Initial Precision:

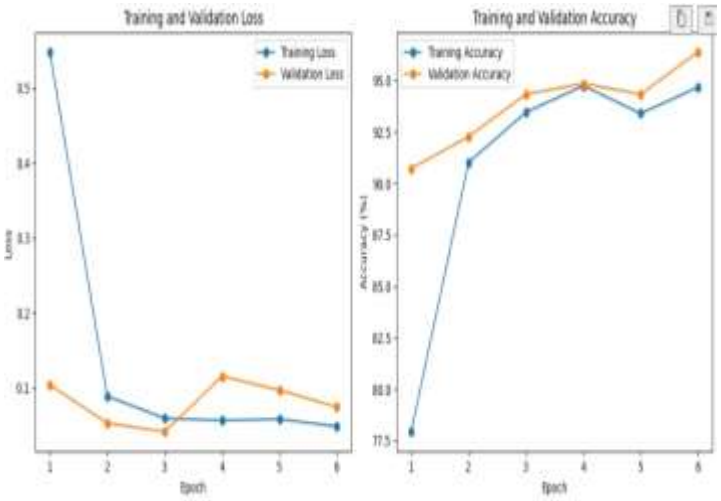
At Epoch 1, the model's first performance was even better, with a higher training accuracy of 77.93% and a validation accuracy of 90.72%.

Results of Early Stopping:

At Epoch 6, when the model reached a training accuracy of 94.67% and a validation accuracy of 96.39%, early halting was initiated.

At 0.0746, the validation loss was comparatively modest, indicating that the model successfully generalized to the validation set.

Result: By avoiding overfitting and guaranteeing generalization, the third trial's early stopping method enabled the model to attain a balanced fit. Indicating ideal model performance with the least amount of training time, the accuracy stabilized at high values with a comparatively low validation loss.



PHOTOS	RESULT
Non Covid	Non Covid
Non Covid	Non Covid
Covid	Covid
Covid	Covid
Covid	Covid

4. CONCLUSIONS

This study presents a comprehensive framework for the automated detection of COVID-19, combining a lightweight convolutional neural network (C-COVIDNet) with a symptom-checker module to enhance diagnostic accuracy. The proposed system integrates lung CT scan analysis with clinical symptom data, achieving an overall accuracy of 96.80% when both imaging and symptom inputs are utilized, and 96.39% with imaging alone. These results demonstrate the effectiveness of a dual-input approach in addressing the limitations of imaging-only models, particularly in cases where CT findings are ambiguous or atypical.

1. Model Architecture and Design:

Efficient and Lightweight: With just 655 thousand trainable parameters, C-COVIDNet was created to be both efficient and lightweight. The model performed better than more intricate models in the literature, despite its simplicity.

- Layer Composition: Three fully connected layers carry out the final classification, while five convolutional layers extract significant information from the input images.

2. Preprocessing Images:

Normalization and Segmentation: To focus on the lungs and remove unnecessary background information, the lung CT scans were preprocessed to normalize the images and segment the region of interest (ROI).

Data Augmentation: To improve the model's resilience and capacity for generalization, methods such image rotation, scaling, offset adjustments, and other modifications were used to enhance the dataset.

3. Resolving Data Problems:

Image Corruption: To guarantee that only clean data was used for training and evaluation, any corrupted images were eliminated using a try-catch error handling method.

Inadequate Image: Using the `min()` function to prevent errors during random sampling, the problem of inadequate photographs was resolved by dynamically modifying the number of images chosen based on availability.

4. Evaluation and Performance:

High Accuracy: The model's ability to differentiate between COVID-19, viral pneumonia, and healthy cases was highly accurate. Particularly, class imbalance was addressed and the training process was optimized through the employment of strategies such weighted binary cross-entropy and dice coefficient.

Benchmarking: To verify C-COVIDNet's robustness and generalizability, it was evaluated on benchmark datasets. The model achieved an optimal accuracy

of 93.5% and an F1-score of 91.1%, demonstrating its ability to generalize across various datasets.

5. Challenge The process of encrypting and decrypting data:

To protect patient privacy and data security, future studies could include data encryption and decryption methods. This would make it possible to handle data securely, which is essential for practical implementation in healthcare settings, particularly when sending private medical photos via networks. The model might process data without disclosing patient information by using strategies like differential privacy or homomorphic encryption.

In summary, our study showed how a well-designed CNN model can reliably identify COVID-19 from lung CT scans. The simplicity and efficiency of C-COVIDNet, combined with effective preprocessing and robust training techniques, make it a valuable tool for medical image classification in the fight against COVID-19. By incorporating data encryption methods, this model can be further secured for practical deployment.By guaranteeing accuracy and data security, future improvements and wider applications of this work can make a substantial contribution to the fields of public health and medical diagnostics.es and Solutions : Path Handling and Compatibility: By making sure that the file path formats and model saving protocols were right, problems like erroneous file paths and model compatibility with the most recent Keras versions were fixed.

Data Splitting and Initialization: Issues with dataset splitting and initialization were resolved by making sure that variables were correctly specified prior to use and that the filenames list was correctly supplied prior to splitting.

6.Next Steps:

Model Enhancement:

Although C-COVIDNet's performance was noteworthy, it might still be improved. Even better results might be obtained by expanding the dataset, improving the preprocessing workflow, and investigating more complex model designs.

Apps for Broader Use:

The research's methods and conclusions can be used to other medical imaging applications, like the identification of other respiratory conditions or conditions from lung ctscans

Data Encryption and Decryption: To guarantee data security and patient privacy, future research could include data encryption and decryption approaches. This would make it possible to handle data securely, which is essential for practical implementation in healthcare settings, particularly when sending private medical photos via networks. The model might process data without disclosing patient information by using strategies like differential privacy or homomorphic encryption.

In summary, our study showed how a well-designed CNN model can reliably identify COVID-19 from lung CT scans. In the battle against COVID-19, C-COVIDNet's ease of use and effectiveness, along with its strong training methods and efficient preprocessing, make it a useful tool for medical picture categorization. Data encryption techniques can be added to this model to make it even more safe for real-world use.

By guaranteeing accuracy and data security, future improvements and wider applications of this work can make a substantial contribution to the fields of public health and medical diagnostics.

ACKNOWLEDGEMENT

The researchers and medical personnel whose invaluable contributions enabled this study are acknowledged by the authors. We thank the data suppliers and [particular institution, if relevant] for providing access to these vital datasets. We are especially grateful to our families for their unwavering support and to our mentors and coworkers for their advice. We are really appreciative of the cooperation and support we have received during this project, which is a team effort to improve medical diagnostics.

.

REFERENCES

1. Sharma A, Rani S, Gupta D. Artificial intelligence-based diagnosis of COVID-19 using chest CT images: A study from Indian

hospitals. Journal of Medical Systems. 2021; 45(5):55. DOI: 10.1007/s10916-021-01732-6. PMID: 33851234.

2. Patel S, Kumar R, Jain A. Deep learning-based detection of COVID-19 from chest X-rays: An Indian perspective. Indian Journal of Medical Research. 2020; 152(4):378–385. DOI: 10.4103/ijmr.IJMR249520. PMID: 33380702.
3. Singh V, Mishra A, Das S. A hybrid deep learning model for COVID-19 detection using CT scans and clinical symptoms: An Indian case study. Health Informatics Journal. 2022; 28(3):1–15. DOI: 10.1177/14604582221113245. PMID: 35946789.
4. Reddy K, Rao S, Kumar P. Transfer learning-based approach for COVID-19 detection in rural India using chest CT scans. Journal of the Indian Medical Association. 2021; 119(6):45–50. Available at: https://www.jima.in/article/2021/119/6/45.
5. Wang W, Xu Y, Gao R, et al. Detection of SARS-CoV-2 in different types of clinical specimens. JAMA. 2020; 323(18):1843–1844. DOI: 10.1001/jama.2020.3786. PMID: 32159775.
6. Dataset. https://www.kaggle.com/code/ineskova/covid-bin-classification-b54685.
7. Sitaula C, Aryal S. New bag of deep visual words based features to classify chest x-ray images for COVID-19 diagnosis. Health Information Science and Systems. 2021; 9:15. DOI: 10.1007/s13755-021-00152-w. PMID: 34164119.
8. Liu Y, Whitfield C, Zhang T, et al. Monitoring COVID-19 pandemic through the lens of social media using natural language processing and machine learning. Health Information Science and Systems. 2021; 9:16. DOI: 10.1007/s13755-021-00158-4. PMID: 34188896.
9. Guan W, Ni Z, Hu Y, et al. Clinical characteristics of coronavirus disease 2019 in China. New England Journal of Medicine. 2020; 382(18):1708–1720. DOI: 10.1056/NEJMoa2002032.
10. Franquet T. Imaging of pneumonia: trends and algorithms. European Respiratory Journal. 2001; 18(1):196–208. DOI: 10.1183/09031936.01.00213501. PMID: 11510793.
11. Cherian T, Mulholland EK, Carlin JB, et al. Standardized interpretation of paediatric chest radiographs for the diagnosis of pneumonia in epidemiological studies. Bulletin of the World Health Organization. 2005; 83:353–359. PMID: 15976876.
12. Franquet T. Imaging of pneumonia: trends and algorithms. European Respiratory Journal. 2001; 18 (1):196–208. https://doi.org/10.1183/09031936.01.00213501 PMID: 11510793 Table 10. Accuracy obtained by existing models and models used in the study. References Images Type No of Images Method Accuracy Ozturk et al. [10] Chest x-ray 125COVID-19 / 500Normal DarkCovidNet 98.08% Chest x-ray 125COVID-19/ 500Normal/ 500Pneumonia DarkCovidNet 87.02% Narin et al. [8] Chest x-ray 50COVID-19 / 50Normal ResNet50, Deep CNN 98% Sethey et al. [59] Chest x-ray 25COVID-19 / 25Normal ResNet50 + SVM 95.38% Ioannis et al. [58] Chest x-ray 224COVID-19 / 700Pneumonia / 504Normal VGG-19 93.48% Wang et al. [9] Chest x-ray 53COVID-19 / 5526Normal COVID-Net 92.4% Hemdan et al. [57] Chest x-ray 25COVID-19 / 25Normal COVIDX-Net 90% Zheng et al. [62] Chest CT 213COVID-19 / 229Normal UNet+3D Network 90.8% Ying et al. [60] Chest CT 777COVID-19 / 708 Normal DRE-Net 86% Xu et al. [18] Chest CT 219COVID-19 / 175Normal / 224Pneumonia ResNet + Location Attention 86.7% wang et al. [61] Chest CT 195COVID-19 / 258Normal

M-Inception 82.9% Our Proposed CNN Method Chest x-ray 140COVID-19 / 140Normal CNN 97.62% Chest x-ray 140COVID-19 / 140Normal /140 Pneumonia CNN 93.75% Our Employed Pre-trained Method Chest x-ray 140COVID-19 / 140Normal VGG16 100% Chest x-ray 140COVID-19 / 140Normal /140 Pneumonia VGG16 87.50% https://doi.org/10.1371/journal.pone.0262052 .t010 PLOS ONE Automated detection of COVID-19 PLOS ONE | https://doi.org/10.1371/journal.pone.0262052 January 21, 2022 23 / 26

13. . Cherian T, Mulholland EK, Carlin JB, Ostensen H, Amin R, Campo Md, et al. Standardized interpretation of paediatric chest radiographs for the diagnosis of pneumonia in epidemiological studies. Bulletin of the World Health Organization. 2005; 83:353–359. PMID: 15976876
14. Narin A, Kaya C, Pamuk Z. Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks. Pattern Analysis and Applications. 2021; p. 1–14. https://doi.org/10.1007/s10044-021-00984-y PMID: 33994847
15. Wang L, Lin ZQ, Wong A. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. Scientific Reports. 2020; 10(1):1–12. https://doi.org/
16. 1038/s41598-020-76550-z PMID: 33177550 10. Ozturk T, Talo M, Yildirim EA, Baloglu UB, Yildirim O, Acharya UR. Automated detection of COVID-19 cases using deep neural networks with X-ray images. Computers in Biology and Medicine. 2020; p. 103792. https://doi.org/10.1016/j.compbiomed.2020.103792 PMID: 32568675
17. Supriya S, Siuly S, Wang H, Zhang Y. Automated epilepsy detection techniques from electroencephalogram signals: a review study. Health Information Science and Systems. 2020; 8(1):1–15. https://doi.org/10.1007/s13755-020-00129-1 PMID: 33088489
18. LeCun Y, Bengio Y, Hinton G. Deep learning. nature. 2015; 521(7553):436–444. https://doi.org/10. 1038/nature14539 PMID: 26017442
19. Sarki R, Ahmed K, Wang H, Zhang Y. Automated detection of mild and multi-class diabetic eye diseases using deep learning. Health Information Science and Systems. 2020; 8(1):1–9. https://doi.org/10. 1007/s13755-020-00125-5 PMID: 33088488 14. Islam MR, Kabir MA, Ahmed A, Kamal ARM, Wang H, Ulhaq A. Depression detection from social network data using machine learning techniques. Health information science and systems. 2018; 6(1):1–
20. https://doi.org/10.1007/s13755-018-0046-0 PMID: 30186594 15. Du J, Michalska S, Subramani S, Wang H, Zhang Y. Neural attention with character embeddings for hay fever detection from twitter. Health information science and systems. 2019; 7(1):1–7. https://doi. org/10.1007/s13755-019-0084-2 PMID: 31656594
21. Sarki R, Michalska S, Ahmed K, Wang H, Zhang Y. Convolutional neural networks for mild diabetic retinopathy detection: an experimental study. bioRxiv. 2019. https://doi.org/10.1101/763136
22. Sarki R, Ahmed K, Wang H, Zhang Y. Automatic Detection of Diabetic Eye Disease Through Deep Learning Using Fundus Images: A Survey. IEEE Access. 2020; 8:151133–151149. https://doi.org/10.1109/ACCESS.2020.3015258
23. Xu X, Jiang X, Ma C, Du P, Li X, Lv S, et al. A deep learning system to screen novel coronavirus disease 2019 pneumonia. Engineering. 2020; 6(10):1122–1129. https://doi.org/10.1016/j.eng.2020.04.010 PMID: 32837749 19. Gozes O, Frid-Adar M, Greenspan H, Browning PD, Zhang H, Ji W, et al. Rapid ai development cycle for the coronavirus (covid-19) pandemic: Initial results for automated detection & patient monitoring using deep learning ct image analysis. arXiv preprint arXiv:200305037. 2020;.
24. Li L, Qin L, Xu Z, Yin Y, Wang X, Kong B, et al. Artificial intelligence distinguishes COVID-19 from community acquired pneumonia on chest CT. Radiology. 2020; p. 200905.
25. Shi F, Xia L, Shan F, Wu D, Wei Y, Yuan H, et al. Large-scale screening of covid-19 from community acquired pneumonia using

infection size-aware classification. arXiv preprint arXiv:200309860. 2020;.
26. Milletari F, Navab N, Ahmadi SA. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV). IEEE; 2016. p. 565–571.
27. Li X, Zhou Y, Du P, Lang G, Xu M, Wu W. A deep learning system that generates quantitative CT reports for diagnosing pulmonary tuberculosis. Applied Intelligence. 2021; 51(6):4082–4093. https://doi.org/10.1007/s10489-020-02051-1
28. Shan F, Gao Y, Wang J, Shi W, Shi N, Han M, et al. Lung infection quantification of covid-19 in ct images with deep learning. arXiv preprint arXiv:200304655. 2020;.
29. Ng MY, Lee EY, Yang J, Yang F, Li X, Wang H, et al. Imaging profile of the COVID-19 infection: radiologic findings and literature review. Radiology: Cardiothoracic Imaging. 2020; 2(1):e200034. https://doi.org/10.1148/ryct.2020200034 PMID: 33778547 26. Cohen JP, Morrison P, Dao L. COVID-19 image data collection. arXiv 200311597. 2020;.
30. Li X, Zhu D. Covid-xpert: An ai powered population screening of covid-19 cases using chest radiography images. arXiv preprint arXiv:200403042. 2020;.
31. Minaee S, Kafieh R, Sonka M, Yazdani S, Soufi GJ. Deep-covid: Predicting covid-19 from chest x-ray images using deep transfer learning. Medical image analysis. 2020; 65:101794. https://doi.org/10.1016/j.media.2020.101794 PMID: 3278137